

Asynchronous Stochastic Dynamics and the Spatial Prisoner’s Dilemma Game

Carlos Grilo¹ and Luís Correia²

¹ Dep. Eng. Informática, Escola Superior de Tecnologia e Gestão
Instituto Politécnico de Leiria

Morro do Lena, 2411-901 Leiria, Apartado 4163, Portugal
grilo@estg.ipoleiria.pt

² LabMag, Dep. Informática, Faculdade Ciências da Universidade de Lisboa
Edifício C6, Campo Grande, 1749-016 Lisboa
Luis.Correia@di.fc.ul.pt

Abstract. We argue that intermediate levels of asynchronism should be explored when one uses evolutionary games to model biological and sociological systems. Usually, only perfect synchronism and continuous asynchronism are used, assuming that it is enough to test the model under these two opposite update methods. We believe that biological and social systems lie somewhere between these two extremes and that we should inquire how the models used in these situations behave when the update method allows more than one element to be active at the same time but not necessarily all of them. Here, we use an update method called Asynchronous Stochastic Dynamics which allows us to explore intermediate levels of asynchronism and we apply it to the Spatial Prisoner’s Dilemma game. We report some results concerning the way the system changes its behaviour as the synchrony rate of the update method varies.

1 Introduction

The explanation of how cooperation could ever emerge on nature and human societies by means of natural evolution has been a difficult problem to solve [1]. Evolutionary game theory [15] has been largely used as a tool to study this problem. In this area interactions between agents are usually modeled as a game and the Prisoner’s Dilemma game is one of the most used metaphors to study the evolution of cooperation. In this game there are two possible strategies: Cooperate (C) or Defect (D). Figure 1 shows the payoff matrix of the game, where the following conditions must be met: $T > R > P > S$.

When panmictic populations are used¹ and when the players play the game just once on each encounter without remembering what happened on previous encounters, theory says that the C strategy is completely dominated by the D one, until complete extinction [15]. However, in [13] Nowak and May showed that

¹ On panmictic populations each agent can interact with any other agent in the population.

	C	D
C	R, R	S, T
D	T, S	P, P

Fig. 1. Payoff matrix for the Prisoner’s Dilemma game. The row player gets the first value of each matrix element.

cooperation can be maintained when the game is played in a two-dimensional spacial grid, in which agents can only interact with their immediate neighbours. This work was almost immediately contested in [7], the reason being that these results were only possible because a synchronous system was used which, according to the authors, is an artificial feature. In a synchronous system all the agents in the population interact and are updated exactly at the same time. Instead, the authors of this work used a continuous asynchronous updating method called *uniform choice* [14] in which, at each time step, only a randomly chosen individual, with reposition, is able to interact with its neighbours, and the reported output was that cooperation is no longer sustainable even if only a D agent exists in the initial population. Given that these results were obtained with a single combination of payoff values, Nowak and colleagues counter-answered in [12], testing several conditions, namely, different payoff values, synchronous and asynchronous (*uniform choice*) update methods, and different levels of determinism in the transition rule (see parameter m in section 3). The results show that cooperation can be maintained for many different conditions, including asynchronism, but they are presented through system snapshot images, which render difficult to measure the exact way they are affected by the modification from a synchronous to a continuous asynchronous discipline. In spite of that, this and other subsequent works [10][17] allowed spatial structure to be viewed as a feature that, in certain situations, can be beneficial to the evolution of cooperation. Also, since the criticism made in [7], it’s common to see papers [16][6][11] where both synchronous and continuous update methods are used.

In this paper we argue that both perfect synchronism and continuous asynchronism are equally artificial ways of simulating the global dynamics of a population of interacting agents. This doesn’t mean that the above mentioned practice of presenting results achieved with these two methods is not a positive one. However, it takes for granted that it is enough to test the system under these two opposite methods. We believe that biological and social systems lie somewhere between these two extremes and that we should inquire how the models used in these situations behave when the update method allows more than one element to be “active” at the same time but not necessarily all of them. In order to do that, we used one update method named *asynchronous stochastic dynamics* [4] that allows us to cover all the spectrum between synchronous and continuous updating. We applied the method to a model similar to the one used in [12] and we report some results obtained.

