Chapter 49
Evolutionary Dynamics of the Spatial Prisoner's Dilemma with Single and Multi-Behaviors: A Multi-Agent Application

Carla Silva, Welma Pereira, Jan Knotek, and Pedro Campos

Abstract This work explores an application of the spatial prisoner's dilemma in two situations: when all agents use the same type of behavior and when they use a mix of behaviors. Our aim is to explore the evolutionary dynamics of this game to analyze the dominance of one strategy over the other. We also investigate, in some possible scenarios, which behavior has better performance when they all coexist in the same environment.

49.1 Introduction

Game theory is an important way of understanding the dynamics of certain behaviors and to analyze the evolution of the components involved. The prisoners' dilemma is a game that raises the problem of cooperation in a stark form: two strategies are available (cooperate or defect). The payoff to mutual cooperation exceeds the payoff to mutual defection (P). Much of the literature on the evolution of cooperation following Axelrod's seminal contribution [2] has sought to identify the factors that influence the possibility of cooperative behavior emerging in populations of boundedly-rational agents playing the repeated prisoners' dilemma (RPD). Hoffmann and Waring [7] have studied the problem of the localization of the agents in the RPD. One important contribution to this area is due to Nowak and May [17]. The authors study a population of RPD playing cellular automata distributed on squares displayed on a torus which are capable only of the Always Defect (ALL-D) and Always Cooperate (ALL-C) strategies. Individual agents interact with

C. Silva (✉)
Faculty of Economics, University of Porto, Porto, Portugal
e-mail: cmap.silva@ymail.com

W. Pereira, J. Knotek, and P. Campos
LIAAD-INESC Porto and Faculty of Economics, University of Porto, Porto, Portugal
e-mail: welma.pereira@gmail.com, jan.knotek@gmail.com, pcampos@fep.up.pt

M.M. Peixoto et al. (eds.), *Dynamics, Games and Science II*, Springer Proceedings in Mathematics 2, DOI 10.1007/978-3-642-14788-3_49,

all neighbors on their eight diagonally and orthogonally adjacent squares and are able to imitate the strategy of any better-performing one among them. Nowak and May have found that the distribution of strategies on the torus depends on the relative size of the RPD-payoffs.

Both the static and dynamic perspectives of evolutionary game theory provide a basis for equilibrium variety as we notice in our multiple simulations. According to Szabó and Fáth [20] there is a static and a dynamic perspective of evolutionary game theory, in the next sections we deeply explore this game dynamics perception. In Zimmermann and Eguluz [25] research the concluding equilibrium solution is composed mostly by cooperative agents, in a prisoner's dilemma with adaptive local interactions, which focus cooperative behavior among a group of agents assuming adaptive interactions. In our case we do not deeply focus on the agent leadership's issue acquire after perhaps existing adaptive interactions or rules. We could analyze this question with specific measures in the link analysis method, but our main goal was to focus on the strategy leadership's issue, view the dominance of one strategy in a spatial environment, focusing assorted issues. In our simulations we had a diversity of situations, as in the more significant payoffs changes occurs when one strategy was trying to be the leader innovation strategy.

In this work, we use Agent-based simulation to analyze the prisoner's dilemma with three types of behavior. We will use three types of behaviors: copy best player (greedy), copy best strategy (conformist) and Pavlovian. After presenting them individually, we join all the three in the same playground. We will explore the game dynamics when the parameters or the initial conditions change. The evaluation is made using statistics and link analysis. Our model is based in physical properties of the automata. Spatial interaction of autonomous actors selects actions from their own logical set based on its own state and on its neighbor's states. We apply this social interaction to 100 agents, each one having one of the three ways of action (behaviors), so that the outcome depends on the choices of all the players based on the Moore Neighborhood, played with eight neighbors (as in [7]). The game dynamics is determined locally since the neighborhood is defined in a finite region. These geographic effects are represented by placing agents in territorial structures and restricting them to interact and learn within certain geographic regions. We consider that in real-world, these agents can be seen as companies. Each agent has an initial strategy which can be either cooperate or defect. Once the dynamics of this game gives rise to clusters of collaborators and/or defectors. We also decided to use social network analysis in order to capture the link relation in the networks of firms that emerged in the different scenarios.

We concluded that Pavlovian behavior scores better than Greedy and Conformist when b (the payoff that corresponds to the defection of a player and the cooperation of the other) is higher. Additionally we can see that when we have more companies with collaborative strategy (small b) we get higher average payoffs in all behaviors. The maximum individual payoff is also higher for small values of b when companies use behavior Greedy or Conformist.

The work is structured as follows: in Sect. 49.2 we define the prisoners' dilemma and give a short overview of game theory. We also define the gain matrix and the strategies involved in the model. Section 49.3 contains the implementation of the model. Section 49.4 describes the experiments. Data analysis is in Sect. 49.5, in Sect. 49.6 we describe possible dynamics GIS application and in Sect. 49.7, we discuss the corresponding results. Finally, in Sect. 49.8, we introduce some future work.

49.2 The Prisoner's Dilemma: A Short Overview of Game Theory and Definition of the Strategies

The prisoner's dilemma is the name given by Albert W. Tucker [6] to the following problem in game theory: Suppose the situation in which there are two suspects (let's say P1 and P2) of a crime that are arrested in separated cells by the police. A prosecutor meets with the prisoners separately and offers the same deal to both of them. They can either testify against the other or to remain silent. If prisoner P1 decides to testify and prisoner P2 remains silent, P1 goes free and P2 receives a 10-years full sentence. Likewise, if prisoner P2 testifies while P1 remains silent, P2 will be the one to be set free while P1 stays in prison for the same 10-years time. If they both testify they both stay in prison for a shorter sentence of 7-years. If in the last case, they both remain in silent, P1 and P2 are sentenced to only 1 year in prison. The problem of the prisoners is to decide weather to remain silent or to testify against the other. Since the prisoners are both isolated, neither of them would know the other's decision. This story is usually generalized to analyze similar situations.

Analyzing the options individually if the other remains in silent, the best is to testify against him (defect) so that you can be set free. But if the other decides to defect the best is also to defect, otherwise you will stay in jail while the other leaves free. In the other hand the best solution for both of them is to remain in silent because in this situation, even though neither of them is set free, they still get a much shorter sentence. The dilemma is that they do not know what the other will do. A common view is that this puzzle illustrates a conflict between individual and group rationality. If each player has chosen a strategy and no player can benefit by changing his or her strategy while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute a Nash equilibrium (Table 49.1).

Table 49.1 Original payoff matrix [6]

	Cooperate (C)	Defect (D)
Cooperate (C)	7,7	10,0
Defect (D)	0,10	1,1

49.2.1 Companies and Strategies

We decided to apply our model to the situation where there is a bunch of companies in the same market business competing with each other. The companies are randomly disposed in a matrix and they interact with their closest eight neighbors surrounding it. For example, the companies can be restaurants in the city of Porto and their business is to sell "francesinhas", the typical dish of this Portuguese city. The neighbors can be seen as the nearest restaurants that competes in the same physical area or district. Another possible application is to see the proximity in the matrix as the companies with most similar activities that competes with each other not necessarily in the same region but in the same kind of business.

Each company can choose to cooperate with their neighbor or not. If they cooperate they form a cartel, i.e. to sell the "francesinha" for the same price and therefore their profit will be quite similar. If they decide not to cooperate they can sell it for cheaper price. In this case they can attract more customers and earn more. This can be seen as the situation of defect in the prisoner's dilemma problem.

The options of collaboration or defect can be seen as two strategies that can be used by the companies. This business market can then be divided into this two types of strategies followed by companies, the ones that follow the strategy of collaboration, let's call it strategy A, the companies that follow the strategy of defection, that will be called strategy B. In this situation the companies can change the strategies at any time. Our aim is to explore the evolution of this game in different scenarios and analyze if one of the strategies dominates the other.

To make this situation more real and more competitive we also decided to introduce a reward policy that gives to the cooperators some advantage. Each time a company decides to cooperate, it wins a reward based on its own status and on the number of collaborators in its neighborhood.

49.2.2 Spatial-Temporal Definitions

We start by defining our game geometry. Each point of our spatial lattice has a state. The possible states are: Cooperate (C) or Defect (D). The states of the cells are updated after each round according to agents current state, the states of its eight nearest neighbors and the agents behavior. Implemented behaviors are described later. All cells in the lattice are updated synchronously. We can define the neighborhoods depending on the system we pretend to model. Concerning the two dimensional lattice the following definitions are common (Fig. 49.1):

The Von Neumann neighborhood can be called also as 4-neighborhood as it contains four cells: the cell above, below, to the right and to the left. The radius of this definition is one. The Moore neighborhood is an enlargement of the Von Neumann neighborhood containing the diagonal cells too, the radius is also one. The Moore neighborhood is also called 8-neighborhood. The extended Moore neighborhood is equivalent to the description of Moore neighborhood, but it reaches over the

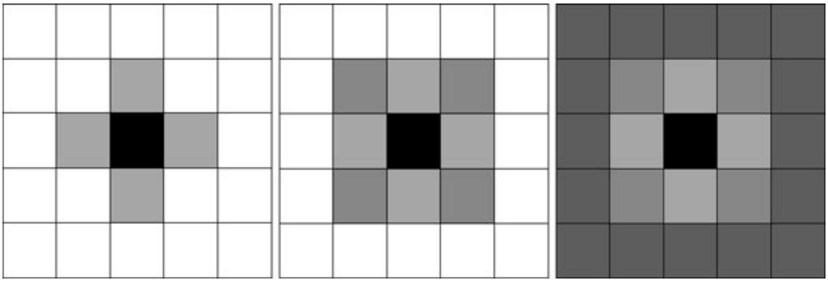


Fig. 49.1 Von Neumann neighborhood, Moore neighborhood and Extended Moore neighborhood

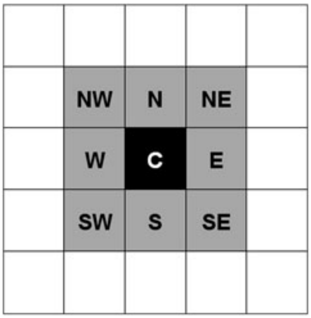


Fig. 49.2 The gray cells are the neighbors of the central one. As previous discussed the states of these cells are used to calculate the subsequently state of the center cell according to the rule defined

distance of the next adjacent cells, in this case the radius is two. Extended Moore Neighborhood can be also called as 25-neighborhood.

In our simulations, we decided to use the technique of Moore neighborhood. Comprising eight cells surrounding a central cell in a square lattice. We could also have used (among others) the technique of Von Neumman neighborhood. The spatial games can vary in many ways its geometry (Fig. 49.2).

Initially random positions are generated. Each cell of the network is occupied by an agent (player, company). To avoid edge effects, the edges of the network (or endpoints of the line) are glued together (Fig. 49.3).

Nowak and May [17] studied a population of agents distributed on cells of a 2-dimensional torus which are capable only of Always – Cooperate (C) or Always – Defect (D) strategies. In our paper we are using the same type of strategies but we rename them as (A) or (B). Nowak and May [17] in their study found out that the distribution of strategies on the torus depends on the relative size of the payoffs. They refer that the future state of each cell depends on the current state of the cell and the states of the cells in the neighborhood, the development of each cell is defined by rules. As we had referred before. In our case rules fall into three basic strategy behaviors. In our game 100 agents play during n time units, which means one or three kind of behaviors in a playground being able to change in each round. The evolution of these rules leaded us to observe dynamical patterns. The survival cooperative and/or defect behavior.

