On the Level of Cooperative Behavior in a Local Interaction Model *

Alexander F. Tieman[†], Harold Houba[‡] and Gerard van der Laan[§]

1998

Abstract

We study local interaction within a population located on a connected graph. Subjects engage in several bilateral interactions at each round in a generalized Prisoners' Dilemma (PD). In each round of play one randomly selected player gets the possibility to update the action he plays in this PD. All individuals use the update rule 'Win Cooperate, Lose Defect', a multi-player variant of Tit-for-Tat. Theoretical results on the set of stable states of the associated dynamics are provided for the cases with and without rare mutations. Simulations provide insight into the probability distribution over these stable states. In both cases a rather high probability is assigned to stable states with a moderate level of cooperation, implying that dominated strategies are used. Furthermore, the probability of reaching the stable state with Nash equilibrium play is small.

Keywords: *Evolution, Local Interaction, Cooperation, Prisoner's Dilemma, Markov processes.*

JEL-code: C78.

First Version: February 1998.

*This is a revised version of the paper Cooperation in a Multi-Dimensional Local Interaction Model, Tinbergen Institute Discussion Paper TI 98-024/1, 1998. The authors would like to thank participants of the Spring Meeting of Young Economists 1998 in Berlin, of the Third Spanish Meeting on Game Theory and Applications in Barcelona, the Ninth Summer Festival on Game Theory at Stony Brook, the colloquim 'Theory of Markets and their Functioning' hosted by the Royal Academy of Arts and Sciences in Amsterdam and seminar participants at the University of Bergen, Norway and the anonymous referees for comments.

†Tinbergen Institute and Free University, Department of Econometrics, De Boelelaan 1105, 1081HV Amsterdam, The Netherlands. E-mail: xtieman@econ.vu.nl, Phone: +31-20-4446022, Fax: +31-20-4446020, URL: http://www.econ.vu.nl/medewerkers/xtieman/ .

‡Free University.

§Tinbergen Institute and Free University.

1. Introduction

Cooperative behavior is often observed in situations in which there is repeated interaction between individuals. This phenomenon is most interesting in a setting where the individuals would be strictly better off by non-cooperative behavior when the interaction would only take place once, as is the case in e.g. the Prisoners' Dilemma. Many repeated game models have tried to explain cooperative behavior in such a setting. These models often take (hyper)rationality of the agents in the game as their point of departure, i.e. they state that agents are able to foresee or predict (most of) the consequences of current behavior on future payoffs and that they are able to calculate the optimal strategy, given their predictive powers.

Sometimes the driving force behind models explaining cooperative behavior is some kind of revealed preference argument: If an individual cooperates with his opponent, although he seemingly has nothing to gain from that, it must be the case that he has an unobservable preference for cooperating over other actions. This is the same as saying that the model is actually misspecified in the sense that the agents play a game which is different from the one presented.

Recently, a different approach is taken by papers in the field of evolutionary game theory, see e.g. Eshel, Samuelson & Shaked (**?**), Schlag (**?**), Binmore & Samuelson (**?**, **?**), the imaginary discussion between representatives of different economic views in Selten (**?**) or the survey article by van der Laan & Tieman (**?**). This paper provides such an alternative model. We abandon the assumption that agents are rational utility-maximizers. Instead we propose that the agents follow simple behavioral rules or heuristics in updating their action, as is seen often in the field of behavioral game theory (see e.g. Camerer (**?**)). Furthermore this paper is in the line of *local interaction models*, see also e.g. Ellison (**?**, **?**). The population consists of agents located on a connected graph with undirected edges. Agents get to play the stage game only with agents in a subgroup of the population, called their *neighbors*. The group of neighbors consists of all agents located in positions on the graph directly adjacent to the location of the agent. The group of neighbors is different for each agent, although there may be substantial overlaps between the groups of neighbors of different agents.

In this paper we consider a model in which in each round of play, a randomly selected player plays the stage game with each of his neighbors. At each stage game a player plays one of k possible actions, labelled from the non-cooperative action 1 to the cooperative action k. Based upon the outcomes of the stage game the selected subject plays with his neighbors, he updates the action he will play in the next stage game by comparing his average payoff of playing the game with all his neighbors to the average payoff his neighbors got from playing the game with him. Put in the terminology of a Prisoners' Dilemma (PD), the update rule can be presented as follows. If the payoff for the agent is relatively high, he will act more cooperatively in the next round of play. If the outcome is relatively low, he will tend to display less cooperative behavior in the next round. This update rule is referred to as *'Win Cooperate, Lose Defect'* [1] is a standard one and that it does not imply that players who 'win' are (in the long run) better off than players who 'lose'. (WCLD). It is often encountered in experiments, both in economics (see e.g. Offerman, Sonnemans & Schram (**?**)) and in sociology (see e.g. Messick & Liebrand (**?**)). This update rule is a multi-player variant of the well know *Tit-for-Tat* (TfT) update rule.

An alternative rationale for this update rule can be found in the literature on aspiration levels (see e.g. **?**), Palomino & Vega-Redondo (**?**), Rabin (**?**), **?**), **?**), Thibaut & Kelley (**?**) or Kelley & Thibaut (**?**)). An *aspiration level* is the minimum payoff an agent requires in order to play a certain action. If his payoff falls short of his aspiration level, he will play a certain other action. The update rule WCLD above is the result of setting the aspiration level to be the average payoff of the neighbors an agent plays against. The aspiration level is thus endogenous. Whenever the agent gets a higher payoff than the average of his neighbors, he will tend to cooperation. When his payoff falls short of this number, he will play less cooperatively in the next round of play.

In this paper we show that the WCLD dynamics admit precisely as many stable states as there are actions in the stage game. In each stable state all players use the same action over and over again. Furthermore, if we start the dynamics in a random initial state, with each state having positive probability to be selected as initial state, each stable state has positive

[1] Note that the label *'Win Cooperate, Lose Defect'* is a standard one and that it does not imply that players who 'win' are (in the long run) better off than players who 'lose'.

probability of being reached. Simulations are performed for three stage games of different size (different numbers of actions), for two different payoff matrices and for two spatial structures. We find that the size does have an impact upon the simulation result: The more actions the less probable it becomes to reach the stable states in which everybody plays the low and the high labelled actions. The stable state with the lowest labelled action corresponds to the unique (strict) Nash equilibrium and, therefore, increasing the number of actions means the Nash equilibrium will be less probable to reach. Similar, the stable state with the highest labelled action corresponds to the stable state with full cooperation. So, increasing the number of actions makes full cooperation less probable also. However, an increase in the number of actions makes the actions around the median (labelled action), which can be regarded as representing moderate levels of cooperation, more probable. We run simulations with payoff matrices belonging to two different classes. The first class of matrices represents the situation in which the exact payoffs are irrelevant and only the subject's own action and the average action of his neighbors matter. For this class of stage games the probability distribution over all stable states is symmetric around the median action(s). For the second class of payoff matrices considered, the payoffs do matter. We show that this class of payoff matrices induces a bias toward cooperative behavior. This bias is confirmed in the simulations where skewed probability distributions over all stable states are observed. The spatial structure, being a circle and a torus, has no significant impact upon the simulation results.

In the literature on evolutionary game theory the notion of mutants has received much attention. The robustness of the above results if rare mutations are introduced to the WCLD dynamics is investigated. The mutation process we think of is one in which mutants are so rare that the expected time in between two mutants is very much greater than the expected time the WCLD dynamics need to reach one of the stable states. So, when a mutant occurs, typically the standard WCLD dynamics bring the system to one of the stable states before another mutant occurs. We show that under the presence of such rare mutations there exists a unique recurrent set, being the set of all stable states of the process without mutations. So, it is impossible to select one of the stable states as being the unique stochastically stable state, see e.g. **?**) for a definition. This result differs qualitatively from other results in the literature, where only one stable state is stochastically stable, e.g. **?**) or **?**). We apply a recent result in **?**) in order to obtain some theoretical results and also to implement a meaningful simulation to compute the probability distribution over the stable states in the recurrent set. These simulations will be referred to as long-run, in contrast with the first set of simulations mentioned above, to which we will refer as medium-run. The long-run simulations yield similar distributions over stable states as the medium-run simulations, although the interpretation of these results is different. In the medium-run, the model converges to a stable state and stays there forever after. The distribution over stable states thus only indicates the a priori probability of convergence to each stable state. The long-run distribution can be interpreted as the long-run fraction of the time the model will spend in each of the stable states, since occasional (indirect) transitions from one stable state to another stable state take place.