The paper is structured as follows: in Sect. 2 we explain why we think that synchronism and continuous asynchronism are both equally artificial and why methods like *asynchronous stochastic dynamics* should be explored when studying

populations of interacting agents. In Sect. 3 we describe the model we used in our experiments and in Sect. 4 we present and discuss some results. Finally, some conclusions are drawn and future work is advanced.

2 Asynchronous Stochastic Dynamics

Beyond the evolution of cooperation, the influence of the update method has been studied in areas like, for example, cellular automata [8][3] and evolutionary algorithms [9][5]. In some of these works, especially the ones about cellular automata and evolution of cooperation, the utilization of synchronous update methods is criticized, the argument being that the real world is not synchronous and, so we can not entirely rely on results achieved that way. As an alternative, continuous asynchronous methods are usually used (see [14] for an analysis of several asynchronous methods). This procedure corresponds to choosing small enough time intervals so that at each one exactly one element of the population can interact with its neighbours and be updated [7].

We think that continuous asynchronism when applied to biological and socio-logical environments can be considered as artificial as perfect synchronism. There are two reasons for this: the first reason is that, when we have a population of interacting agents, many interactions can be occurring at the same time. If interactions were an instantaneous phenomena we could model the dynamics of the system as if interactions occurred one after another but that is not the case. Interactions can take some time, which means that their output is not available to other ongoing interactions. Even if we consider interactions as being instantaneous, the time that information takes to be transmitted and perceived implies that interactions' consequences are not immediately available. Another reason is the determinism of continuous update methods: on what basis can we say that, at each time step, exactly 1, 2, or n elements are “active”? Even if we have an idea of the level of activity of the system being modeled, it's doubtful that it is always exactly the same. This, of course, is not a problem for evolutionary algorithms, where, given a problem to solve, the goal is to achieve the best solution as fast as possible, independently of the methods we use. But social systems are not as predictable and some sort of nondeterminism should be used.

A feasible alternative to perfect synchronism and continuous asynchronism is a method named *asynchronous stochastic dynamics* (ASD) in which, at each time step, each element of the population has a given probability $0 < \alpha \leq 1$ of being selected to interact with its neighbours, after which it is updated using a given transition rule. The α parameter is called the *synchrony rate* and is the same for all the elements of the population. After this selection procedure both the interactions and the application of the transition rule are done as if they occurred simultaneously, i.e., synchronously. The stochastic nature of the method implies that the number of selected elements may vary from time step to time step. Also, the α parameter allows us to explore intermediate levels of asynchronism. This was done, for example, in [4] for studying the robustness of elementary cellular automata to asynchronism. The authors found that some

automata are very sensitive to small changes in the α parameter. Besides, it can happen that a given automata has a similar behaviour to, for example, $\alpha = 1$ and $\alpha = 0.1$, which are values at almost opposite sides of the α domain, but has a very different behaviour say, for example, for $\alpha = 0.5$ ². This made us question if this could be the case for spatial evolutionary games, since these models resemble cellular automata in many aspects [13]. Evolutionary games are used as metaphors to model real situations. It's difficult to know exactly the update discipline of the modeled system's elements. Therefore, it's useful to know if the model is robust to changes in the update method. Even if it's robust, in the sense that it doesn't change significantly its behaviour to small changes on the update method, it can gradually change its behaviour as we change the synchrony rate. This change can be such that the system has a very different behaviour under the two extremes of the update method (synchronous vs. continuous) and the ASD method gives us the possibility of knowing how this change happens.

3 The Model

The model we used in the experiments is very similar to the one used in [12] so that the results could be compared. Agents are placed in a toroidal two dimensional grid so that each agent occupies one cell. Each time step can be divided in three stages: the activation stage, the interaction stage and the update stage.