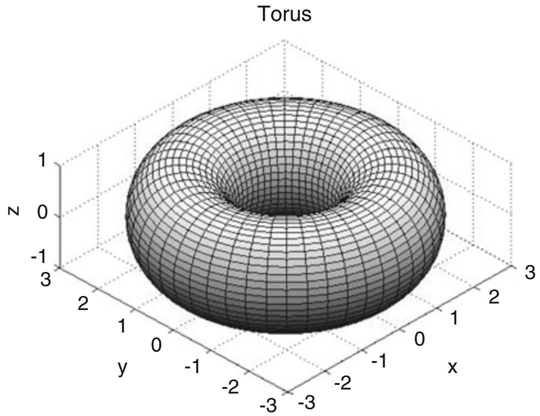


Fig. 49.3 By connecting square lattice from left to right side and from up to bottom side we obtain a torus topology, where all cells are equal (all have the same count of neighbors). Here we have an example image made in *Matlab*, using *surf* command, which draws 3-D shaded surface plot

As Jun and Sethi refer in [9], the survival of cooperative behavior in populations in which each person interacts only with a small set of social neighbors, the individuals adjust their behavior over time by shortsightedly imitating more successful strategies within their own neighborhood. All this process leaded us to a called Demographic Game, as Epstein [5] refer seems an appropriate name for this class of models because they involve spatial, evolutionary, and population dynamics. In Epstein's research [5], each agent is an object whose one main attributes is his vision. He considers vision like the distance an agent can see, looking north, south, east, or west. In our case we assume that all of our agents have a peripheral vision, looking 360° around, so we also include north–west, south–west, north–east and south–east. Our agents move around this space, interacting with Moore Neighbors (all with peripheral vision).

49.2.3 Gains Matrix

In what follows, a player must choose between two strategies. Isaac [8] refers that by altering a single entry of the payoff matrix could demonstrate that payoff cardinality is crucial to prisoner's dilemma outcomes on an evolutionary grid. And also refers that the evolutionary processes are fundamental to cooperation in social situations and have been an enduring theoretical problem in diverse areas, like biological, sociological, and geographical. Using Nowak [16] calculating gains according to his definition, we build for each round a gain matrix, which gives us payoffs values, the ones that determine if the agent will stay with the same strategy or will change according to his neighbors payoffs (Table 49.2).

Table 49.2 Nowak [16] defines the payoffs values according to the above rules

	Cooperate (C)	Defect (D)
Cooperate (C)	1	0
Defect (D)	b	ϵ

Calculation of Gain:

1. If C find D, then D obtains b>1 and C gets 0
2. If D satisfies D, D gets ϵ
3. Cooperate if both C and C, then each gets 1

Where $1<b<2$ and $\epsilon \to 0$.

Nowak and May [17] presented the seminal work that showed how the spatial effects of the interactions between simple agents in a cellular automaton model of the iterated prisoner's dilemma was sufficient enough for the evolution of cooperation. A similar cellular automaton model was built that simulated cooperation through the behavioral adaptation of Pavlovian agents as they adjusted their cooperation by mimicking the most successful player in a neighborhood. In this present paper we observe this cooperation or non-cooperation behaviors varying the b value between 1 and 2 and establishing ϵ as 0.01.

49.2.4 Game Dynamics

Cooperation is frequently observed in real-life psycho-economic experiments. This result either suggests that the abstract Prisoner's Dilemma game is not the right model for the situation or that the players do not fulfill all the premises. Indeed, there is good reason to believe that many realistic problems, in which the effect of an agent's action depends on what other agents do are far more complex that perfect rationality of the players could be postulated, Szabó and Fáth [20]. Nevertheless, the standard deductive reasoning loses its appeal when agents have non-negligible cognitive limitations, there is a cost of gathering information about possible outcomes and payoffs. Like Xianyu [23] refer this in his recent paper. We also agree that agents have incomplete information on other agents' strategies, so the agents need to learn and develop their own strategies in this unknown environment. Mind necessarily becomes an endogenous dynamic variable of the model. This kind of bounded rationality may explain that in many situations people respond instinctively, play according to heuristic rules and social norms rather than adopting the strategies indicated by rational game theory. In our simulation we compute three rational behaviors.

49.2.5 Three Behaviors Strategies

According to the payoff and to the strategy agent chooses if he wants to change strategy or not. This means changing from A to B (Cooperate to Defect) or vice-versa,

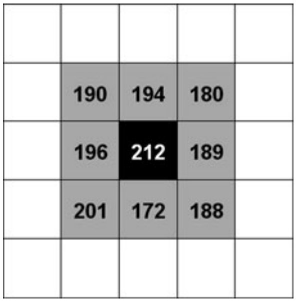


Fig. 49.4 Each cell of the network is occupied by a player, and each one has one associated payoff value calculated using the gain matrix

or else no change at all. In the following example the player won't change strategy because his payoff is bigger than the payoff of its neighbors (Fig. 49.4).

We chose to implement in R following three behaviors: Copy Best Player (greedy), Copy Best Strategy (conformist) and Pavlovian. Three kinds of social preference theories have been tested. As Oliver Kirchkamp [11] we apply the idea of evolution to a spatial model, were prisoners' dilemmas or coordination games are played repeatedly within neighborhoods where players instead of optimizing in each round, prefer to copy successful strategies. Discriminative behavior of players is introduced representing strategies as small automaton, which can be in different states against different neighbors. These personality types represent certain simple aspects of actual human behavior. Pavlovian agents are the most realistic automaton for the investigation of the evolution of cooperation, because they are simple enough to know nothing about their rational choices but intelligent enough to follow an action that produces a satisfactory state of affairs tends to reinforce the repetition of that particular action.

According to Power [18], greedy is an agent who imitates the neighbor with the highest reward. Then conformist is an agent who imitates the action of the majority in the social unit. And Pavlovian is an agent with a coefficient of learning whose probability of cooperation changes by an amount proportional to the reward/penalty it receives from the environment.

49.2.5.1 Copy Best Player (Oliver Kirchkamp [10])

Greedy, a learning player can simply look around in the neighborhood which he observes and determine the player with the highest payoff. A learning player that uses the rule "copy best player" will pick the strategy of the most successful player. Of course, it could well be that there is more than a single player who has the maximal payoff. Then let players use the following tie breaking rule. Define the set of most successful players in neighborhood N_L^i of player i as

$$M^{i,t} \leftarrow argmax_{j \in N_L^i} \left(\frac{\pi_\epsilon^{j,t}}{n_\epsilon^{i,t}} \right) \tag{49.1}$$

where $\pi_\epsilon^{j,t}$ is player's j payoff in time t and $n_\epsilon^{i,t}$ is a count of players in the neighborhood.

The probability that player i choose strategy s in period $t+1$ is determined as

$$P(x^{i,t+1}=s) \leftarrow \begin{cases} 1 & \text{if } x^{i,t} \in \left\{x^{j,t} | j \in M^{i,t}\right\} \textit{ and } s = x^{i,t} \\ 0 & \text{if } x^{i,t} \in \left\{x^{j,t} | j \in M^{i,t}\right\} \textit{ and } s \neq x^{i,t} \\ \frac{\sum_{j \in M^{i,t} \wedge x^{j,t}=s} n_\epsilon^{j,t}}{\sum_{j \in M^{i,t}} n_\epsilon^{j,t}} & \textit{otherwise} \end{cases} \tag{49.2}$$

where $M^{i,t}$ is a set of best players in neighborhood of player i in time t and $n_\epsilon^{j,t}$ is a count of neighbors of player j in time t.

Thus the player that is to be copied is chosen randomly with probabilities that are proportional to the number of interactions the respective best players had. In the special case where the player's own strategy is among the best strategies, we assume that the player prefers to keep his own strategy.

In our experiments our greedy agent just changes his strategy based in the highest payoff neighbor.

49.2.5.2 Copy Best Strategy (Oliver Kirchkamp [10])

A learning player is a Conformist when it look at the average payoffs of strategy s at time t in the neighborhood of player i which we designate by $f_s^{i,t}$:

$$f_s^{i,t} \leftarrow \begin{cases} \frac{\sum_{j \in \cup_s^{i,t}} \pi_\epsilon^{j,t}}{\sum_{j \in \cup_s^{i,t}} n_\epsilon^{j,t}} & \text{if} \sum_{j \in \cup_s^{i,t}} n_\epsilon^{j,t} > 0 \\ -\infty & \textit{otherwise} \end{cases} \tag{49.3}$$

where $\cup_s^{i,t}$ is a set of players in neighborhood of player i with strategy s in time t, $\pi_\epsilon^{j,t}$ is payoff of player j in time t and $n_\epsilon^{j,t}$ is a count of players in neighborhood of player j in time t.

If a strategy is not used in a neighborhood, we define its fitness to be $-\infty$ to make sure that it will be never selected by an evolutionary process. A learning player that uses the rule "copy best strategy" switches to the strategy with the highest average payoff. Again there could be more than one strategy with maximal payoff. Then we use the following tie breaking rule: define the set of most successful strategies as:

$$N^{i,t} \leftarrow argmax_s \left(f_s^{i,t}\right) \tag{49.4}$$

As for the "copy best player", two strategies could achieve exactly the same average payoff. The probability that player i uses strategy s in the next period is then

$$P(x^{i,t+1}=s) \leftarrow \begin{cases} 1 & \text{if } x^{i,t} \in N^{i,t} \textit{ and } s = x^{i,t} \\ 0 & \text{if } x^{i,t} \in N^{i,t} \textit{ and } s \neq x^{i,t} \\ \dfrac{\sum_{j \in \cup_s^{i,t}} n_\epsilon^{j,t}}{\sum_{\sigma \in N^{i,t}} \sum_{j \in \cup_\sigma^{i,t}} n_\epsilon^{j,t}} & \textit{otherwise} \end{cases} \tag{49.5}$$

where $\cup_s^{i,t}$ is a set of players in neighborhood of player i with strategy s in time t, $N^{i,t}$ is a set of strategies with highest mean payoff and $n_\epsilon^{j,t}$ is a number of players in neighborhood of player j in time t.

If a player already uses one of the best strategies, he adopts one of the best strategies randomly with probabilities proportional to the number of interactions the users had with the respective strategies.

In our experiments the conformist agent just changes his strategy to the one, which has higher average payoff in players neighborhood.

49.2.5.3 Pavlovian [18]

Like Power [18] we also define the agents as stochastic learning automata with Pavlovian personalities and attitudes.

By definition Pavlov works according to the following algorithm: Szabó and Fáth [20] “repeat your latest action if that produced one of the two highest possible payoffs, and switch to the other possible action, if your last round payoff was one of the two lowest possible payoffs”. As such Pavlov belongs to the more general class of Win-Stay-Lose-Shift strategies, which define a direct payoff aspiration level for strategy change. An alternative definition frequently appearing in the literature is “cooperate if and only if you and your opponent used the same move in the previous round”, and this translates into the same rule of the Prisoner’s Dilemma.

Pavlovian strategies are formulated as a weighted payoff, an average production function, and a three-step memory coefficient of learning. Given an agent, the weighted payoff is defined as:

$$RPwt = \sum_{i=1}^{3} Mc_i \cdot w_i \tag{49.6}$$

$RPwt \leftarrow$ weighted payoff of agent in last three rounds.

W_i is a weighting parameter such that all weights sum to one, and Mc_i is the history payoff. Assuming that the effects of memory decrease with time, $w_1 \geq w_2 \geq w_3$ and $w_1 + w_2 + w_3 = 1$. Let $S(t)$ be the strategy of the player in time t. Then parameter α–learning rate is after each round for each Pavlovian player set in a following manner:

$$\alpha_i(t+1) \leftarrow \begin{cases} \alpha_i(t) + 0.15 & \text{, if } (S(t) = S(t-1)) \wedge (S(t-1) = S(t-2)) \\ \alpha_i(t) + 0.10 & \text{, if } (S(t) = S(t-1)) \wedge (S(t-1) \neq S(t-2)) \\ \alpha_i(t) - 0.10 & \text{, if } (S(t) \neq S(t-1)) \end{cases} \tag{49.7}$$

The probability of cooperation for agent i at time $t+1$ is:

$$p(t+1) \leftarrow \begin{cases} p(t) + (1-p(t)) \cdot \alpha_i & \text{, for } S(t) = C \text{ and } RPwt > pf_{avg} \\ (1-\alpha_i) \cdot p(t) & \text{, for } S(t) = C \text{ and } RPwt \leq pf_{avg} \end{cases} \tag{49.8}$$

For every t there is $q(t) = 1 - p(t)$. If previous action is D:

$$q(t+1) \leftarrow \begin{cases} q(t) + (1-q(t)) \cdot \alpha_i & \text{, for } S(t) = D \text{ and } RPwt > pf_{avg} \\ (1-\alpha_i) \cdot q(t) & \text{, for } S(t) = D \text{ and } RPwt \leq pf_{avg} \end{cases} \tag{49.9}$$

The state of agent i is updated contingent on its previous state, the average neighborhood production function, and the probabilities for both C and D. The neighborhood production function for time t is the cooperation payoff for the group following:

$$pf(t) = \frac{\sum C_j}{N} \tag{49.10}$$

C_j is the payoff value for agent j and N is the total number of agents in the neighborhood. The average neighborhood function for three memory events is given by:

$$pf_{avg} = \frac{\sum_i^3 pf_i}{3} \tag{49.11}$$

pf_{avg} – average payoff in neighborhood in last three rounds.