This paper is organized as follows. Section 2 presents the basic model and discusses some preliminary properties of two classes of payoff matrices used in the simulations. The theoretical results for the WCLD dynamics, i.e. the medium-run in our terminology, and the corresponding simulation results are presented in Section 3 respectively Section 4. Section 5 and Section 6 contain the theoretical respectively simulation results for the WCLD dynamics in case of rare mutations, i.e. the long-run.

2. The basic model

We consider a population of N players. These players are located at the vertices of a connected graph. Each player i, $i = 1, \ldots, N$ has $2 \leq m \leq N-1$ distinct edges to m different other players on the graph. We call these m players the neighbors of player i. We impose that the edges of the graph are undirected, i.e. whenever i is a neighbor of j, then j is also a neighbor of i. The class of such populations includes as special cases populations on circles and populations on tori.[2]

[2] For some integer $n \geq 3$, take $N = n^2$. Then the population of these N players is said to be located on a torus if for each player the location is given uniquely by a pair $x = (x_1, x_2)$ with $x_l \in \{1, \ldots, n\}$, $l = 1, 2$ and the player on location $x = (x_1, x_2)$ has the 8 players on the locations $y = (y_1, y_2)$ with $y_l = (x_l - 1) \bmod n$, x_l, $(x_l + 1) \bmod n$, $l = 1, 2$, except $y = x$ as his neighbors.

Each player only interacts directly with his neighbors. Since we want to focus on local interaction we typically suppose that $m \ll N$, i.e. the group of neighbors is only a small subset of the entire population. Interaction takes place during infinitely many rounds, labelled $t = 0, 1, 2, \ldots$. In each round of play one of the players is selected randomly with each individual being equally likely to be selected.[3] The selected player at time t is called the subject at time t. This subject at time t interacts with his m neighbors by playing the stage game with each of his neighbors. This stage game is taken to be a symmetric 2-player $k \times k$ bimatrix game (A, A') characterized by a payoff matrix $A = (\alpha_{a,b})_{a,b=1}^{k}$. So, playing action a against an opponent playing b gives a payoff $\alpha_{a,b}$ to the subject and a payoff $\alpha_{b,a}$ to his opponent. The subject plays the same action to each of his neighbors. The payoff matrix A is assumed to be a Prisoners' Dilemma matrix (PD-matrix).

Definition 2.1. *For $k \geq 2$, a $k \times k$ payoff matrix A such that*

(i) $\alpha_{a+1,a+1} > \alpha_{a,a},\ a = 1, \ldots, k-1,$

(ii) $2\alpha_{a+1,a+1} > \alpha_{a,b} + \alpha_{b,a},\ a = 1, \ldots, k-1,\ b = 1, \ldots, k,$

(iii) for any pair (a, b), $a, b = 1, \ldots, k-1$, the triple $(\alpha_{a+1,b}; \alpha_{a,b}; \alpha_{a,b+1})$ satisfies $\alpha_{a+1,b} < \alpha_{a,b} < \alpha_{a,b+1}$

is called a PD matrix.

So, for $k = 2$ the matrix A is a standard Prisoners' Dilemma and for $k > 2$ it has a 'Prisoners' Dilemma structure'. In a PD matrix the action pair $(1, 1)$ is the unique Nash equilibrium, and the action pair (k, k) maximizes the sum of the payoffs to both players and is the unique symmetric Pareto efficient outcome. Furthermore action $a,\ a = 1, \ldots, k-1$, dominates action $a+1$. Two strict subclasses of the class of PD matrices are the class of *PD BiLinear matrices* (**PDBL-matrix**) and the class of *PD Quadratic-Linear matrices* (**PDQL-matrix**).

[3] Although the process is in discrete time, a continuous time interpretation can easily be given by assigning identical i.i.d. Poisson processes to every player in the population. Now, let every player be selected at the times given by the Poisson process. Since the probability that two players are selected at the same point in time is 0, this selection procedure yields the same results as the discrete time process.

A PDBL matrix is a matrix A in which for given parameters $\gamma > 0$ and $\delta > 1$ it holds that $\alpha_{a+1,a} = \alpha_{a,a} - \gamma$ and $\alpha_{a,a+1} = \alpha_{a,a} + \delta \cdot \gamma$, resulting in $\alpha_{a,b} = \alpha_{1,1} - \gamma \cdot (a-1) + \delta \cdot \gamma \cdot (b-1)$, i.e. the payoff to the subject is linear decreasing in his own action and linear increasing in the action of his opponent. Well-known examples of PDBL matrices are so-called public goods games, also called voluntary contribution mechanisms (VCM). In a VCM, players allocate their endowment between a private and a public account. Then, the total amount in the public account is multiplied by a factor (> 1) and subsequently the money in the account is divided over all players. These games are often been studied in experimental economics, see e.g. **?**) for a specific experiment or **?**) for a general survey on public goods games.

A $k \times k$ matrix A is a PDQL matrix when for given positive parameters δ_1, δ_2 , δ_3 and δ_4 satisfying $\delta_1 > \delta_2 + (k-1)\delta_3$ and $(k-1)\delta_3 > \delta_4$ it holds that

$$\alpha_{a,b} = \alpha_{11} - \delta_1(a-1)^2 + \delta_2(a-1) + \delta_3(a-1)(b-1) - \delta_4(b-1),$$

i.e. the payoff to the subject is quadratic-concave in his own action and is linear increasing in the action of the opponent. The restrictions on the parameters δ_i, $i = 1, \ldots, 4$, see to it that the game is iteratively dominance solvable with a unique Nash equilibrium at $(1,1)$. Note that both the class of PDBL matrices and the class of PDQL matrices are strict subclasses of the class of supermodular matrices as defined in Topkis (**?**), i.e. the players' actions in the stage game are strategic complements. An example of a PDQL matrix is obtained when the stage game between the subject and his opponent is a Cournot quantity competition duopoly. We also provide an example featuring Bertrand price competition. This game does not satisfy the third condition of Definition 2.1, but only $\alpha_{a+1,a+1} < \alpha_{a,a+1}$, $a = 1, \ldots, k-1$, and $\alpha_{a,b} < \alpha_{a,b+1}$, $a = 1, \ldots k$, $b = 1, \ldots, k-1$. Nevertheless, this condition is sufficient for all the results obtained in this paper. We call a matrix satisfying this payoff structure a generalized PDQL matrix.

Example 2.2. We consider a symmetric duopoly game in which two suppliers i and j in a heterogeneous good market each choose an action out of a discrete set of prices. For given parameters $\gamma_0 > 0$ and $\gamma_1 > \gamma_2 > 0$, let the demand function of player $i = 1, 2$ be given by $D_i(p_i, p_j) = \gamma_0 - \gamma_1 p_i + \gamma_2 p_j$, with p_i the price of player i and p_j, $j = 1, 2$, $j \neq i$, the price of his opponent j. So, the demand for player i's product is declining in i's price, rising in j's price and it is more sensitive to player i's own price than it is to the price of his competitor. Furthermore, suppose that for both players the cost function is given by $C(q) = q$ with q the production. Then, for given prices p_i and p_j the profit of player i is equal to

$$\pi^i(p_i, p_j) = D_i(p_i, p_j) \cdot p_i - C(D_i(p_i, p_j)) = -\gamma_1 (p_i)^2 + (\gamma_0 + \gamma_1) p_i + \gamma_2 p_i p_j - \gamma_2 p_j - \gamma_0,$$

which is quadratic and concave in p_i and linear in p_j. Let p^N be the price set by both players in the (symmetric) Bertrand-Nash equilibrium price and let $p^C > p^N$ be the cartel price which is set by both players when maximizing their joint profits. Now, suppose that both players agree to set the same price $p_i = p_j = p$. Then the profit of a player i is decreasing in his own price p_i for p above p^N, but increasing in p_i for p below p^N. Hence an agreement below p^N is not reasonable. On the other hand, the profits of both players is decreasing in p when $p > p^C$ and increasing in p when $p < p^C$. So, also an agreement above p^C is not reasonable. Therefore, we restrict attention to prices in the interval $\left[p^N, p^C\right]$. For suitable values of the parameters we obtain a generalized PDQL payoff matrix when prices to be chosen by the players are restricted to a discrete set of k prices within this interval, for instance when action a of player i corresponds to setting its price equal to $p = \frac{k-a}{k-1} p^N + \frac{a-1}{k-1} p^C$, so that action $a = 1$ yields the Bertrand-Nash price p^N and action $a = k$ the cartel price p^C. Observe that the grid becomes finer when k increases. Indeed the payoff structure is linear increasing in the action of the opponent and quadratic-concave in the own action of the player. This concludes the example.