In the first stage we decide, using the ASD method, which agents will be active at the current time step, i.e., which are the agents that will interact with their neighbours and, therefore, will be updated in the following stages. The utilization of this update method is the main modification we made to the original model. Recall that in the original model the update methods used were synchronous update and *uniform choice*. The ASD method equals synchronous update when $\alpha = 1$ and approaches *uniform choice* as $\alpha \rightarrow \frac{1}{n}$ where n is the population size.

In the second stage, the selected agents play a one round Prisoner's Dilemma game with all their 8 surrounding neighbours. This type of neighbourhood is usually called the *Moore* neighbourhood. In some works the neighbourhood is allowed to include the agent itself. This is justified by considering that each cell can represent not a single agent but a set of similar agents that may interact with each other. Here, we do not consider self-interaction since we are interested in modeling cells as individual agents.

Agents can only play C or D and the only way they can change their strategy is by way of the application of the transition rule in the third stage. It is common practice to define the game's payoff values as $R = 1$, $T = b$ ($b > 1$) and $S = P = 0$. The b parameter represents the advantage of D players over C ones when

² The measure used to compare the behaviour of an automaton under different conditions was the mean number of cells with value 1 during a given sampling period after a transient period.

these play the game with each other. Defining payoff values this way has the advantage that the game can be characterized by just one parameter.

Besides the regular two dimensional grid, we also made experiments with small-world networks (SWNs) [18] in order to verify if the results were dependent on the underlying interaction topology. We build SWNs as in [16]: first, a regular two dimensional grid is built so that each agent is linked to its 8 surrounding neighbours by undirected links; then, with probability p , each link is replaced by another one linking two randomly selected agents. Parameter p is called the rewiring probability. During the rewiring process we do not allow the creation of self links, as we do not allow self-interaction. Repeated links and disconnected graphs are also avoided. The rewiring process may create long range links connecting distant agents. For simplicity, we will call neighbours to all interconnected agents, even if they are not located at adjacent cells. By varying p from 0 to 1 we are able to build from completely regular networks to random ones. SWNs have the property that, even for very small values of the rewiring probability, the mean path length between any two nodes is much smaller than in a regular network, maintaining however a high clustering coefficient observed in many real systems including social ones.

The third stage is used to model the fact that agents tend to imitate the most successful agents they know. It can also be interpreted as the selection step of an evolutionary process in which the least successful strategies tend to be replaced by the most successful ones. This is done by synchronously applying a transition rule to the agents selected in the first stage. The transition rule used here (and also in [12]) is a generalization of the proportional update rule. Let G_i be the average payoff earned by agent i in the interaction stage, N_i be the set of neighbours of agent i , s_i be equal to 1 if i 's strategy is C and 0 otherwise, and m a positive number. The probability that in the next time step agent i adopts C as its strategy is then given by

$$p_C = \frac{\sum_{l \in N_i \cup i} s_l G_l^m}{\sum_{l \in N_i \cup i} G_l^m}. \quad (1)$$

The m parameter acts as a weight that favors the most successful neighbour's strategy B in the update process: the bigger m , the larger is the probability that i adopts B . When $m = +\infty$ we have a deterministic best neighbour rule such that i always adopts B as its next strategy. When $m = 1$ we have the proportional update rule. It can be viewed, as well, as the deterministic degree of the transition rule. We use average payoffs instead of total payoffs because the rewiring process used to build small-world topologies may result in agents having a different number of neighbours.

4 Simulations and Results

As in [12], the simulations were done with populations of $80 \times 80 = 6400$ agents. When the system is running synchronously, i.e., when $\alpha = 1$ we let the system run during a transient period of 200 iterations. After this, we let the system

run during 100 more iterations, and at the end we take as output the average proportion of cooperators during this period, which is called the sampling period. When $\alpha \neq 1$ the number of selected agents at each time step may not be equal to the size of the population and it may vary between two consecutive time steps. In order to guarantee that the runs with $\alpha \neq 1$ are equivalent to the synchronous ones in what concerns to the total number of individual updates, we let the system first run until $200 * 6400 = 1280000$ individual updates have been done. After this, we sample the proportion of cooperators during $100 * 6400 = 640000$ individual updates and we average by the number of time steps needed to do these updates.