Thus, the state of agent i at time $t+1$ with $S(t)$, where $Ru \in [0, 1]$ is a uniform random value:

For $S(t) = C$:

$$S(t+1) \leftarrow \begin{cases} D & \text{if } RPwt \text{ for agent } i < pf_{avg} \text{ and } p(t+1) < q(t+1) \\ & \text{and } q(t+1) > Ru \\ C & \text{if conditions for } D \text{ are not satisfied} \end{cases} \tag{49.12}$$

For $S(t) = D$:

$$S(t+1) \leftarrow \begin{cases} C & \text{if } RPwt \text{ for agent } i < pf_{avg} \text{ and } q(t+1) < p(t+1) \\ & \text{and } p(t+1) > Ru \\ D & \text{if conditions for } C \text{ are not satisfied} \end{cases} \tag{49.13}$$

Pavlovian strategies according to Kraines and Kraines [12], are quite stable even in a noisy environment. Although this strategy cooperates and retaliates, as does Tit-For-Tat [2], it is not tolerant. The Pavlovian behavior will exploit altruistic strategies until he is punished by mutual defection. Pavlovian strategies are natural models for many real life conflict-of-interest.

49.3 Implementation

The experiment was implemented in R [19]. Its main advantage is that there is a large library of packages to compute many statistics, draw graphs etc. R is an open source software and its distributed for free.

49.3.1 Definition of the Game

In our experiments, we used two dimensional arrays of agents as it is done in [17]. Our implementation involves the iterated n-person prisoner's dilemma.

For most of the statistics and simulations, a ten by ten square lattice is used. Boundary conditions are solved periodically – square lattice is bended by two sides into a torus. Periodic conditions simplifies the problem, because each cell (agent) has exactly the same conditions – eight neighbors. Behaviors and rewards are then influenced only by the configuration of strategies around the agent.

In each round, each agent is playing with all agents in a Moore eight-neighborhood and also with himself. Rewards are driven by a simplified evaluation system:

Table 49.3 shows the rewards for each combination of strategies, that can be played by two agents. When Cooperator meets Defector, Defector receives a payoff equal to constant b from interval $\langle 1, 2\rangle$ and Cooperator receives 0. Cooperators meeting each other receive reward 1 and finally two Defectors playing together receive $\epsilon \to 0$.

In our experiments, ϵ is set to 0.01 and b is the variable which we are modifying to see the changes in process. These rules of the game are implemented in the function *play*:

```
play<-function(a, b,payoffB, payoffEps){
 if(a== 0 && b== 0) {return (1)}
 if(a== 0 && b== 1) {return (0)}
 if(a== 1 && b== 0) {return (payoffB)}
 if(a== 1 && b== 1) {return (payoffEps)}
}
```

49.3.2 Initialization

The whole simulation process is started by function *PlayDilemma*, which takes as parameters initial strategy matrix, behavior matrix, Payoff B, Payoff ϵ, number of rounds and size of playground.

Table 49.3 Payoff matrix of the two agents game Nowak [16]

	Cooperate (C)	Defect (D)
Cooperate(C)	1	0
Defect (D)	b	ϵ

Initial strategy matrix contains the first strategy, which agents will play. "Zeros" mean Cooperate, "ones" are interpreted as Defect.

Behavior matrix contains 1 for Greedy behavior, Conformist behavior and Pavlovian agent.

49.3.3 One Round

A typical round works as follows:

1. Calculate payoffs for all agents according to their current strategies (Cooperate/Defect)
2. Evaluate last rounds payoffs, compute learning rate α and probability of cooperation (Pavlovian agent)
3. Change strategies by last rounds payoffs and strategies in agents neighborhood using agent's predefined behavior

As the calculation of payoffs for each behavior is the same, the only aspect which one implemented differently for each behavior is the function *changeStrategies*, which is called after each round. For Pavlovian agents, there are also two matrices to be evaluated – Learning rate α matrix and probability matrix. This is described in detail in Pavlovian behavior Sect. 49.3.4.3.

49.3.4 Implemented Behaviors

After each round, agents are evaluating the payoffs and if they need, they change strategy – cooperate or defect, for the next round. Evaluations for changing the strategy are implemented as three types of behavior. In this paper, we are also evaluating how these behaviors are influencing the distribution of complete payoffs among the agents after 50 or more rounds.

49.3.4.1 Copy Best Player

Copy Best Player is the simplest behavior we have used. Agent compares its payoff with payoffs of players in neighborhood and adopts the strategy of the most successful player in previous round.

49.3.4.2 Copy Best Strategy

The second behavior we used is Copy Best Strategy. To compare the strategies in neighborhood, agent uses mean of payoffs gained by players using strategy

Cooperate and mean of payoffs gained by players using strategy Defect. Player compares these means and picks the strategy with higher mean. If the player himself has higher payoff than these means, he keeps its last strategy.

49.3.4.3 Pavlovian

Pavlovian behavior is described in [24] and also in [18]. This behavior is much more complex than previous two, and its results are not always the best.

Pavlovian agent is defined as an agent with a coefficient of learning whose probability of cooperation changes by an amount proportional to the reward/penalty it receives from the environment.

We had to make some adaptation for the Pavlovian agent defined in previous papers, because our environment is not giving any penalties. Our Pavlovian agent is comparing his reward to the mean of rewards of the neighborhood.

Each agent has learning rate α and probability of cooperation p. These two variables are adjusted in each round looking three rounds back on a payoff response from the environment.

49.3.4.4 Three Rounds Payoff

The agent uses weights $w_1 \geq w_2 \geq w_3$ and $w_1 + w_2 + w_3 = 1$. Weight w_1 is the largest in order to make last round more important than previous rounds.

The three rounds payoff is weighted for each agent and computed using this equation:

$$RPwt = \sum_{i=1}^{3} Mc_i \cdot w_i \tag{49.14}$$

where Mc_i is a payoff in one round and w_i is the corresponding weight.

49.3.4.5 Neighborhood Three Rounds Payoff

pf_{avg} is the mean of payoffs in agents neighborhood for last three rounds. It is computed as follows:

$$pf(t) = \frac{\sum Mc_j}{n} \tag{49.15}$$

$$pf_{avg} = \frac{\sum_{i=0}^{2} pf(t-i)}{3} \tag{49.16}$$

Mc_j are the payoffs of neighborhood agents and n is the complete count of agents in neighborhood. In our case it is always set to eight because we used Moore neighborhood. From the first equation we get the average production in the neighborhood in one round. To get the three rounds mean, we compute the mean of $pf(t-2)$, $pf(t-1)$, $pf(t)$ using the second equation.

49.3.4.6 Learning Rate α

To know if the player was changing the strategy a lot or he wasn't changing at all, we are introducing parameter α. This parameter allows the Pavlovian agent to start changing the strategy more often, if the constant behavior has low payoff compared to mean payoff of the neighbors or in other situation, agent can remain in similar changing rate as he had in last rounds. Learning rate α is a bounded variable always set to a value from the interval [0, 1]. If α is close to zero, it means, that agent was changing strategies often in the past rounds. If α is close to one, the agent was stable in last rounds. In each round α is adjusted according to the following (49.17):

$$\alpha_i(t+1) \leftarrow \begin{cases} \alpha_i(t) + 0.15 & \text{, if } (S(t) = S(t-1)) \wedge (S(t-1) = S(t-2)) \\ \alpha_i(t) + 0.10 & \text{, if } (S(t) = S(t-1)) \wedge (S(t-1) \neq S(t-2)) \\ \alpha_i(t) - 0.10 & \text{, if } (S(t) \neq S(t-1)) \end{cases} \tag{49.17}$$

Where t means the time of the round, and $S(t)$ is the agent's strategy in round t.

49.3.4.7 Probability of Cooperation

p is the probability of using a cooperate strategy in a given round. Notation $p(t)$ means probability of cooperation in round t. Probability is adjusted in each round by the payoff response of the environment. If previous strategy is C, then probability of Cooperation is computed as:

$$p(t+1) \leftarrow \begin{cases} p(t) + (1-p(t)) \cdot \alpha_i & \text{, for } S(t) = C \text{ and } RPwt > pf_{avg} \\ (1-\alpha_i) \cdot p(t) & \text{, for } S(t) = C \text{ and } RPwt \leq pf_{avg} \end{cases} \tag{49.18}$$

Note that for every t there must be $q(t) = 1 - p(t)$. This is used in the implementation to have only one matrix for probabilities.

The same set of equations is used for updating the action probabilities when the previous action is D. Probability q of defect is computed as:

$$q(t+1) \leftarrow \begin{cases} q(t) + (1-q(t)) \cdot \alpha_i & \text{, for } S(t) = D \text{ and } RPwt > pf_{avg} \\ (1-\alpha_i) \cdot q(t) & \text{, for } S(t) = D \text{ and } RPwt \leq pf_{avg} \end{cases} \tag{49.19}$$

Final strategy is chosen by probability of cooperation and defection and also by the last round's payoff. Conditions we used are slightly modified from conditions of previous paper [18]. To make the Pavlovian agent a non-random agent, in our (49.18) and (49.19) we removed the condition containing random number Ru, which was compared with cooperation/defection probability in equations in Sect. 49.2.5.3 from [18]. This was needed to have the chance to repeat each play in the same way.

For $S(t) = C$:

$$S(t+1) \leftarrow \begin{cases} D & \text{if } RPwt \text{ for agent } i < pf_{avg} \text{ and } p(t+1) < q(t+1) \\ C & \text{if conditions for } D \text{ are not satisfied} \end{cases} \tag{49.20}$$

For $S(t) = D$:

$$S(t+1) \leftarrow \begin{cases} C & \text{if } RPwt \text{ for agent } i < pf_{avg} \text{ and } q(t+1) < p(t+1) \\ D & \text{if conditions for } C \text{ are not satisfied} \end{cases} \tag{49.21}$$

Pavlovian agent is the most sophisticated agent we used, because it takes longer to compute each round. This higher demand for time should have been rewarded by better results in experiments, which were not so good as we expected.

49.3.5 Cooperation Reward Policy

We decided to use some reward policy to give some advantage to the cooperators. We only use it though in our multi-behaviors situation. Here is how it works: every time a player decides to cooperate it will receive a reward that will be added to his payoff. The reward is defined as follows:

$$0.1 \times \mathit{Size.Group} \tag{49.22}$$

where the Size.Group represents the number of cooperators among its eight neighbors. The idea behind is that group of cooperators will get benefited proportionally to its group size.

49.4 Experiments

This section analyzes the dynamics of the spatial prisoner's dilemma that we have implemented. The game is played repeatedly for twenty rounds for each behavior described before. The evolution of the strategies and their respective payoffs are displayed through graphics.

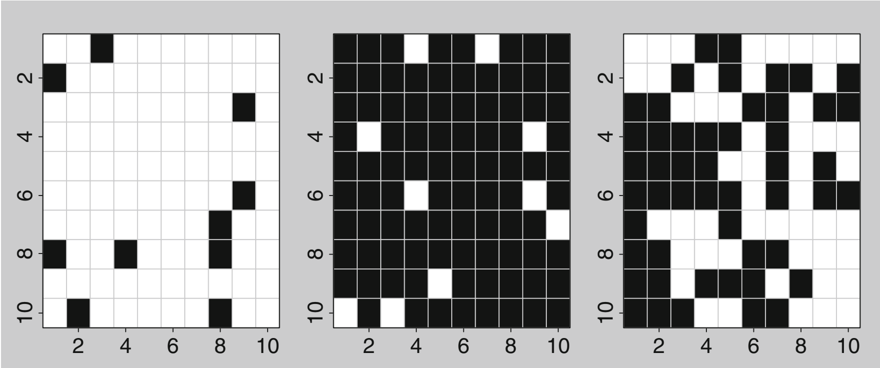


Fig. 49.5 The initial matrices (M1, M2 and M3)

We investigate our application in the situation where the companies have all the same behavior (Single behavior situation) and when all three behaviors coexist in the same playground (Multi-behaviors situation).

