Max-Planck-Institut für Mathematik in den Naturwissenschaften Leipzig

Reinforcement Learning in Complementarity Game and Population Dynamics

by

Jürgen Jost and Wei Li

Preprint no.: 13 2013

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
(Dated: December 17, 2012)

We let different reinforcement learning schemes compete in a complementarity game [1] played between members of two populations. This leads us to a new version of Roth-Erev (NRE) reinforcement learning. In this NRE scheme, the probability of choosing a certain action k for any player n at time t is proportional to the accumulated rescaled reward by playing k during the time steps prior to t. The formula of the rescaled reward is a power law of payoffs, with the optimal value of the power exponent being 1.5. NRE reinforcement learning outperforms the original Roth-Erev-, Bush-Mosteller-, and SoftMax reinforcement learning when all of them choose optimal parameters. NRE reinforcement learning also performs better than most evolutionary strategies, except for the simplest ones which have the advantage of most quickly converging to some favorable fixed point.

PACS numbers: 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 [2], reinforcement learning is a goal-directed learning scheme. It depends on the agent's interaction with the environment, including other agents [3]. 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 some 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 modelling 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 between several strategies, labelled by k. As a normalization, at time $t = 1$, none of the players has any experience, and each player n has propensity $p_{nk}(1)$ to play the k-th strategy. A reinforcement learning rule then prescribes how a player should update her propensity in subsequent rounds depending on the reward her actions yielded in previous rounds. One of the most successful versions of reinforcement learning is Roth and Erev learning (RE) [4], which goes like this. If at time t player n plays strategy k and gets a payoff x, then the propensity to play k is updated to be $q_{nk}(t+1) = q_{nk}(t) + x$. For all other strategies i, $q_{ni}(t+1) = q_{ni}(t)$. So, the probability $p_{nk}(t)$ for player n to play strategy k at time t is

$$p_{nk}(t) = q_{nk}(t)/\sum_i q_{ni}(t), \tag{1}$$

where the sum is taken over all strategies that are available for 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 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 [5] which allow one to introduce some additional parameters. The first parameter is a "cutoff" parameter which prevents events with negiliblily 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.

---

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

Bush and Mosteller (BM) [6] introduced a rather different version of reinforcement learning in which the past payoffs are completely forgotten. 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, of the distance between the original probability and 1. Whereas the alternative action probabilities are decremented correspondingly so that the sum of all probabilities is always 1. 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 [3]. Thus, the probability of player i for action s at time t is

$$p_s^i(t) = \frac{e^{\lambda A_s^i(t)}}{\sum_{k\in S} e^{\lambda A_k^i(t)}}, \tag{2}$$

where S is the set of all strategies available to the player and $A_s^i(t)$ is the expected value, or propensity, for player i to choose $s \in S$ 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. [7] studied the dynamics of learning in coupled oscillators with delayed reinforcement. Xie et al. [8] used reinforcement of irregular spiking of neurons to study the learning in neural networks. Potapov et al. [9] 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. [10] used a group of reinforcement-learning agents to derive coupled replicator equations that describe the dynamics of collective learning in multi-agent systems. More literature regarding the use of reinforcement learning can be found in Refs. [11–15].

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 pay-offs 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 [1] so that each player on one hand plays against other players from an opposing population and on the other hand has its pay-offs compared with other players from his 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 pay-offs.

II. A NEW 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 that 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 pay-off. 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 pay-offs 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 the buyers. 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 $k \in \{0, 1, 2..., K\}$ is called an action. Besides standard Roth-Erev and Bush-Mosteller reinforcement learning, we shall also evaluate a modified Roth-Erev scheme. In Roth-Erev type learning, the propensity of player n to choose action k at time t is of the form

$$Q_{nk}(t) = Q_{nk}(1) + SR_{nk}(t-1), \tag{3}$$

where $Q_{nk}(1)$ is the initial propensity of player n to choose action k, and $SR_{nk}(t-1)$ is the rescaled reward she (he) has received from the periods up to time $t-1$ in which she (he) has chosen action k. In the most basic case $Q_{nk}(1)$ is the same for all actions. The order of magnitude of $Q_{nk}(1)$, 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 $SR_{nk}(t-1)$ as this will determine the speed of learning. In RE reinforcement, $SR_{nk}(t-1) = AR_{nk}(t-1)C_{nk}(t-1)$, where $AR_{nk}(t-1)$ and $C_{nk}(t-1)$ are the average reward received and the number of times that player n choose action k in the periods up to time $t-1$, respectively. For our purposes, we adopt the more general definition

$$SR_{nk}(t-1) = R_{nk}^{\tau} C_{nk}(t-1), \tag{4}$$

where R_{nk} is the reward (payoff) of player n when choosing action k in the periods up to time $t-1$, and τ is some non-negative real number. Since in our game there is no reward for unsuccessful interactions, we only need to count the successful ones. Hence R_{nk} is simply $K-k$ if player n is a buyer and k if she/he is a seller. The probability $P_{nk}(t)$ for player n to choose k in the new scheme then is

$$P_{nk}(t) = \frac{R_{nk}^{\tau} C_{nk}(t-1)}{\sum\limits_{i} R_{ni}^{\tau} C_{ni}(t-1)}. \tag{5}$$

The order of τ determines the learning speed. If $\tau = 0$, then the players are always exploring; if $\tau = \infty$ then a fixed action will be taken. And for $\tau = 1$, the standard RE scheme is restored. Here, we wish to find the optimal value of τ for our game. Extensive simulations indicate an optimal value of $\tau = 1.5$, which, in fact, agrees with the finding in [5]. From now on we fix the value of τ to be 1.5 and call this new reinforcement scheme NRE.