Each point in the charts of Fig. 2 is the result for a combination of the b , m and α values and $p = 0$ (regular grid), averaged over 30 runs. We used 10 different b values varying from 1.02 to 1.7. α values vary from 1 to 0.1 by steps of 0.1. For the m parameter we used values $+\infty$, 100, 10, 8, 6, 4, 2 and 1.

In [4] a given cellular automata is said to be robust if the average density of 1's in the sampling period doesn't change by more than 0.1 when the synchrony rate α parameter is changed by a small value (they also change α by 0.1 steps). If we take these same values, but using the frequency of cooperators instead of the density of 1's as the measure of interest, the first conclusion we can derive from Fig. 2 is that, in general, the system is robust to small changes of the α value. There are, however, some situations of non-robustness: when $m = +\infty$ ($b = 1.15$, $b = 1.35$ and $b = 1.61$) and $m = 100$ ($b = 1.55$ and $b = 1.61$) and several when $m = 2$ and $m = 1$. The big jumps that can be observed for the $m = 2$ and $m = 1$ cases are due to the large difference in the frequency of cooperators for opposite values of α : in order to get from one point to the other, big jumps must be made. This is not the real justification when $b = 1.02$. In this case we can see that the real reason for the non-robustness is that the frequency of cooperators doesn't change uniformly as we change the α value: when we move from $\alpha = 1$ to $\alpha = 0.1$ the frequency of cooperators grows very quickly until it stabilizes in the maximum value. This means that, for this particular combination of the m and b values, cooperation hegemony is the dominant result for a significant fraction of the α domain.

Another result that we can derive from the charts is that, as we change the α parameter, the variation of the frequency of cooperators is not always monotonic. For example, there are some situations in which, when we move from $\alpha = 1$ to $\alpha = 0.1$, the frequency of cooperators first decreases but then, at some point, it starts increasing. The most significant of these situations happens for ($m = 8$, $b = 1.35$) and ($m = 6$, $b = 1.3$). In the first situation, the difference in the frequency of cooperators obtained with $\alpha = 1$ and $\alpha = 0.1$ is 0.022, but the difference of the values obtained with $\alpha = 1$ and $\alpha = 0.5$ is 0.231. Excepting these situations, which happen for a relatively low value of m , non-monotonicity happens mainly for large values of m , that is, when the probability that an agent imitates its most successful neighbour is high. Nevertheless, we can say that, in general, the system responds monotonically as we change α from one extremity to the other.

