Reinforcement learning in complementarity game and population dynamics

Jürgen Jost and Wei Li[*]

Max Planck Institute for Mathematics in the Sciences, Inselstr. 22, 04103 Leipzig, Germany

(Received 17 December 2012; revised manuscript received 19 December 2013; published 11 February 2014)

We systematically test and compare different reinforcement learning schemes in a complementarity game [J. Jost and W. Li, Physica A **345**, 245 (2005)] played between members of two populations. More precisely, we study the Roth-Erev, Bush-Mosteller, and SoftMax reinforcement learning schemes. A modified version of Roth-Erev with a power exponent of 1.5, as opposed to 1 in the standard version, performs best. We also compare these reinforcement learning strategies with evolutionary schemes. This gives insight into aspects like the issue of quick adaptation as opposed to systematic exploration or the role of learning rates.

DOI: 10.1103/PhysRevE.89.022113 PACS number(s): 02.50.Le, 89.75.Fb, 89.90.+n

I. INTRODUCTION

Reinforcement learning is a well-established and rather ubiquitous learning scheme. Its aim is to select and reinforce those actions that lead to high rewards and to avoid the others. It is especially powerful in solving problems in the fields of robotics, optimal control, and artificial intelligence. Unlike standard supervised learning [1], reinforcement learning is a goal-directed learning scheme. It depends on the agent's interaction with the environment, including other agents [2]. In reinforcement learning, learners are not told in advance which action to choose but rather have to try to maximize their rewards (mostly delayed and stochastic) by trial-and-error or more elaborate learning schemes. Reinforcement learning features on-line performance, which involves finding a good balance between exploration and exploitation. There are basically two problems in reinforcement learning, a statistical problem and a decision problem. The statistical problem is concerned with modeling the environment. The decision problem is about converting the reward expectation into an action.

Since its inception, many different schemes have been introduced in order to implement the idea of reinforcement learning. In the setting that we are going to model, players can choose among several strategies, labeled k. As a normalization, at time $t = 0$, none of the players has any experience, and each player n has nonzero propensity $Q_{nk}(0)$ to play the k-th strategy. A reinforcement learning rule then prescribes how a player should update his or her propensity in subsequent rounds depending on the reward his or her actions yielded in previous rounds. One of the most successful versions of reinforcement learning is Roth and Erev learning (RE) [3], which goes as follows: If at time t player n plays strategy k and gets a payoff $R_{nk}(t)$, then the propensity to play k is updated to be $Q_{nk}(t+1) = Q_{nk}(t) + R_{nk}(t)$. For all other strategies i, $Q_{ni}(t+1) = Q_{ni}(t)$. So the probability $P_{nk}(t+1)$ for player n to play strategy k at time $t+1$ is

$$P_{nk}(t+1) = Q_{nk}(t+1) / \sum_{i \in S} Q_{ni}(t+1), \tag{1}$$

where the sum is taken over all strategies in the set S that are available to player n. So strategies which have proved to be more successful tend to be played with greater frequency than those which have been less successful. In RE, the learning can be fast initially but then it slows down. In this simple model the players are not allowed to observe the full strategies of other players or to make calculations based on other players' payoffs. So it can be applied to the kinds of game in which players only observe one another's choices. In addition to the basic model, there were some modifications [4] which allow one to introduce some additional parameters. The first parameter is a "cutoff" parameter which prevents events with negligibly small probabilities from influencing the outcome. The second parameter prevents the probability of a strategy from approaching zero if it is in the vicinity of a successful strategy. The third parameter prevents the sum of any player's propensities from going to infinity. All three parameters are usually given quite small values.

Bush and Mosteller (BM) [5] introduced a rather different version of reinforcement learning in which the past payoffs are completely forgotten. What is relevant is the payoff immediately prior to the current action and not the payoffs of all earlier periods. In BM reinforcement, the probability of choosing a rewarded act is incremented by adding some fraction, the product of the reward and some learning parameter r ($0 \leqslant r \leqslant 1$), of the distance between the original probability and 1. Whereas the alternative actions' probabilities are decremented correspondingly so the sum of all probabilities is always 1. Hence, at time $t+1$ the probability of player n choosing action k, and the sum of probabilities of him choosing actions other than k are given, respectively, by

$$P_{nk}(t+1) = P_{nk}(t) + [1 - P_{nk}(t)] r_{nk}(t) r \tag{2}$$

and

$$P_{ni}(t+1) = P_{ni}(t)[1 - r_{nk}(t) r], i \neq k, \tag{3}$$

where the sum is taken over all actions other than k, and $r_{nk}(t)$ is the rescaled value of the actual reward $R_{nk}(t)$ of choosing action k for player n at time t. If $R_{\max}$ is the maximum reward of all possible actions, then $r_{nk}(t) = R_{nk}(t)/R_{\max}$. If the learning parameter r is small, then the learning is slow, and if it is large, then the learning is fast. Hence, in BM, the learning never slows down, unless one changes the value of r during the process.

Another well-known reinforcement learning scheme is the so-called SoftMax (SM) reinforcement in which the probabilities for players to choose certain actions are taken from a Gibbs-Boltzmann distribution [2]. Thus, the probability

[*] On leave from the Complexity Science Center, Huazhong Normal University, Wuhan 430079, P.R. China.

of player n for action k at time t is

$$P_{nk}(t) = \frac{e^{\lambda A_k^n(t)}}{\sum_{i \in S} e^{\lambda A_i^n(t)}}, \tag{4}$$

where S is the set of all strategies available to the player and $A_k^n(t)$ is the expected value, or propensity, for player n to choose $k \in S$ at time t. If the "inverse temperature" (in the jargon of statistical physics) λ is infinite, we get greedy learning, in which only the action with the highest propensity is taken. This is called exploitation. If λ is zero then all actions have equal probability, which is then termed exploration. The key then is to find a value of λ that achieves a reasonable trade-off between exploitation and exploration.

Reinforcement learning has been extensively studied in dealing with various tasks, both simple and complex. For instance, Trevisan *et al.* [6] studied the dynamics of learning in coupled oscillators with delayed reinforcement. Xie *et al.* [7] used reinforcement of irregular spiking of neurons to study the learning in neural networks. Potapov *et al.* [8] investigated the convergence rate of reinforcement learning algorithms and found that the choices of parameters such as learning steps, discount rate, and exploration degree may drastically influence the convergence of the techniques of reinforcement learning. Yuzuru *et al.* [9] used a group of reinforcement-learning agents to derive coupled replicator equations that describe the dynamics of collective learning in multiagent systems. More literature regarding the use of reinforcement learning can be found in Refs. [10–14].

The purpose of the present paper is to compare different reinforcement learning schemes in a dynamic setting in which also other players are learning and thereby continually shifting the payoffs of each player. Therefore, learning players need to adapt to the results of the learning of others. We shall utilize the complementarity game introduced in Ref. [15] so each player on one hand plays against other players from an opposing population and on the other hand has its payoffs compared with other players from his or her own population. Members of both populations may learn, by reinforcement or other schemes, and by equipping the two opposing populations with different schemes, we can then systematically compare which scheme performs better. We can also compare individual learning with evolutionary schemes where only the population as a whole learns across generations because its members reproduce based on their accumulated payoffs.

II. A ROTH-EREV TYPE SCHEME

Let us introduce our game. We have two populations simply called *buyers and sellers*. At each round, a buyer i is randomly paired to a seller j. The seller asks an amount k_j, and the buyer offers k_i, with both (integer-valued) bids ranging between 0 and some large integer K. If k_i is larger than, or at least equal to, k_j, then a deal concludes and the buyer wins $K - k_i$, and the seller k_j. Otherwise, the interaction fails and both gain nothing. Thus, the buyer is interested in making the offer as small as possible so the deal is just concluded but not smaller. The seller faces the reverse situation. Therefore, both players wish to drive as hard a bargain as possible, but if they push too hard, then the transaction will fail and both lose. Any value between 0 and K is a Nash equilibrium for the mutual offers. At $K/2$, the situation is symmetric in the sense that both players receive the same payoff. When players can learn from their experience in previous rounds and adapt, the actions should converge to some equilibrium value. As an alternative to individual learning or in addition to such learning, we can also insert this game into an evolutionary framework. The fitness of players is then measured by their accumulated payoffs over a fixed number of rounds and the players reproduce according to their fitness to generate the next generation. Again, the game then can be expected to settle at some Nash equilibrium, and as long as the setting is kept symmetric between the two populations, that equilibrium value should be around $K/2$. The speed of convergence towards that equilibrium will naturally depend on the strategies available to the players, in particular how and to what degree they are allowed to learn or coordinate. Our key point then is to break the intrinsic symmetry between the two populations by giving them different strategic options. We can then see which type of strategy is better in the sense that it leads to a more favorable equilibrium value for the corresponding population. If that value is larger than $K/2$, then the sellers do better, otherwise it is the buyers who do better. In general, we find that simpler and more flexible strategies lead to superior results at the population level because they can process the information in a more efficient way, which speeds up the convergence rate [16].