III. SIMULATIONS

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 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. 1, BM learns quite slowly. The optimal learning rate for BM is found to be around 0.0005 in our setting. With a higher learning rate, the system gets stuck in some sub-optimal equilibrium. SM learns very quickly but only leads to a sub-optimal 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 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 on NRE can achieve more favorable equilibrium values than RE can do. As we see from the plots, NRE can surpass RE after around 10000 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. 2). 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. 3. NRE is also superior to SM by driving the equilibrium value to 8 (12).

We now turn to an estimate of the learning speed for RE and NRE. Let us start from RE. Denote by $P_b(k_i, t)$ the probability of choosing action k_i at time t for buyers, and by $P_s(k_j, t)$ the probability of choosing action k_j at time t for sellers. We then have

$$P_b(k_i, t+1) = \frac{\sum_{k_j=0}^{k_i} P_b(k_i, t) P_s(k_j, t)(K - k_i)}{\sum_{k_i=0}^{K} \sum_{k_j=0}^{k_i} P_b(k_i, t) P_s(k_j, t)(K - k_i)}, \tag{6}$$

and

$$P_s(k_j, t+1) = \frac{\sum_{k_i=k_j}^{K} P_b(k_i, t) P_s(k_j, t) k_j}{\sum_{k_j=0}^{K} \sum_{k_i=k_j}^{K} P_b(k_i, t) P_s(k_j, t) k_j}. \quad (7)$$

For convenience of computation, in the above two equations 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 infinity). The sums in both equations can be replaced by integrals when K goes to ∞. The initial conditions are $P_b(k_i, 0) = P_s(k_j, 0) = 1/(K+1), k_i, k_j = 0, 1, 2..., K$. The probabilities as Eqs. (6) and (7) 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 $1/(K+1)$. 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. 2 where NRE learns more slowly than RE does in the beginning.

We now briefly analyze why BM and SM reinforcement learning do not perform well in our game. First in BM the learning never slows down. This means 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 that 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, 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 that a single peak will remain. As in the beginning the players are always exploring a little, the consequence of the double peaks is that the optimal action cannot emerge as the two major remaining actions will co-exist. If a greedy learning method is taken (by having a high temperature) 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. An efficient learning should be neither too hot (exploration) nor too cold (exploitation) [17].

We now widen the perspective and compare NRE reinforcement to other evolutionary schemes that we have studied before [1, 16]. First we introduce the 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.

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

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

2. for $m = 1$, that strategy is called 1-round opponent: each player utilizes the offer of his opponent in the most recent round

3. average-friend-opponent: the average of one's friends' opponents' bids in the most recent round (here, each player has a certain number of friends (usually small in comparison with the population size) within his own population)

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

5. average-successful-friend: the average of one's friends' successful bids 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. 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 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 the 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. 4, when the buyers choose NRE reinforcement learning and the sellers choose 1-round opponent evolutionary strategy, the buyers gain advantage by offering 8 eventually. Here, evolution proceeds much faster than NRE reinforcement: 8 has been reached for the sellers after around 10000 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 time step 100000 approximately. This also indicates that reinforcement learning is overall much slower than evolution, all other conditions being equal. We also notice that in NRE reinforcement learning the increase is very steep in the beginning but then gets flatter, as in the "Law of Practice" in psychology. We tracked the time-dependent distributions of offers for both buyers and sellers during a certain time period. The heterogeneity of buyers' actions is very significant. For the same time period as given for the buyers, the sellers who take 1-round opponent strategy are more homogeneous as the centralized (most-frequent) offer is already 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. 5). When faced with 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 [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.

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 simple average-friend-opponent strategy (Fig. 6). The reason is as before that simple averaging is more efficient in information processing. Unlike the direct information in average-previous-opponent strategy, in average-friend-opponent strategy indirect information from the friends is used. NRE reinforcement learning can beat average-all-friend strategy as well since the latter uses no information from the other population. NRE reinforcement learning ties simple average-all-friend, 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 information at all; each player chooses a fixed random offer that will be updated through the selection [16]. For 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. 7).

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 1-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 average-friend-opponent, and transitivity holds among the three. But NRE loses 4, and RE also loses 4, to simple average-successful-friend, and the transitivity fails here. The reason might be that in these two cases simple average-successful-friend strategy dominates the interactions so that reinforcement learning schemes have to adapt to the same level of equilibrium.

BM reinforcement learning is found equal to 1-round- and 2-round opponent strategy, average-friend-opponent strategy and average-all-friend strategy. BM can defeat 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 1-round opponent and there is transitivity between these three. But there is no transitivity among BM, 1-round opponent, and 2-round opponent. Because we found already 1-round opponent can defeat 2-round opponent.