We now consider the dynamics of the model. At each round of play, the action of each player is given and determined by history. The state of the system at time t is described by the N-tuple $s^t = \left(s_1^t, s_2^t, \ldots, s_N^t\right)$ where $s_i^t \in \{1, 2, \ldots, k\}$ denotes the action of player i at time t. Initially (at $t = 0$) the population starts in a given state $s^0 \in S$, where $S = \{1, 2, \ldots, k\}^N$ is the state space. At each time t only the subject selected by the random mechanism gets the possibility to update his action, a so called *learning draw*. Note that at each date only one player gets the learning draw. The underlying assumption here is that at each moment in (continuous, real world) time, with probability 0 this moment will be the moment of a potential change in behavior of more than one agent. The selected player first plays the stage game with his neighbors. Then, to update his action, the subject compares the average payoff he got from playing the game once with each of his m neighbors, denoted by π_{self}, with the average payoff his m neighbors got from playing the game with the subject, denoted by π_{nbs}.[4] For the ease of discussion, in the following we say that a player is in a 'win' ('lose') situation, whenever his own payoff is higher (lower) than the average payoff of his neighbors. Recall however from footnote 1 that this does not mean that players who 'win' are (in the long run) better off than players who 'lose'. Observe that the subject plays the same action against each of his neighbors. The rationale behind this comparison is the assumption that players only interact with a limited group of other individuals from the population and thus only observe the results of these interactions. Based on the comparison of the payoffs, the subject updates his action using the update rule 'Win Cooperate, Lose Defect'. Whenever $\pi_{self} > \pi_{nbs}$, the subject is in a 'win' situation and sets his action s_i^t to $s_i^t + 1$ if $s_i^t < k$ and sticks to his current action s_i^t if $s_i^t = k$. When $\pi_{self} < \pi_{nbs}$, the subject is in a 'lose' situation and updates his action s_i^t to $s_i^t - 1$ if $s_i^t > 1$ and to s_i^t if $s_i^t = 1$. When both payoffs are exactly equal, the subject will stick to the action he is playing at present in the next round. This update rule can be formalized as follows. Let μ^t be the k-dimensional vector of integers describing the neighborhood of the subject at time t by defining $\mu_b^t \in \{0, \ldots, m\}$ as the number of neighbors of the subject playing action b, $b = 1, \ldots, k$, in state s^t. Let i be the subject selected at t. Then it holds that

$$\pi_{self} = \frac{1}{m} \sum_{b=1}^{k} \mu_b^t \alpha_{s_i^t, b} \text{ and } \pi_{nbs} = \frac{1}{m} \sum_{b=1}^{k} \mu_b^t \alpha_{b, s_i^t}.$$

Hence it follows that

$$\pi_{self} \gtrless \pi_{nbs} \Leftrightarrow \left(A - A'\right)_{s_i^t} \cdot \mu^t \gtrless 0,$$

where $(A - A')_a$ is the a-th row of the matrix $A - A'$. Therefore, according to WCLD the state s^{t+1} is given by

$$s_i^{t+1} = \begin{cases} s_i^t + 1, & \text{when } (A - A')_{s_i^t} \cdot \mu^t > 0 \text{ and } s_i^t < k, \\ s_i^t - 1, & \text{when } (A - A')_{s_i^t} \cdot \mu^t < 0 \text{ and } s_i^t > 1, \\ s_i^t, & \text{all other cases,} \end{cases}$$

[4] We implicitly assume that different payoffs can be added up. This is justified in e.g. the case that payoffs are in monetary terms and players are risk-neutral.

for the subject i and by $s_j^{t+1} = s_j^t$ for all players $j \neq i$. The expression for s_i^{t+1} makes explicit that the update depends crucially on the group of players the subject interacts with.

The update rule WCLD is closely related to the well known Tit-for-Tat (TfT) update rule. In fact, for $k = 2$ WCLD is exactly TfT in a two player PD game. In case of a multi-player game several versions of TfT are known. Here we refer to the TfT-rule which says that in a 2-action PD game a player will play 'Cooperative' in the next round if and only if an a priori required fraction of his opponents plays 'Cooperative' in the current round of play (see e.g. **?**)). This multi-player version of TfT becomes more forgiving if the fraction required to switch to the cooperative action is lowered. When this fraction is taken to be one, this TfT-rule is quite severe in the sense that all opponents are required to play cooperatively to make a player switching to cooperative behavior in the next round of play. For the case $k = 2$, in our model of local interaction it holds that a subject playing cooperatively learns that some of his neighbors defected on him when his average payoff is lower than that of his neighbors. According to WCLD, he then will play non-cooperatively in the next round of play. A subject that plays the non-cooperative action himself and gets a higher average payoff than his neighbors, reasons that at least one neighbor is playing cooperatively and will then switch to cooperative behavior. So, for a 2-action multi-player PD game the update rule WCLD is a version of TfT which depends on the subject's own action. The rule administers severe punishment when the subject himself plays cooperatively, because then one non-cooperatively playing neighbor is sufficient to switch to non-cooperative behavior. On the other hand, the rule is forgiving in the sense that when playing non-cooperatively himself, the subject will switch to cooperative behavior as soon as one neighbor plays cooperatively. Thus WCLD is a highly reciprocal update rule.

For $k > 2$ the reasoning above still holds qualitatively in the sense that when the subject observes that his own average payoff is higher (lower) than the average payoff of his neighbors, he reasons that at least some of his neighbors are playing a more (less) cooperative action than his own action and therefore he also updates his own action from s_i^t to $s_i^t + 1$ ($s_i^t - 1$), unless already $s_i^t = k$ ($s_i^t = 1$). This can be made more precise for the subclass of PDBL matrices, because in this case the payoff to the subject of the stage game is increasing linearly in the action of his opponent. So, for A being a PDBL matrix we have that $(A - A')_{s_i^t} \cdot \mu^t > 0$

(respectively < 0) and hence that the subject i is in a 'win' (lose) situation, whenever his own action is lower (higher) than the 'average action' $\frac{1}{m}\sum_{b=1}^{k} b\mu_b^t$ of his neighbors. Thus, similar as with TfT, the extent to which one's opponents act cooperatively determines whether one will act cooperatively in the next round. However, in contrast with TfT, the update also depends on the own action of the subject. When handed the learning draw, a player becomes more forgiving in the next round if he plays more defectively in the current round.

Within the subclass of generalized PDQL matrices, the update rule is even more forgiving than it is in the subclass of PDBL matrices in the sense that the above reasoning still holds with respect to the 'win' situation, but is not necessarily true for the 'lose' situation. So, in case of generalized PDQL matrices the subject is certainly in a 'win' situation when his own action is equal to or lower than the average action of his neighbors, and he may even be in a 'win' situation when his own action is higher than the average action of his neighbors. This is stated in the following corollary.

Corollary 2.3. *Let A be a generalized PDQL matrix. If player i has the learning draw at time t and $s_i^t \leq \frac{1}{m}\sum_{b=1}^{k} b\mu_b^t$, then the subject i is in a 'win' situation.*

Proof

First, we prove the statement for $k = 3$ and $s_i^t = 2$. In this case it follows from $s_i^t = 2 \leq \frac{1}{m}\sum_{b=1}^{k} b\mu_b^t = \frac{1}{m}\left(\mu_1^t + 2\mu_2^t + 3\mu_3^t\right)$ that $\mu_3^t \geq \mu_1^t$. Furthermore, we have that

$$m(\pi_{self} - \pi_{nbs}) = (A - A')_2 \cdot \mu^t = (\alpha_{2,1} - \alpha_{1,2})\mu_1^t + (\alpha_{2,3} - \alpha_{3,2})\mu_3^t.$$

Because of the conditions of a generalized PDQL matrix, both $\alpha_{2,1}-\alpha_{1,2} > 0$ and $\alpha_{2,3}-\alpha_{3,2} > 0$. So, the subject is in a 'win' situation if

$$\mu_3^t > \frac{\alpha_{1,2} - \alpha_{2,1}}{\alpha_{2,3} - \alpha_{3,2}}\mu_1^t.$$

Since $\mu_3^t \geq \mu_1^t$ it is sufficient to show that $\frac{\alpha_{1,2}-\alpha_{2,1}}{\alpha_{2,3}-\alpha_{3,2}} < 1$. Therefore, rewrite $\frac{\alpha_{1,2}-\alpha_{2,1}}{\alpha_{2,3}-\alpha_{3,2}} = \frac{(\alpha_{2,2}-\alpha_{2,1})+(\alpha_{1,2}-\alpha_{2,2})}{(\alpha_{2,3}-\alpha_{2,2})+(\alpha_{2,2}-\alpha_{3,2})}$. Now, note that $\alpha_{2,2} - \alpha_{2,1} = \alpha_{2,3} - \alpha_{2,2}$, since the payoff to the subject in the stage game is linear in the action of the opponent. Furthermore, since the payoff is quadratic-concave in one's own action, it holds that $\alpha_{1,2} - \alpha_{2,2} < \alpha_{2,2} - \alpha_{3,2}$, leading to the conclusion that indeed $\frac{\alpha_{1,2}-\alpha_{2,1}}{\alpha_{2,3}-\alpha_{3,2}} < 1$. For general values of k and s_t^i, a similar argument on the linear and quadratic nature of the payoff function yields the above result. □

In the above model we focus on a selected player playing the stage game with everyone of his neighbors. The model can easily be extended to allow for a subject playing the stage game with only a randomly chosen nonempty subset of his neighbors. Hence it also allows for a setup where a player has a group of people he can interact with, but where the actual interaction takes place only with a limited number of the members from this group. On top of this model, in Section 5 we will introduce noise in the form of mutations. In the next two sections we discuss the dynamic features of the model without mutations. This can be seen as the medium-run behavior of the dynamic process.