We shall now utilize this method to compare different reinforcement learning schemes. The two populations each have N players. Choosing any integer $k \in \{0,1,2,\ldots,K\}$ is called an action k. Besides standard Roth-Erev, Bush-Mosteller, and SoftMax reinforcement learning, we shall also evaluate a modified Roth-Erev scheme. In this modified Roth-Erev-type learning, the propensity of player n to choose action k at time $t+1$ is of the form

$$Q_{nk}(t+1) = Q_{nk}(0) + S_{nk}(t), t \geqslant 1, \tag{5}$$

where $Q_{nk}(0)$ is the initial propensity of player n to choose action k and $S_{nk}(t)$ is the sum of rescaled payoffs, proportional to the cumulated payoff $T_{nk}(t)$ she (he) has received from the periods up to time t in which she (he) has chosen action k. In the most basic case $Q_{nk}(0)$ is the same for all actions. The order of magnitude of $Q_{nk}(0)$, however, has to be set carefully as this will have a long-term effect on the learning process. The key issue here is the formula for $S_{nk}(t)$ as this will determine the speed of learning. In RE reinforcement, $S_{nk}(t) = T_{nk}(t) = A_{nk}(t)C_{nk}(t)$, where $A_{nk}(t)$ and $C_{nk}(t)$ are the average payoff and the number of times that player n chooses action k in the periods up to time t, respectively. For our purposes, we adopt the more general definition

$$S_{nk}(t) = \sum_{t'=1}^{C_{nk}(t)} [R_{nk}(t')]^{\tau}, \tag{6}$$

where $R_{nk}(t')$ is the payoff to player n for choosing action k at time step t' $(0 \leqslant t' \leqslant t)$ and τ is some non-negative real number. The probability $P_{nk}(t+1)$ for player n to choose k in the new scheme then is

$$P_{nk}(t+1) = \frac{\sum_{t'=1}^{C_{nk}(t)} [R_{nk}(t')]^{\tau}}{\sum_{i=0}^{K} \sum_{t'=1}^{C_{ni}(t)} [R_{ni}(t')]^{\tau}}. \tag{7}$$

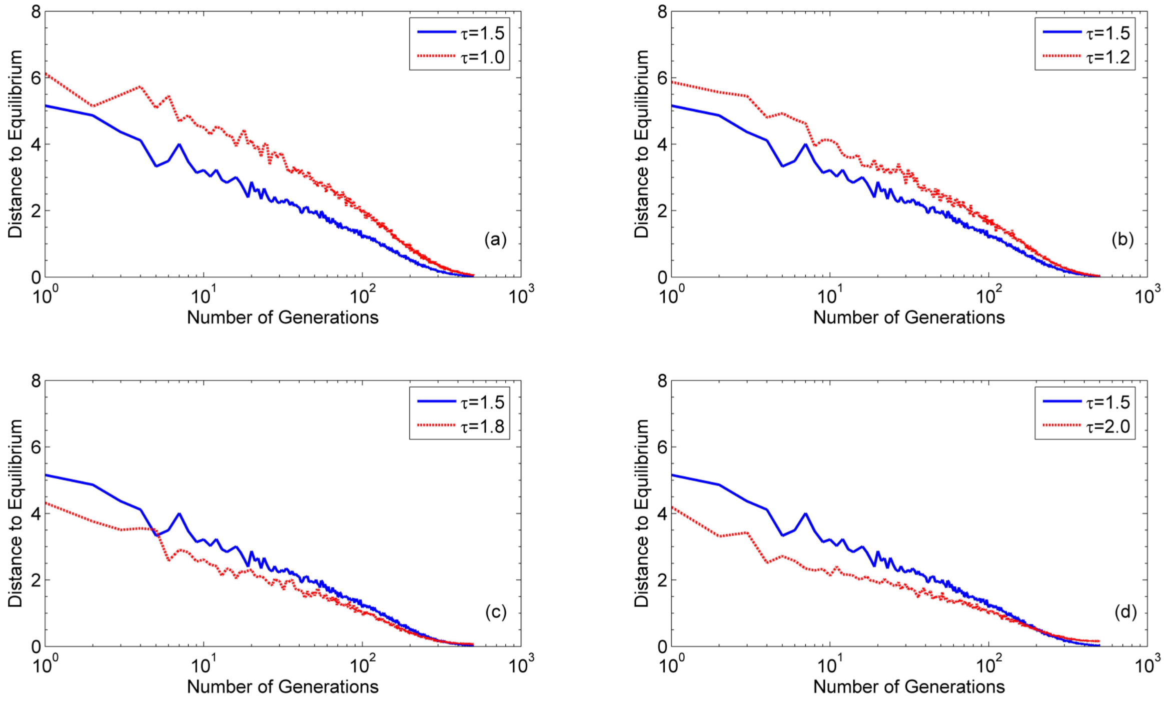

FIG. 1. (Color online) The comparison of convergence curves for power-formed Roth-Erev reinforcement learning with different exponents. Here the exponent τ takes 1.0, 1.2, 1.5, 1.8, and 2.0, respectively. $\tau = 1.5$ curve does not converge in the fastest way until nearly after 300 generations. Here one generation consist of 1000 time steps. (a) $\tau = 1.5$ vs $\tau = 1.0$; (b) $\tau = 1.5$ vs $\tau = 1.2$; (c) $\tau = 1.5$ vs $\tau = 1.8$; (d) $\tau = 1.5$ vs $\tau = 2.0$.

If $\tau = 1$, then $S_{nk}(t) = \sum_{t'=1}^{C_{nk}(t)} R_{nk}(t') = A_{nk}(t)C_{nk}(t)$, and the standard RE scheme is restored.

The order of magnitude of τ determines the learning speed. If $\tau = 0$, then the players are always exploring; if $\tau = \infty$, then a sole action will be taken. Here, we wish to find the optimal value of τ for our game. One way of determining the exact value of the optimal τ is by comparing the convergence of different power-formed RE learning. Namely all the members of the two populations will choose the same power-formed RE learning strategy, i.e., the one with the same given τ. Then the difference of the offers between buyers and sellers will be tracked. As known, in equilibrium, the difference is zero under symmetric situations. Hence, by varying the value of τ, one could obtain different curves of convergence towards equilibrium. As shown in Fig. 1, the $\tau = 1.5$ RE is not the best in the beginning stage. But after nearly 300 generations (here one generation consists of 1000 rounds of interactions), it becomes the best by beating the rest four RE's with τ either above or below 1.5. Extensive simulations also indicate an optimal value of $\tau = 1.5$, which, in fact, agrees with the finding in Ref. [4]. This observation also tells us clearly that the convergence speed shall not be too fast or too slow. A moderate convergence rate is more suitable in most cases. From now on we call this new reinforcement scheme NRE and fix the value of τ at 1.5 in the following simulations without further mention.

III. EQUATIONS FOR REINFORCEMENT LEARNING

A. Roth-Erev reinforcement learning

We first consider the RE reinforcement learning in our model. The technique can be readily applied to new Roth-Erev (NRE) and SoftMax (SM) learning schemes. Assume that both buyers and sellers can choose offers between 0 to K (arbitrarily large integer). At any given time step t, a buyer who chooses action i, namely offering i $(0 \leqslant i \leqslant K)$, meets a seller who chooses action j $(0 \leqslant j \leqslant K)$, namely asking for j. If $i \geqslant j$, then the payoff is $K - i$ for the buyer and j for the seller. Otherwise both receive 0 as payoffs. Denote $B_i(t)$ and $S_j(t)$ the cumulated reinforcements to the actions of i and j for buyers and sellers at time t, respectively. Here $B_i(t)$ and $S_j(t)$ are defined analogously to $T_{nk}(t)$ in the previous section. The probability that action i (j) is chosen for buyer (seller) is simply

$$P_{b,i}(t+1) = \frac{B_i(t)}{B(t)}, i = 0,1,\dots,K, \tag{8}$$

$$P_{s,j}(t+1) = \frac{S_i(t)}{S(t)}, j = 0,1,\dots,K, \tag{9}$$

where $B(t) = \sum_{i=0}^{K} B_i(t)$ and $S(t) = \sum_{j=0}^{K} S_j(t)$, respectively. To avoid the nonzero denominator, both $B_i(0)$ and $S_j(0)$ are taken to be positive for all i's and j's.

We now focus attention on the long-run behavior of average reinforcements. Consider the time averages of the reinforcements for both buyers and sellers. Namely

$$b_i(t) = \frac{B_i(t)}{t}, i = 0,1,\dots,K, \tag{10}$$

$$s_j(t) = \frac{S_j(t)}{t}, j = 0,1,\dots,K. \tag{11}$$

We also have $b(t) = \sum_{i=0}^{K} b_i(t)$ and $s(t) = \sum_{j=0}^{K} s_j(t)$.

It has been proven [17] that the mean-field version of Roth-Erev learning is similar to Maynard Smith's replicator dynamics [18]. Imitating the replicator dynamics [18], one can write down the following equations for RE reinforcement learning in our model,

$$\frac{dP_{b,i}(t)}{dt} = \frac{P_{b,i}(t)[E_{b,i}(t) - E_b(t)]}{b(t)}, i = 0,1,\dots,K, \quad (12)$$

$$\frac{dP_{s,j}(t)}{dt} = \frac{P_{s,j}(t)[E_{s,j}(t) - E_s(t)]}{s(t)}, j = 0,1,\dots,K, \quad (13)$$

where $E_{b,i}(t) = (K-i)\sum_{j'=0}^{i} P_{s,j'}(t)$ and $E_{s,j}(t) = j\sum_{i'=j}^{K} P_{b,i'}(t)$ are expected payoffs to buyer action i and seller action j at time t, respectively. Furthermore, $E_b(t) = \sum_{i=0}^{K} P_{b,i}(t)E_{b,i}(t)$ and $E_s(t) = \sum_{j=0}^{K} P_{s,j}(t)E_{s,j}(t)$ are expected payoffs of buyers and sellers at time t, respectively. The terms $b(t)$ and $s(t)$ here are no longer constants as in Maynard Smith's replicator dynamics in which reinforcements are always equal to current payoffs. In RE learning, the reinforcements approach current payoffs in a steady way. Therefore, it is reasonable to assume that in the continuous limit the change rate of reinforcements with respect to time is only related to the difference between reinforcements and current payoffs. That is,

$$\frac{db(t)}{dt} = c_b[E_b(t) - b(t)], \quad (14)$$

$$\frac{ds(t)}{dt} = c_s[E_s(t) - s(t)], \quad (15)$$

where c_b and c_s are constants independent of time. Combining $P_{b,i}(t) = b_i(t)/b(t)$ and $P_{s,j}(t) = s_j(t)/s(t)$, as well as Eqs. (12), (13), (14), and (15), we obtain $c_b = c_s = 1$ and

$$\frac{db_i(t)}{dt} = -b_i(t) + P_{b,i}(t)E_{b,i}(t), i = 0,1,\dots,K, \quad (16)$$

$$\frac{ds_j(t)}{dt} = -s_j(t) + P_{s,j}(t)E_{s,j}(t), j = 0,1,\dots,K. \quad (17)$$