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 RE, BM and SM reinforcement learning schemes. NRE reinforcement learning also wins against most evolutionary strategies evolving look-up tables but loses to the simple version of those strategies without look-up tables.It remains to evaluate the performance of this NRE reinforcement learning in other learning tasks.

Acknowledgements

W.L. would like to thank Prof. Jost for the hospitality during his stay in Leipzig where this work was done. 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] J. Jost and W. Li, Physica A **345**, 245-266 (2005).
[2] T. Hastie, R. Tibshirani and J. Friedman, *Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer-Verlag (2008).
[3] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT, Cambridge (1998).
[4] A.E. Roth and I. Erev, Games Econ. Behav. **8**, 164-212 (1995).
[5] I. Erev and A.E. Roth, The American Economic Review **88 (4)**, 848-881 (1998).
[6] R.B. Bush and F. Mosteller, Psychol. Rev. **58 (5)**, 313-323 (1951).
[7] M. A. Trevisan, S. Bouzat, I. Samengo, and G. B. Mindlin, Phys. Rev. E **72**, 011907 (2005).
[8] Xiaohui Xie and H. Sebastian Seung, Phys. Rev. E **69**, 041909 (2004).
[9] A. Potapov and M. K. Ali, Phys. Rev. E **67**, 026706 (2003).
[10] Yuzuru Sato and James P. Crutchfield, Phys. Rev. E **67**, 015206 (2003).
[11] Sabino Gadaleta and Gerhard Dangelmayr, Phys. Rev. E **63**, 036217 (2001).
[12] M. Sperl, A. Chang, N. Weber, and A. Hbler, Phys. Rev. E **59**, 3165 (1999).
[13] Shigetoshi Nara and Peter Davis, Phys. Rev. E **55**, 826 (1997).
[14] Masanori Kushibe, Yun Liu, and Junji Ohtsubo, Phys. Rev. E **53**, 4502 (1996).
[15] Dimitris Stassinopoulos and Per Bak , Phys. Rev. E **51**, 5033 (1995).
[16] J. Jost and W. Li, Adv. Com. Sys. **11 (6)**, 901-926 (2008).
[17] B. Skyrms, *Signals: Evolution, Learning, and Information*, Oxford University Press, NY (2010).