3. Medium-run dynamic behavior

The main result of this section is stated in Theorem 3.2 and says that the only stable states of the system are the states in which all players play the same action. An important observation underlying this result is that the dynamics are *boundary preserving*, that is, an individual may change his action only when one of his neighbors uses an action different from his own (see e.g. Eshel, Sansone & Shaked (**?**)). To see this, all we need is the observation that if all neighbors of a subject play the same action as the subject, all these players get exactly the same payoff in all of the stage games that are played. Therefore the comparison of the subject's average payoff to the average payoff of his neighbors results in equality and the subject will not change his action. So, only when at least one of the neighbors of the subject uses a different action than the subject itself, it is possible that the average payoff of the subject differs from the average payoff of his neighbors. Such situations can only arise at the boundaries of clusters, where a cluster is defined as a group of adjacent players that play the same action and a member of a cluster is on the boundary of the cluster when at least one of his neighbors plays another action. At a boundary a subject playing action a either does not change his action and the clusters remain unchanged or he changes his action to $a+1$ or $a-1$, and consequently the subject joins another cluster or forms a new cluster of size 1 by himself. When changing his action, the size of the cluster the subject belonged to decreases by one and the size of the cluster the subject switches to increases by one. This argument, together with Lemma 3.1 below, implies that when the subject is in an a-cluster (i.e. a cluster consisting of players all playing action a) and has at least one of its neighbors in another cluster, either the subject stays in the a-cluster, or joins one of the other neighboring clusters, or forms a new a'-cluster of size 1 with either $a' = a + 1$ or $a' = a - 1$. In the latter case we must have that at least one of the neighbors belongs to an $\widetilde{a}$-cluster with $\widetilde{a} > a'$ when $a' = a+1$, respectively $\widetilde{a} < a'$ when $a' = a-1$. Thus, in a population where all players use the same action, no new clusters will emerge. So, to show convergence of the system to a stable state it is sufficient to show that there are sample paths with positive probability that lead to a decrease of the number of clusters to 1. To prove the main result, we first state the following lemma.

Lemma 3.1. *Let A be a $k \times k$ PD matrix. Then, for all a, $a < k$ (respectively $a > 1$), every a-player facing a set of opponents containing at least one $\widetilde{a}$-player, $a < \widetilde{a} \leq k$ ($1 \leq \widetilde{a} < a$) and not containing any $\widehat{a}$-player, $1 \leq \widehat{a} < a$ ($a < \widehat{a} \leq k$), will change his action into $a+1$ ($a-1$) when this player gets the learning draw.*

Proof

Suppose player i with $s_t^i = a$, $a < k$, faces a set of opponents containing at least one $\widetilde{a}$-player, $a < \widetilde{a} \leq k$, and not containing any $\widehat{a}$-player, $1 \leq \widehat{a} < a$, and gets the learning draw. Then it holds that $\mu_b^t = 0$ for $b < a$ and $\mu_b^t > 0$ for at least one of the components $b = a+1, \ldots, k$. Hence

$$\pi_{self} - \pi_{nbs} = \frac{1}{m}\left(A - A'\right)_a \mu^t = \frac{1}{m}\sum_{b=a}^{k} \mu_b^t \left(\alpha_{a,b} - \alpha_{b,a}\right) > 0,$$

since iterated application of the inequality $\alpha_{a+1,b} < \alpha_{a,b} < \alpha_{a,b+1}$ gives us $\alpha_{k,a} < \ldots < \alpha_{a+2,a} < \alpha_{a+1,a} < \alpha_{a,a} < \alpha_{a,a+1} < \alpha_{a,a+2} < \ldots < \alpha_{a,k}$. Hence, according to WCLD player i updates his action to $s_i^{t+1} = a+1$. An analogous reasoning shows that the subject updates his action to $a-1$ when he faces a set of opponents containing at least one $\widetilde{a}$-player, $1 \leq \widetilde{a} < a$, and not containing any $\widehat{a}$-player, $a < \widehat{a} \leq k$. $\square$

For $a = 1, \ldots, k$, let $s(a) \in S$ denote the state in which all players use action a, i.e. $s_i(a) = a$ for all $i = 1, \ldots, N$. We now prove the following theorem, which says that a state is stable if and only if all players use the same action.

Theorem 3.2. *For $k \geq 2$, let the $k \times k$ matrix A be a PD matrix. Then the set of stable states is given by $\{s(a) \in S | \ a = 1, \ldots, k\}$. Furthermore, from any initial state s^0, all stable states $s(a)$ with $\min_i s_i^0 \leq a \leq \max_i s_i^0$ can be reached with positive probability.*

Proof

First, from the boundary preservingness, it is trivial that any state $s(a)$, $a = 1, 2, \ldots, k$, is stable. Second, in any other state at least one player will be in the position indicated in Lemma 3.1 and consequently he will update when handed the learning draw. Therefore, no states but those in the set $\{s(a) \in S | \ a = 1, \ldots, k\}$ are stable.

Now consider the following sequence of learning draws to reach the stable state $s(a)$ for some $a = 1, \ldots, k$. First give the learning draw to a player on the boundary of a 1-cluster. According to Lemma 3.1 this player will update his action to 2. Continue to do so until no 1-players are left. Then, give the learning draw sequentially to all 2-players. In absence of any 1-players, 2-players will update their action to action 3. When all 2-players have done so, give the learning draw to all 3-players, and so forth, until there are no players left using an action $b < a$. Then give the learning draw sequentially to all k-players on the boundary of a k-cluster. They will update to action $k - 1$ according to Lemma 3.1. When there are no k-players left, start giving the learning draw to $(k-1)$-players on the boundary, and so forth, until there are no players left who use an action $b > a$. Then the stable state is reached in which all players use action a. This particular sequence of learning draws occurs with positive probability. Thus we have shown that each stable state in which all players use action a can be reached from any initial state containing at least one $\widetilde{a} \leq a$ player and at least one $\widehat{a} \geq a$ player. $\square$

From Theorem 3.2 it follows directly that when all actions are present in the initial state s^0, all stable states can be reached with positive probability. Note that there are many more paths than the one described in the proof that lead to stable states. Therefore it is not necessarily the case that the occurrence of a stable state becomes increasingly rare when the number of actions k increases. In section 4 we show that convergence times do increase with k, but not dramatically. Furthermore, from Lemma 3.1 it immediately follows that no stable state can be *asymptotically* stable, because one or more mutations can take the population out of a stable state and lead it to another stable state with positive probability. We postpone WCLD dynamics with rare mutations until Section 5.

The next two corollaries follow immediately from the proof of Theorem 3.2. For given state $s \in S$, let s' be defined by $s'_i = k + 1 - s_i$ for all $i = 1, \ldots, N$, i.e. the states s and s' are each other's mirror image in the sense that player i plays action $k + 1 - a$ in s' if he plays action a in s. Note that for any pair of stable states $s(a)$ and $s(k + 1 - a)$, $a = 1, \ldots, k$, it holds that they are each other's mirror image. Now, a probability distribution over all states is said to be symmetric if for every $s \in S$ it assigns the same probability to s and to its mirror image s'.

In the proof of the following Corollary 3.3, let p^{∞}_{s,s^*} be the probability that the system converges in the limit to state s^*, given that the initial state is s. From Theorem 3.2 it is immediately clear that p^{∞}_{s,s^*} can only be positive iff s^* is an element of the set of stable states, i.e. $s^* = s(a)$ for some $a = 1, \ldots, k$.