Since K is relatively large, we shall utilize a continuum approximation. Thus, the offers now vary continuously between 0 and 1 (by rescaling from $[0,K]$ to the unit interval $[0,1]$). The set of equations for RE learning now can be approximated as, after some simple algebra,

$$\begin{aligned}
\dot{P}_{b,i}(t) &= \frac{P_{b,i}(t)}{b(t)}\Bigg[\int_0^1 i'P_{b,i'}(t)di' \int_0^{i'} P_{s,j}(t)dj \\
&\quad - i\int_0^i P_{s,j}(t)dj\Bigg], i \in [0,1],\\
\dot{b}(t) &= -b(t) + \int_0^1 (1-i)P_{b,i}(t)di \int_0^i P_{s,j}(t)dj,\\
\dot{P}_{s,j}(t) &= \frac{P_{s,j}(t)}{s(t)}\Bigg[j\int_j^1 P_{b,i}(t)di - \int_0^1 j'P_{s,j'}(t)dj' \\
&\quad \times \int_{j'}^1 P_{b,i}(t)di\Bigg], j \in [0,1],\\
\dot{s}(t) &= -s(t) + \int_0^1 jP_{s,j}(t)dj \int_j^1 P_{b,i}(t)di. \quad (18)
\end{aligned}$$

Now we consider a special case in which the buyers always offer m $(0 \leqslant m \leqslant K)$, namely $P_{b,i}(t) = \delta(i-m), i = 0,1,\dots,K$. We want to see how the sellers will respond. The equations for the seller choice probability are now

$$\dot{P}_{s,j}(t) = -\frac{P_{s,j}(t)}{s(t)}\sum_{j=0}^{m} P_{s,j'}(t)j'dj', m < j \leqslant K, \quad (19)$$

$$\dot{P}_{s,j}(t) = \frac{P_{s,j}(t)}{s(t)}\left[j - \sum_{j=0}^{m} P_{s,j'}(t)j'dj'\right], 0 \leqslant j \leqslant m. \quad (20)$$

Equation (19) tells that the probability for the sellers to ask for higher than m tends to decrease, which is very natural as buyers are all offering m. Denote $m' = \sum_0^m P_{s,j}(t)jdj$. The maximum value of m' is m, which is achieved when $P_{s,j} = \delta(j-m)$. This is exactly the solution of the stability condition of Eq. (20), namely $\dot{P}_{s,j}(t) = 0, 0 \leqslant j \leqslant m$. Hence, at the equilibrium, the sellers all have to only ask for m.

B. Other reinforcement learning schemes

The equations for NRE reinforcement learning schemes are nearly the same as those for RE reinforcement learning, except for the forms of $E_{b,i}(t)$, $E_{s,j}(t)$, $E_b(t)$, and $E_s(t)$. In NRE, the reinforcements gained at each round are no longer directly payoffs but some functions of the payoffs. Therefore, $E_{b,i}(t)$ and $E_{s,j}(t)$ no longer represent expected payoffs but expected reinforcements instead for buyers and sellers, respectively. Hence for NRE, we have

$$E_{b,i}(t) = (K-i)^\tau \sum_{j'=0}^{i} P_{s,j'}(t), i = 0,1,\dots,K, \quad (21)$$

$$E_{s,j}(t) = j^\tau \sum_{i'=j}^{K} P_{b,i'}(t), j = 0,1,\dots,K, \quad (22)$$

where τ $(\tau > 0)$ is the exponent of the power form of NRE reinforcement learning. Of course, the form of $E_b(t)$ $(E_s(t))$ changes according to $E_{b,i}(t)$ $(E_{s,j}(t))$.

For SM reinforcement learning, Eqs. (14) and (15) remain, but Eqs. (12) and (13) need to be revised based on the nature of SM learning scheme. In SM, the evolution of choice probability distributions can be written as

$$\dot{P}_{b,i}(t) = \lambda(\dot{E}_{b,i}(t) - \hat{E}_B(t))P_{b,i}(t), i = 0,1,\dots,K, \quad (23)$$

$$\dot{P}_{s,j}(t) = \lambda(\dot{E}_{s,j}(t) - \hat{E}_S(t))P_{s,j}(t), j = 0,1,\dots,K, \quad (24)$$

where $\hat{E}_B(t) = \frac{\sum_{i=0}^{K} \dot{E}_{b,i}(t)e^{\lambda E_{b,i}(t)}}{\sum_{i=0}^{K} e^{\lambda E_{b,i}(t)}}$ and $\hat{E}_S(t) = \frac{\sum_{i=0}^{K} \dot{E}_{s,j}(t)e^{\lambda E_{s,j}(t)}}{\sum_{j=0}^{K} e^{\lambda E_{s,j}(t)}}$. Here λ is the parameter for SM, which has been given when SM was first introduced.

BM reinforcement learning cannot be placed into the above framework. It has been pointed out [19], however, that the mean-field version of BM is also a replicator dynamics. We define $\Delta P_{b,i}(t) = P_{b,i}(t+1) - P_{b,i}(t)$ and $\Delta P_{s,j}(t) = P_{s,j}(t+1) - P_{s,j}(t)$. If both buyers and sellers use BM, the movement of state can be learned by studying the expectations of $\Delta P_{b,i}(t)$ and $\Delta P_{s,j}(t)$ conditional on the action probability

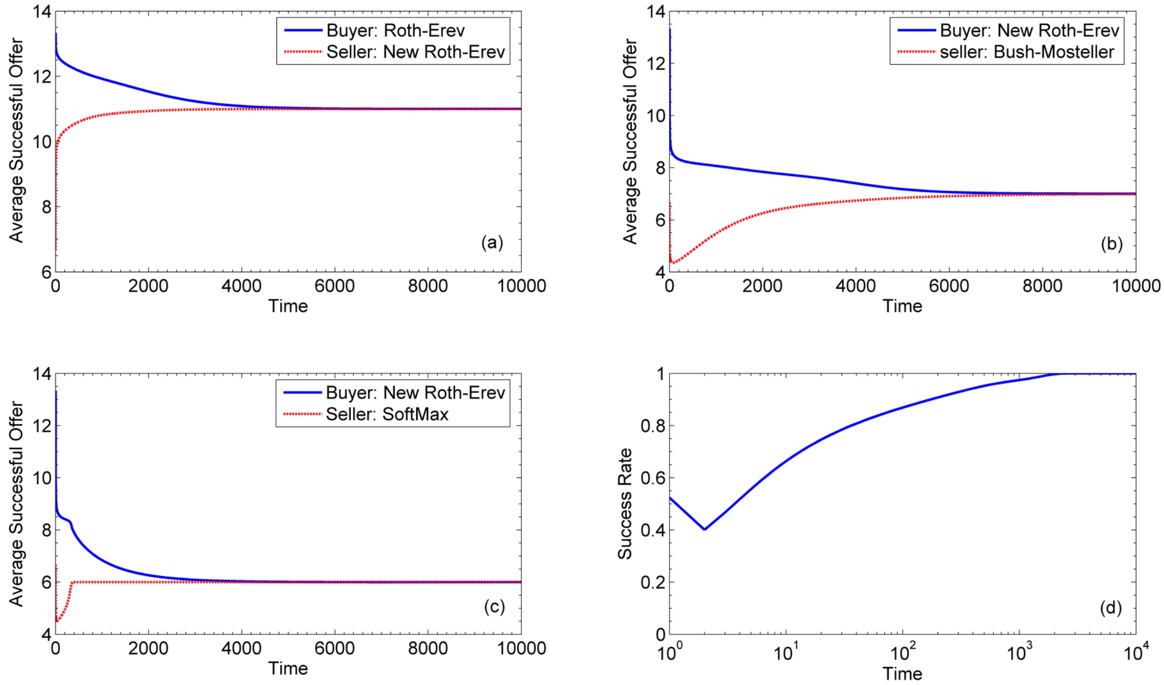

FIG. 2. (Color online) Numerical solutions of differential equations for reinforcement learning comparison. (a) Roth-Erev (RE) vs new Roth-Erev (NRE); (b) new Roth-Erev vs Bush-Mosteller (BM); (c) new Roth-Erev vs SoftMax (SM); (d) success rate for the case in (a).

distributions at time t, say, $P(t)$. Therefore, we have

$$E(\Delta P_{b,i}(t)|P(t)) = P_{b,i}(t)[E_{b,i}(t) - E_b(t)]r/K, \quad i = 0,1,\ldots,K, \tag{25}$$

$$E(\Delta P_{s,j}(t)|P(t)) = P_{s,j}(t)[E_{s,j}(t) - E_s(t)]r/K, \quad j = 0,1,\ldots,K, \tag{26}$$

where r ($0 < r < 1$) is the learning rate, and $E(\Delta P_{b,i}(t)|P(t))$ and $E(\Delta P_{s,j}(t)|P(t))$ are expectations of of $\Delta P_{b,i}(t)$ and $\Delta P_{s,j}(t)$ conditional on $P(t)$, respectively.

C. Numerical study of equations for reinforcement learning

For certain two-player game like the one in our model, the differential equations for reinforcement learning are strongly coupled, and, hence, no explicit solutions are available. Therefore we study those equations numerically. To compare the numerical results to the Monte Carlo simulations presented afterward, we need to introduce some quantities. Those are *average successful offer* (denoted by O_{AS}) and *success rate* (denoted by R_S), whose definitions have been given in Ref. [15]. Of course, all the quantities at time t are expectations conditional on the probability distributions $P(t)$. We have

$$R_S(t) = \sum_{i=0}^{K} P_{b,i}(t) \sum_{j=0}^{i} P_{s,j}(t) \tag{27}$$

or, equivalently,

$$R_S(t) = \sum_{j=0}^{K} P_{s,j}(t) \sum_{i=j}^{K} P_{b,i}(t). \tag{28}$$

For the buyer,

$$O_{\mathrm{AS}}(t) = \frac{\sum_{i=0}^{K} i\, P_{b,i}(t) \sum_{j=0}^{i} P_{s,j}(t)}{\sum_{i=0}^{K} P_{b,i}(t) \sum_{j=0}^{i} P_{s,j}(t)}, \tag{29}$$

and for the seller,

$$O_{\mathrm{AS}}(t) = \frac{\sum_{j=0}^{K} j\, P_{s,j}(t) \sum_{i=j}^{K} P_{b,i}(t)}{\sum_{i=0}^{K} P_{s,j}(t) \sum_{i=j}^{K} P_{b,i}(t)}. \tag{30}$$

As we can see from Eqs. (29) and (30), the overall performance of a population is determined not only by its own players but also by the players of the opposite population. After these preparations, we can then compare NRE to the other learning schemes. The numerical results are shown in Fig. 2. Here $K = 20$. We see that NRE beats RE and the equilibrium is 9 (11). NRE defeats BM by setting the equilibrium at 7 (13). NRE is also much better than SM by setting the equilibrium at 6 (14). In the next section, we will see that these numerical studies conform to the simulations by evolutionary updates. In contrast to the evolutionary updates which are subject to noise, the numerical solutions are exact within the error bounds of the numerical scheme employed. Nevertheless, if we want, we can also perturb the differential equations by noise. This will be investigated in more detail elsewhere.