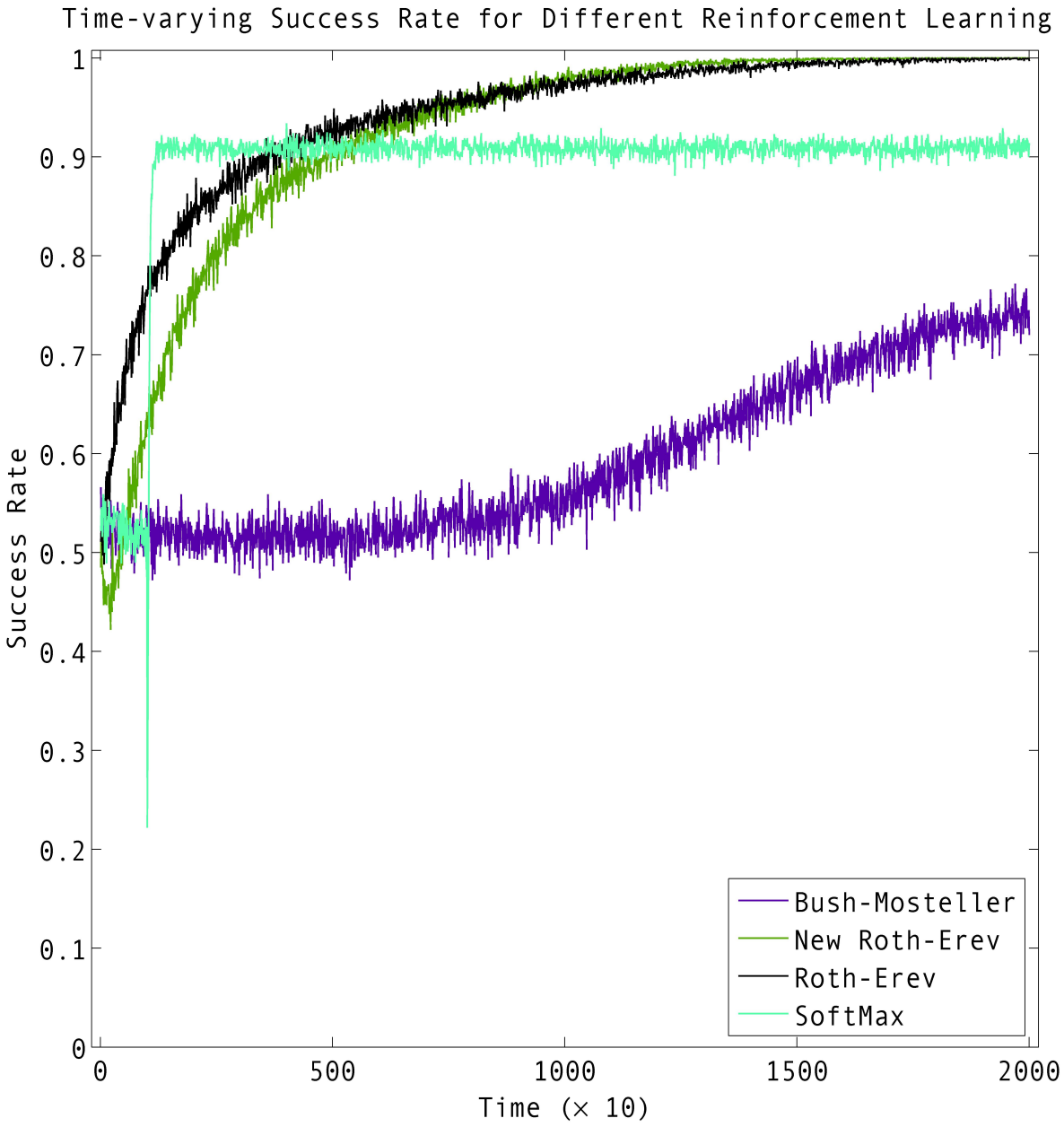

FIG. 1: The time-varying success rates for the four different reinforcement learning: Roth-Erev (RE), Bush-Mosteller (BM), SoftMax (SM), and New Roth-Erev (NRE). The parameters: RE: initial inclination: 12, forgetting rate 0.01; BM: learning rate: 0.0005; SM: temperature 3.5; NRE: initial inclination: 50, power exponent: 1.5, forgetting rate: 0.01.

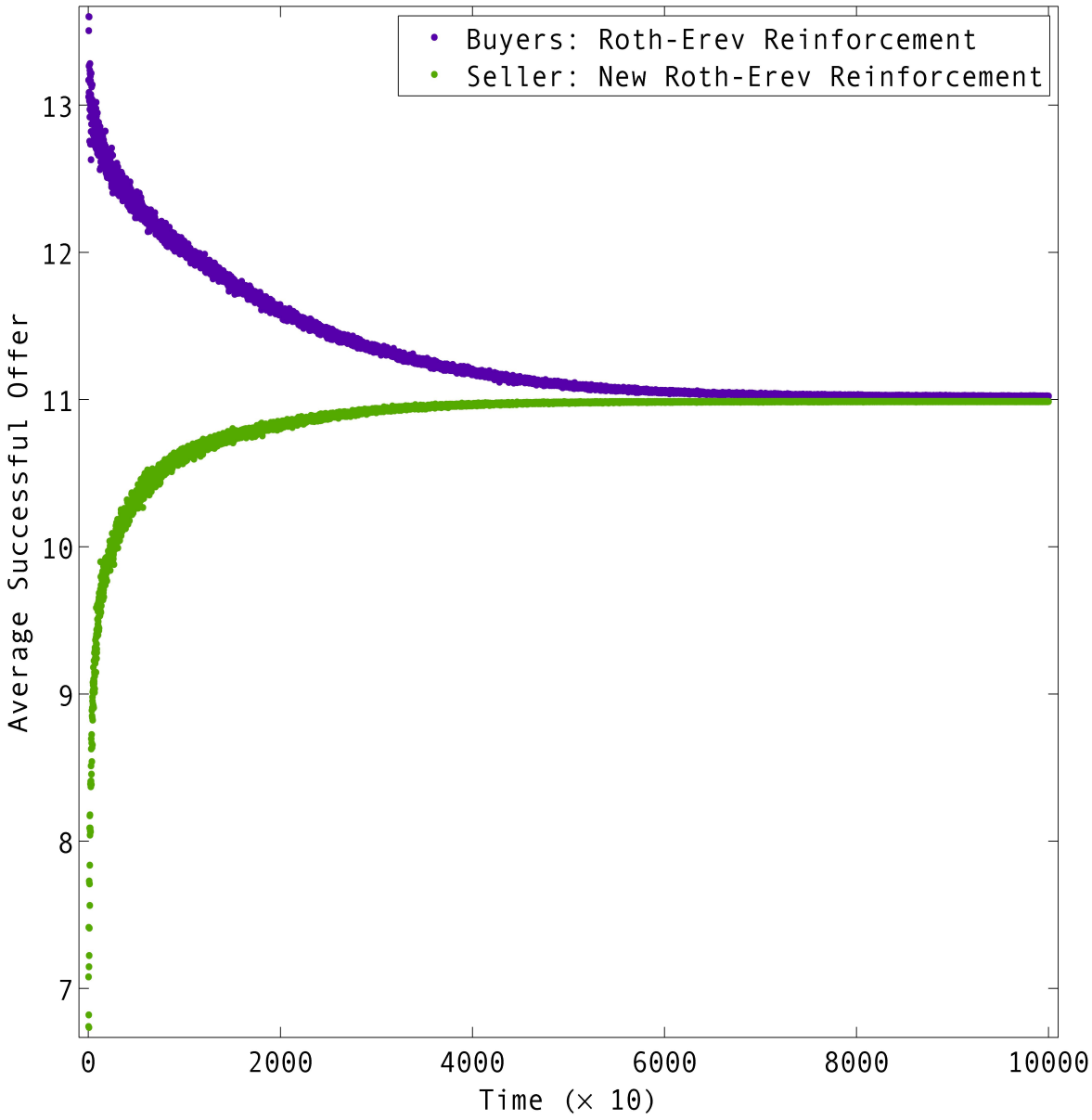

FIG. 2: NRE reinforcement gains advantage over RE reinforcement when both choose optimal parameters. When the buyers take RE and the sellers NRE, the eventual equilibrium is 11, greater than 10.

