# What Makes the Spatial Prisoner's Dilemma Game Sensitive to Asynchronism?

Carlos Grilo[1,2], Luís Correia[2]

[1]Dep. Eng. Informática, Escola Superior de Tecnologia e Gestão
Instituto Politécnico de Leiria
[2]LabMAg, Dep. Informática, Faculdade Ciências da Universidade de Lisboa
grilo@estg.ipleiria.pt

### Abstract

We investigate aspects that control the Spatial Prisoner's Dilemma game sensitivity to the synchrony rate of the model. Based on simulations done with the *generalized proportional* and the *replicator dynamics* transition rules, we conclude that the sensitivity of the game to the synchrony rate depends almost exclusively on the transition rule used to model the strategy update by the agents. We then identify the features of these transition rules that are responsible for the sensitivity of the game. The results show that the Spatial Prisoner's Dilemma game becomes more and more sensitive for noise levels above a given noise threshold. Below this threshold, the game is robust to the noise level and its robustness even slightly grows, compared to the imitate the best strategy, if a small amount of noise is present in the strategy update process.

## Introduction

Spatial evolutionary games are used as models to study, for example, how cooperation could ever emerge in nature and human societies (Smith, 1982). They are also used as models to study how cooperation can be promoted and sustained in artificial societies (Oh, 2001). In these models, a structured population of agents interacts during several time steps through a given *game* which is used as a metaphor for the type of interaction that is being studied. The population is structured in the sense that each agent can only interact with its neighbors. The underlying structure that defines who interacts with whom is called the *interaction topology*. After each interaction session, some or all the agents, depending on the *update dynamics* used, have the possibility of changing their strategies. This is done using a so called *transition rule* that models the fact that agents tend to adapt their behavior to the context in which they live by imitating 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.

The discussion about using synchronous or asynchronous dynamics on these models started with a paper by Huberman and Glance (1993). Synchronous dynamics means that, at each time step, the revision of strategies happens for all agents simultaneously, while this is not the case for asynchronous dynamics. In that paper the authors contested the results achieved by Nowak and May (1992) who showed that cooperation can be maintained when the Prisoner's Dilemma game is played on a regular 2-dimensional grid by agents which do not remember their neighbors' past actions. Huberman and Glance criticized the fact that the model used in (Nowak and May, 1992) was a synchronous one, which is an artificial feature. They also presented the results of simulations where cooperation was no longer sustainable when an asynchronous dynamics were used. After this work, Nowak et al. (1994) tested their model under several conditions, including synchronous and asynchronous dynamics and showed that cooperation can be maintained for many different conditions, including asynchronism. However, the results are presented through system snapshot images, which render it difficult to measure the way they are affected by the modification from synchronous to asynchronous dynamics. Recently, in (Newth and Cornforth, 2007), a similar scenario was studied using various asynchronous update methods besides synchronous dynamics. The authors found that the synchronous updating scheme supports more cooperators than the asynchronous ones.

On the contrary, in (Grilo and Correia, 2007) we found that, in the Spatial Prisoner's Dilemma game, asynchronous updating supports, in general, more cooperators than synchronous updating. This conclusion was only possible because a large number of conditions was tested. Namely, we used small-world networks as interaction topologies so that the whole spectrum between regular and random networks could be explored. We also used the *generalized proportional* transition rule (see Section III), which allows us to tune the level of noise present in the strategy update process. We consider that there is noise when an agent fails to imitate the strategy of its most successful neighbor. We found that asynchronous updating is detrimental for cooperation only for very small noise values. That is, for the majority of the noise domain, asynchronous updating benefits cooperation. Also, as we go from regular to random networks, asynchronous updating becomes beneficial to cooperation even for very small noise values. In (Grilo and Correia, 2008) we showed that the conclusions do not change if scale-free networks (Barabasi and Albert, 1999) are used. We also showed that the final outcome of the model is basically the same whether a deterministic or a stochastic asynchronous dynamics is used, which is in contrast with results reported in (Gershenson, 2002) for random boolean networks.

The proportion of cooperating agents eventually achieved in a spatial evolutionary game can be influenced by, for example, the game that is being used, the interaction topology, the transition rule or the update dynamics. The influence of some of these aspects has previously been studied. For example, in (Pacheco and Santos, 2005) the influence of the interaction topology is examined. Also, in (Tomassini et al., 2006) the influence of the interaction topology, the transition rule and the update dynamics in the Hawk-Dove game are studied.

But, as far as we know, prior to this work, there has been no explanation of the influence of the update dynamics in the outcome of spatial evolutionary games. This work is a step in that direction. Here, we identify the aspects that control the Spatial Prisoner's Dilemma game sensitivity to asynchronism. Based on previous simulations performed with the *generalized proportional* transition rule and new ones done with the *replicator dynamics* transition rule, we first conclude that the sensitivity of the Spatial Prisoner's Dilemma game to asynchronism depends almost exclusively on the transition rule. We then identify the features of these transition rules that are responsible for the sensitivity of the game.