IV. SIMULATIONS

A. Reinforcement versus reinforcement learning

We first compare the efficiency of the four different reinforcement learning schemes in our game, i.e., RE, BM, SM, and NRE, at the population level. We simply equip the members of one population with one learning strategy and the members of the opposite one with another strategy and check

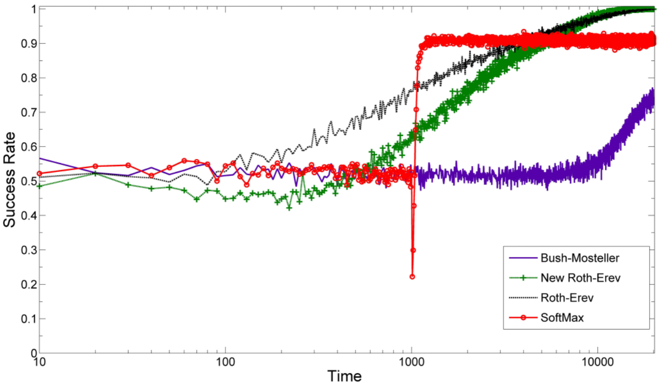

FIG. 3. (Color online) The time-varying success rates for the four different reinforcement learning schemes: RE, BM, SM, and NRE. The parameters are as follows: RE: initial inclination, 12, forgetting rate, 0.01; BM: learning rate, 0.0005; SM: $\lambda = 3.5$; NRE: initial inclination, 50, power exponent, 1.5, forgetting rate, 0.01.

for which the equilibrium value eventually reached is more favorable. Analogously, we can also equip both populations with the same type of strategy, but with different parameter values, in order to find the optimal value of that parameter. In order to see the basic picture, with issues like speed of convergence, however, we first equip both populations with the same strategies and the same parameters before we move to the comparison of different parameters or strategies.

In the simulations, we take $N = 1000$ and $K = 20$. The success rate is simply defined as the ratio of the number of successful deals to the number of pairs (that is, N, the population size).

As indicated in Fig. 3, BM learns quite slowly. The optimal learning rate is found to be around 0.0005 for BM in our setting. With a higher learning rate, the system gets stuck in some suboptimal equilibrium. This optimal value does not depend on the system size, which is shown in Fig. 4. Here the convergence towards equilibrium is studied via tracking the difference of offers between the buyers and the sellers, who all choose the BM with the same given learning rate. When the system sizes take 100, 200, 400, 800, and 1600, respectively, all the curves of convergence collapse onto each other. Of course, the optimal value of the learning rate might depend on the complexity of the task in models other than ours. But in the setting of our study there is no such dependence. SM learns very quickly, which only leads to a suboptimal equilibrium as seen from the low success rate eventually reached (around 0.9). This is so because some potentially effective actions may have been eliminated at rather early stages. RE and NRE learn much better than both BM and SM do as the learning rates are moderate and so achieve a good balance between exploration and exploitation. Interestingly, in the beginning NRE lags behind RE by learning relatively slowly but later NRE can achieve more favorable equilibrium values than RE.

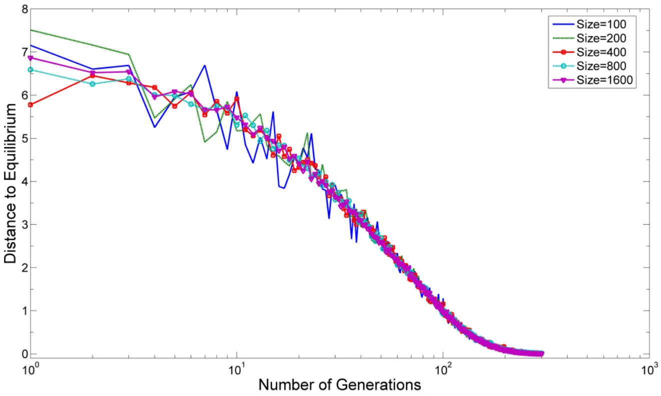

FIG. 4. (Color online) Size effects on convergence to equilibrium for BM reinforcement learning. All curves with sizes equaling 100, 200, 400, 800, and 1600, respectively, collapse onto each other. In all sizes, the learning rate is taken to be 0.0005. Therefore, there is almost no dependence of learning rate on system size.

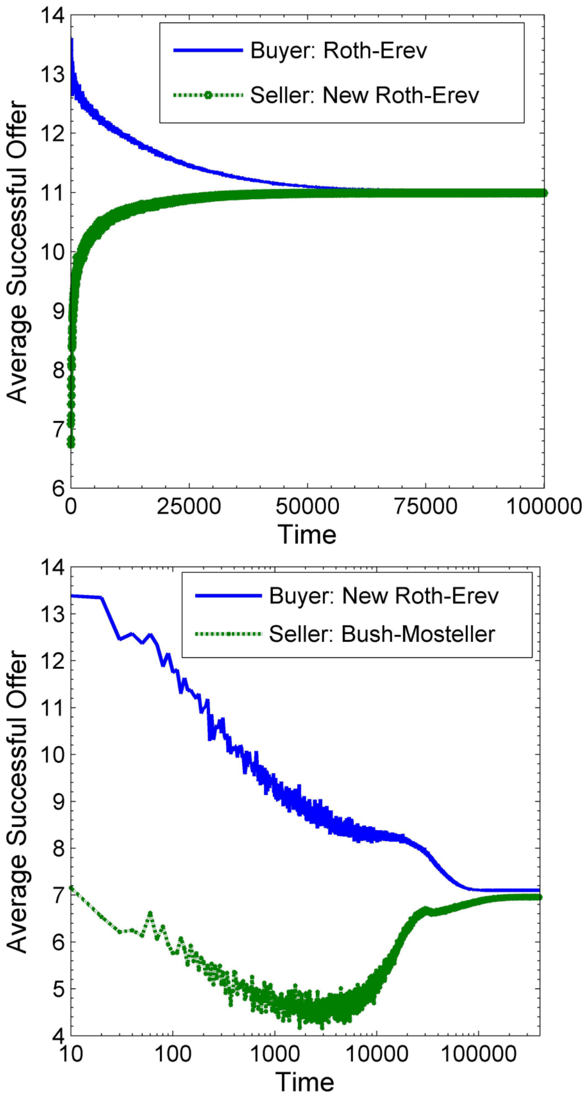

FIG. 5. (Color online) Top: NRE reinforcement gains advantage over RE reinforcement when both choose optimal parameters as indicated in Fig. 3. When the buyers take RE and the sellers NRE, the eventual equilibrium is 11, which is greater than 10. Bottom: NRE reinforcement gains advantage over BM reinforcement when both choose optimal parameters. When the buyers take NRE and the sellers BM, the eventual equilibrium is 7, which is much less than 10.

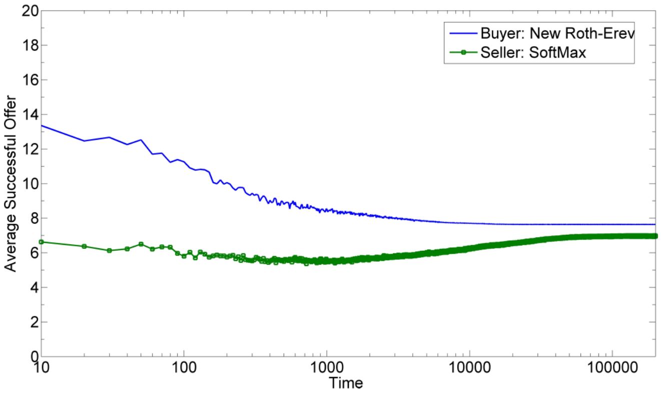

FIG. 6. (Color online) NRE reinforcement (buyer) defeats SM reinforcement (seller). The convergence takes a long time. The numerical study indicates the final equilibrium is 6. Parameters are the same as in Fig. 3.

As we see from the plots, NRE can surpass RE after around 10 000 time steps.

We now describe simulations of round-robin comparisons among the four different strategies. If we assign the buyers NRE and the sellers RE, the eventual equilibrium value is 9, which means the buyers are more favored (Fig. 5, top panel). If the buyers choose RE and the sellers choose NRE, then, by symmetry, the equilibrium value becomes 11, which is better for the sellers. Comparison between NRE and BM yields an equilibrium value of 7 (13) when the buyers (sellers) take NRE, and the other side takes BM (see Fig. 5, bottom panel). NRE is also superior to SM, see Fig. 6. It takes very long to approach the equilibrium, which is 6 here, as predicted by the numerical solution. Apparently, the Monte Carlo simulations conform to the numerical solutions of the differential equations in the previous section.

B. Learning speed

We now turn to an estimate of the learning speed for RE and NRE. Let us start from RE. Denote by $P_{b,i}(t)$ the probability of choosing action i at time t for buyers and by $P_{s,j}(t)$ the probability of choosing action k_j at time t for sellers. We then have

$$P_{b,i}(t+1) = \frac{\sum_{j=0}^{i} P_{b,i}(t)P_{s,j}(t)(K-i)}{\sum_{i=0}^{K}\sum_{j=0}^{i} P_{b,i}(t)P_{s,j}(t)(K-i)} \qquad (31)$$

and

$$P_{s,j}(t+1) = \frac{\sum_{i=j}^{K} P_{b,i}(t)P_{s,j}(t)j}{\sum_{j=0}^{K}\sum_{i=j}^{K} P_{b,i}(t)P_{s,j}(t)j}. \qquad (32)$$

For convenience of computation, in Eqs. (31) and (32) we have assumed that the initial inclination of each action is infinitely small and can be treated as zero (this claim is reasonable as K goes to ∞). The initial conditions are $P_{b,i}(0) = P_{s,j}(0) = 1/(K+1), i,j = 0,1,2\ldots,K$. The probabilities as Eqs. (31) and (32) are coupled, but after some algebra we obtain $P_{b,K/2}(1) = 1.5/(K+1)$. As $P_{b,K/2}(0) = 1/(K+1)$, hence, the probability of choosing $K/2$ for the buyers is increased by $0.5/(K+1)$ after the first learning step. Therefore the learning speed for RE is proportional to $1/(K+1)$ at the start. When K is large, the learning time should be proportional to K, though the learning speed in RE is not constant. A rather similar calculation can be applied to NRE to obtain $P_{b,K/2}(1) = 1.5K/(K+1)^2$. Thus, the increase of the probability of choosing action $K/2$ for the buyers is less than $0.5/(K+1)$, the counterpart for RE reinforcement. This simple calculation confirms the simulation results in Fig. 3 where NRE learns more slowly than RE does in the beginning.