The initial matrix was M3 (Fig. 49.5). M3 was generated randomly with 50–50% of the companies using strategies A and B. In the single behavior situation, we always used the same initial matrix so that allows us to compare our results. In the multi-behaviors situation, the experiments were performed in different scenarios with the 3 types of initial matrices.

49.4.1 Single Behavior Situation

The experiments performed in this section analyze the dynamics of a situation where all the companies have the same type of behavior.

Table 49.4 Mean Payoffs of the three behaviors

Payoffs (bs)	1.1	1.5	1.9
Greedy Behavior	169.3	130.5	8.7
Conformist Behavior	162.9	84.8	50.5
Pavlovian Behavior	99.9	87.9	90.2

Figure 49.6 shows the evolution of strategies A and B when all the companies use the behavior greedy the b is equal to 1.1. After stage 4 it doesn't change anymore. We represent the companies with strategy A by black squares and the companies with strategy B by white squares. We can see that the number of blacks increases until only one white company survives in the end.

For $b = 1.5$, Fig. 49.7 shows that even though more blacks can be seen in stage 10, the dynamics of the game for this value of b is more intense and therefore the companies change the strategies more often. If we increase the value of b to 1.9 (very close to the limit of 2), all the companies choose strategy B just after the second interaction.

With the conformist behavior, a similar situation happens. We can see in Fig. 49.9 that after five interactions two companies with strategy B survives in the world of A's with a low b. For b = 1.5 the instability occurs once again and for a high b, the "whites" are the majority. But there are a few "blacks" that survive, more than in the greedy behavior situation.

As for the Pavlovian behavior Figs. 49.12–49.14 show that there are even more changing for all values of b. Figure 49.15 shows how the frequencies of A's (solid line) and B's (dashed line) change. The frequencies start at the same point (0.5) as we start with an initial matrix with a 50–50% of A's and B's. As time passes, the solid line goes up and the dashed line goes down, showing that there are more A's and B's for a b of 1.1. As the value of the b increases, the dashed line exceeds the solid line for all behaviors. We notice though that the competitions between A's and B's are harder for the smarter companies. This can seen as looking on how close (and how much they move up and down) the solid and dashed lines are for the situation where companies use the Pavlovian or the conformist behavior in relation to the situation where they use the greedy behavior (our less intelligent behavior).

Concerning the payoffs we notice from the Table 49.4 that in general the average payoff of all companies decreases as we increase the value of b. We can also notice that our smarter behavior (Pavlovian) scores better in situations with higher values of b. For b = 1.9, Pavlovians have an average payoff of 90.2 while the conformists have 50.5 and the greedy agents have only 8.7. Additionally we can see that when we have more companies with strategy A (small b), ie. selling a product for the same price, we get higher average payoff for all behaviors. The maximum individual payoff is also higher for small values of b when the companies use behavior greedy or conformist but when they use the Pavlovian the maximum is reached when b is high (1.9).

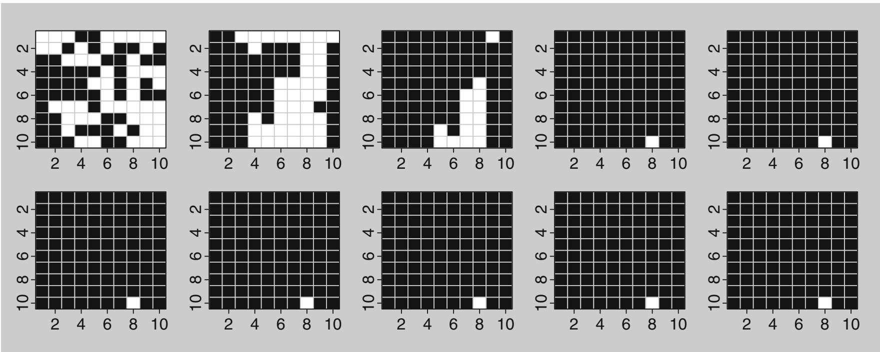


Fig. 49.6 The evolution of strategies A (*black*) and B (*white*) using the Greedy behavior and $b = 1.1$

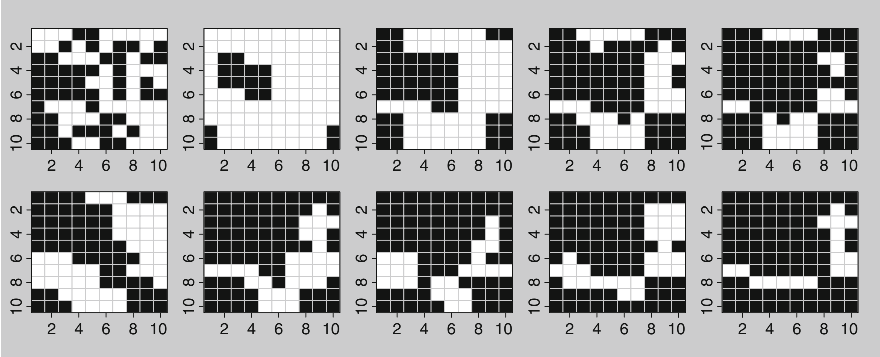


Fig. 49.7 The evolution of strategies A (*black*) and B (*white*) using the Greedy behavior and $b = 1.5$

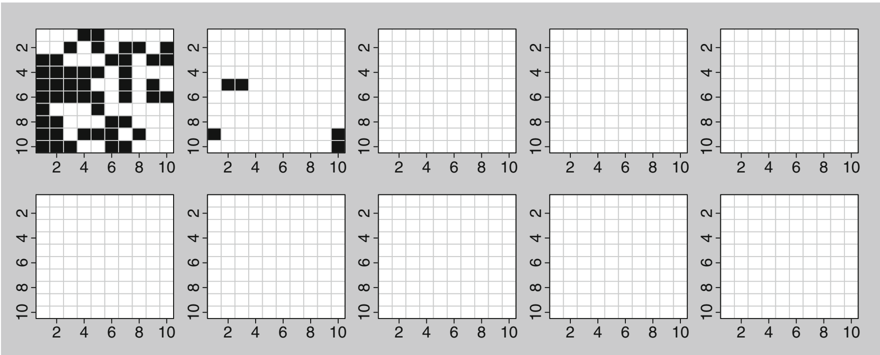


Fig. 49.8 The evolution of strategies A (*black*) and B (*white*) using the Greedy behavior and $b = 1.9$

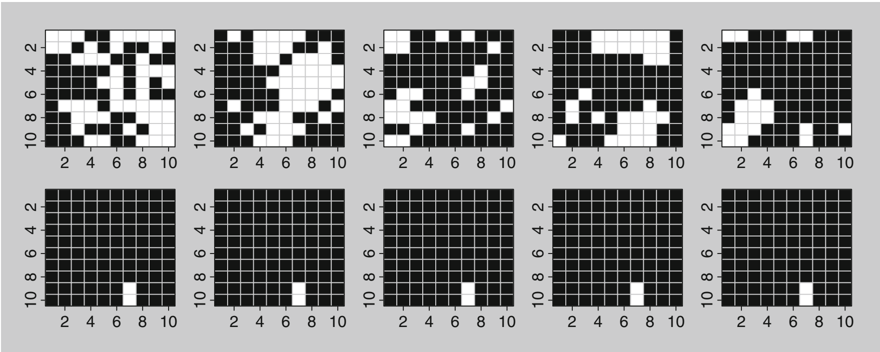


Fig. 49.9 The evolution of strategies A (*black*) and B (*white*) using the Conformist behavior and $b = 1.1$

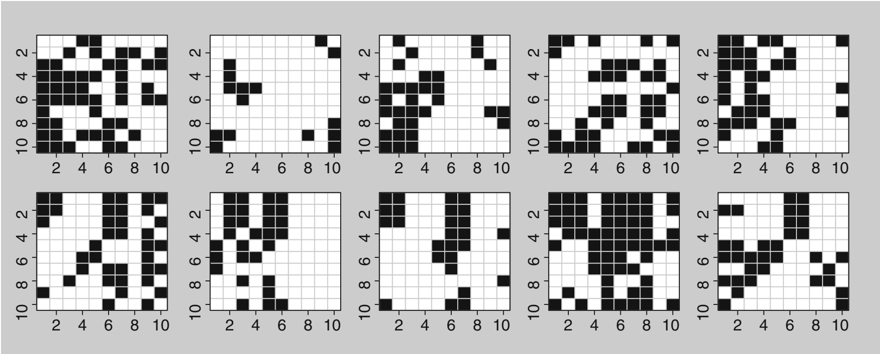


Fig. 49.10 The evolution of strategies A (*black*) and B (*white*) using the Conformist behavior and $b = 1.5$

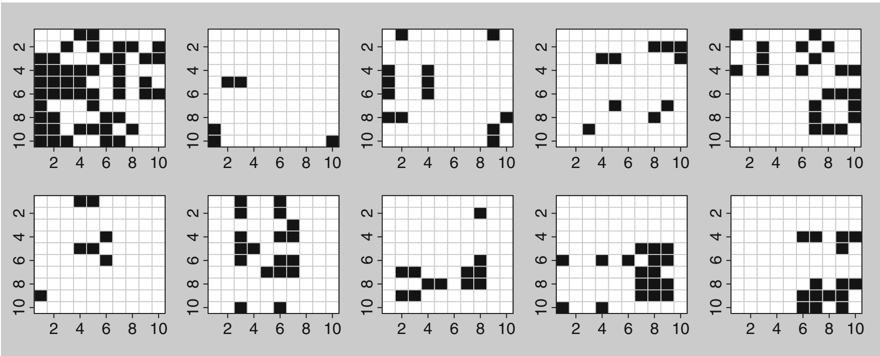


Fig. 49.11 The evolution of strategies A (*black*) and B (*white*) using the Conformist behavior and $b = 1.9$

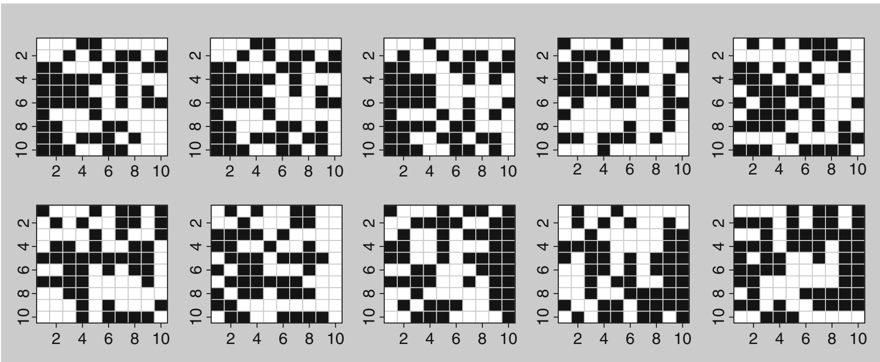


Fig. 49.12 The evolution of strategies A (*black*) and B (*white*) using the Pavlovian behavior and $b = 1.1$

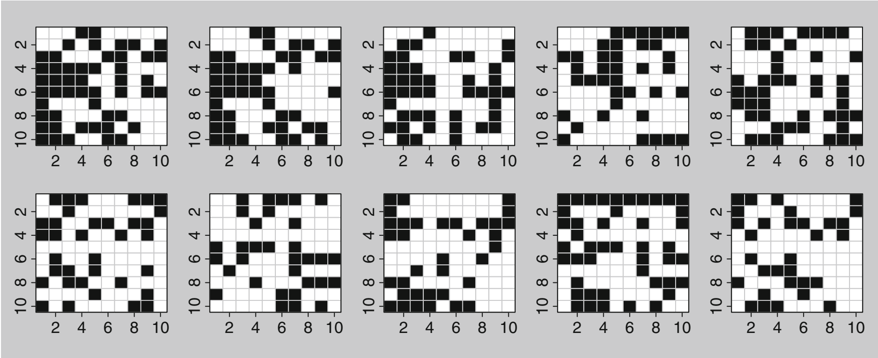


Fig. 49.13 The evolution of strategies A (*black*) and B (*white*) using the Pavlovian behavior and $b = 1.5$