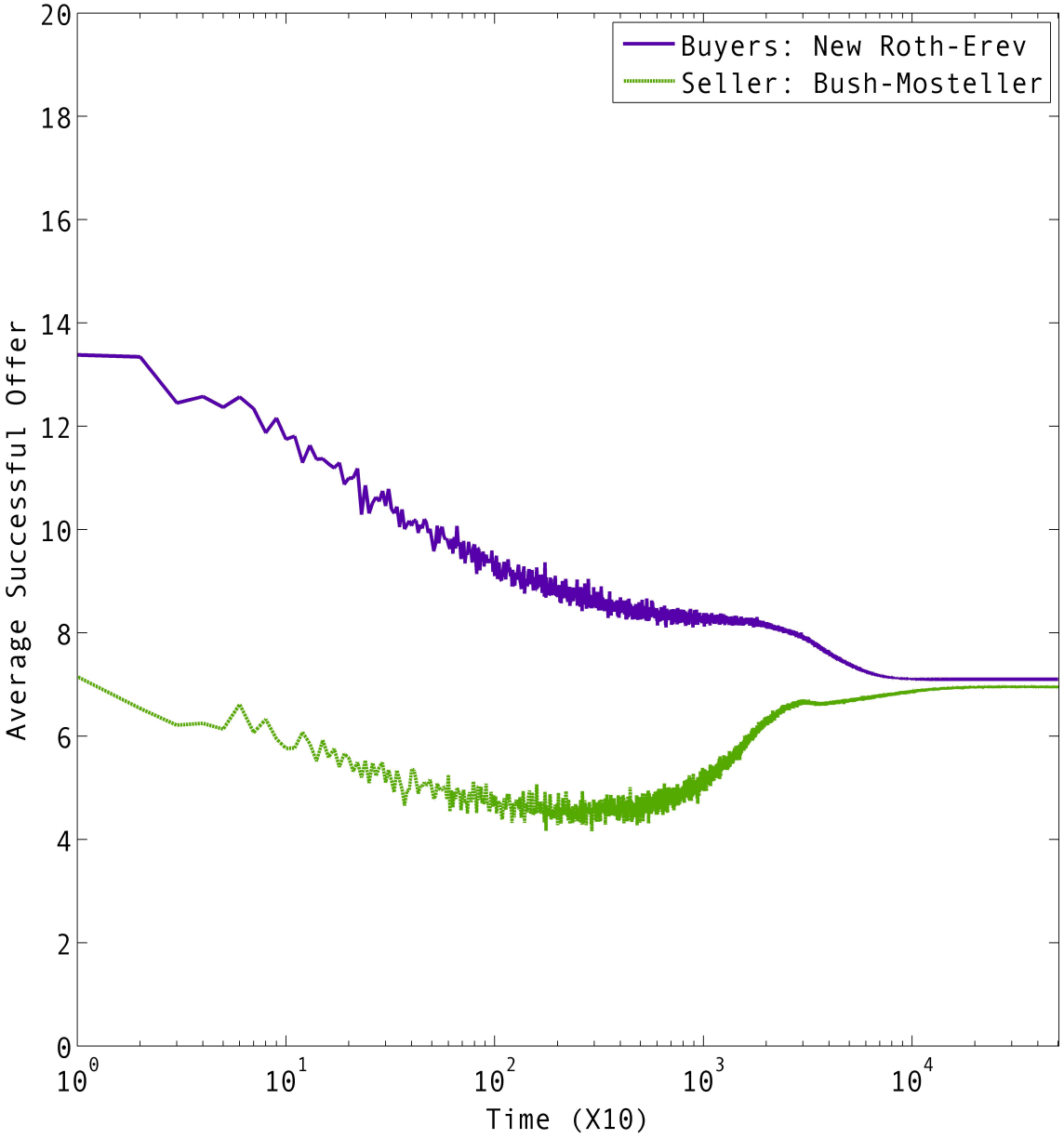

FIG. 3: NRE reinforcement gains advantage over BM reinforcement when both choose optimal parameters. When the buyers take NRE and the sellers BM, the eventual equilibrium is 7, much less than 10.