Corollary 3.3. *At time $t = 0$, let the initial state be drawn from a symmetric probability distribution over all states. Then, for $k = 2$ the a priori (i.e. before initialization of the state at $t = 0$) probability of convergence to each one of the two stable states is exactly $\frac{1}{2}$ for any 2×2 PD matrix A.*

Proof

Note that $s(2) = s'(1)$. The proof of Theorem 3.2 shows that there is positive probability to end up in either one of these two stable states from all other states. Lemma 3.1 shows that a 1-player, facing a set of opponents containing at least one 2-player, will change his action and become a 2-player and that a 2-player, facing a set of opponents containing at least one 1-player, will become a 1 -player. Hence $p^{\infty}_{s,s(1)} = p^{\infty}_{s',s(2)}$ for any initial state s. Since the initial state is drawn from a symmetric probability distribution, both stable states are reached with equal probability. $\square$

A symmetry argument analogous to the one used in the proof above yields the following corollary. Note that Corollary 3.3 holds for all PD matrices, while the next corollary only holds for PDBL matrices.

Corollary 3.4. *At time $t = 0$, let the initial state be drawn from a symmetric probability distribution over all states and let A be a PDBL matrix. Then, for any $k \geq 2$, the a priori probability distribution of convergence of the process to each one of the k stable states is symmetric around $\frac{k+1}{2}$.*

Proof

Since A is a PDBL matrix, we know from Section 2 that a subject is in a 'win' ('lose') situation if his action is lower (higher) than the average action of his neighbors. Hence it follows that $p^{\infty}_{s,s(a)} = p^{\infty}_{s',s(k+1-a)}$, $a = 1, 2, \ldots, k$ for any initial state s. Since the initial state is drawn from a symmetric probability distribution, the a priori probability of reaching state $s(a)$ is equal to that of reaching state $s(k+1-a)$ for all $a = 1, \ldots, k$. $\square$

4. Medium-Run Simulations

In Corollary 3.3 we have seen that for $k = 2$ the frequency distribution of reaching either one of the two stable states will be exactly $\frac{1}{2}$, when the initial state is selected according to a symmetric probability distribution. Furthermore, for $k \geq 2$, we see in Corollary 3.4 that the frequency distribution will be symmetric around $\frac{k+1}{2}$ when the payoffs are bilinear. To gain more insight in the actual shape of the frequency distribution for $k > 2$, simulations have to be performed in order to study the behavior of the process. To do so, we consider populations in which the spatial structure is taken to be either a circle or a torus, i.e. in the graph reflecting this spatial structure each player has two or eight neighbors respectively. Because of computing power restrictions in the case of a torus, we restrict simulations to $N = 16$, i.e. to 16 players located on the circle or to a 4×4 torus. We perform simulation runs for different values of k. Furthermore, all of simulation runs are performed for two different payoff matrices A. The first matrix A we study is a PDBL matrix. The second matrix A is a generalized PDQL matrix generated by Bertrand price competition on a discrete set of prices with the Bertrand-Nash price p^N as the lowest price and the cartel price p^C as the highest price, see Example 2.2.

To restrict the number of simulations, we only perform simulations to estimate the probability that a population ends up in a certain stable state, given that the initial state is drawn from the (symmetric) uniform distribution over all states, i.e. each state has equal probability of being selected as the initial state. To get the a priori probability (before initialization at time $t = 0$) of reaching each stable state, we record how often each of the stable states is reached for a large number of initializations. This gives us an estimate of the frequency distribution of convergence to the stable states.

Each simulation consists of the following steps. To initialize the simulation an initial state s^0 is drawn from the uniform distribution by assigning to each player i a random action s_i^0 with $\Pr\left(s_i^0 = a\right) = \frac{1}{k}$, for all $a = 1, 2, \ldots, k$. Then, at each time $t \geq 0$, a player is chosen at random (uniformly) from the population and gets the learning draw. Since the dynamics are boundary preserving, choosing a player that is not on a boundary of a cluster does not result in a change of the state.[5] The selected player plays the $k \times k$ stage game (A, A') against all his neighbors and updates his action through the update rule WCLD. We proceed until the system has converged, i.e. until all players play the same action, and record the obtained stable state. We run $1,000$ of these simulations and call these $1,000$ simulations a *simulation run*. After a simulation run, we report the frequency distribution over stable states as the outcome. The law of large numbers implies that this frequency distribution converges to the true probability distribution over stable states when the number of simulations in each simulation run is taken to infinity.

The medium-run simulation results are reported in tables A.1 to A.3 in appendix A. We draw attention to several interesting features of the results. The first of these features is that in all of the medium-run simulations there is a high probability that the system ends up in a state with a moderate level of cooperation. When the payoff matrix is PDBL, the stable state in which all players use the median action $\frac{k+1}{2}$ (the median stable state) is reached most often,

[5] Therefore convergence time would decrease dramatically when we would only select agents on the boundary.

and a lot of mass in the frequency distribution is piled up at the stable states corresponding to the actions $\frac{k-1}{2}$, respectively $\frac{k+3}{2}$, at either side of this median stable state. The intuition for this result is twofold. First, in a state where many different actions are being played, it is more likely for an 'extreme' action to disappear than it is for an action closer to the median action. Under WCLD dynamics, a single learning draw handed to a player using action 1 (k) on a boundary is enough to make this player update to action 2 ($k-1$), an action closer to the median action. However, a player on a boundary using an action close to the median action is approximately as likely to update to a higher labelled action as he is to update to a lower labelled action. Thus, the probability that a player on a boundary that uses an action close to the median action will remain using an action close to the median action over time, is much higher than the probability of a player on a boundary using an extreme action staying close to this action over time. Second, once there are no players left using action 1 (k), this action can not reappear, in contrast to other actions $a \in \{2, 3, \ldots, k-1\}$, which can reappear as long as there is at least one player left who uses an action $b < a$ and one player who uses an action $c > a$. These two arguments imply that there are many more paths to stable states in which all players use an action close to the median action (i.e. a stable state 'close to' the median stable state) than there are paths to stable states in which all players use an action close to one of the extreme actions 1 or k.

When the number of actions in the stage game k increases, there are three clear effects in the simulation outcomes. First, we observe symmetry around the median action in the frequency distribution for PDBL matrices. Here the explanation is provided by Corollary 3.4. Second, the distribution is spread out over more actions, and as a consequence the frequency of convergence to each individual action decreases. Third, the frequency distribution is not spread out to an extend that, for large values of k, the model converges to each stable state at least once in the simulations. The frequency of the 'extreme' actions, i.e. the actions close to actions 1 and k, decreases rapidly when k increases. From Theorem 3.2 we know that all stable states have positive probability of being reached. Apparently, the probability of reaching a stable state in which all players use an action close to one of the extreme actions 1 or k is positive but very small. To quantify the two last mentioned effects, we calculate the variance of the frequency distribution that results from the simulations [6]. Note that the second effect increases the variance, while the third effect decreases the variance of the distribution. We see that the variance decreases when k increases. This clearly means that the tendency that 'extreme' actions have lower frequencies dominates the effect of spreading out of the distribution when k increases. From this we conclude that as k increases, the total frequency of the actions around the median action increases, making moderate levels of cooperative behavior more probable.

Another interesting feature we focus on in the simulation results is the skewness of the frequency distribution towards the higher labelled action when the payoff matrix is generalized PDQL. As we have shown in Corollary 2.3, a subject is in a 'win' situation more easily when the payoffs are quadratic-linear than when the payoffs are bilinear. A subject in a 'win' situation updates to a higher labelled action. Therefore, it is evident that the generalized PDQL matrix imposes a bias toward higher labelled actions compared to the outcome for a PDBL matrix. However, the magnitude of this effect is not a priori clear. The simulation results reveal a moderate skewness for the parameter values we simulated with. Since these parameter values yield realistic values of the price and cross-price elasticities in the underlying Bertrand price competition game, our simulation results can be interpreted as support for findings that in real world settings people tend to moderate levels of cooperation. More specifically, our findings support the claim that although people will not take the fully cooperative action, they will use an action that is between the median action and the fully cooperative action most of the time.

The spatial structure on which the players are located does not have much of an effect on these results. We see that in most simulation outcomes the frequency of convergence to the median stable state is slightly less when the players are located on a torus than when they are located on a circle. However, this difference is not pronounced enough to regard it as anything but an artefact of the simulations.