The paper is structured as follows: in Section II we describe the model used in our simulations. In Section III we first compare the results achieved with the *generalized proportional* and the *replicator dynamics* transition rules and then we identify the features of these rules that influence the sensitivity of the model to asynchronism. Finally, in Section IV some conclusions are drawn and future work is advanced.

## The Model

### The Prisoner's Dilemma Game

In the Prisoner's dilemma game (PD), players can cooperate (C) or defect (D). The payoffs are the following: R to each player if they both play C; P to each if they both play D; T and S if one plays D and the other C, respectively. These values must obey $T > R > P > S$ and $2R > T+S$. It follows that there is a strong temptation to play D. But, if both play D, which is the rational choice or the Nash equilibrium of the game, both get less payoff than if they both play C, hence the dilemma. For practical reasons, the payoffs are usually defined as $R = 1, T = b > 1$ and $S = P = 0$, where $b$ represents the advantage of D players over C ones when they play the game with each other. This has the advantage that the game can be described by only one parameter without losing its essence (Nowak et al., 1994).

### Interaction Topology

We use *small-world networks* (SWNs) (Watts and Strogatz, 1998) as the interaction topology. We build SWNs as in (Tomassini et al., 2006): first, a toroidal regular 2-dimensional grid is built so that each node is linked to its 8 surrounding neighbors by undirected links; then, with probability $\phi$, each link is replaced by another one linking two randomly selected nodes. Parameter $\phi$ is called the *rewiring probability*. Some works (Nowak et al., 1994) allow self-links because it is considered that each node can represent not a single agent but a set of similar agents that may interact with each other. Here, we do not allow self-interaction since we are interested in modeling nodes as individual agents. Repeated links and disconnected graphs are also avoided. The rewiring process may create long range links connecting distant agents. For simplicity, we will refer to interconnected agents as neighbors, even if they are not located at adjacent nodes. By varying $\phi$ 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 average 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.

### Interaction and Strategy Update Dynamics

On each time step, agents first play a one round PD game with all their neighbors. Agents are pure strategists which can only play C or D. After this interaction stage, each agent updates its strategy with probability $\alpha$ using a *transition rule* (see next section) that takes into account the payoff of the agent's neighbors. The update is done synchronously by all the agents selected to engage in this revision process. The $\alpha$ parameter is called the *synchrony rate* and is the same for all agents. This type of update dynamics is called *asynchronous stochastic dynamics* (Fatès and Morvan, 2005). It allows us to cover all the spectrum between synchronous and sequential dynamics. When $\alpha = 1$ we have a synchronous model, where all the agents update at the same time. As $\alpha \to \frac{1}{n}$, where $n$ is the population size, the model approaches sequential dynamics, where exactly one agent updates its strategy at each time step.

Asynchronous stochastic dynamics models the fact that, at each moment, more than one agent, but not necessarily all of them, can update their strategy. Usually, asynchronism is understood as sequential dynamics. As an example, in all the works mentioned above, asynchronous dynamics means sequential updating. However, the reality seems to lie somewhere between synchronism and sequentiality and, so, both types of dynamics can be considered as artificial. In a population of interacting agents, many decision processes can occur at the same time but not necessarily involving all the agents. If these were instantaneous phenomena we could model the dynamics of the system as if they occurred one after another but that is not usually the case. These processes can take some time, which means that their output is not available to other ongoing decision processes. Even if we consider them as being instantaneous, the time that information takes to be transmitted and perceived implies that their consequences are not immediately available to other agents. Asynchronous stochastic dynamics also models the fact that, at each time step, the number of agents updating their strategy is not always the same, which is a reasonable assumption. With this type of dynamics, this number follows a binomial distribution with mean $\alpha$. Apart from these considerations, as we will see in the following sections, the fact that the $\alpha$ parameter allows us to explore intermediate levels of asynchronism is also useful in the analysis of the influence of this feature.

### Simulations Setup

All the simulations were performed with populations of $50 \times 50 = 2500$ agents, randomly initialized with 50% of Cs and 50% of Ds. When the system is running synchronously, i.e., when $\alpha = 1$, we let it first run during a period of 900 iterations which, we confirmed, is enough to pass the transient period of the evolutionary process. After this, we let the system run for 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 these runs are equivalent to the synchronous ones in what concerns to the total number of individual updates, we let the system first run until $900 \times 2500$ individual updates have been done. After this, we sample the proportion of cooperators during more $100 \times 2500$ individual updates and we average it by the number of time steps needed to do these updates. For each combination, 30 runs were made and the average of these runs is taken as the output.