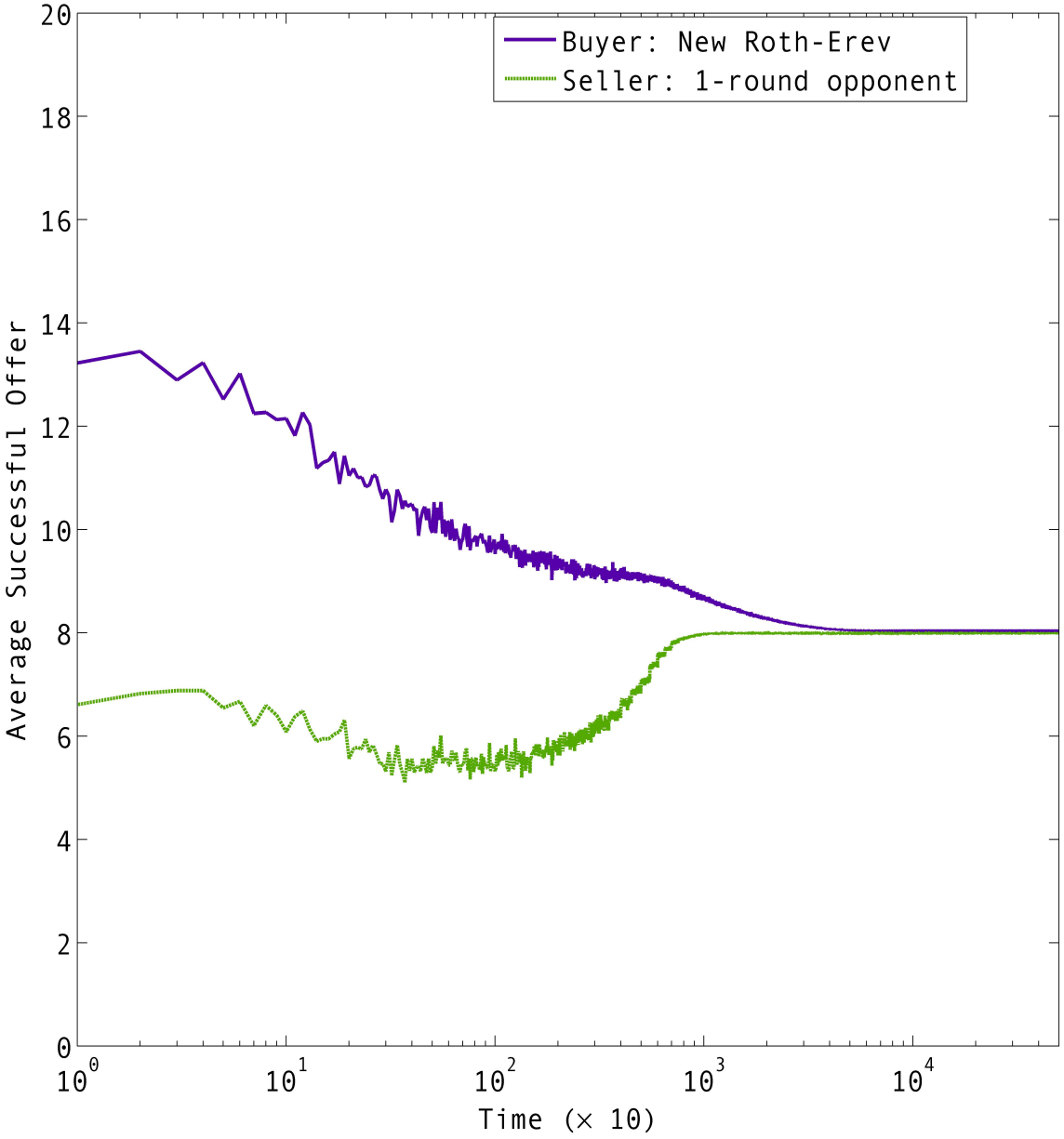

FIG. 4: NRE reinforcement gains advantage over 1-round opponent strategy when both choose optimal parameters. When the buyers take NRE and the sellers 1-round opponent, the eventual equilibrium is 8, less than 10.