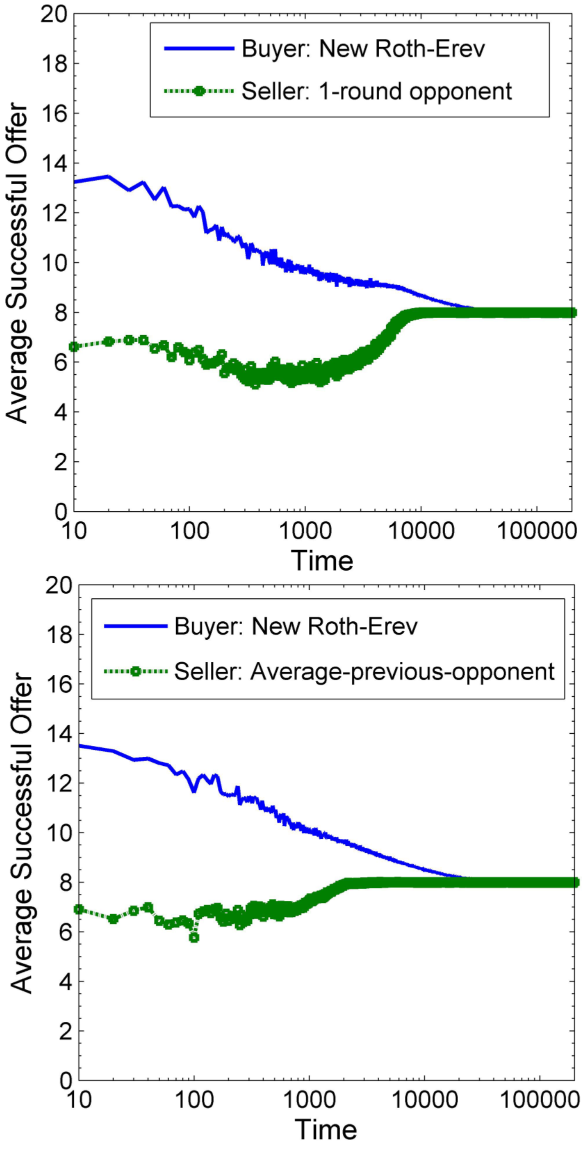

FIG. 7. (Color online) Top: NRE reinforcement gains advantage over the one-round opponent strategy when both choose optimal parameters. When the buyers take NRE strategy and the sellers the one-round opponent strategy, the eventual equilibrium is 8, which is less than 10. Bottom: NRE reinforcement gains advantage over average-previous-opponent strategy when both choose optimal parameters. When the buyers take the NRE strategy and the sellers the average-previous-opponent strategy, the eventual equilibrium is 8, which is less than 10.

We now briefly analyze why BM and SM reinforcement learning do not perform well in our game. First, in BM the

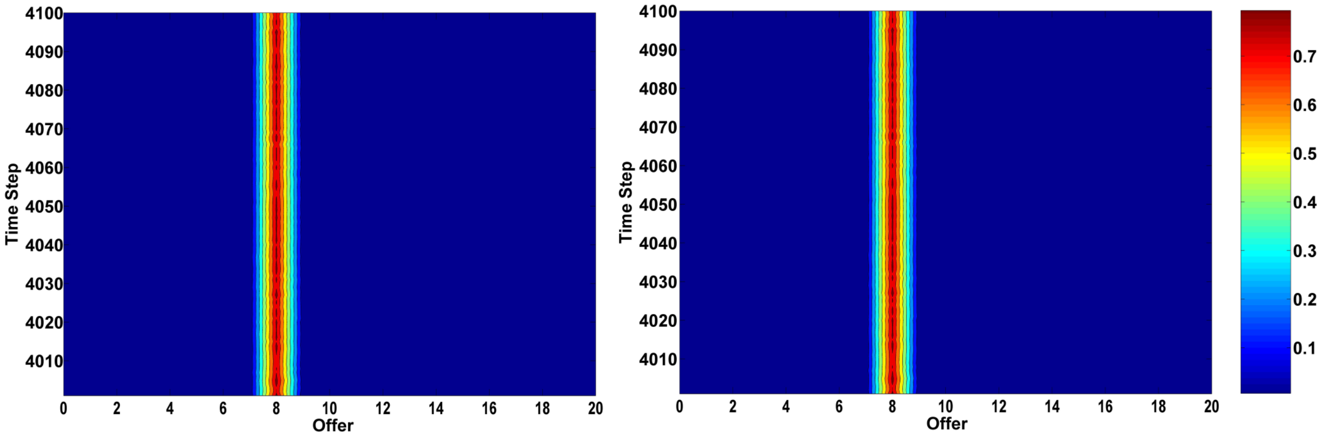

FIG. 8. (Color online) The time-varying distributions of offers for (a) buyers who take the NRE strategy and (b) sellers who the take one-round opponent strategy. The buyers' actions are more heterogeneous, whereas the sellers' actions are more homogeneous.

learning never slows down. This means that if a worse action is chosen, then there is no way to change. So for BM to fit into our game, the learning rate r has to be set very carefully. If r is too high, then the learning is very fast and it is very likely that an unfavored equilibrium will be reached. If r is low, then the learning takes very long, which will constitute a disadvantage when confronted with a quicker learner. This is indeed the dilemma between high and low learning rates. Our extensive simulations suggest that the learning rate should not be greater than 0.001 so the fair equilibrium, with value $K/2$, can be attained. But we already know from previous study that simpler and flexible strategies are more favored. Hence, when faced with other well-performing strategies, be they reinforcement learning or evolutionary schemes, BM will lose out as its convergence speed is very small at its optimal performance ($r = 0.0005$). A higher learning rate of BM will be even worse in the competition with other learning schemes. The process of exploration dominates the learning of BM when the learning rate is small.

SM reinforcement learning is another story. In SM, the probability of choosing action k is based on the average or expected reward (payoff), not on the total reward. Suppose we use $E(k)$ to denote the expectation reward of choosing action k. If the players are always exploring without learning, then $E(k) = (k+1)(K-k)/(K+1)$ for buyers and $k(K+1-k)/(K+1)$ for sellers. If K is an odd integer, then we have $E(K/2-1/2) = E(K/2+1/2)$; if K is an even integer then we have $E(K/2-1) = E(K/2+1)$ for both buyers and sellers. That is, there may exist two peaks in the action probability distribution which we wish to train. It is not easy to separate the double peaks so a single peak will remain. As in the beginning the players are always exploring, the consequence of the double peaks is that the optimal action cannot emerge as the two major remaining actions will coexist. If a greedy learning method is taken (by having a high "temperature," namely a large λ) at an early stage, then the good actions may get eliminated first, which is even worse. Whereas RE and NRE are situated between a slow-learning BM and a fast-learning SM and therefore behave more effectively. Efficient learning should be neither too hot (exploration) nor too cold (exploitation) [20].

C. Reinforcement versus evolutionary learning

We now widen the perspective and compare NRE reinforcement to other evolutionary schemes that we have studied before [15,16]. First, we introduce the following parameters for the evolutionary scheme of replacing a population of players by a new one composed of possibly mutated members of the present one with a fitness based selection:

(1) generation length (time): the number of rounds (time steps) played between two consecutive selections (if applicable);

(2) selection percentage: the percentage of the players who will be chosen as parents to generate the offspring during the evolutionary process;

(3) mutation rate: the rate of random mutation during the evolutionary process.

In the selection process, the fitness for a certain player is simply the payoffs he or she accumulate during the interactions between two consecutive selections. All the payoffs will then be compared and ranked among the members within the same population. In our setting, the top 50% fittest players will be kept as parents to produce the next generation. The whole selection process is implemented in the form of a genetic algorithm.

Next we list the five strategies in the pool, classified on the basis of the types of information they use as follows:

(1) average-previous-opponent: the average of one's opponents' offers in the previous, say, m (limited and usually much smaller than the generation length), rounds;

(2) for $m = 1$, that strategy is called one-round opponent: Each player utilizes the offer of his or her opponent in the most recent round;

(3) average-friend-opponent: the average of one's friends' opponents' offers in the most recent round [here each player has a certain number of friends (usually small in comparison with the population size) within his or her own population];

(4) average-all-friend: the average of one's friends' offers in the most recent round (thus, here, in contrast to the previous strategies, no information about the other population is used during each generation); and

(5) average-successful-friend: the average of one's friends' successful offers in the most recent round (here information

from the other population is used indirectly, but selectively, because their offers decide which of the friends are successful).

Each strategy can have two variants, either directly employing the value computed according to the chosen strategy as the next own offer or using that value as the input in a look-up table whose output then is that next offer. The look-up table then is itself an object of evolution. Hence, the evolutionary scheme refers to replacing a population with current look-up tables by a new one with evolved look-up tables. During the whole evolutionary process, the strategy that each population has chosen in the beginning will be always kept. In fact, since the look-up table has K input entries and has to provide an output for each of them, evolution will take quite some time to test it out thoroughly. To distinguish these two variants, we can simply put "simple" in front of the strategy that is not using look-up tables. For the one-round opponent strategy, the only efficient variant is the one with evolving look-up tables [16]. For the strategies that involve friends, we will introduce friendship networks of different topologies, with the average degree of each being fixed to, say, 5.

To have a stable setting, in our major simulations with evolving look-up tables, the generation length, the selection percentage and the mutation rate are 500, 0.5, and 0.01, respectively. In this paper, for the "simple" strategies without evolving look-up tables, the generation length has been taken to be 1, 4, or even larger in various simulations.