## Simulation Results

In our first simulations (Grilo and Correia, 2007, 2008), we used, a generalization of the *proportional* transition rule (GP) proposed in (Nowak et al., 1994). Let $G_x$ be the average payoff earned by agent $x$, $N_x$ be the set of neighbors of $x$ and $c_x$ be equal to 1 if $x$'s strategy is C and 0 otherwise. According to this rule, the probability that an agent $x$ adopts C as its next strategy is

$$p_C(x, K) = \frac{\sum_{i \in N_x \cup x} c_i (G_i)^{\frac{1}{K}}}{\sum_{i \in N_x \cup x} (G_i)^{\frac{1}{K}}}, \quad (1)$$

| Parameter | Values |
|---|---|
| $\phi$ | 0 (reg.), 1 (rand.), SW: 0.01, 0.05, 0.1 |
| $\alpha$ | 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1 |
| $b$ | 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2 |
| $K$ | 0, 1/100, 1/10, 1/8, 1/6, 1/4, 1/2, 1 |

Table 1: Parameter values used in the simulations.

where $K \in ]0, +\infty[$ can be viewed as the noise present in the strategy update process. Noise is present in this process if there is some possibility that an agent imitates strategies other than the one used by its most successful neighbor. Small noise values favor the choice of the most successful neighbors' strategies. Also, as noise diminishes, the probability of imitating an agent with a lower payoff becomes smaller. When $K \to 0$ we have a deterministic best-neighbor rule such that $i$ always adopts the best neighbor's strategy. When $K = 1$ we have a simple proportional update rule. Finally, for $K \to +\infty$ we have random drift where payoffs play no role in the decision process. For the moment, our analysis considers only the interval $K \in ]0, 1]$. In this interval the decision process is strongly guided by the payoffs earned by the agents.

Each simulation is a combination of the $\phi$, $\alpha$, $b$ and $K$ parameters, and all the possible combinations of the values shown in Table 1 were tested. As Fig. 1 illustrates, when the GP rule is used, in situations where both cooperation and defection coexist, the level of cooperation can change significantly as we change $\alpha$. For given $\alpha$ and $b$ values, the levels of cooperation may be different when distinct $\phi$ and $K$ values are used. Also, the exact way how the model reacts to $\alpha$ changes may change as well. However, no matter the $\phi$ and $K$ values used, there is a common qualitative behavior: the model is sensitive to changes in the synchrony rate $\alpha$. Due to this and space limitations we only show results for $\phi = 0.1$, inside the small world regime.

After experimenting with the GP rule, we also ran simulations with one of the most popular transition rules, the *replicator dynamics* rule (RD) (Hofbauer and Sigmund, 1998), which, when used on structured populations, is defined in the following way (Tomassini et al., 2006): the probability $p(s_x \to s_y)$ that an agent $x$, with strategy $s_x$ and average payoff $G_x$, imitates a randomly chosen neighbor $y$, with strategy $s_y$ and average payoff $G_y$, is equal to:

$$p(s_x \to s_y) = f(G_y - G_x) = \begin{cases} \frac{G_y - G_x}{b} & \text{if } G_y - G_x > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where $b$ is the largest possible payoff difference between two players in a one shot PD game. As Fig. 2 illustrates,

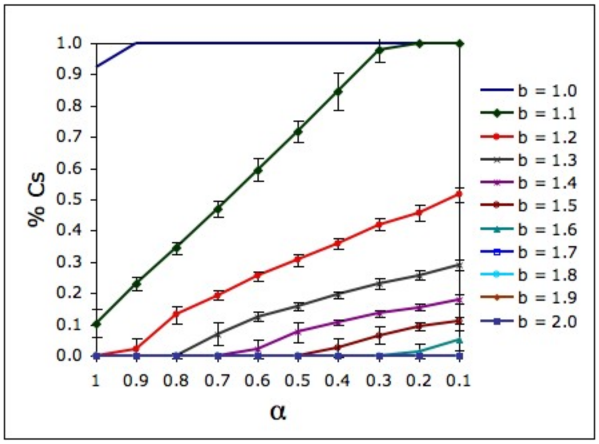


Figure 1: % of cooperators for $\phi = 0.1$ and $K = 1$ (GP rule).

when the RD rule is used, the level of cooperation is approximately constant as we change the synchrony rate $\alpha$. As for the GP rule, the qualitative behavior of the model does not change no matter the interaction topology used.

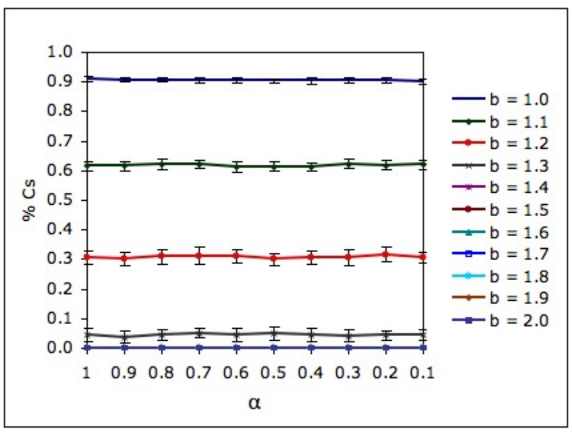


Figure 2: % of cooperators for $\phi = 0.1$ (RD rule).