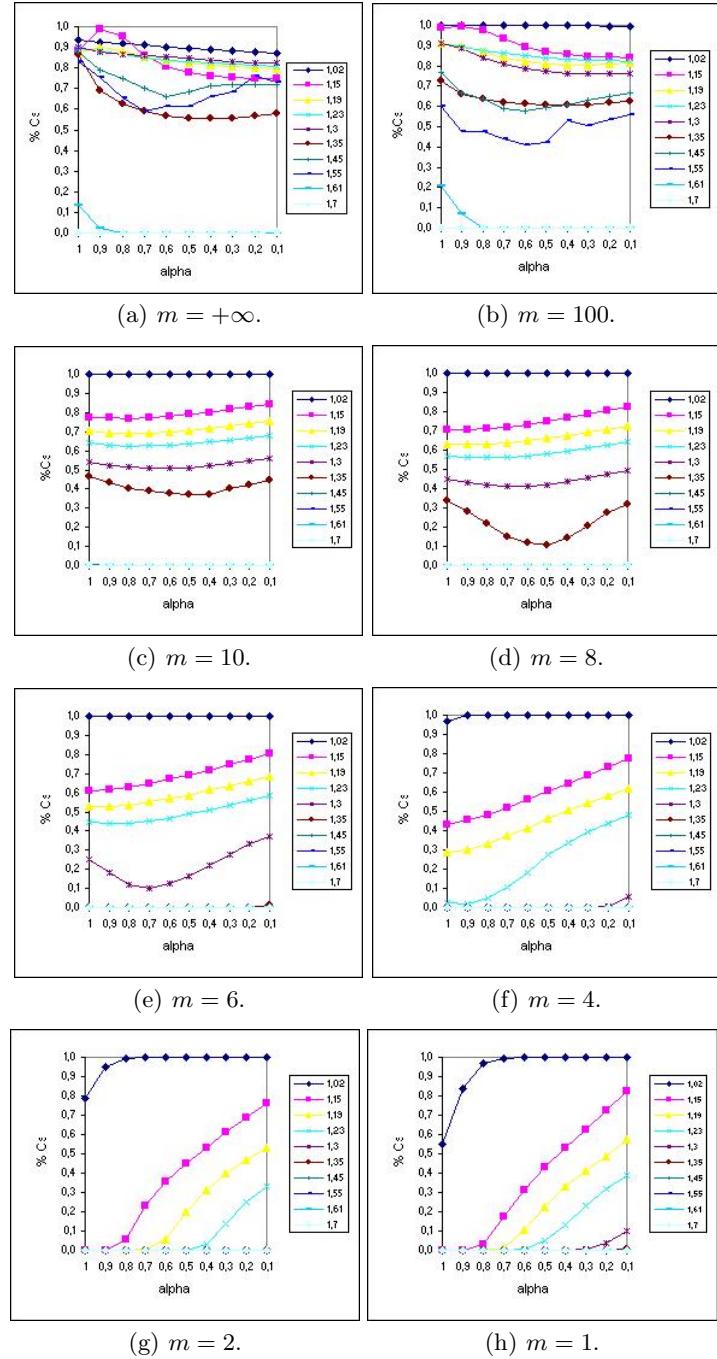


Fig. 2. % of cooperators for $p = 0$ and different combinations of m , b and α

There is a somewhat unexpected result that we can also derive by looking at the charts. It can be phrased like this: the lower the value of the m parameter, the more is cooperation favoured when we decrease the value of α . That is, for high values of m , there is a general tendency for a decrease in the frequency of cooperators as we decrease the value of α . But, as we decrease the value of m , lower values of α become more beneficial to the emergence of cooperation. In the first set of simulations we have done, we used only the values $+\infty$, 100, 10, and 1 as in [12]. But, after verifying the behaviour of the system for $m = 1$, we decided to experiment with some more m values between 1 and 10 in order to confirm if this was a consistent behaviour, which in fact is. We will call to this phenomenon the *small determinism degree and small synchrony rate* phenomenon.

As we said before, besides the regular grid, we also did some simulations with SWNs in order to verify how much the results were dependent of the underlying interaction topology and how could they change as we change the rewiring probability. We used the following p values: 0 (regular grid), 0.01, 0.05, 0.1 and 1.0 (random network). Due to space restrictions, we only show the results achieved with $p = 0.1$ and $p = 1$ (Figs. 3 and 4 in the Appendix). For a start, we may observe that, as we increase the rewiring probability, the system progressively becomes more robust to changes in α . For example, when $p = 0.1$, only for the cases $(m = 4, b = 1.02)$, $(m = 1, b = 1.02)$ and $(m = 2, b = 1.02)$ we have jumps larger than 0.1. In what concerns monotonicity, we also observe that the system becomes progressively more monotonous as the rewiring probability is increased. When $p = 0.1$ some cases of non-monotonicity remain but, excepting two situations ($(m = +\infty, b = 1.61)$ and $(m = 100, b = 1.61)$), the difference between the largest value and the lower one is very small. For $p = 1$, non-monotonicity doesn't exist.