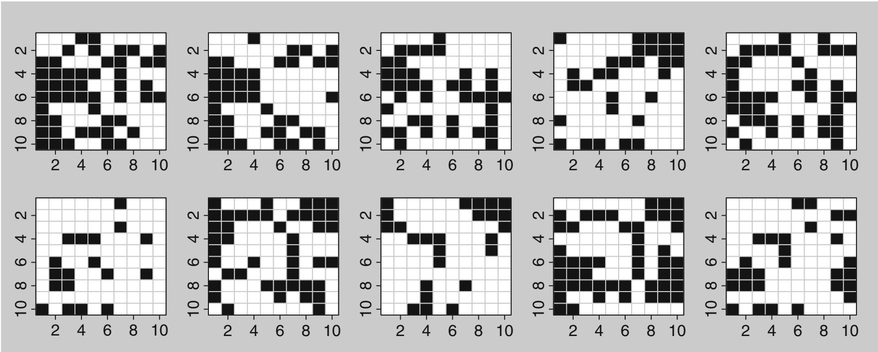


Fig. 49.14 The evolution of strategies A (*black*) and B (*white*) using the Pavlovian behavior and $b = 1.9$

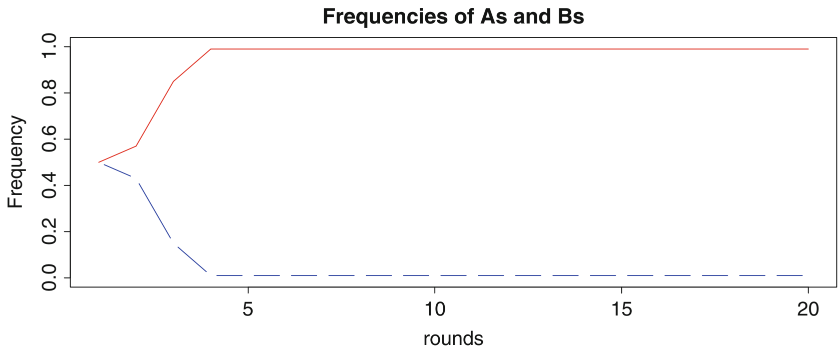


Fig. 49.15 The frequencies of strategies A and B for the Greedy behavior ($b = 1.1$)

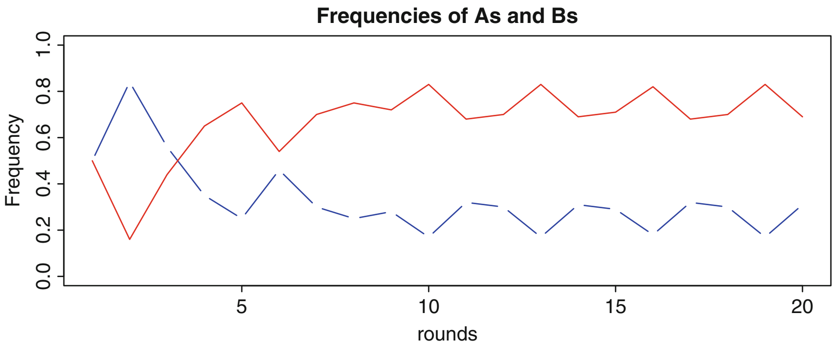


Fig. 49.16 The frequencies of strategies A and B for the Greedy behavior ($b = 1.5$)

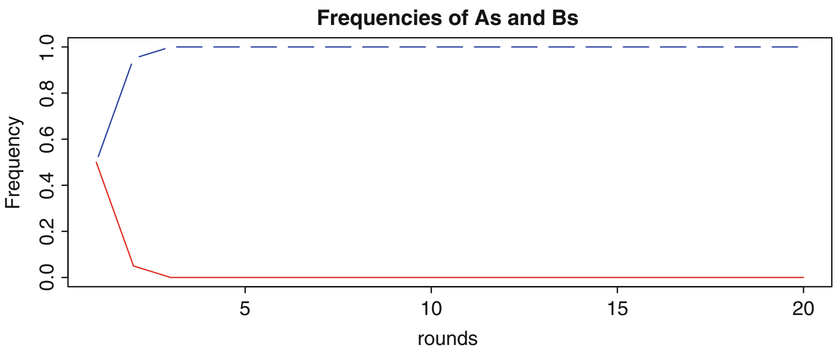


Fig. 49.17 The frequencies of strategies A and B for the Greedy behavior ($b = 1.9$)

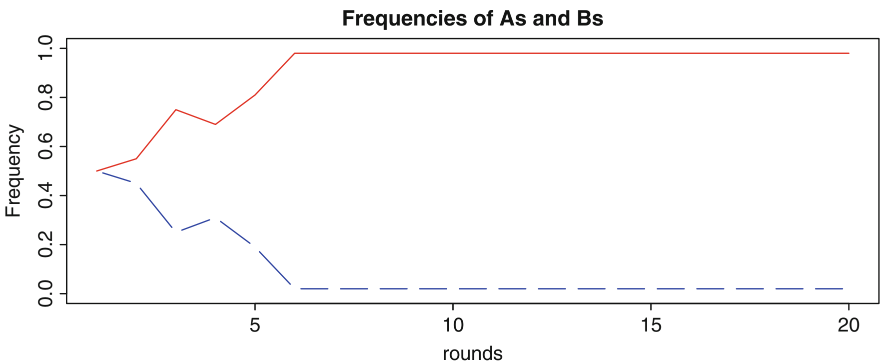


Fig. 49.18 The frequencies of strategies A and B for the Conformist behavior ($b = 1.1$)

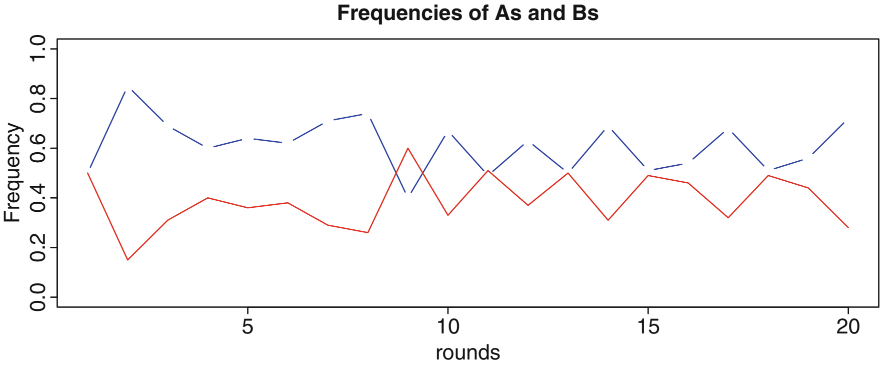


Fig. 49.19 The frequencies of strategies A and B for the Conformist behavior ($b = 1.5$)

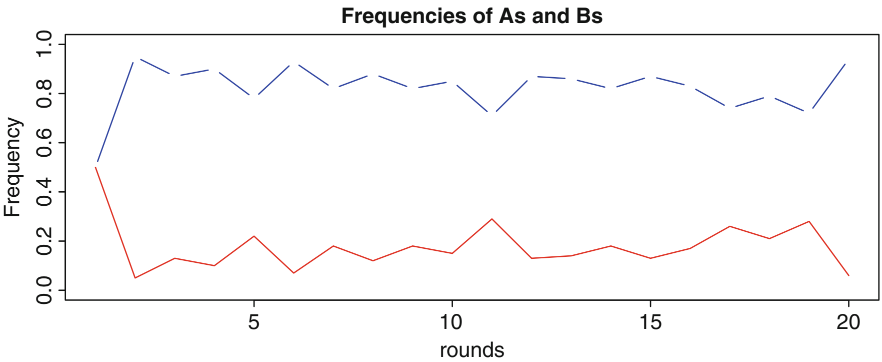


Fig. 49.20 The frequencies of strategies A and B for the Conformist behavior ($b = 1.9$)

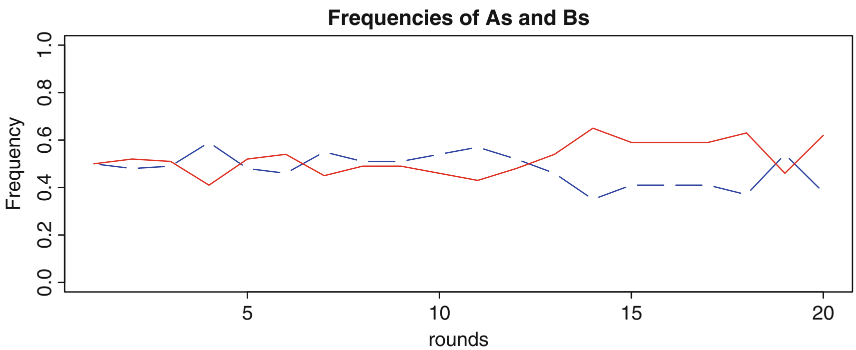


Fig. 49.21 The frequencies of strategies A and B for the Pavlovian behavior ($b = 1.1$)

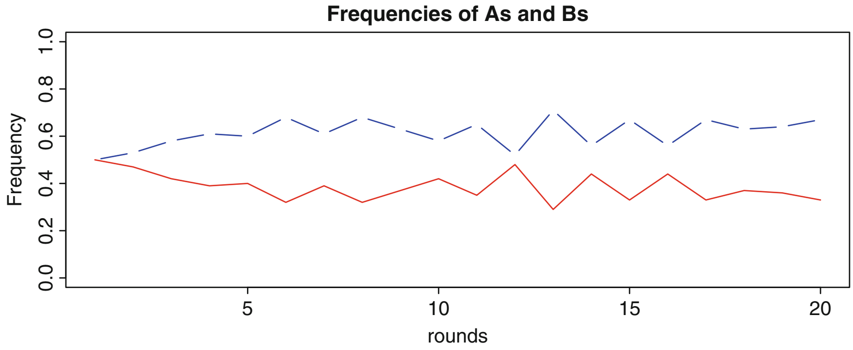


Fig. 49.22 The frequencies of strategies A and B for the Pavlovian behavior ($b = 1.5$)

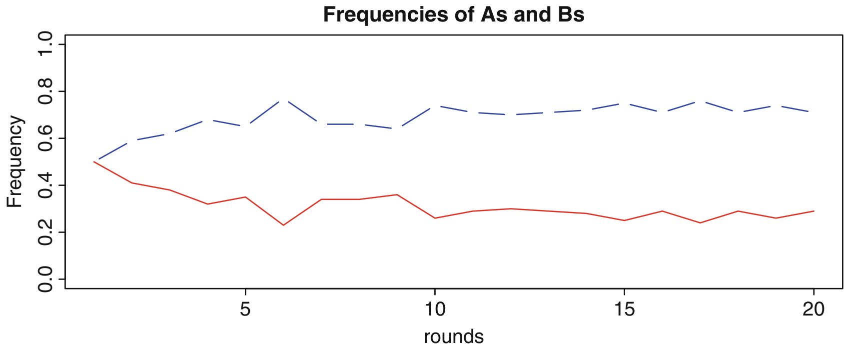


Fig. 49.23 The frequencies of strategies A and B for the Pavlovian behavior ($b = 1.9$)

49.4.2 Multi-Behaviors

In this section we investigate the situation where there are groups of companies with different behaviors and we check which group of behavior gets the highest payoff (wins the game). Our motivation comes from the fact that not all companies have the same behaviors, in reality. Even though, this is still a simplification of the reality that can give some insight into the competition between firms.