From these results, it follows that the sensitivity of the model to the synchrony rate depends almost entirely on the transition rule that is used. This brings us to the question we try to answer with this work: which features of these transition rules are responsible for the Spatial PD's game sensitivity to the synchrony rate? After describing the function we use to measure the sensitivity of the model to the synchrony rate, we will start by looking to one of these features: *payoff monotonicity*.

## Sensitivity Measure

We want to measure the sensitivity to the synchrony rate for situations like, for example, the one of Fig. 1, where $\phi$ and $K$ are fixed. Let $C(\phi, R, b_i, \alpha_j)$ be the proportion of cooperators achieved for specific input parameters, where $R$ represents the input parameter set of the transition rule (for example, for the GP rule $R = \{K\}$). We first compute, for each $b$ value, the standard deviation of the proportion of cooperators achieved along all $\alpha$ values. We then sum these standard deviations, which gives us the overall sensitivity for a specific combination of $\phi$ and $R$ values:

$$s(\phi, R) = \sum_{i=0}^{10} \sqrt{\frac{1}{10} \sum_{j=1}^{10} (C(\phi, R, b_i, \alpha_j) - \overline{C}(\phi, R, b_i))^2}, \quad (3)$$

where $b_i = 1 + 0.1i$ and $\alpha_j = 0.1j$. This measure compresses the results obtained for given $\phi$ and $R$ parameters in a single value, which may lead to some loss of information. Therefore, whenever necessary, we will complement the results obtained with equation 3 with an analysis of the data from which the sensitivity values were derived.

## Payoff Monotonicity

A transition rule is said to be *payoff monotonic* if it forbids the imitation of agents with smaller payoffs (Szabó, 2007). Looking at equations (1) and (2) we easily see that, while the RD rule is payoff monotonic, the GP rule is not (except when $K \rightarrow 0$). Given this, we first modified the RD rule in order to turn it into a non-payoff monotonic rule. The modified rule is as follows:

$$p(s_x \rightarrow s_y) = f(G_y - G_x, M) =$$

$$\begin{cases} (1 - \frac{1}{M})\frac{G_y - G_x}{b} + \frac{1}{M} & \text{if } G_y - G_x > 0 \\ \frac{1}{M} - \frac{1}{M}\frac{G_x - G_y}{b} & \text{otherwise,} \end{cases} \quad (4)$$

where $\frac{1}{M}$ is the probability that $x$ imitates $y$ when $G_x = G_y$. $M \in [1, +\infty[$ can be viewed as the payoff monotonicity degree: the bigger $M$, the smaller the probability that $x$ imitates an agent with a lower payoff. We refer to this rule as *non-payoff monotonic* RD (NPMRD).

Fig. 3 shows the sensitivity of the model calculated as in equation 3. It shows that the sensitivity grows up to $\frac{1}{M} = 0.3$ and decreases after this value, although staying higher than the sensitivity of the standard RD rule. This means that the RD rule becomes sensitive to the synchrony rate only if it is non-payoff monotonic. But, if we look at Fig. 4, where the proportion of cooperators is depicted for $b = 1$, we can see that, for situations where cooperators and defectors coexist, the sensitivity continues to grow even for $\frac{1}{M} > 0.3$. That is, in these situations the influence of the synchrony rate in the output of the system grows as $\frac{1}{M}$ grows.

After this, we modified the GP rule in order to verify if its sensitivity to the synchrony rate is also due to the fact

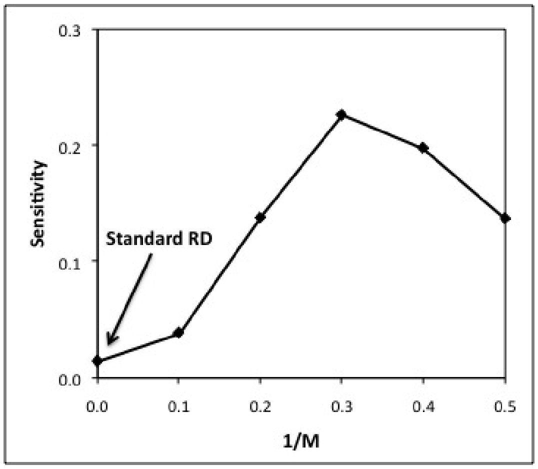


Figure 3: Sensitivity of the NPMRD rule to the synchrony rate as a function of $\frac{1}{M}$, for $\phi = 0.1$.