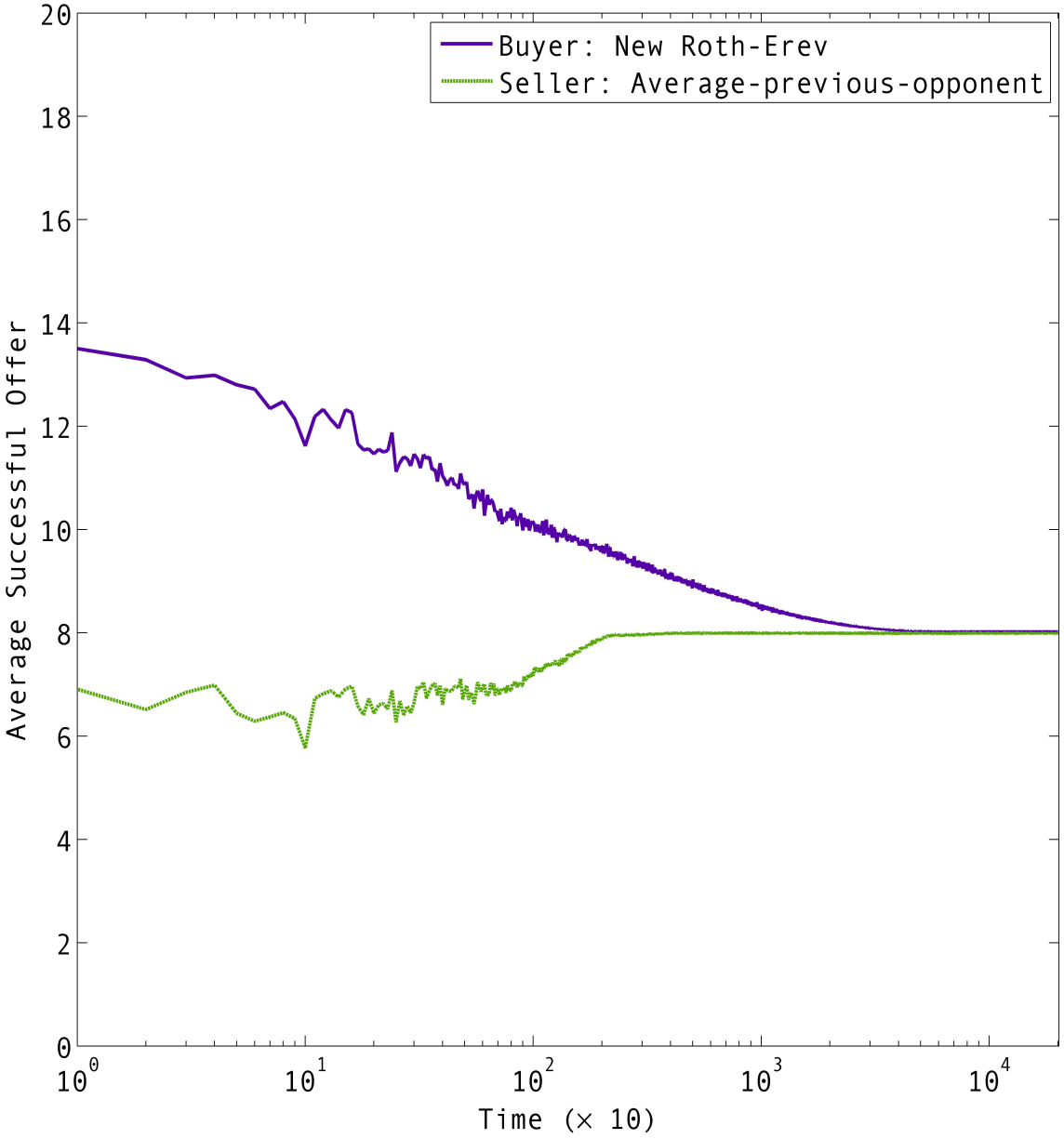

FIG. 5: NRE reinforcement gains advantage over average-previous-opponent strategy when both choose optimal parameters. When the buyers take NRE and the sellers average-previous-opponent, the eventual equilibrium is 8, less than 10.