We created and tested the following scenarios:

- Scenario 1: The initial matrix is M1: there are 90% of companies using strategy B and 10% using strategy A
- Scenario 2: The initial matrix is M2: there are 90% of companies using strategy A and 10% using strategy B
- Scenario 3: The initial matrix is M3: there are 50/50% of companies using strategies A/B

For each scenario we change the value of b (1.2, 1.5 and 1.9) and create a game with three possible percentages of behaviors (Fig. 49.5):

- Balanced: similar percentage of behaviors (33% of greedy, 33% of conformist and 34% of Pavlovian)
- More greedy (90% of greedy, 5% of conformist and 5% of Pavlovian)
- More conformist (90% of conformist, 5% of greedy and 5% of Pavlovian)
- More Pavlovian (90% of Pavlovian, 5% of conformist and 5% of greedy)

The results can be seen in Table 49.5. This table shows the average payoffs of the sets of companies with one of the three behaviors. In scenario 1 and more greedy

Table 49.5 Average payoffs of the multi-behaviors situation

	Balanced	More Greedy	More Conformist	More Pavlovian
Scenario 1	(b = 1.2)			
Pavlovian	127.259	13.054	135.182	91.978
Conformist	132.244	7.714	148.697	89.706
Greedy	126.233	7.979	157.482	91.162
	(b = 1.5)			
Pavlovian	73.747	12.486	92.894	79.452
Conformist	77.639	8.558	90.121	90.752
Greedy	79.195	9.003	96.132	88.482
	(b = 1.9)			
Pavlovian	50.476	12.806	12.630	81.573
Conformist	50.706	10.318	10.875	93.422
Greedy	60.863	10.896	10.872	92.566
Scenario 2	(b = 1.2)			
Pavlovian	164.789	194.738	182.158	120.832
Conformist	171.726	192.414	183.389	121.764
Greedy	170.108	192.699	184.828	124.860
	(b = 1.5)			
Pavlovian	79.289	139.104	112.756	111.431
Conformist	81.822	154.916	110.912	120.320
Greedy	82.152	150.755	110.558	115.842
	(b = 1.9)			
Pavlovian	65.849	89.724	61.702	99.994
Conformist	71.692	127.056	69.413	122.212
Greedy	79.091	112.997	70.626	123.532
Scenario 3	(b = 1.2)			
Pavlovian	125.418	175.840	159.234	113.844
Conformist	137.533	185.380	171.578	115.448
Greedy	130.555	173.732	169.214	118.768
	(b = 1.5)			
Pavlovian	81.914	133.956	104.500	88.550
Conformist	86.944	143.686	104.207	100.012
Greedy	87.807	129.863	98.998	105.756
	(b = 1.9)			
Pavlovian	57.749	17.238	51.898	88.642
Conformist	58.934	17.038	58.291	86.646
Greedy	68.102	16.101	60.890	116.872

environment we can see the advantage of the group of Pavlovians as their mean payoffs are always higher than for the other behaviors for all values of b. For the other possible combinations it is harder to make conclusions because the mean values fluctuates fairly often.

49.5 Data Analysis

According to Epstein [4], simulation is a particularly tool, when the aim is to establish that some set of micro assumptions is sufficient to generate a macro phenomenon of awareness. The behavior emerged from the complexity generated by the moving agents in our spatial geometry, is analyzed by implementing experiments and link analysis methods. Masuda and Aihara [15] found that for intermediate values of b, small-world architecture realizes a quasi-optimal behavior in the sense of rapid convergence to a good equilibrium. Here it is implicitly measured by a hierarchy of states.

49.5.1 Link Analysis

As Luo, Chakraborty, Sycara [14] in their research about the Prisoner's Dilemma game in a graph, we also use multiple types of agents. As them, we assume there are different types of agents forming the nodes of the graph. Prisoner's Dilemmas game in graphs with synchronized strategy update, is a game where the graph topology is assumed to be essential to analyze. So we decide to construct the network and observe the results using a special library present in R a package called "sna" – Social Network Analysis tool. We use an undirected graph $G = (V, E)$ to represent the agents of groups and their connections, where $V = v_i | i = 1, ..., n$ is a set of n nodes representing the set of n agents, and $E = (v_i, v_j) | i \neq j, i, j \in 1, ..., n$ represents a set of edges so that $(v_i, v_j) \in E$ if v_i and v_j are connected to each other (Fig. 49.21).

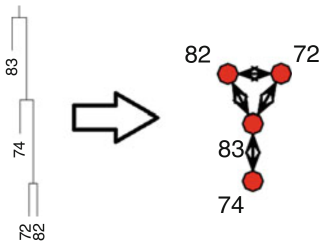
Szabó and Fáth [20] in evolutionary games on graphs define its evolutionary form when the interacting agents are linked in a specific social network, the core solution concepts and methods are very similar to those applied in non-equilibrium statistical physics. To also evaluate this situation, we produced simulations to compare the three b values (payoffs). In this section we build an adjacency matrix. Our aim is being able to according the strategy matrix, for each round, connect cooperating strategy agents. These connected agents form graph components. For this link analysis, we play 20 rounds, with 100 agents using the innovations strategies A and B. It is worth mentioning that in the evolution of spatial games, social preference has received negligible attention, according to Xianyu [23] although it has been accepted that the structure of agent interaction indeed plays an important role in a significant number of spatial games. In our study we are reverting this situation giving absolutely attention to social behaviors (Fig. 49.22).

49.5.2 The Graphs Payoff-Based Link Analysis

Jun and Sethi [9] visualize an interesting implication where there is a sense in which dense networks are more conducive to the evolution of cooperation than sparse networks. In our study we explore the graph density and according to the results for b = 1.1 (density = 0.06909091), for b = 1.5 (density = 0.01151515) and finally for b = 1.9 (density = 0.001616162). We also verify that the graph for b = 1.1 has higher density as cooperation strategy is used much more (Fig. 49.23).

We observe that for b = 1.9 we almost did not pick significant information because link analysis is stronger when cooperators are winning, which means lowest payoffs. We observe centrality positions in lowest b values. The stronger behavior from each strategy was furthermore reviewed finding strong components, seeking who is in what component, what are the component sizes, and which is the largest component. We observe that for higher b values (highest payoffs), most of all agents are in diverse components. When the payoffs are smaller (lower b values), the agents are in less components. We finally verify that the number of agents getting into the largest component increases with the decreasing of b value. That's because number of defecting agents is negatively affected by lowering the b value. (Figs. 49.24–49.30).

Fig. 49.24 Data example: Cluster Dendrogram and Graph for corresponding cooperator cluster, following Innovation Strategy A



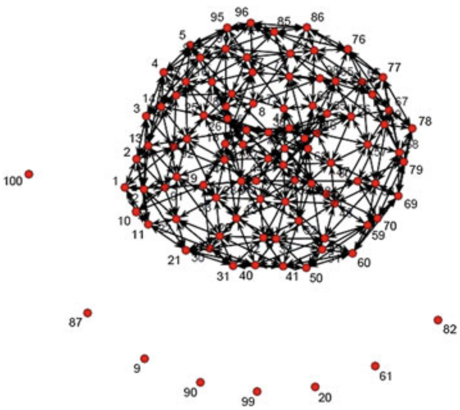


Fig. 49.25 Graph plot for $b = 1.1$ – network composed by 92 Cooperate Agents (Strategy A) and 8 Defect Agents (Strategy B)

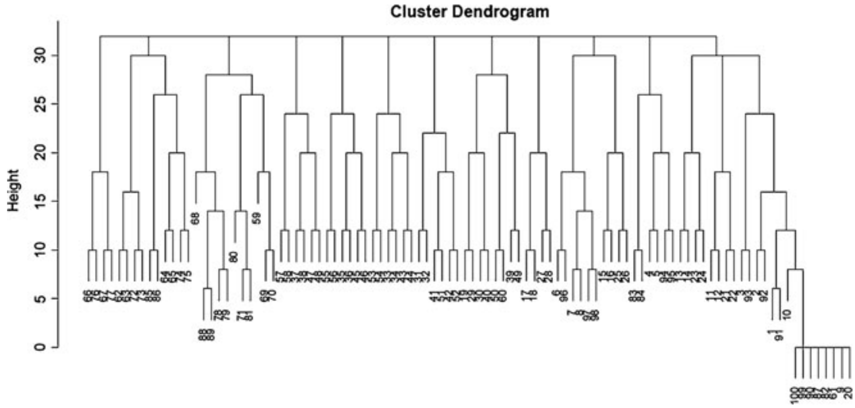


Fig. 49.26 Cluster Dendrogram for $b = 1.1$ – network composed by 92 Cooperate Agents (Strategy A) and 8 Defect Agents (Strategy B)

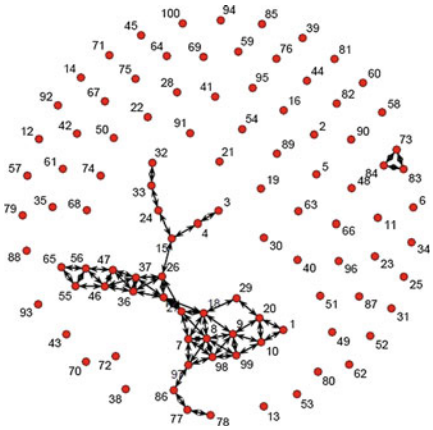


Fig. 49.27 Graph plot for $b = 1.5$ – network composed by 35 Cooperate Agents (Strategy A) and 65 Defect Agents (Strategy B)

Through link analysis we observe, like we did with statistics analysis, that the innovation Strategy A (cooperate) is dominant for lowest payoffs, and Strategy B (defect) dominant for high payoffs.

In Figs. 49.31–49.33 we see three different scenarios, each one having three different payoffs, where we observe (by link analysis) graphs representing different strategy matrices. We change the initial strategy A and strategy B portion.

Then in Fig. 49.34 we use the same initial strategy matrix, but agents use different behaviors all of them coexisting in the same playground. We change Greedy (G), Conformist (C) and Pavlovian (P) behaviors portion this time. The down-right plot shows the more balanced situation. The up-right filled with almost all cooperators, shows the more Conformist, the up-left more Greedy and the down-left more Pavlovian situations.

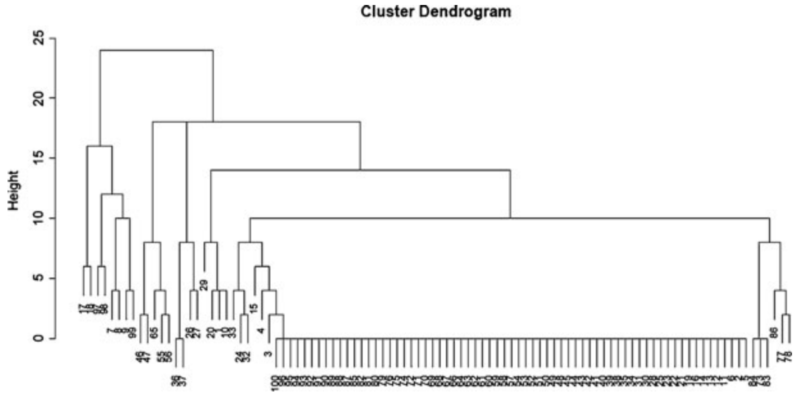
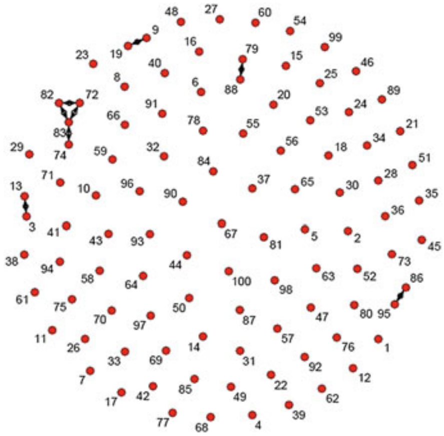


Fig. 49.28 Cluster Dendrogram for $b = 1.5$ – network composed by 35 Cooperate Agents (Strategy A) and 65 Defect Agents (Strategy B)

Fig. 49.29 Graph plot for $b = 1.9$ – network composed by 12 Cooperate Agents (Strategy A) and 88 Defect Agents (Strategy B)