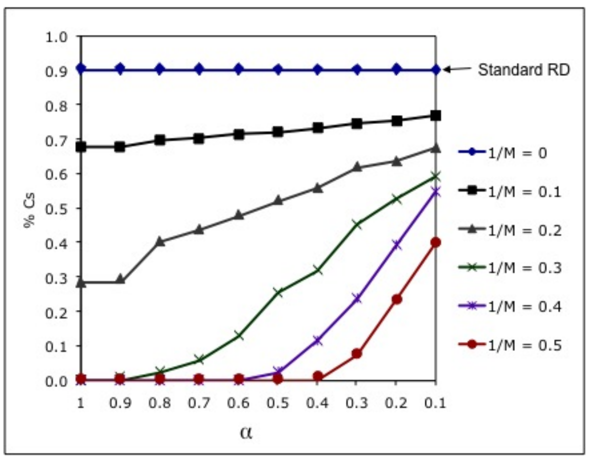


Figure 4: % of cooperators for $\phi = 0.1$ and $b = 1$ (NPMRD).

that agents can imitate a neighbor with a lower payoff. We will refer this rule as *payoff monotonic* GP (PMGP). Before describing PMGP, we recall that the GP rule takes the sum of the payoffs of C/D agents instead of treating the strategy/payoff of each neighbor individually. Putting it another way, the GP rule models a competition between two strategies (C and D) so that the winning probability is proportional to the sum of the payoffs of the agents using each strategy. The PMGP rule applies the original GP rule, eq. (1), only if one of the two following conditions is true:

$$\text{if } G_{Cs} < G_{Ds} \text{ and } s_x = C, \tag{5}$$

$$\text{if } G_{Cs} > G_{Ds} \text{ and } s_x = D, \tag{6}$$

where $G_{Cs}$ and $G_{Ds}$ are, respectively, the sums of the payoffs of C and D neighbors (including the payoff of the agent to be updated $x$), each one powered to $\frac{1}{K}$. Agent $x$ keeps its strategy if none of these conditions is true.

Fig. 5 shows that the PMGP rule becomes much less sensitive to $\alpha$ changes than the original version for $K > 0.1$. Just as an example, compare Fig. 1 with Fig. 6: even taking into account some significant standard deviations in the payoff monotonic case, the difference in sensitivity between the two situations is clear. The divergence for $K > 0.1$ means that, above this value, payoff monotonicity also plays an important role in the insensitivity of the GP rule to changes in the synchrony rate, as it does with the standard RD rule. Stating it from the opposite perspective, when agents are allowed to imitate less successful strategies, the model's sensitivity grows as this possibility increases. Given that the probability of choosing less successful strategies grows with the noise level, this means that high noise levels increase the model sensitivity to the synchrony rate.

But, Fig. 5 also shows that, for $K <= 0.1$, the sensitivity of the PMGP rule stops diverging from the sensitivity of the original GP rule. It also shows that payoff monotonicity is not the only force that influences the sensitivity of the model to the synchrony rate. Notice that the sensitivity of the PMGP rule also varies as we change the noise level. That is, even when we prevent the imitation of less successful strategies, the model's sensitivity continues to vary with noise: it grows as the noise level decreases. Therefore, there must be another feature related to the noise level that also influences the model's sensitivity, although less than payoff monotonicity. We address this problem in the next section.

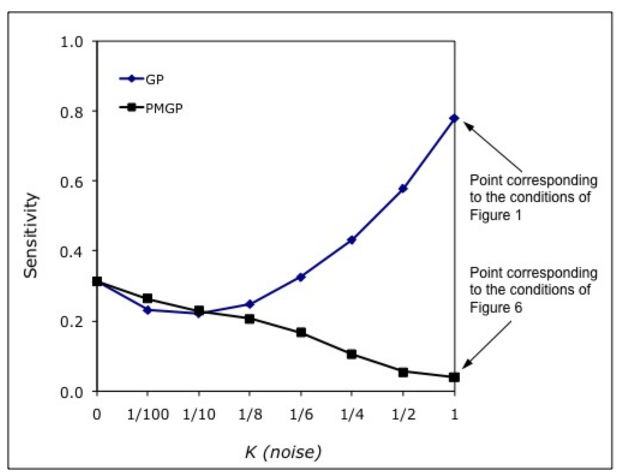


Figure 5: Sensitivity of the GP and PMGP rules to the synchrony rate as a function of $K$, for $\phi = 0.1$.

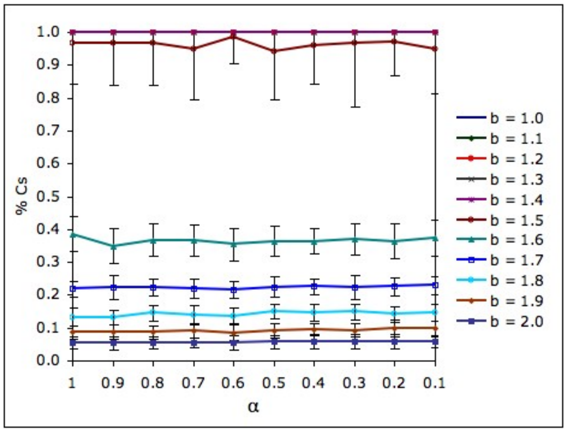


Figure 6: % of cooperators for $\phi = 0.1$ and $K$ = 1 (PMGP rule).