The last point we focus on in the simulation outcomes are the convergence times. In general we see that outcomes with a lower frequency tend to display higher convergence time. A stable state that takes many periods for the system to reach is not reached often, since in the (many) periods during which the system has not yet converged, there is always a probability that the

[6] The values of the variances can be found in the tables in appendix A.

system moves away from the stable state that is hard to reach, and move towards another stable state. We also see that the convergence times slightly increase with k. This means that although for larger k there are more stable states the system can converge to and there is a large probability that there are more different actions present in the initial state s^0, this does not have a substantial influence on the speed of convergence. Furthermore, there is not much difference in convergence times between the simulations performed with PDBL and PDQL payoffs. However, we note a large increase in convergence time when the spatial structure is a torus instead of a circle. This effect can be illustrated by looking at (for the case $k = 2$) the number of neighbors the last player playing action a, $a = 1, 2$, has in a population where all other players use action b, $b = 1, 2$, $b \neq a$. This a-player has 2 neighbors playing an action b when located on a circle and 8 neighbors playing action b on a torus. Whenever the a-player gets the learning draw, he will switch his action and from then on the population will be in stable state $s(b)$. However, when one of his neighbors gets the learning draw, this player will switch to a and the state moves further away from the stable state $s(b)$. Clearly on the circle the probability that a neighbor of the a-player gets the learning draw is 2 times as large as the probability that the a-player gets the learning draw, but 8 times as large when the population is located at the torus. Thus, when only one a-player is left, the probability of moving away from the stable state $s(b)$ is larger than that of reaching the stable state, but these probabilities become more unfavorable for reaching the stable state when the number of neighbors of each player increases. This effect accounts for the rapidly increasing convergence time when the number of neighbors m increases. In fact, the increase in convergence time is exponential when we increase the dimension of the neighborhood, as is done in the transition from a circle to a torus. From the simulation outcomes, we see that the effect is stronger than one would predict solely on the basis of this illustration. This is due to the fact that the number of neighbors also influences the probability of getting one step closer to a stable state, when the state is not very close to a stable state yet.

5. Long-Run dynamic behavior.

In this section we study the long-run effects of adding rare mutations to the system. We introduce mutations in the original Markov process and we characterize the unique recurrent set of this process with mutations, which turns out to be the set of all stable states of the WCLD dynamics without mutations. Since this set is not a singleton, the question arises what the distribution of the long-run fraction of time spent in each of the different states in the recurrent set looks like. Note that the long-run fraction of time spent in a state is equal to the probability to be in this state at an arbitrary time t, far enough away from the time origin 0. Our analysis is closely related to the standard literature on stochastic stability, e.g. **?**), Young (**?**), (**?**) and **?**).

Following **?**), we introduce an artificial Markov process representing mutations on the set of stable states and show that this process has a unique invariant distribution. This distribution only puts positive weight on the elements of the unique recurrent set, which we characterize. We apply a result from **?**) to prove that this invariant distribution is equal to the limit of the invariant distribution of the original Markov process with a very small probability of mutation, when this probability of mutation is taken to zero in the limit. Subsequently we show that the long-run simulation output converges to this unique invariant distribution.

We think of mutations as being rare, i.e. the probability of a mutation is small and vanishes in the limit. This means that the expected time that passes between the occurrence of two mutations becomes very large. Since the expected time for the standard WCLD dynamics to converge is not affected, this simply means that in the limit the expected time between two mutations exceeds the convergence time of the WCLD dynamics by a large factor. In other words, the expected time in between mutations is much larger than the expected time the WCLD dynamics need to reach one of the stable states. Thus we can simply neglect rare mutations that occur during the period of time when the model is not in a stable state. This means that we can suffice with perturbing the stable states of the dynamics. Nevertheless, some of our theoretical arguments are valid in a more general framework, which allows for several mutants at the same time also.

Since each player has k actions, the number of states in the system is equal to $K = k^N$. We enumerate the states $s \in S$ by $j = 1, 2, \ldots, K$ and we associate $j = 1, 2, \ldots, k$ with the k stable states $s(j)$ of the medium-run dynamics. Also for $j > k$ the j-th state is denoted by $s(j)$. We now introduce the transition matrices $P = (p_{jh})_{j,h=1}^{K}$, representing the Markov process specified by the original WCLD dynamics and $Q = (q_{jh})_{j,h=1}^{K}$ representing mutations from stable states only. Specifically, we focus attention on one-shot mutations from stable states. A *one-shot mutation* from a stable state $s(j)$, $j = 1, 2, \ldots, k$, yields a state in which one player plays a randomly chosen action $a \in \{1, 2, \ldots, k\}$, $a \neq j$, and all other players in the population still play action j. We call a state that can be reached from a stable state by a single one-shot mutation *adjacent* to this stable state. The reason for the focus on one-shot mutations is that (a sequence of) one-shot mutations is enough to get the system from any stable state to any other stable state with positive probability, as indicated in Lemma 5.1. As indicated above, we focus on a probability of mutation that is small compared to the speed of convergence of the dynamics P, i.e. mutations that only play a role once the dynamics P have settled down. We incorporate this feature through explicitly only perturbing stable states. Thus, for all $j > k$ we set $q_{jj} = 1$ and hence $q_{jh} = 0$ for all $h \neq j$. Moreover, we assume that, among other mutations, Q contains one-shot mutations from stable states, i.e. for any stable state j, $1 < j \leq k$, that $q_{jh} > 0$ for at least one adjacent state $s(h)$ with, for exactly one player i, $s_i(h) < j$ and for any stable state j, $1 \leq j < k$, that $q_{jh'} > 0$ for at least one adjacent state $s(h')$ with, for exactly one player i, $s_i(h') > j$. So, mutations from a stable state $s(j)$ may lead to actions either below or above j. Now, let $P(\varepsilon) = (1 - \varepsilon) P + \varepsilon Q$ be the Markov process obtained from the medium-run Markov process P by adding a probability $\varepsilon > 0$ of a mutation taking the system from state j to state h with positive probability iff $q_{jh} > 0$. Thus $P(\varepsilon)$ is an ergodic Markov process of the type commonly used in the literature, e.g. **?**) and Young (**?**, **?**). The assumption stated above on the entries of Q guarantees that with positive probability the system can go from any stable state to any other stable state, i.e. all of the stable states $s(j)$, $j = 1, \ldots, k$, are in the same recurrent set, as defined in **?**) (p. 220). The next lemma says that the set of stable states is the unique recurrent set of the process $P(\varepsilon)$.

Lemma 5.1. *The set $\{s(j) \in S | j = 1, \ldots, k\}$ of stable states is the unique recurrent set of the system $P(\varepsilon)$.*

Proof.

By the assumptions on Q, for any stable state $s(a)$, $a > 1$ $(a < k)$ a one-shot mutation to a state in which one mutant plays an action $b < a$ $(b > a)$ can occur. Then, following the proof of Theorem 3.2 with positive probability there is a path that converges to any stable state $s(c)$ with $b \leq c \leq a$ $(a \leq c \leq b)$. Thus every stable state is connected with every other stable state through a series of one-shot mutations. Therefore, all the k stable states $s(j)$, $j = 1, \ldots, k$, are in the same recurrent set. Since the process P has no other stable states, this is the unique recurrent set. □

Since $P(\varepsilon)$ is ergodic, it has a unique invariant distribution over the set of states. For given $\varepsilon > 0$, let $\lambda(\varepsilon, Q) \in \mathbb{R}^K$ denote the invariant distribution of the Markov process $P(\varepsilon)$ over the set of states S, with $\lambda_j(\varepsilon, Q) \in \mathbb{R}$ the average long-run time that the system is in state $j \in S$. From Lemma 5.1 it follows that $\lambda_j(\varepsilon, Q)$ goes to zero for all $j > k$ when ε goes to zero. So, in the limit the invariant distribution only assigns positive measure to the stable states in the recurrent set. To find this limiting invariant distribution we consider the artificial Markov process $Q \cdot (P)^\infty = QP^\infty$. The specification of the entries of matrix Q guarantees that QP^∞ implicitly defines an ergodic $k \times k$ Markov process on the stable states and therefore it implicitly defines a unique invariant distribution on the set of stable states of P. This distribution contains the first k entries of the unique eigenvector $\lambda^*(Q) \in \mathbb{R}^K$ of the matrix $(QP^\infty)'$ with eigenvalue one, i.e. $(QP^\infty)' \lambda^*(Q) = \lambda^*(Q)$. Note that the transpose of $\lambda^*(Q)$, i.e. $\lambda^*(Q)'$, is the invariant distribution of QP^∞. The following lemma states that in the invariant distribution $\lambda^*(Q)'$ only the stable states of the process P have positive measure.

Lemma 5.2. *The unique invariant distribution $\lambda^*(Q)'$ of the Markov process QP^∞ satisfies $\lambda_j^*(Q) = 0$ for all $j > k$ and $\lambda_j^*(Q) > 0$ for all $j \leq k$.*

Proof.