As we can see in Fig. 7 (top panel), when the buyers choose NRE reinforcement learning and the sellers choose the one-round opponent evolutionary strategy, the buyers gain an advantage by offering 8 eventually. Here evolution proceeds much faster than NRE reinforcement: 8 has been reached for the sellers after around 10 000 time steps, but the buyers are still offering 9. But the sellers cannot utilize this advantage by demanding amounts higher than 8 and have to wait until the buyers converge to 8 at approximately time step 100 000. Note that by taking the one-round opponent strategy, the sellers are not directly copying 9, the most-recent offer of the opponents, namely buyers, as their offers. Rather, they use 9 as an input into their own evolved look-up tables for an output for their next-round offers. This is exactly why their offers are 8 instead. We also notice that in NRE reinforcement the learning is very fast in the beginning but then gets flatter, as in the "law of practice" in psychology. In Fig. 8 we tracked the time-dependent distributions of offers for both buyers and sellers during the same time period. The heterogeneity of buyers' actions is very significant as buyers' offers are distributed between 7 and 11. For the same time period as given for the buyers, the sellers who take the one-round opponent strategy are more homogeneous as their offers are only distributed between 7 and 9. The centralized (most-frequent) offer for the sellers is nearly approaching the eventual equilibrium value, which is 8 in our setting. It is exactly the heterogeneity that helps NRE win. NRE reinforcement learning can also defeat average-previous-opponent strategy when m is larger, say, 5 (Fig. 7, bottom panel). When faced with a simple average-previous-opponent strategy ($m = 5$), however, NRE has no chance, as the former converges speedily to equilibrium after nearly 100 time steps. We already learned from Ref. [16] that averaging is a good strategy that can dampen the fluctuations in actions and therefore speed up the convergence. Simple averaging without look-up tables is even better than the one that evolves the look-up tables as the complicated evolutionary schemes need extra time for elaboration.

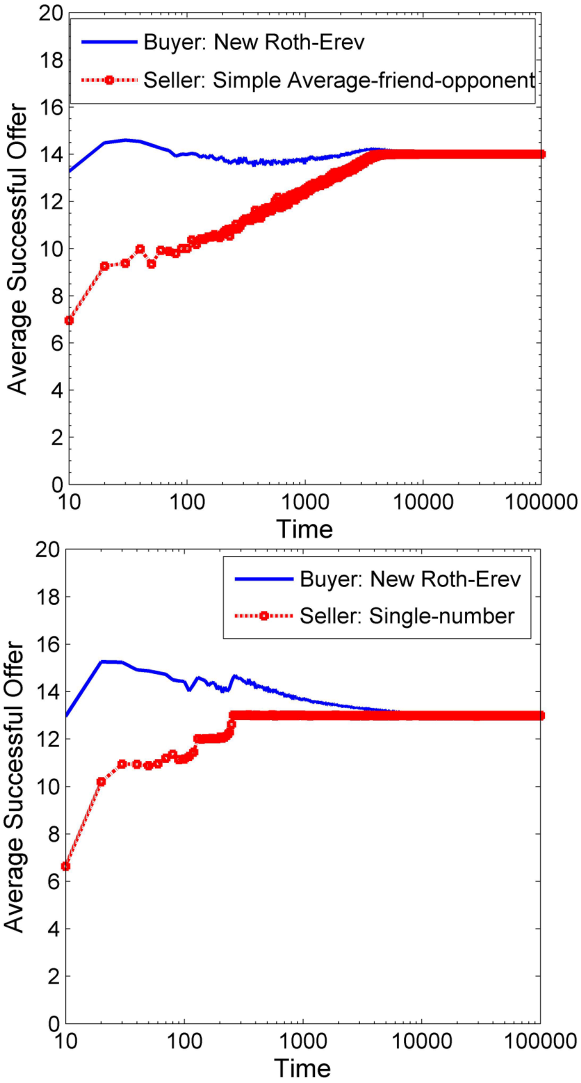

FIG. 9. (Color online) Top: NRE reinforcement loses advantage to the simple average-friend-opponent strategy when both choose optimal parameters. When the buyers take the NRE strategy and the sellers take the simple average-friend-opponent strategy, the eventual equilibrium is 14, which is much greater than 10. Bottom: NRE reinforcement loses advantage to the single-number strategy when both choose optimal parameters. When the buyers take the NRE strategy and the sellers the single-number strategy, the eventual equilibrium is 13, which is much greater than 10.

NRE reinforcement learning wins against the average-friend-opponent strategy (the average number of friends per player is 5) but will lose when confronted with the simple average-friend-opponent strategy (Fig. 9, top panel). The reason is as before that simple averaging is more able to adapt when dealing with simpler tasks. But this does not mean that the simple averaging strategy will also win in more sophisticated environments or given more complicated tasks. The reason is that in such cases, more feedback or information needs to be collected in order to deal with the higher complexity of the systems. In those occasions powerful strategies

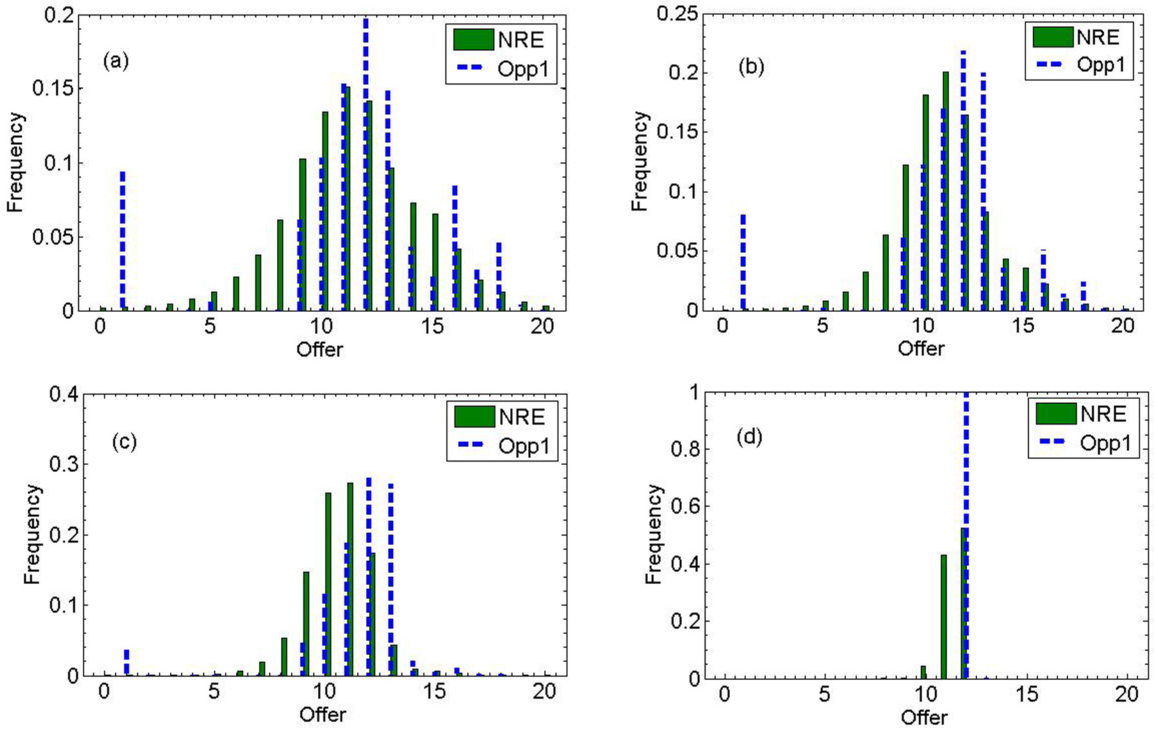

FIG. 10. (Color online) Distributions of offers for buyers who take Opp1 and for sellers who take NRE after (a) 3 generations, (b) 5 generations, (c) 10 generations, and (d) 50 generations. Here one generation consists of 500 time steps. Note the heterogeneity of distributions of offers for NRE players.

might not be very simple. Unlike the direct information in the average-previous-opponent strategy, in the average-friend-opponent strategy indirect information from friends is used. NRE reinforcement learning can beat the average-all-friend strategy as well since the latter uses no information from the other population. NRE reinforcement learning ties the simple average-all-friend strategy, with both reaching $K/2$, the symmetrical equilibrium value. This happens simply because the simple average-all-friend strategy converges quickly to $K/2$, which NRE then has to follow. NRE reinforcement learning can beat average-successful-friend strategy, whether simple or not. The reason is that following the successful experience will make the players' offers more timid, which is not good against a population that can try more ambitious offers.

In fact, the simplest is also the most successful strategy, the single-number strategy: players use no direct information at all; each player chooses a fixed random offer that will be updated through the selection based on fitness [16]. The more successful offer will be played with higher frequency and is more likely to spread within the population. Eventually a given offer might be chosen by all the members of the same population if the stochastic effects wash out. For the single-number strategy, it turns out that the minimal generation length, 1, is optimal. The population can then evolve most quickly. Here when we compare NRE reinforcement learning to the single-number strategy when both employ optimal parameters, the latter wins by converging very quickly after nearly a few hundred time steps (Fig. 9, bottom panel). Again, the superiority of simpler strategies may not extend to more complex environments. The advantage may only reflect the adaptability of simpler strategies in simple settings.

We have also compared RE reinforcement learning with all the evolutionary strategies we have devised, with similar, but somewhat less significant, results as for NRE. For example, the equilibrium value of the competition between RE reinforcement learning and the one-round opponent strategy is 9 (11), 1 less than 8 (12). This is consistent as we have found that the advantage of NRE reinforcement over RE reinforcement is just 1 in our game. Our simulations indicate that in most cases this advantage is transitive but there are a few exceptions. For instance, NRE wins 2, and RE wins 1, over the average-friend-opponent strategy, and transitivity holds among the three. But NRE loses 4, and RE also loses 4, to the simple average-successful-friend strategy, and the transitivity fails here. The reason might be that in these two

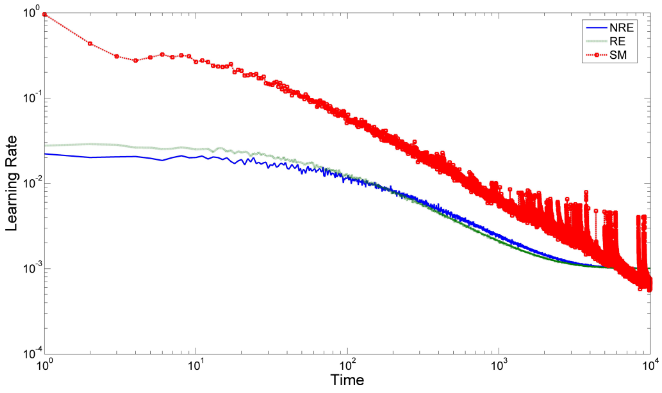

FIG. 11. (Color online) Comparison of learning rates for the three different reinforcement learning schemes, RE, NRE, and SM.