## Imitate the Best Tendency

Given that, with the PMGP rule, agents cannot imitate less successful strategies, what other forces influence the sensitivity of the Spatial PD game? If we analyze the original GP rule, we see that the probability of choosing a strategy with a lower payoff becomes very low as $K$ approaches 0. That is, the payoff monotonicity degree increases as $K$ decreases. On the other hand, as $K$ decreases, the tendency to imitate the wealthiest neighbors is increased for both the original and the modified GP rule. Therefore, the two rules become more and more similar as $K$ is decreased. In fact, when $K \rightarrow 0$, the two rules become one and the same deterministic rule: choose the strategy used by the best neighbor (see Fig. 5). This explains why the the two rules' sensitivities are similar for $K < 0.1$.

The above reasoning suggests that, besides payoff monotonicity, the "imitate the best tendency" level also influences the sensitivity of the Spatial PD game to the synchrony rate. More specifically, it suggests that the sensitivity of the model increases with the "imitate the best tendency" level. This could explain why the sensitivity of the model slightly increases for $K$ values near 0 when the original GP rule is used (see Fig. 5). In order to verify this hypothesis, and given that it is based only on results achieved with the GP rule, we now turn our attention again to the RD rule. The goal is to verify if the "imitate the best tendency" level also influences the sensitivity of the model when this rule is used.

The first modification we have done to the RD rule was to change the way the neighbor $y$ is chosen: each neighbor of the updating agent $x$ has a given probability $0 < \theta \leq 1$ of entering a tournament. After this, the wealthiest agent in the tournament is selected and becomes the candidate neighbor $y$. $\theta$ represents the tendency of $x$ to select its best neighbors. For example, when $\theta = 1$, $y$ is always the wealthiest neighbor of $x$.

Once defined the way of choosing $y$, we still have no total control on $x$'s "imitate the best tendency". Notice that, in the standard RD rule, $p(s_x \rightarrow s_y)$ only depends on the difference $G_y - G_x$. That is, we have no control on the sensitivity of $x$ to the payoff difference between the two agents. Given this, we further modified $p(s_x \rightarrow s_y)$ in the following way:

$$p(s_x \rightarrow s_y) = f(G_y - G_x, S) =$$

$$\begin{cases} \left(\frac{G_y - G_x}{b}\right)^{\frac{1}{S}} & \text{if } G_y - G_x > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where the sensitivity of $x$ to $G_y - G_x$ is given by $S \in [1, +\infty[$: for the same payoff difference, the larger $S$, the bigger the probability that $x$ imitates $y$. With these two modifications we can cover all the space between the best neighbor rule ($\theta = 1$, $S = +\infty$) and the standard RD rule ($\theta \approx \frac{1}{|N_x|}$, $S = 1$). We will refer to this rule as *extended* RD (ERD).

Fig. 7 shows the sensitivity of the ERD rule calculated as in equation 3. As can be seen in the chart, excepting some small fluctuations, the sensitivity of the model when the ERD rule is used grows as both $\theta$ and $S$ are increased. This means that, as for the GP rule, a strong "imitate the best tendency" level also increases the RD's rule sensitivity to the synchrony rate.

## Neighborhood Monitoring

There is yet another feature in which GP and RD differ: while the GP rule models a complete monitoring of the neighborhood (because all the neighbors' payoffs are considered), the RD rule models a partial neighborhood monitoring (only the payoff of one neighbor is considered). Notice that, despite the fact that the above described variant ERD allows a variable neighborhood monitoring, it considers only the payoff of one agent. Thus, we also modified the two rules in order to verify if this feature has some influence on the sensitivity to $\alpha$.

The GP rule was modified in the following way: each neighbor of $x$ has a given probability $\beta$ of being considered in equation 1 (the updating agent $x$ is always considered). The $\beta$ parameter can be viewed as the neighborhood monitoring level. We will refer to this rule as *partial neighborhood monitoring* GP (PNMGP). Fig. 8 shows that the PNMGP rule is less sensitive to the synchrony rate than the original GP rule by a factor of approximately $1/2$, maintaing, however, a similar qualitative behavior.

The RD rule was modified so that, as in the case of the original GP rule, the payoff of all the neighbors contribute to the decision of the updating agent $x$. According to the *complete neighborhood monitoring* RD rule (CNMRD), the