Since the process P converges to one of the stable states, we have that only the first k columns of the matrix P^∞ can contain positive entries. All other columns of P^∞ only contain zeroes and so do the corresponding columns of QP^∞. So, the components $j > k$ of the unique eigenvector with eigenvalue 1 of $(QP^{\infty})'$ are necessarily 0, since the associated rows of $(QP^{\infty})'$ contain zeroes only.

The components of the invariant distribution have to sum to 1. Thus, one of the components $\lambda_j^*(Q)$, $j \leq k$ has to be positive. Since all (stable) states j, $j \leq k$, are in the same recurrent set, they either all have zero measure or they all have positive measure. Thus, we conclude that $\lambda_j^*(Q) > 0$ for all $j \leq k$. $\square$

We now proceed with the claim that the invariant distribution $\lambda^*(Q)'$ is the limit of a sequence $\lambda(\varepsilon, Q)'$ as $\varepsilon \downarrow 0$, a result due to **?**).

Theorem 5.3. *Let* $\lambda(\varepsilon, Q)'$ *be the invariant distribution of the ergodic Markov matrix* $P(\varepsilon)$ *as specified above, then* $\lim_{\varepsilon\downarrow 0} \lambda(\varepsilon, Q)' = \lambda^*(Q)'$.

Proof.

Follows directly from **?**), Proposition 2, 3 and 4 and Theorem 2. $\square$

We now describe the setup of the long-run simulations. For each stable state $s(a)$, $a = 1, \ldots, k$, we take each state $s(j)$ with $q_{aj} > 0$ as the (fixed) initial state of a *simulation run*. As before, a simulation run consists of T simulations. At each simulation of the simulation run, the process P is run until the system has converged to a stable state. The simulation run over T simulations provide us with estimates $\widehat{p}_{jb}^T$ for the transition probabilities p_{jb}^{∞} in P^{∞} from state $s(j)$ to each of the stable state $s(b)$, $b = 1, \ldots, k$. Note that since the system always ends up in a stable state, $\widehat{p}_{ji}^T = 0$ for all $i > k$. Weighting the estimates $\widehat{p}_{jb}^T$ with q_{aj} and adding up over j results in estimates $\widehat{r}_{ab}^T = \sum_j \left(q_{aj} \cdot \widehat{p}_{jb}^T\right)$ of transition probabilities from stable state $s(a)$ to stable state $s(b)$ of the Markov process QP^{∞}. Calculating the unique eigenvector at eigenvalue 1 (with components summing to 1) of the $k \times k$ matrix of estimates $\widehat{R}' = \left((\widehat{r}_{ab}^T)_{a,b=1}^k\right)'$ results in estimates of the first k entries of the distribution $\lambda^*(Q)$, which are denoted by $\lambda_j^T(Q)$, $j = 1, 2, \ldots, k$, to indicate that the system $\widehat{R}'$ is generated by the simulation runs of length T. As noted before, the remaining entries of $\lambda^T(Q)$ are 0. A law of large numbers guarantees that $\lambda^T(Q)$ converges to $\lambda^*(Q)$ as T goes to infinity.

It should be noted that although the technique developed above holds for quite general specifications of the matrix of mutations Q, the distribution $\lambda^*(Q)$ depends on this specification. In the long-run simulations we focus on a matrix $Q = Q_{LR}$ that only contains one-shot mutations from the stable states.

The actual simulation runs are done in order to estimate the transition probabilities $\widehat{p}_{jb}^{\infty}$ mentioned above. Each simulation consists of the following steps. First we initialize the simulation with a state j adjacent to a stable state. Then we perform the simulation in the same way as described above in section 4. A player is chosen at random (uniformly) from the population and gets the learning draw. The selected agent plays the $k \times k$ bimatrix stage game (A, A') against all his neighbors, looks at the outcomes and updates his action through the update rule WCLD. Then time progresses one period and a second player is selected at random to get the learning draw, etc. until the system has converged, i.e. until all players play the same action. We record the stable state to which the system has converged and start the next simulation, which is initialized with the same state as the first simulation. We perform simulation runs of length $T = 1,000$. After we have run an entire simulation run, we use the frequency distribution over stable states $\widehat{p}_{jh}^{1000}$, $h = 1, 2, \ldots, k$, as estimates for the transition probabilities p_{jh}^{∞}. For each stable state, we perform a simulation run for all states j that are adjacent to this stable state. For each stable state i, we include the outcome of a simulation run in which each simulation is initialized with the stable state itself. This run has already converged at the start-up phase and therefore we know that the transition probability p_{ii}^{∞} will be 1 for sure, and we needn't actually perform the simulation run. With the estimates of the transition probabilities obtained from simulating, we calculate the invariant distribution of the matrix R, which is an estimate of the distribution $\lambda^*(Q)$.

As before, the simulations are performed for different values of k and for both the spatial structure of a circle and a torus. Furthermore, all of the simulations are performed for a PDBL matrix and for a generalized PDQL matrix generated by Bertrand price competition as described in example 2.2.

At this point we note that the medium-run simulations can be placed in the same framework as the long-run simulations. This requires specifying the medium-run matrix Q_{MR} as $q_{jh} = \frac{1}{K}$, for $j = 1, 2, \ldots, k$, $h = 1, 2, \ldots K$, and $q_{jj} = 1$, $j > k$, i.e. each of the first k rows of Q_{MR} contains the uniform distribution over all initial states and the other rows of Q_{MR} only contain a 1 on the diagonal and zeroes elsewhere. Since $Q_{MR} \neq Q_{LR}$, also $\lambda^* (Q_{MR}) \neq \lambda^* (Q_{LR})$ and $\lambda^T (Q_{MR}) \neq \lambda^T (Q_{LR})$. Nevertheless, Lemma 5.2 and Theorem 5.3 both apply to $\lambda^* (Q_{MR})$ and, moreover, we are able to characterize $\lambda^* (Q_{MR})$ explicitly.

Corollary 5.4. *The unique invariant distribution $\lambda^* (Q_{MR})'$ of the Markov process $Q_{MR}P^\infty$ is given by $\lambda^* (Q_{MR})' = \left(\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}\right) \cdot P^\infty$.*

Proof.

The claim follows from straightforward matrix algebra.

$$\begin{aligned}
\left(\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}\right) \cdot P^\infty \cdot Q_{MR}P^\infty &= \left(\sum_{i=1}^{K} \frac{1}{K} (P^\infty)_{ij}\right)_{j=1}^{K} \cdot Q_{MR}P^\infty \\
&= \left(\sum_{j=1}^{K} \left\{\sum_{i=1}^{K} \frac{1}{K} (P^\infty)_{ij} \cdot \sum_{i=1}^{K} \frac{1}{K} (P^\infty)_{ih}\right\}\right)_{h=1}^{K} \\
&= \left(\sum_{i=1}^{K} \frac{1}{K} (P^\infty)_{ih} \cdot \sum_{j=1}^{K} \left\{\sum_{i=1}^{K} \frac{1}{K} (P^\infty)_{ij}\right\}\right)_{h=1}^{K}.
\end{aligned}$$

Since the vector $\left(\sum_{i=1}^{K} \frac{1}{K} (P^\infty)_{ij}\right)_{j=1}^{K}$ is a probability distribution, $\sum_{j=1}^{K} \left\{\sum_{i=1}^{K} \frac{1}{K} (P^\infty)_{ij}\right\} = 1$, which yields

$$\left(\sum_{i=1}^{K} \frac{1}{K} (P^\infty)_{ih}\right)_{h=1}^{K} = \left(\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}\right) \cdot P^\infty.$$

This concludes the proof. □

Note that this implies that $\lambda_j^* (Q_{MR}) = 0$, for $j > k$, since all columns j, $j > k$, of the matrix P^∞ only contain zeroes.

6. Results of Long-Run Simulations

The long-run simulation results are reported in tables A.4 to A.6 in appendix A. In this section we discuss these results. Note that the tables containing the results of the long-run simulations do not contain convergence times. The reason for this lies in the procedure for the long-run simulations as described above. The individual simulation runs performed to estimate $\widehat{p}_{jh}^{1000}$, $h = 1, 2, \ldots, k$, converge as fast as their medium-run counterparts. However, calculating the invariant distribution from these estimates $\widehat{p}_{jh}^{1000}$ makes it impossible to give a meaningful statistic for convergence time associated with the entries of the invariant distribution.

We see from the tables that the frequency distributions resulting from the long-run simulations look much the same as those associated with the medium-run outcomes. Indeed, in section 5 we already showed that all of the stable states of the model without mutations are stochastically stable states in the model with mutations. Now we see that the frequency distribution over the states in the set of stochastically stable states is also very similar to the medium-run outcome. However, note that the interpretation of the long-run distribution is different from that of the medium-run distribution. The entries of the long-run distribution represent both the probability of being in each one of the stable state at a given time and the long-run fraction of the time that the system spends in each one of the stable states. The medium-run distribution only gives probabilities of convergence to each stable state. After this initial convergence, in the absence of mutations, the system remains in the stable state it converged to forever.