As to the *small determinism degree and small synchrony rate* phenomenon, the first thing we can conclude is that, not only it appears in all the tested topologies, but also that, as p is increased, its effects start to appear earlier, i.e, for larger values of m . For example, for $p = 1$, the combination $(m = +\infty, b = 1.02)$ is the only one where the frequency of cooperators decreases as we decrease the value of α . Also, as p is increased, cooperation becomes possible for an increasing part of the b domain. As to this, we would like to stress that there are many combinations (m, b) for which cooperation is not present when $\alpha = 1$ but for which it is present when $\alpha = 0.1$. Once more, the utilization of the ASD method allows us to analyze how this change happens. One of these cases occurs for $(p = 0.1, m = +\infty, b = 1.7)$, where cooperation becomes possible only when α is approaching 0.1. However, for the most part of these cases cooperation is made possible as soon as we depart from the synchronous discipline. This may suggest that the existence of some degree of cooperation is the most probable result in these situations since it exists for almost the entire α domain. These results also show that, for the most part of the parameters' combinations, synchronism renders difficult the emergence of cooperation, which may have some relevant social consequences. For example, this may suggest that two negotiators submitting

their proposals by the way of a mediator will have more difficulty at arriving to an agreement than if they do it asynchronously.

5 Conclusion and Future Work

In this work we argued that intermediate levels of asynchronism should be explored in the study of models such as spacial evolutionary games, since we believe that real systems, and specially sociological ones, lie somewhere between perfect synchronism and continuous updating. We used an update method called *asynchronous stochastic dynamics* which allows the exploration of all the space between these two extremes and applied it to spacial versions of the Prisoner's Dilemma game. This method allows us derive some results concerning, for example, robustness and monotonicity of the system that are not possible to derive if only synchronous and continuous updating are used. We found that, in general, the spacial Prisoner's Dilemma game responds robustly and monotonically to changes in the synchrony rate. This behaviour is usually taken for granted but results obtained in recent works on elementary cellular automata systems, with which spatial evolutionary games have many resemblances, show that the behaviour of these systems may be very different on intermediate levels of the update discipline. This method also allows us to analyze how the synchrony rate affects the system behaviour as other system parameters change. We found, for example, that lower values of the synchrony rate become more beneficial to the evolution of cooperation as the level of determinism of the generalized proportional update transition rule diminishes. Finally, in situations in which some degree of cooperation exists under one of the extremes of the update rule but not under the other, this method allows us to understand how the change from one type of result to the other happens and what is the most common output throughout the synchrony rate spectrum.

One of our first future extensions to this work will be to explore the ASD method with other very used games in order to verify if some of the results achieved with the Prisoner's Dilemma game as, for example, the *small determinism degree and small synchrony rate* phenomenon, can be generalized. The results achieved in [16] with the Snowdrift game, where the best-neighbour (equivalent to $m = +\infty$) and the simple proportional update ($m = 1$) transition rules, as well as synchronous and continuous updating were used, seem to indicate that this is the case. However, only by exploring intermediate levels of asynchronism and intermediate levels of determinism of the transition rule we can confirm this. We also plan to use scale free networks [2] as the underlying interaction topology. This will allow us to compare, for example, the robustness of other types of topologies to changes in the synchrony rate.

References

1. Axelrod, R.: *The Evolution of Cooperation*. Penguin Books (1984)
2. Barabasi, A.-L., Albert, R.: Emergence of scaling in random networks. *Science* 286, 509 (1999)