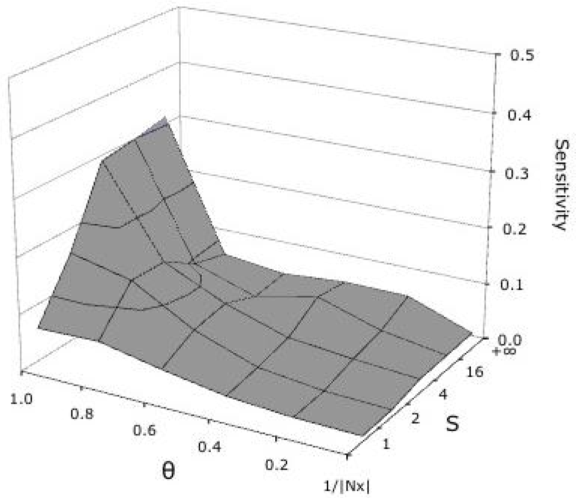


Figure 7: Sensitivity of the ERD rule for $\phi = 0.1$ as a function of $\theta$ and $S$. $\theta = \frac{1}{|N_x|}$ means that only a candidate neighbor $y$ is randomly chosen as in the standard RD rule. Therefore, the point $s(\theta = \frac{1}{|N_x|}, S = 0)$ corresponds to the sensitivity of the standard RD rule (Fig. 2). $s(\theta = 1, S = +\infty) = 0.312$, which is very close to $s(K = 0) = 0.314$ of Fig. 5. Both points correspond to the best neighbor rule.

probability $p(s_x \rightarrow s_a)$ that an agent $x$, with strategy $s_x$, changes its strategy to an alternative strategy $s_a$, where $s_a = D$ if $s_x = C$ and vice-versa, is equal to:

$$p(s_x \rightarrow s_a) = \begin{cases} \frac{G_X - G_A}{bk_A} & \text{if } G_X - G_A > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where $G_X$ and $G_A$ are the sum of the average payoffs earned by the neighbors of $x$ playing, respectively, strategy $s_x$ and $s_a$ (including $x$), and $k_A$ is the number of neighbors with strategy $s_a$. Fig. 9 shows the proportion of cooperators achieved with this rule when $\phi = 0.1$. The sensitivity to the synchrony rate for this situation is equal to 0.030, which is about the double of the sensitivity of the standard RD rule, 0.014, for the same situation (Fig. 2). This result is consistent with the one achieved with the PNMGP and GP rules. However, for the two situations, that is, for the GP *versus* PNMGP and the RD *versus* CNMRD rules, the difference in sensitivity is partly due to the fact that, with the complete neighborhood monitoring versions, there are more $b$ values for which Cs and Ds coexist (compare, Fig. 2 and Fig. 9) than for the partial neighborhood monitoring versions. Therefore, more work must be done, namely exploring intermediate levels of neighborhood monitoring, in order to determine the real influence of the neighborhood monitoring level over the model's sensitivity to the synchrony rate.

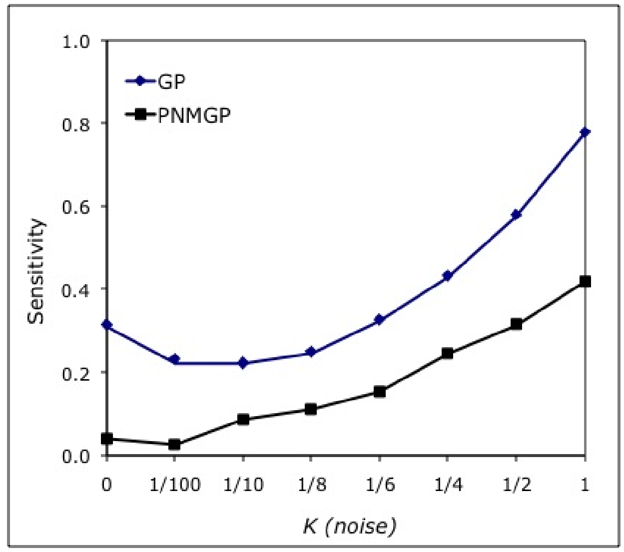


Figure 8: Sensitivity of the PNMGP rule to the synchrony rate as a function of $K$, for $\phi = 0.1$ and $\beta = 0.1$.

## Conclusions and Future Work

In this work we identified the features that determine the sensitivity of the Spatial Prisoner's Dilemma game to the synchrony rate. We first found that the sensitivity of the model depends almost completely on the transition rule used to model the strategy update process. For this, we used the *generalized proportional* and the *replicator dynamics* rules which are, respectively, sensitive and insensitive to the synchrony rate no matter the interaction topologies used in the simulations. We then used some variants of these rules in order to identify the features that make them responsible for the sensitivity of the model.

The results can be summarized in the following way: the lower the payoff monotonicity degree and the higher the "imitate the best tendency" level, the more sensitive is the game to the synchrony rate. But, given that these are just consequences of the noise level, we can state the results in the following way: on the one hand, the Spatial Prisoner's Dilemma game becomes more and more sensitive for noise levels above a given noise threshold (0.1 in the GP transition rule). On the other hand, the game is robust to small noise levels, and its robustness even grows, compared to the imitate the best strategy, if a small amount of noise is present in the strategy update process. The line corresponding to the original GP rule in Fig. 5 illustrates this well. As far as we know, this is the first time such a result is achieved. We stress that these results are the same for all the interaction topologies we used in the simulations, which go from regular to random networks.