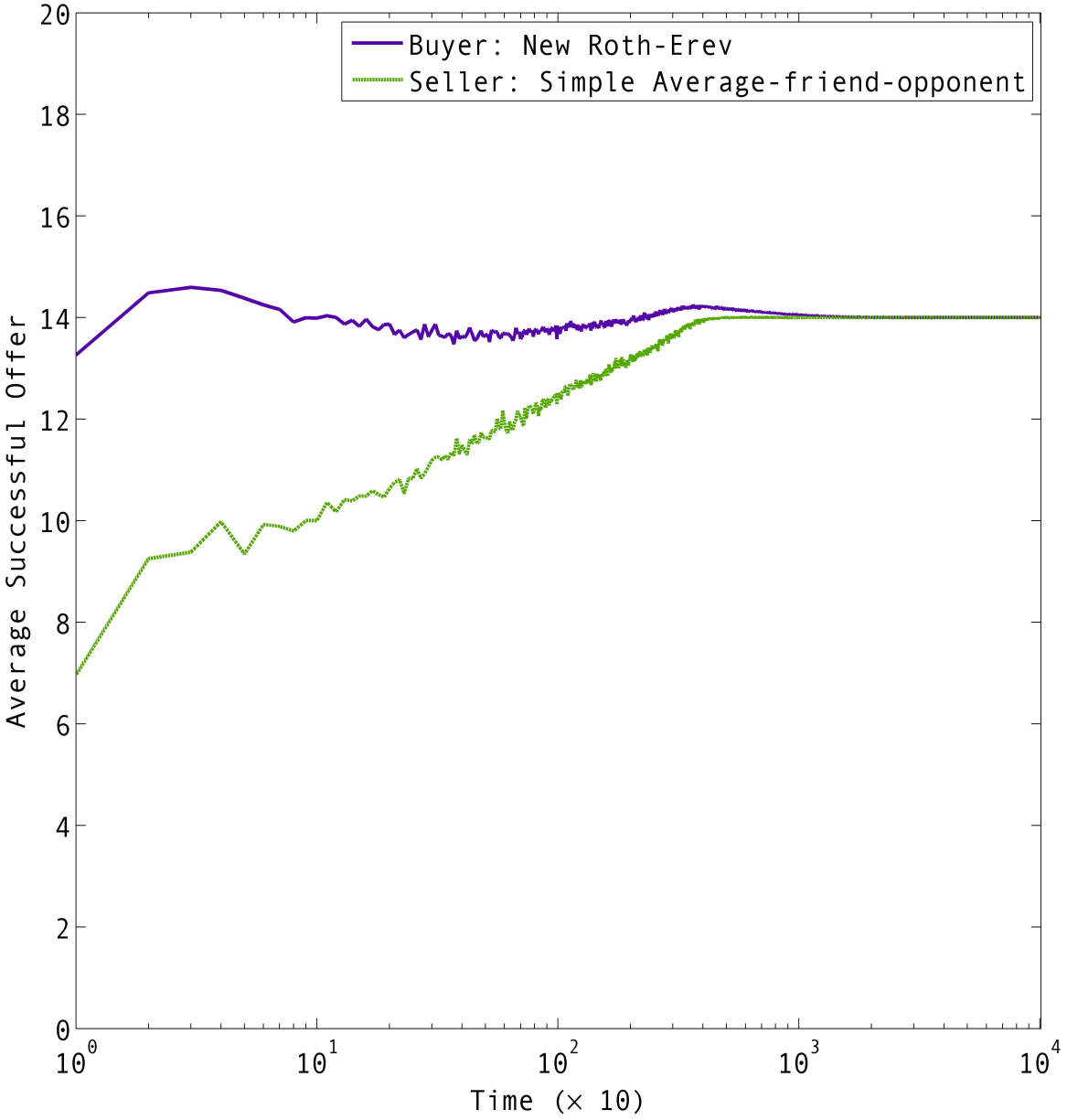

FIG. 6: NRE reinforcement loses advantage to simple average-friend-opponent strategy when both choose optimal parameters. When the buyers take NRE and the sellers simple average-friend-opponent, the eventual equilibrium is 14, much greater than 10.

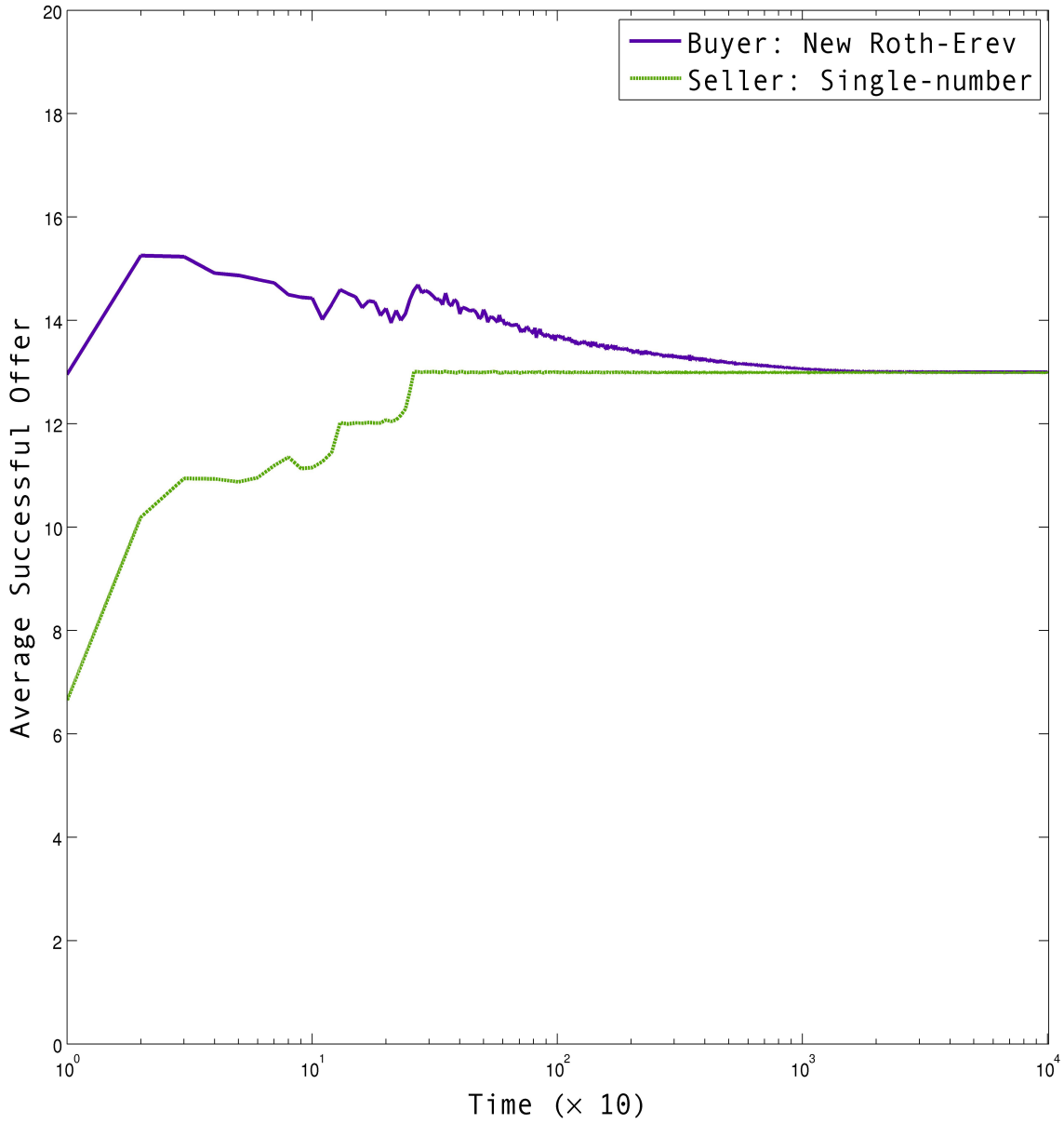

FIG. 7: NRE reinforcement loses advantage to single-number strategy when both choose optimal parameters. When the buyers take NRE and the sellers single-number, the eventual equilibrium is 13, much greater than 10.