3. Bersini, H., Detours, V.: Asynchrony induces stability in cellular automata based models. In: Maes, P., Brooks, R. (eds.) *Proceedings of the Artificial Life IV Conference*, pp. 382–387. MIT Press, Cambridge (1994)
4. Fatés, N.A., Morvan, M.: An experimental study of robustness to asynchronism for elementary cellular automata. *Complex Systems* 16(1), 1–27 (2005)
5. Giacobini, M.: *Artificial Evolution on Network Structures: How Time and Space Influence Dynamics*. PhD thesis, Université de Lausanne, Switzerland (2005)
6. Hauert, C., Doebeli, M.: Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature* 428, 643–646 (2004)
7. Huberman, B., Glance, N.: Evolutionary games and computer simulations. *Proceedings of the National Academy of Sciences USA* 90, 7716–7718 (1993)
8. Ingerson, T.E., Buvel, R.L.: Structure in asynchronous cellular automata. *Physica D Nonlinear Phenomena* 10, 59–68 (1984)
9. De Jong, K.A., Sarma, J.: Generation gaps revisited. In: Whitley, L.D. (ed.) *Foundations of Genetic Algorithms 2*, pp. 19–28. Morgan Kaufmann, San Mateo, CA (1993)
10. Killingback, T., Doebeli, M.: Spatial Evolutionary Game Theory: Hawks and Doves Revisited. *Royal Society of London Proceedings Series B* 263, 1135–1144 (1996)
11. Luthi, L., Giacobini, M., Tomassini, M.: Synchronous and asynchronous network evolution in a population of stubborn prisoners. In: Kendall, G., Lucas, S. (eds.) *IEEE Symposium on Computational Intelligence and Games*, pp. 225–232 (2005)
12. Nowak, M., Bonhoeffer, S., May, R.M.: More spatial games. *International Journal of Bifurcation and Chaos* 4(1), 33–56 (1994)
13. Nowak, M.A., May, R.M.: Evolutionary games and spatial chaos. *Nature* 359, 826–829 (1992)
14. Schönfich, B., de Roos, A.: Synchronous and asynchronous updating in cellular automata. *BioSystems* 51(3), 123–143 (1999)
15. Smith, J.M.: *Evolution and the Theory of Games*. Cambridge University Press, Cambridge (1982)
16. Tomassini, M., Luthi, L., Giacobini, M.: Hawks and doves on small-world networks. *Physical Review E* 73(1), 016132 (2006)
17. Vainstein, M.H., Arenzon, J.J.: Disordered environments in spatial games. *Physical Review E* 64, 051905 (2001)
18. Watts, D., Strogatz, S.H.: Collective dynamics of small-world networks. *Nature* 393, 440–442 (1998)