This result indicates that the noise level may play an important role in the robustness of real dynamical systems where social dilemmas exist. More precisely, it suggests that

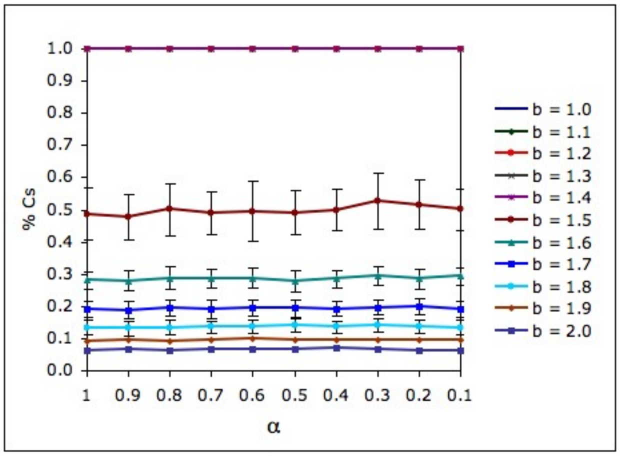


Figure 9: % of cooperators for $\phi = 0.1$ (CNMRD rule).

a moderate noise level can enhance the system's robustness to small variations on the underlying conditions. On the other hand, significant noise levels make a dynamical system too sensitive to small perturbations. More work must be done, however, in order to verify if this can be generalized to perturbations other than the ones related to the synchrony rate.

Future extensions to this work will explore *asynchronous stochastic dynamics* with other games in order to verify if the results achieved with the Prisoner's Dilemma game can be further generalized. The results achieved in (Tomassini et al., 2006) with the Hawk-Dove game, where the *best-neighbor* ($K \rightarrow 0$), the simple *proportional* ($K = 1$) and the *replicator dynamics* transition rules, as well as synchronous and sequential updating were used, seem to indicate that, also in this game, the transition rule is what determines the sensitivity of the model. However, only by exploring intermediate asynchronism and noise levels we can confirm this. Other transition rules, as the Sigmoid transition rule (Szabó, 2007) and interaction topologies, as the scale-free network model, will also be explored.

Finally, even if we now know that the noise level of the transition rule is the key feature in what concerns the sensitivity of the Spatial Prisoner's Dilemma game to the synchrony rate, we still do not know why it influences the sensitivity of the model as it does. Trying to explain this will be one of the main directions of our future work.

## Acknowledgements

We thank the reviewers of the paper for their useful comments. We also thank the GruVA members Pedro Santana, Vasco Santos and Luis Simões for useful discussions and comments. This work was partially supported by FCT/MCTES grant No. SFRH/BD/37650/2007.

## References

Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286:509.

Fatès, N. and Morvan, M. (2005). An experimental study of robustness to asynchronism for elementary cellular automata. *Complex Systems*, 16(1):1–27.

Gershenson, C. (2002). Classification of random boolean networks. In *Proceedings of the Eight International Conference on Artificial Life*, pages 1–8. MIT Press.

Grilo, C. and Correia, L. (2007). Asynchronous stochastic dynamics and the spatial prisoner's dilemma game. In *Proceedings of the 13th Portuguese Conference on Artificial Intelligence, EPIA 2007*, pages 235–246. Springer-Verlag.

Grilo, C. and Correia, L. (2008). The influence of asynchronous dynamics in the spatial prisoner's dilemma game. In *Animals to Animats - 10th International Conference on the Simulation of Behavior (SAB'08)*. Springer-Verlag. To appear.

Hofbauer, J. and Sigmund, K. (1998). *Evolutionary Games and Population Dynamics*. Cambridge University Press.

Huberman, B. and Glance, N. (1993). Evolutionary games and computer simulations. *Proceedings of the National Academy of Sciences*, 90:7716–7718.

Newth, D. and Cornforth, D. (2007). Asynchronous spatial evolutionary games: spatial patterns, diversity and chaos. In *Proceeding of the 2007 IEEE Congress on Evolutionary Computation*, pages 2463–2470.

Nowak, M., Bonhoeffer, S., and May, R. M. (1994). More spatial games. *International Journal of Bifurcation and Chaos*, 4(1):33–56.

Nowak, M. and May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359:826–829.

Oh, J. C. (2001). Cooperating search agents explore more than defecting search agents in the internet information access. In *Proceedings of the 2001 Congress on Evolutionary Computation, CEC2001*, pages 1261–1268. IEEE Press.

Pacheco, J. M. and Santos, F. C. (2005). Network dependence of the dilemmas of cooperation. *Science of Complex Networks: From Biology to the Internet and WWW, CNET 2004*, 776:90–100.

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

Szabó, G. (2007). Evolutionary games on graphs. *Phys. Rep.*, 446:97–216.

Tomassini, M., Luthi, L., and Giacobini, M. (2006). Hawks and doves on small-world networks. *Physical Review E*, 73(1):016132.

Watts, D. and Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393:440–442.