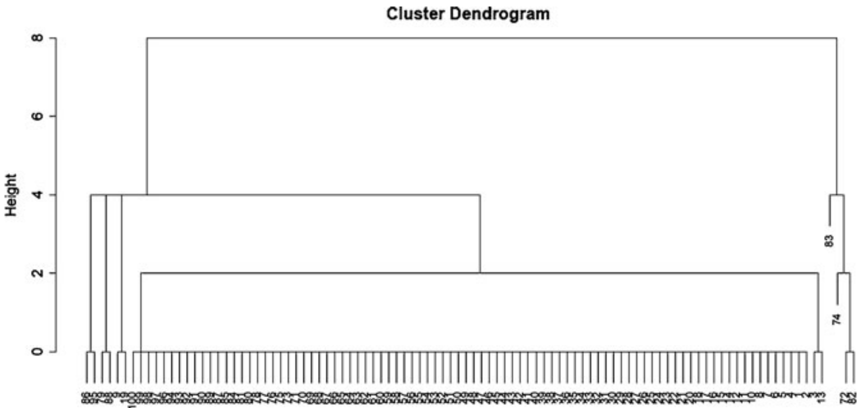


Fig. 49.30 Cluster Dendrogram for $b = 1.9$ – network composed by 12 Cooperate Agents (Strategy A) and 88 Defect Agents (Strategy B)

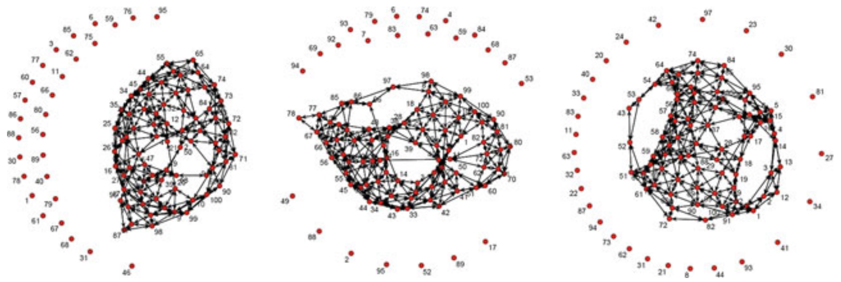


Fig. 49.31 Same behaviors, 20 rounds, payoff = 1.2. Different initial strategy matrices (90% - A, 10% - B); (10% - A, 90% - B); (50% - A, 50% - B)

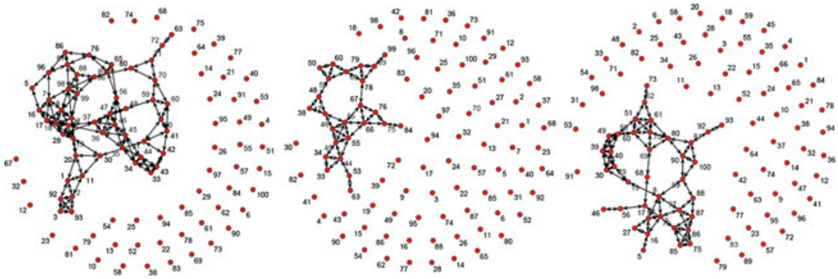


Fig. 49.32 Same behaviors, 20 rounds, payoff = 1.5. Different initial strategy matrices (90% - A, 10% - B); (10% - A, 90% - B); (50% - A, 50% - B)

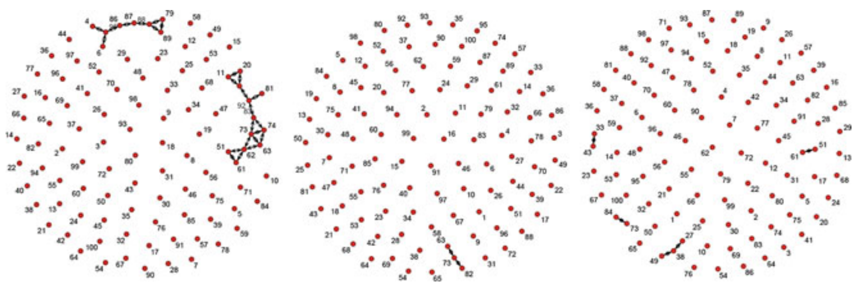


Fig. 49.33 Same behaviors, 20 rounds, payoff = 1.9. Different initial strategy matrices (90% - A, 10% - B); (10% - A, 90% - B); (50% - A, 50% - B)

In this paper our main concern is to analyze how the graph structure of interactions can modify and enrich the representation of behavioral patterns emerging in evolutionary games. One of our innovation strategies leading this research was based on using social network analysis, to evaluate b values, not just statistically, as it usually appears in most of the Prisoner's Dilemma studies.

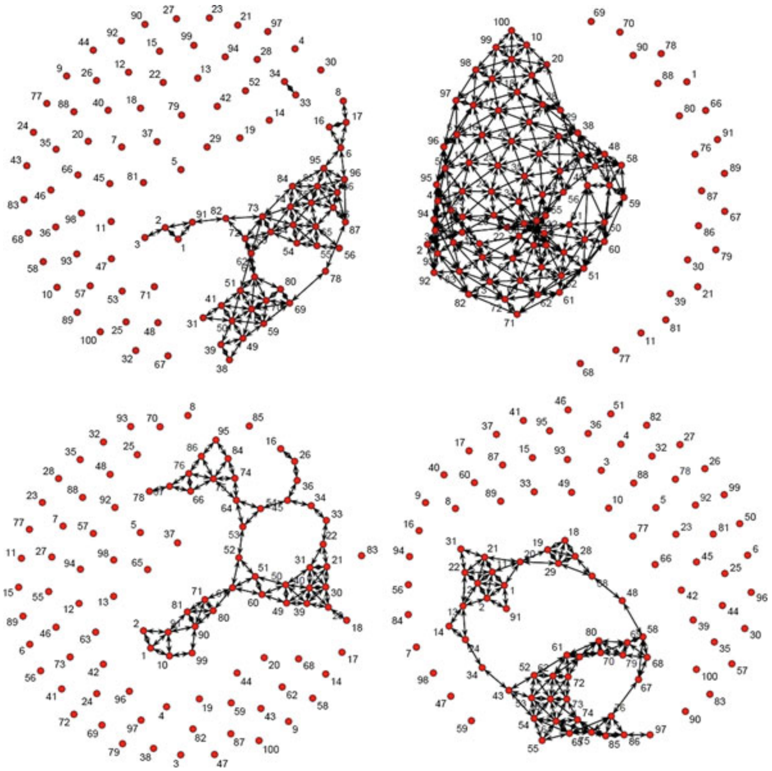


Fig. 49.34 Same initial strategy matrix, 20 rounds, payoff = 1.5. Different behaviors: (90% - G, 5% - C, 5% - P); (5% - G, 90% - C, 5% - P); (5% - G, 5% - C, 90% - P);(33% - G, 33% - C, 34% -P)

49.6 Dynamics GIS

Following the new Essays on Geography and GIS on GIS Best Practices, ESRI article [1], we felt the need to develop a GIS tool which allows us to acquire information from a base map of real-world locations. Bringing us the possibility to combine different data sets performed dynamically, we thought about a space-time combination which could provide a spatiotemporal understand and the possibility to predict reality. Dynamics GIS described by May Yuan [1] seemed the perfect tool to achieve our aim once we need to perform knowledgeable decisions, and analyze adaptation strategies for this dynamic world multi-agent-based.

Using especially agent-based modeling, with spatiotemporal analysis we became able to extract spatiotemporal information, survey clusters and change detection. Knowing that the complexity of reality can be understood by analyzing spatial and social organizations, by representing an dynamics GIS we can understand the structure connecting subsystems and supersystems, according to May Yuan [1] and also detecting existent hierarchies, like we did in Sect. 49.5.1 with link analysis methods, and now with a spatiotemporal representation.

We build a system, so we can be able to obtain georeferenced information about innovation strategies A and B. To do so, we developed a methodology using a cartographic projection designed to create a coordinate system, since we need to represent objects in space. We start modeling our data because we want to link different datasets together by the fact they relate to specific and also different geographic locations. We set one of the flat representation system used in Portuguese mapping, Hayford-Gauss System. We use the datum for the planimetric geodetic network based on the International Hayford ellipsoid parameters: $a = 6378388$ (semi-major axis), $f = 1/297$ (flattening), $e^2 = 0.00672267002233$ (squared eccentricity). The ellipsoid is positioned at the Central Meridian, $\lambda = 9°07'54.862''W$ (geodetic longitude). We use a file containing the geodetic latitude and longitude of the geodetic border points in Portugal. The longitude of the points was adjusted to the Central Meridian. We use Transverse Mercator Projection (also named as Gauss Projection), is a conformal projection, where the lengths remains along a meridian called the Central Meridian of projection, the origin of the ordinate is at the equator, the origin of abscises is at Central Meridian. It can be defined by the following equations:

In the Central Meridian,

$$\begin{cases} x = 0 \\ y = S\varphi = \int_0^{\varphi} \rho d\varphi \end{cases} \tag{49.23}$$

$$\rho = \frac{a(1-e^2)}{\sqrt{(1-e^2 sin^2\varphi)^3}} \tag{49.24}$$

Formulas for direct transformation of Transverse Mercator projection,

$$S\varphi = a[A_0\varphi - A_2 \sin 2\varphi + A_4 \sin 4\varphi + ...] \tag{49.25}$$

Obtaining the different A values by the following equation,

$$\begin{aligned} A_0 &= 1 - \frac{1}{4}e^2 - \frac{3}{64}e^4 + ... \\ A_2 &= \frac{3}{8}(e^2 + \frac{1}{4}e^4 + ...) \\ A_4 &= \frac{15}{256}(e^4 + ... \end{aligned} \tag{49.26}$$

$$\begin{cases} x = N\cos\varphi * \lambda \\ y = S\varphi + \frac{1}{2}N\cos\varphi * \sin\varphi * \lambda^2 \end{cases} \tag{49.27}$$

Where φ is the Geodetic Latitude, λ Geodetic Longitude, a semi-major axis of the ellipsoid, e eccentricity of the ellipsoid, N 1st Vertical Curvature Radius, ρ Meridian Curvature Radius and finally x and y our agents position according to all this parameters previously defined.

49.6.1 Dynamics Business Parks

Using the equations defined in the Sect. 49.6 we build the algorithm in *Matlab* to perform our dynamics GIS. Modeling geographic dynamics, we obtain each agent or company georeferenced information in dynamic reality, we now know, who is playing, which strategy behavior, where and when.

We set three different Business Parks: North Park, Centre Park and South Park. They are fictitious (not existent business parks) but referred to existent (real-world) positions. Three Business Parks, with 100 agents each, each agent with a calculated (referenced-based) position, based on the parameters establish on the previous section (Fig. 49.35).

In our research the agent or company is located always in the same position, as we established on Sect. 49.2.2 but each agent changes his states according to game dynamics related to psychological geography (and geometry, in the context of the discrete model cellular automata), deeply well-marked in this single and multi-behavior spatial prisoner's dilemma dynamics (Fig. 49.36).

This technique allows us to survey in a long run which is the most used innovation strategy per company, per business park and per country. Scanning in any direction, performing zoom in and out, agents or companies positions can be observed looking into the axis, or by numbers in output coordinates (Fig. 49.37).