A Results for $p = 0.1$ and $p = 1$

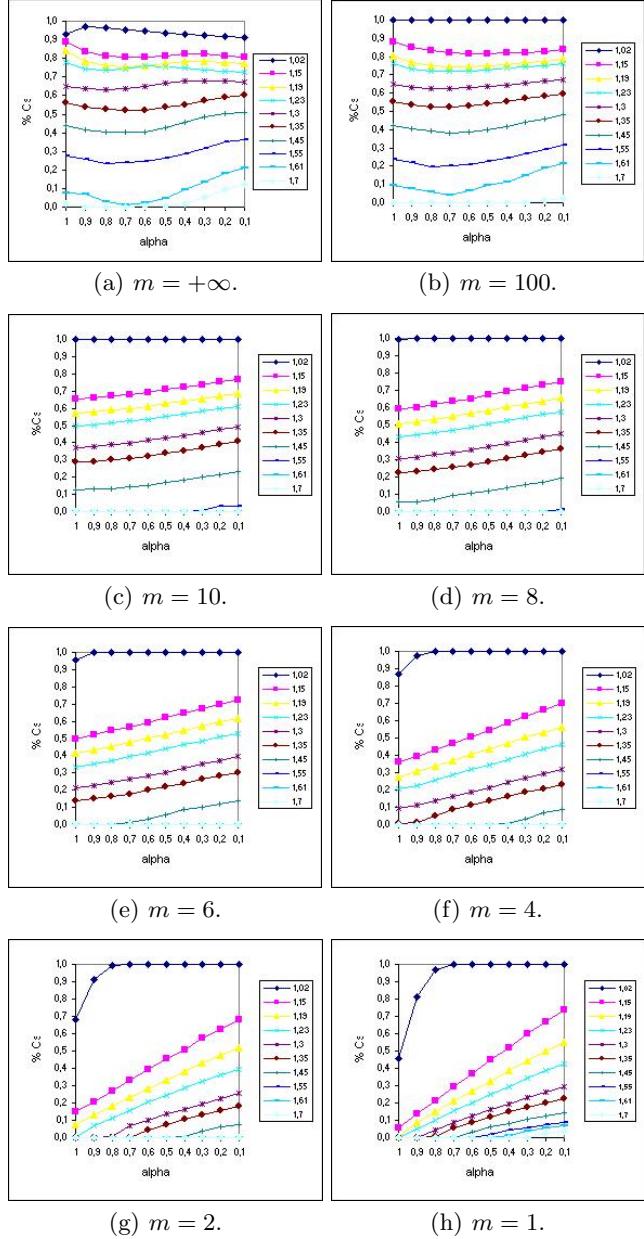
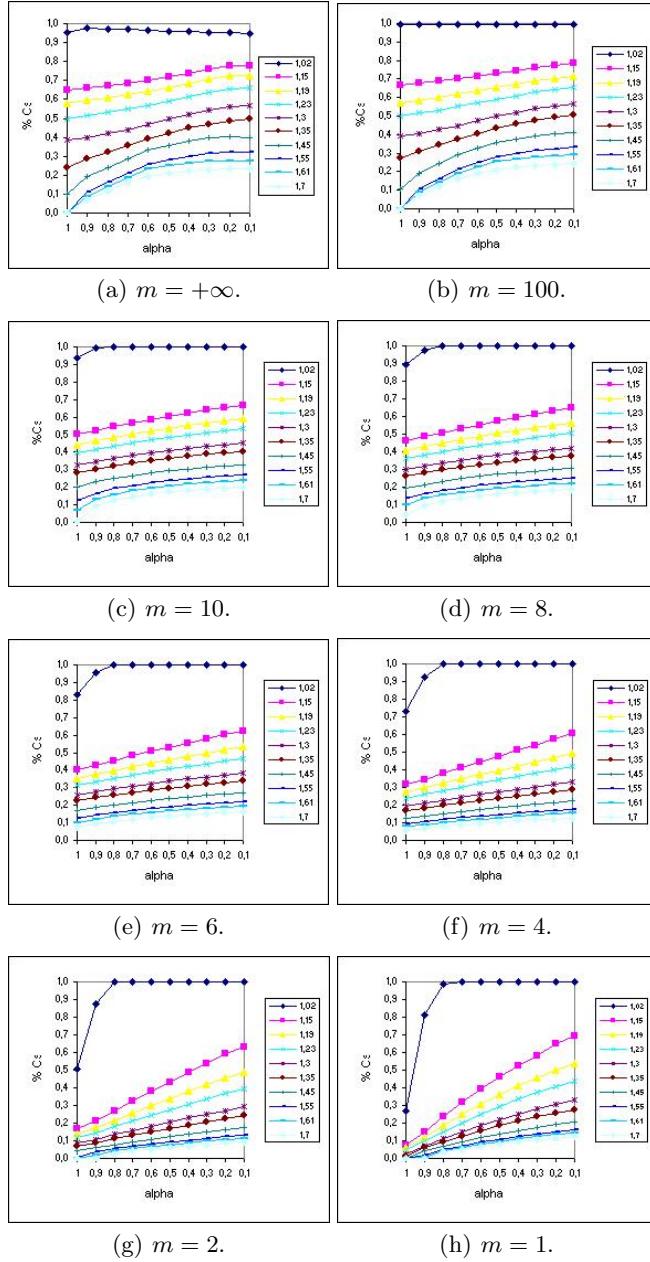


Fig. 3. % of cooperators for $p = 0.1$ and different combinations of m , b and α

**Fig. 4.** % of cooperators for $p = 1.0$ and different combinations of m , b and α