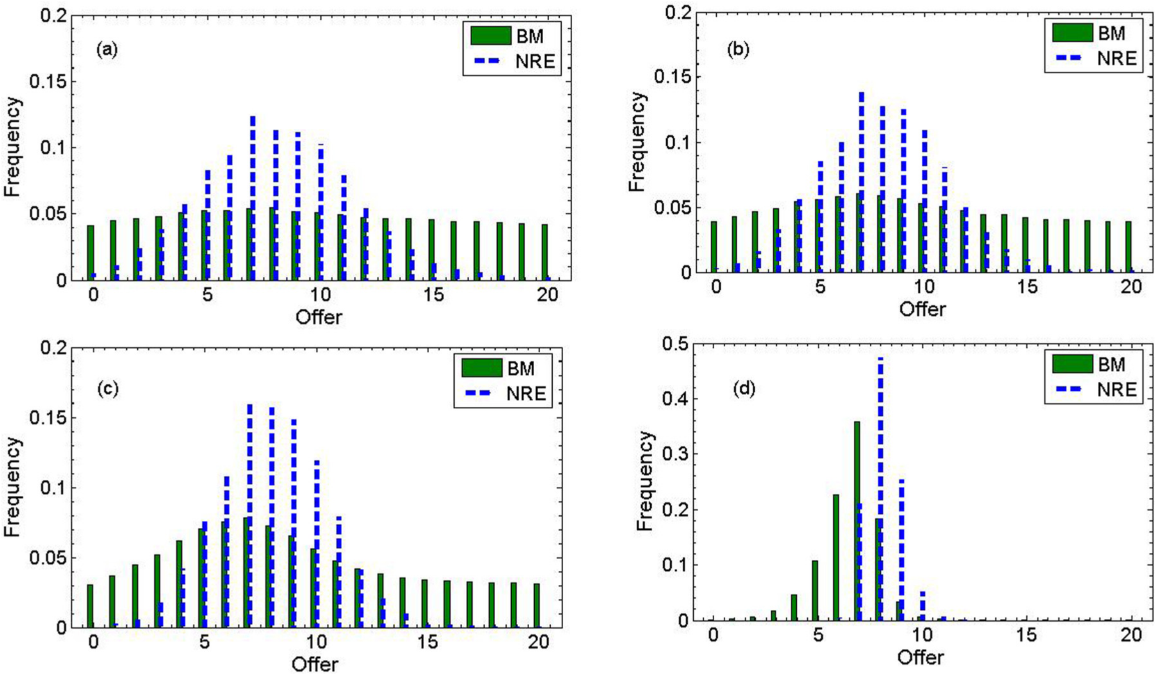

FIG. 12. (Color online) Distributions of offers for buyers who take the NRE scheme and for sellers who take the BM scheme after (a) 3 generations, (b) 5 generations, (c) 10 generations, and (d) 50 generations. Here one generation consists of 500 time steps. Most of the time BM players learn more slowly than NRE players.

cases the simple average-successful-friend strategy dominates the interactions so reinforcement learning schemes have to adapt to the same level of equilibrium.

BM reinforcement learning is found to be equal to the one-round- and two-round opponent strategies, the average-friend-opponent strategy, and the average-all-friend strategy. BM can defeat the average-successful-friend strategy, with or without the look-up tables. But BM loses to the remaining simple strategies in which no look-up table is included. Again here we find that, with certain exceptions, the transitivity of advantage holds. For instance, RE wins 1 over BM, and BM is equal to the one-round opponent strategy and there is transitivity between these three. But there is no transitivity among the BM, one-round opponent, and two-round opponent strategies because we found already that one-round opponent strategy can defeat the two-round opponent strategy.

We now try to understand why NRE performs better than either the one-round opponent strategy or other reinforcement learning schemes within the framework of our model. In general, there are two main aspects that may affect the comparison of the performance of strategies. First, the power of a strategy is related to its ability to deal with various tasks, be they simple or difficult. Second, the flexibility of a strategy is related to its ability to adapt to changing circumstances. Although the rules of our model are extremely simple, the action takes place at three different levels. The first level is information evaluation and learning for individuals. The second level is adaptation and evolution between individuals inside a population. The third level is competition between strategy spaces at the population level. In the present setting, our game focuses more on the adaptive ability of the strategies than on their power. But still the collective dynamics at the population level is a combination of adaptation, learning, and evolution.

We first compare NRE reinforcement learning with the one-round opponent strategy (abbreviated as Opp1 hereafter). As already seen in Fig. 8, the superiority of the NRE results from the heterogeneity of the distribution of its players' offers. To better display this feature of NRE, we tracked the distributions of both NRE and Opp1 players' offers from the beginning until the stage of equilibrium. Such distributions for four different stages are shown in Fig. 10 in which the buyers take Opp1 and the sellers take NRE. Each generation consists of 500 time steps. Figures 10(a), 10(b), 10(c), and 10(d) display the distributions of offers for both populations after 3, 5, 10, and 50 generations, respectively. We clearly see the diversity of offers for NRE at all times, although, of course, learning eventually also narrows down the distribution

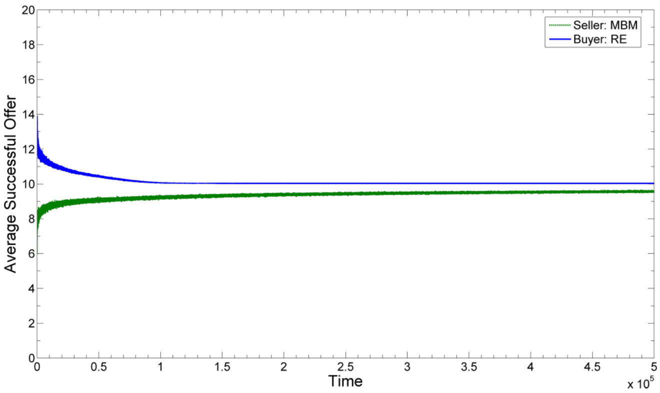

FIG. 13. (Color online) A modified Bush-Mosteller (MBM) in which the learning rate is changing with respect to the time t as $t^{-0.8}$ performs as well as RE. In both schemes the optimized parameters are taken.

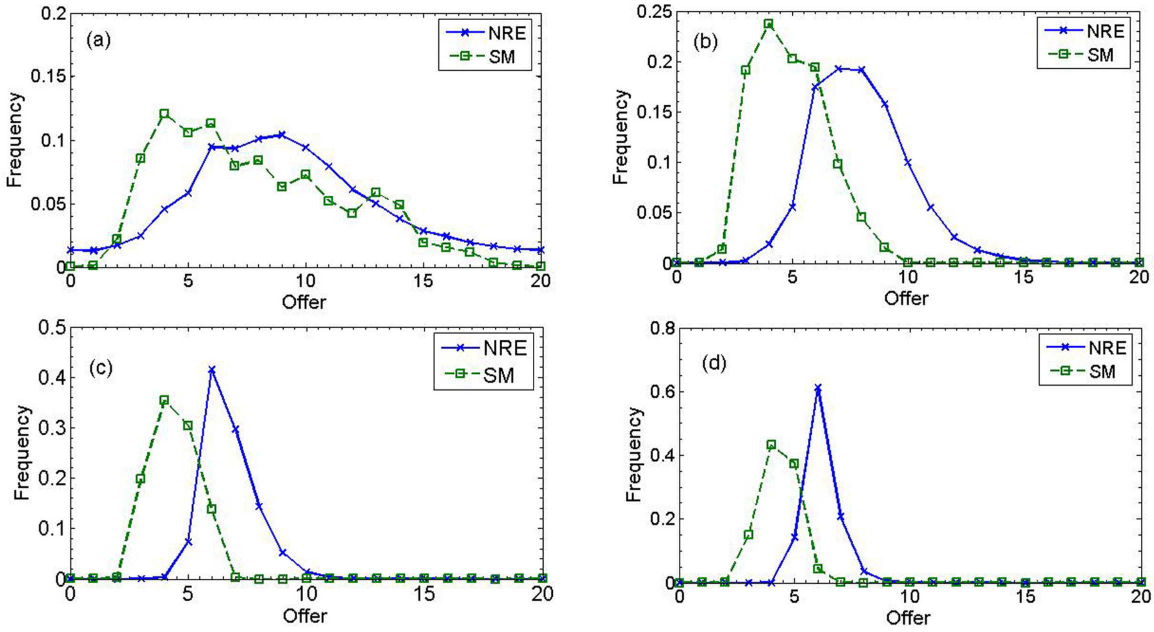

FIG. 14. (Color online) Distributions of offers for buyers who take NRE and for sellers who take SM after (a) 1 generation, (b) 5 generations, (c) 20 generations, and (d) 40 generations. Here one generation consists of 500 time steps. SM players learn faster than NRE players. NRE pushes SM by gradually shifting the highest peak, where the most frequent offer locates, towards its own favorable value.

of offers. In contrast, Opp1 players quickly concentrate at the single offer value 12. The heterogeneity of offers is crucial for learning because it allows enough time for testing all possible values. Without this feature some potentially good values might get eliminated at rather early stages.

Of course, the fact that NRE is on-line learning also contributes to its better performance over Opp1 where analogs of learning effects only occur at the transition between generations. NRE can also take a subtle advantage of the fact that Opp1 tests the entries of look-up tables.