Most features of the long-run results and the intuition behind these features are similar to the ones discussed in section 4. We only briefly mention these outcomes here. First, in the long-run, the model will sustain moderate levels of cooperation. Second, the influence of the number of action k is limited. When k is larger, the frequency distribution is more spread out over the larger number of stable states and the variance of the distribution decreases, leading to a higher probability of being in a moderately cooperative stable state at a given time. Third, the influence of the spatial structure on which the population is located seems negligible, as long as the graph is connected. Fourth, we have the symmetry around the median action for PDBL payoffs and fifth, we see that the frequency distribution over stable states is skewed towards the higher labelled actions when we focus on PDQL payoffs.

Note that these long-run outcomes have been obtained by focussing on the crucial influence of rare mutations, as is done in most of the standard literature on stochastic evolutionary models. Therefore, our results are quite general. In our opinion these general results suggests that the standard literature should focus less on the selection of one (equilibrium) state when players are rational, and devote more attention to the effect of introducing boundedly rational update rules in the selection models. Our results show that this is a fruitful exercise with sometimes surprising results.

A. Simulation Results.

Here we report the results of some of our simulation runs. We present separate tables for different values of k, $k = 3, 7, 11$. For each k we ran four different simulations for two spatial structure, being a circle of size $N = 16$ and a 4×4 torus, and two payoff matrices, being PDBL and generalized PDQL. The parameters for each PDBL matrix are $\alpha_{a,b} = 500 - 10 \cdot (a-1) + 20 \cdot (b-1)$. Note that the values of the parameters $\alpha_{1,1} = 500$, $\gamma = 10$ and $\delta = 2$ are irrelevant because the dynamics are solely driven by the subject's action and his neighbors average action. The generalized PDQL matrix corresponds to Bertrand price competition with $\gamma_0 = 20$, $\gamma_1 = 1$ and $\gamma_2 = \frac{1}{2}$. The first column contains the labels of the actions. The heading of each of the other columns captures the relevant information, e.g. 'PDBL circle' means a PDBL payoff matrix and a circle as spatial structure. Each column shows the frequency distribution over the actions. For the medium-run simulation, the convergence times are included in brackets. Convergence time is the time (number of rounds in the model) it takes the system to reach a stable state when we hand out the learning draw uniformly over all players. Thus, we do not restrict the dynamics to select only players that are on a boundary of a cluster, which would decrease convergence time considerably.

At the bottom of each column, the variance of the frequency distribution in this column is shown. To make the variances in the different tables comparable, we rescale the labels of the actions in the variance calculations. An action with label j, $j = 1, 2, \ldots, k$, gets a value of $\frac{j-1}{k-1}$, so that for all different values of k, the actions are on $[0, 1]$. The reported variance is thus

$$\sum_{j=1}^{k} \Pr(X = j) \left\{ \frac{j-1}{k-1} - \sum_{i=1}^{k} \left(\Pr(X = i) \cdot \frac{i-1}{k-1} \right) \right\}^2 ,$$

where $\Pr(X = j)$ is taken to be the reported frequency of reaching stable state j.

Tables A.1 through A.3 present the results of the medium-run simulations, while tables A.4 through A.6 report the results for the long-run simulations.

action	PDBL Circle	PDQL Circle	PDBL Torus	PDQL Torus
1	.156 (378)	.109 (400)	.242 $(13, 168)$	.091 $(12, 536)$
2	.677 (280)	.625 (292)	.503 $(14, 783)$	.508 $(12, 367)$
3	.167 (403)	.266 (379)	.255 $(13, 354)$	.401 $(14, 375)$
var	.081	.088	.124	.099

Table A.1: Medium-run simulation results for $k = 3$.

action	PDBL Circle	PDQL Circle	PDBL Torus	PDQL Torus
1	.000 $(-)$	.000 $(-)$	.000 $(-)$	.000 $(-)$
2	.026 (464)	.008 (411)	.025 $(19, 551)$	.010 $(17, 467)$
3	.247 (481)	.120 (419)	.247 $(13, 826)$	.163 $(14, 682)$
4	.467 (400)	.394 (416)	.452 $(14, 197)$	.445 $(13, 844)$
5	.232 (407)	.366 (450)	.251 $(15, 528)$	.329 $(13, 715)$
6	.028 (457)	.104 (518)	.025 $(18, 329)$	.053 $(16, 130)$
7	.000 $(-)$	.008 (754)	.000 $(-)$	.000 $(-)$
var	.019	.022	.019	.019

Table A.2: Medium-run simulation results for $k = 7$.

action	PDBL Circle	PDQL Circle	PDBL Torus	PDQL Torus
1	.000 (−)	.000 (−)	.000 (−)	.000 (−)
2	.000 (−)	.000 (−)	.000 (−)	.000 (−)
3	.006 (875)	.001 (756)	.005 (9, 022)	.000 (−)
4	.055 (559)	.016 (463)	.055 (9, 581)	.035 (10, 959)
5	.241 (491)	.107 (514)	.268 (15, 470)	.159 (13, 865)
6	.362 (472)	.274 (488)	.371 (14, 453)	.367 (13, 535)
7	.244 (478)	.366 (486)	.246 (13, 620)	.319 (14, 689)
8	.087 (468)	.181 (533)	.051 (15, 472)	.111 (13, 402)
9	.005 (512)	.045 (596)	.004 (16, 827)	.009 (9, 697)
10	.000 (−)	.009 (587)	.000 (−)	.000 (−)
11	.000 (−)	.001 (1263)	.000 (−)	.000 (−)
var	.011	.013	.010	.010

Table A.3: Medium-run simulation results for $k = 11$.

action	PDBL Circle	PDQL Circle	PDBL Torus	PDQL Torus
1	.194	.174	.237	.0999
2	.596	.595	.512	.505
3	.211	.231	.250	.395
var	.101	.100	.122	.102

Table A.4: Long-run simulation results for $k = 3$.

action	PDBL Circle	PDQL Circle	PDBL Torus	PDQL Torus
1	2.95×10^{-3}	9.21×10^{-4}	4.16×10^{-5}	0
2	5.10×10^{-2}	2.27×10^{-2}	1.89×10^{-2}	6.38×10^{-3}
3	2.39×10^{-1}	1.37×10^{-1}	2.49×10^{-1}	1.62×10^{-1}
4	4.06×10^{-1}	3.39×10^{-1}	4.64×10^{-1}	4.38×10^{-1}
5	2.43×10^{-1}	3.49×10^{-1}	2.48×10^{-1}	3.46×10^{-1}
6	5.58×10^{-2}	1.35×10^{-1}	1.92×10^{-2}	4.72×10^{-2}
7	2.69×10^{-3}	1.64×10^{-2}	8.63×10^{-5}	4.72×10^{-4}
var	2.67×10^{-2}	2.89×10^{-2}	1.81×10^{-2}	1.82×10^{-2}

Table A.5: Long-run simulation results for $k = 7$.

action	PDBL Circle	PDQL Circle	PDBL Torus	PDQL Torus
1	4.43×10^{-5}	3.64×10^{-6}	0	0
2	1.76×10^{-3}	2.08×10^{-4}	9.46×10^{-5}	1.04×10^{-5}
3	1.88×10^{-2}	3.26×10^{-3}	3.87×10^{-3}	1.97×10^{-3}
4	9.13×10^{-2}	2.63×10^{-2}	5.80×10^{-2}	2.96×10^{-2}
5	2.32×10^{-1}	1.02×10^{-1}	2.47×10^{-1}	1.77×10^{-1}
6	3.02×10^{-1}	2.34×10^{-1}	3.83×10^{-1}	3.66×10^{-1}
7	2.34×10^{-1}	3.03×10^{-1}	2.45×10^{-1}	3.12×10^{-1}
8	9.83×10^{-2}	2.21×10^{-1}	5.87×10^{-2}	1.04×10^{-1}
9	2.05×10^{-2}	9.11×10^{-2}	4.60×10^{-3}	9.62×10^{-3}
10	1.91×10^{-3}	1.71×10^{-2}	1.31×10^{-4}	1.86×10^{-4}
11	5.00×10^{-5}	1.12×10^{-3}	0	0
var	1.64×10^{-2}	1.69×10^{-2}	1.04×10^{-2}	1.04×10^{-2}

Table A.6: Long-run simulation results for $k = 11$.