With this dynamics GIS tool and sample analysis we are suggesting a spatiotemporal Business Park classification, this temporal classification plus temporal agents or companies gains, would provide significant information to the construction of some kind of Business Park Dynamics Rating System. If globally applied

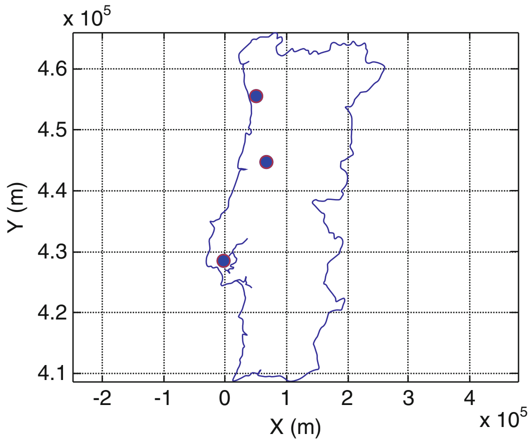


Fig. 49.35 Portugal Transverse Mercator Projection. Three Dynamics Business Parks playing spatial prisoner's dilemma innovation strategies A and B, with single and multi-behaviors. North Business Park, Centre Business Park, South Business Park

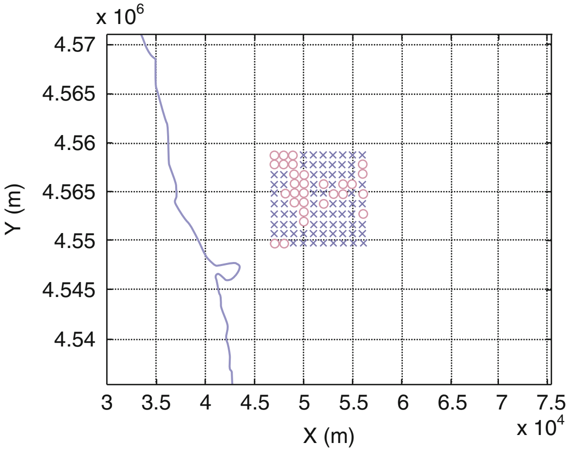


Fig. 49.36 North Business Park, playing spatial prisoner's dilemma innovation strategies A (x) and B (o), with more Pavlovian behavior

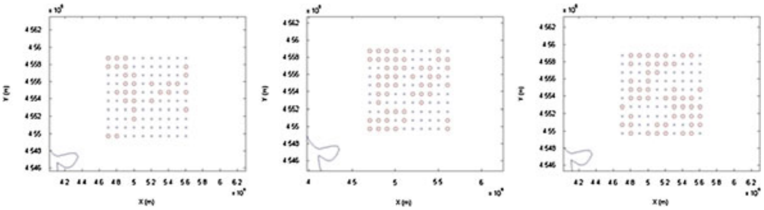


Fig. 49.37 North Business Park playing innovation strategies A (x) and B (o) with more Pavlovian behavior. At time units, *left*: $t = 1$, (strategy A wins), *middle*: $t = 2$ (strategy B wins), *right*: $t = 3$ (no winning strategy, it's balanced). During this three time units analysis we can classify North Business Park as following innovation strategy A and B, a balanced strategy behavior

innovation strategy A or B analyzing who is obtaining bigger gains, not just a payoff-based analysis but also a social analysis we can rate the country following the winning strategy with higher global results, basing our full-model analysis in human growth contribution, welfare and development. Obviously much more parameters enter in this kind of rating; we are just contributing with a small step. Since change and movement are two essential elements in temporal GIS, in this research we didn't define the time topology, as Giacomo Bonanno [3] did, when describing perfect information games time, agents predictions in branching time (also referred by May Yuan [1] on GIS Best Practices), perhaps when we build the future work, in Sect. 49.8, we can explore temporal definitions, associating to each agent or company prediction strategies linked to their spatiotemporal definitions (Fig. 49.38).

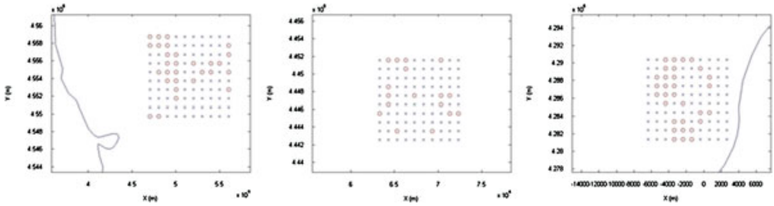


Fig. 49.38 At time unit $t = 1$, each business park is using one kind of behavior when playing innovation strategies A (x) and B (o). *left*: North Business Park (more Pavlovian behavior – strategy A wins), middle: Centre Business Park (more greedy behavior – strategy A wins), right: South Business Park (more conformist behavior – strategy A wins). During this one time unit analysis we can classify Portuguese Business Parks as following innovation strategy A.

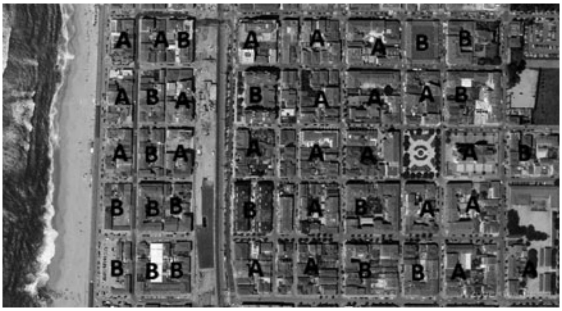


Fig. 49.39 Illustrative fictitious Business Park playing Innovation Strategies A and B. Image font: Google Earth, Espinho, Portugal.

49.6.2 Another Real: World Application

Robert Axelrod and John Holland, mentioned by Hoffmann and Waring [7] as responsible for important discoveries such as the Tit-For-Tat strategy in Prisoners' dilemma (Axelrod) or the genetic algorithms (Holland), consider social tagging as a means of influencing game outcome. In the context of the social sciences, the interactions and learning simulations take place within a neighborhood of players. Neighborhoods can be formed geographically, or in a more abstract sense, such as in the range of partners available for trading.

So we thought in a Real – World Application for our study, based in many few companies clustered in parks, business parks. These parks for many reasons need to be rated and we thought that some kind of rating system should exist. Each company belonging to a park, plays Spatial Prisoner's Dilemma, which means: we could classify the parks with the classification following innovation Strategy A or Strategy B, like we did to each company or agent in the paper.

There are a lot of parameters entering in parks classification based in some theories; we believe our research may well contribute in some way to a Real-World Business Park Dynamics Rating System.

49.7 Discussion

In this work, the experiments are presented in two different ways depending on the type of behavior: single behavior and multi-behavior. In the single behavior situation, we noticed that in general, the average payoff of all companies decreases as we increase the value of b. It is also possible to conclude that Pavlovian behavior scores better then Greedy and Conformist when b is higher. Additionally we can see that when we have more companies with strategy A (small b) we get higher average payoffs in all behaviors. The maximum individual payoff is also higher for small values of b when companies use behavior Greedy or Conformist. In the multi-behavior situation (different percentages of Pavlovian, Greedy and Conformists are defined), and in the scenario 1 (90% of companies use strategy B and 10% use strategy A) in the more greedy environment, we state the Pavlovian behavior dominates for all values of b.

Results of link analysis show that for greater values of b, the density of the network decreases. In addition, it is possible to see that Strategy A is dominant for lower payoffs (defined by the b values) and strategy is dominant for higher payoffs.

In some researches, players in prisoner's dilemma are modeled as learning automaton; such models can be viewed as complicated variants of reinforcement learners. However, following Wakano and Yamamura [21], in our research we were also interested in the behavior of the simplest form of reinforcement learner. During the study we start asking ourselves if Pavlovian would be more efficient than the Conformist, or the simplest one, the Greedy. Our result shows that agents with a simple and instinctively familiar learning rule put up a very efficient and tough cooperation in some situations. At the same time, Kraines and Kraines [13] prove that a society of fast adapting agents may experience conflict and disagreement while another society with slower learning agents will benefit in cooperation. As Alexander Pope was mentioned in Kraines and Kraines [13] leaving the following thought "a little learning is a valuable thing and it is too much learning that is dangerous"–we feel like being in a great adventure, so we will just keep walking into knowledge flow.

49.8 Future Work

Collect geo-spatial data in Prisoner's Dilemma theory game and build real models based on geo-referenced images representing real world situations, using Geographic Information Systems tools. Once the dynamics is implemented, the major concern is the long run behavior of the system: fixed points, cycles, and their stability, chaos, other parameters, and the connection between static concept as Nash equilibrium or evolutionary stability, and dynamic predictions. According to Wolfram [22] even when the general problem is undecidable, the emergence of particular finite sequences in the limit set for a cellular automaton may be decidable. We need revolutionary new concepts and methods for the categorization of the rising complex and self-organizing patterns. One could be the development of dynamics GIS using Multi-Agent Systems with dynamics predictions.

References

1. Yuan. M., Chrisman, N., "Jack" Owens, J.B., Dobson, J.E., Goodchild, M.F., Moll, G., Gallis, M., Millar, H., Getis, A., Wrigth, D.J.: GIS Best Practices. Essays on Geography and GIS. ESRI, Sept (2008)
2. Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984)
3. Bonanno, G.: Branching time, perfect information games and backward induction. Department of Economics, University of California, Nov (1999)
4. Epstein, J.M.: Zones of cooperation in demographic prisoner's dilemma. Complexity **4**, 36–48 (1996)
5. Epstein, J.M.: Zones of cooperation in demographic prisoner's dilemma. Working Papers 97-12-094, Santa Fe Institute, Dec (1997)
6. Tucker, A.W., Kuhn, H.W.: Contributions to the Theory of Games I. Annals of Mathematics Studies, no. 24. Princeton University Press, Princeton (1950)
7. Hoffmann, R., Waring, N.: The localisation of interaction and learning in the repeated prisoner's dilemma. Working Papers 96-08-064, Santa Fe Institute, Aug (1996)
8. Isaac, A.G.: Simulating evolutionary games: a python-based introduction. J. Artif. Soc. Soc. Simul. **11**(3), 8 (2008)
9. Jun, T., Sethi, R.: Neighborhood structure and the evolution of cooperation. J. Evol. Econ. **17**, 623–646 (2007)
10. Kirchkamp, O.: Spatial evolution of automata in the prisoner's dilemma. In: Social Science Microsimulation [Dagstuhl Seminar, May, 1995], pp. 307–358, London, UK. Springer, Berlin (1996)
11. Kirchkamp, O.: Spatial evolution of automata in the prisoners' dilemma. J. Econ. Behav. Organ. **43**(2), 239–262 (2000)
12. Kraines, D., Kraines, V.: Learning to cooperate with pavlov an adaptive strategy for the iterated prisoner's dilemma with noise. J. Theory Decis. **35**(2), 107–150 (1993)
13. Kraines, D., Kraines, V.: The threshold of cooperation among adaptive agents: Pavlov and the stag hunt. In: ECAI '96: Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages, pp. 219–231, London, UK. Springer, Berlin (1997)
14. Sycara, K., Luo, L., Chakraborty, N.: Prisoners dilemma on graphs with heterogeneous agents (2009)
15. Masuda, N., Aihara, K.: Spatial prisoner's dilemma optimally played in small-world networks. Phys. Lett. **13**, 55–61 (2003)
16. Nowak, M.A.: Evolutionary Dynamics: Exploring the Equations of Life. Harvard University Press, Cambridge, MA (2006)
17. Nowak, M.A., May, R.M.: The Spatial Dilemmas of Evolution. University of Oxford, Oxford (1992)
18. Power, C.: A spatial agent-based model of n-person prisoner's dilemma cooperation in a socio-geographic community. J. Artif. Soc. Soc. Simul. **12**, 1 (2009)
19. R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, (2009). ISBN 3-900051-07-0.
20. Szabó, G., Fáth, G.: Evolutionary Games on Graphs. Physics Reports (2007)
21. Wakano, J.Y., Yamamura, N.: A simple learning strategy that realizes robust cooperation better than pavlov in iterated prisoners' dilemma. J. Ethol. **19**, 1–8 (2001)
22. Wolfram, S.: Computation Theory of Cellular Automata. Princeton University Press, Princeton (1984)
23. Xianyu, B.: Social preference, incomplete information, and the evolution of ultimatum game in the small world networks: An agent-based approach. J. Artif. Soc. Soc. Simul. **13**, 2 (2010)

24. Wang, X.-L., Sheng, Z.-H., Hou, Y.-Z., Du, J.-G.: The Evolution of Cooperation with Memory, Learning, and Dynamic Preferential Selection in Spatial Prisoner's Dilemma Game. International Symposium on Nonlinear Dynamics. J. Phys. Conf. Ser. **96**, 1–6 (2007)
25. Zimmermann, M.G., Eguluz, V.M.: Cooperation, social networks, and the emergence of leadership in a prisoner's dilemma with adaptive local interactions. Phys. Rev. E **72**(5), 056118 (2005)