The performance comparison between NRE and other reinforcement learning schemes is related to the learning rate. Our model confirms the well-known fact that learning needs to balance exploration and exploitation. If the learning rate is too large, then the system may easily get stuck in suboptimal states. If the rate is too small, the system is always exploring with little exploiting. Moreover, the learning rate should be adaptive and decrease with time. We have already seen that the constant-learning BM and the fast-learning SM do not fit our model well. Of course, the choice of learning rates is problem specific [8,21–24]. An effective varying learning rate $r(t)$ usually should satisfy the following two constraints:

$$\sum_{t=1}^{\infty} r(t) = \infty, \sum_{t=1}^{\infty} r^2(t) < \infty. \tag{33}$$

Hence, $r(t)$ shall be of the "$1/t^{\alpha}$" type, with $1/2 < \alpha \leqslant 1$. In the RE, NRE, and SM schemes, the learning rate is not explicitly given, but we can employ the definition of Sutton [2]. So the probability of a rewarded action k is updated according to

$$P_{nk}(t+1) = P_{nk}(t) + [1 - P_{nk}(t)]r_k(t). \tag{34}$$

Considering Eqs. (8) and (9) and Eq. (34), one can define the action k-related learning rate $r_k(t)$ for RE and NRE schemes simply as

$$r_k(t) = \frac{E_{b,k}(t)}{B(t)} \tag{35}$$

or, equivalently,

$$r_k(t) = \frac{E_{s,k}(t)}{S(t)}, \tag{36}$$

where $E_{b,k}(t)$ and $E_{s,k}(t)$ are the expected reinforcements to action k chosen by a buyer and a seller, respectively. Equations (35) and (36) are equivalent so we only need to focus on one of them, say, the former. Combining Eqs. (14) and (35), one can obtain, after some algebra,

$$\dot{b}(t) = B(t)\left(r(t) - \frac{1}{t}\right), \tag{37}$$

where $r(t) = \sum_0^K r_i(t)P_{b,i}(t)$ is the mean learning rate for all possible actions at t. Hence, we notice that at the equilibrium $r(t)$ is exactly $1/t$. In the regimes (other than the equilibrium) where $\dot{b}(t) > 0$, $r(t)$ converges faster than $1/t$.

Using Eq. (34) we measured the numerical values of the learning rates for RE, NRE, and SM; these are displayed in Fig. 11. Most noticeably we find SM learns much faster than both RE and NRE before $t = 5000$. In the beginning the learning rate for SM is nearly two orders of magnitude greater than those of RE and NRE. We notice that there exist regimes of plateaus for both RE and NRE due to the effects of initial inclinations, primarily before $t = 100$, where RE learns slightly faster than NRE. But after crossing the $t = 100$ point, NRE learns a little faster than RE. In the intermediate stage, for RE, $r(t) \sim t^{-0.8}$, and for NRE, $r(t) \sim t^{-0.79}$. These learning rates are consistent with the results from Eq. (37) when noise is taken into account. For SM, the dependence of the learning rate on t is complicated. At the start, $r(t)$

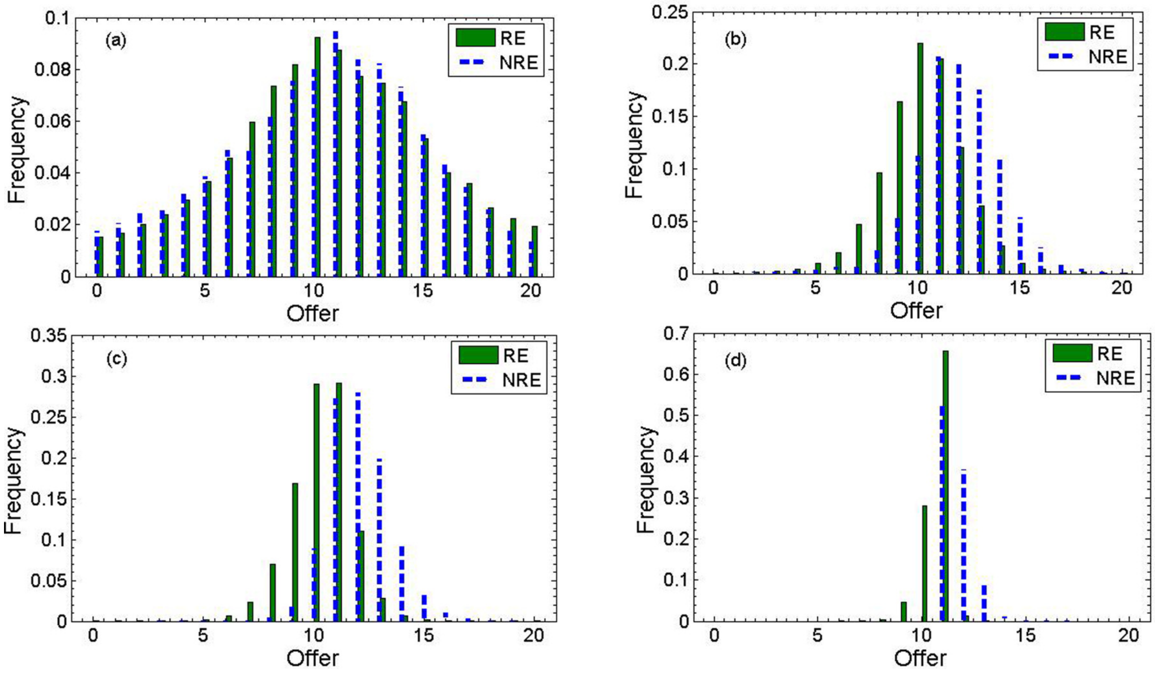

FIG. 15. (Color online) Distributions of offers for buyers who take RE and for sellers who take NRE after (a) 1 generation, (b) 5 generations, (c) 10 generations, and (d) 30 generations. Here one generation consists of 500 time steps. Most of the time RE players learn slightly more slowly than NRE players.

decreases almost linearly with respect to t, so the large learning rate is maintained for a long time. In the middle period $r(t)$ scales as $t^{-0.9}$ and at the latter stage $t^{-1.1}$. Therefore, SM is not an effective scheme for our game because learning is too fast.

The effects of learning rates can also be demonstrated through the time-varying distributions of players' offers. In Fig. 12 we compare the distributions of offers for NRE and BM players. In Fig. 12(a) we see that after three generations, the distribution for NRE players is approaching a Gaussian whereas the counterpart for BM players is still more or less a uniform distribution. After five generations [Fig. 12(b)] the distribution for BM players starts to look like a Gaussian. But still BM players are learning more slowly than NRE players, so the most frequent offer, 7, is more advantageous to the latter as buyers. In Fig. 12(c) and Fig. 12(d) the gap between the performance of the two strategies is shortened as for BM the learning is constant and for NRE the learning slows down gradually. So the main reason that NRE beats BM is that, when both take optimized parameters, the latter learns too slowly in the beginning. With this insight, it is then natural to consider a modified BM with a varying learning rate of $t^{-0.8}$ type. This modified BM can perform as well as RE, see Fig. 13. In another direction, NRE beats SM because the latter learns too fast in the beginning when both strategies are optimized. Figures 14(a), 14(b), 14(c), and 14(d) demonstrate this over time. NRE is better than RE because it uses its information somewhat faster (Fig. 15). The exponent τ might be considered as the strength parameter of payoff. As seen in Fig. 1, $\tau = 1.5$ leads to optimal convergence towards equilibrium.

V. CONCLUSION

We have identified in this paper an efficient reinforcement learning scheme within the framework of our game. This so-called NRE reinforcement learning performs better than the RE, BM, and SM reinforcement learning schemes. NRE reinforcement learning also beats most evolutionary strategies with evolving look-up tables but loses to the simple version of those strategies without look-up tables.

We have analyzed the learning rates of the various strategies in order to explain their different performances. Powerful strategies may not be adaptive and vice versa. The effective learning scheme found in here could be very specific to the setting of our game and may not be very powerful in other more complicated settings. It remains to evaluate the performance of this NRE reinforcement learning in other learning tasks.

ACKNOWLEDGMENTS

The authors thank the two referees for constructive suggestions and comments. W.L. was partially supported by the National Natural Science Foundation of China under Grant No. 10975057 and the Programme of Introducing Talents of Discipline to Universities under Grant No. B08033.

[1] T. Hastie, R. Tibshirani, and J. Friedman, *Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer-Verlag, Berlin, 2008).

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA 1998).

[3] A. E. Roth and I. Erev, Games Econ. Behav. **8**, 164 (1995).

[4] I. Erev and A. E. Roth, Am. Econ. Rev. **88**, 848 (1998).

[5] R. B. Bush and F. Mosteller, Psychol. Rev. **58**, 313 (1951).

[6] M. A. Trevisan, S. Bouzat, I. Samengo, and G. B. Mindlin, Phys. Rev. E **72**, 011907 (2005).

[7] X. H. Xie and H. S. Seung, Phys. Rev. E **69**, 041909 (2004).

[8] A. Potapov and M. K. Ali, Phys. Rev. E **67**, 026706 (2003).

[9] Y. Sato and J. P. Crutchfield, Phys. Rev. E **67**, 015206 (2003).

[10] S. Gadaleta and G. Dangelmayr, Phys. Rev. E **63**, 036217 (2001).

[11] M. Sperl, A. Chang, N. Weber, and A. Hübler, Phys. Rev. E **59**, 3165 (1999).

[12] S. Nara and P. Davis, Phys. Rev. E **55**, 826 (1997).

[13] M. Kushibe, Y. Liu, and J. Ohtsubo, Phys. Rev. E **53**, 4502 (1996).

[14] D. Stassinopoulos and P. Bak, Phys. Rev. E **51**, 5033 (1995).

[15] J. Jost and W. Li, Physica A **345**, 245 (2005).

[16] J. Jost and W. Li, Adv. Com. Sys. **11**, 901 (2008).

[17] A. Beggs, J. Econ. Theo. **122**, 1 (2005).

[18] J. M. Smith, *Evolution and the Theory of Games* (Cambridge University Press, Cambridge, 1982).

[19] T. Börgers and R. Sarin, J. Econ. Theo. **77**, 1 (1997).

[20] B. Skyrms, *Signals: Evolution, Learning, and Information* (Oxford University Press, New York, 2010).

[21] P. Dayan, Mach. Lear. **8**, 341 (1992).

[22] S. Singh and P. Dayan, Mach. Lear. **32**, 5 (1998).

[23] E. Even-Dar and Y. Mansour, J. Mach. Lear. Res. **5**, 1 (2004).

[24] R. S. Sutton, Mach. Lear. **3**, 9 (1988).