Erasmus University Rotterdam

Erasmus School of Economics

Bachelor Thesis: International Bachelor Econometrics and Operations Research & International Bachelor Economics and Business Economics

The Extent to Which Learning Rules Impact the Level of Cooperation in Spatial Price Competition

Author:
Wendelien Bakelaar
Student Number:
369051

Supervisors:
Dr. Rommert Dekker
Dr. Jurjen Kamphorst

July 20, 2016

I would like to express my gratitude to Prof. dr. ir. R. Dekker and Dr. J.J.A Kamphorst for their feedback, insights and diverse perspectives throughout the process of this research.

Abstract

This paper investigates the extent to which the level of cooperation between spatially distributed firms differs with the use of alternative learning rules. The results are derived based on simulations of a structure conform to that presented by Waltman et al. (2013). Their model has been converted to Java in order to improve the efficiency of the model and simultaneously facilitate the comparison of various learning rules. Testing these rules in a homogeneous setting allows for comparability and this paper finds that imitation and reciprocity both lead to forms of cooperation, one on a global scale and the other on a local scale respectively. Furthermore, the results have proven to be robust to the influence of noise, experimentation and the presence of explicit cooperation in the form of a price agreement between two neighbouring firms. The level of information does however have an impact, often decreasing the level of cooperation between firms.

Contents

1 Introduction

Despite cooperation often being in the joint best interest of everyone involved, agents frequently fall prey to the desire to defect on their partners in order to obtain (temporary) higher payoffs. The conditions under which cooperation or altruism can persist in the long run, rather than being eradicated by egoistic behaviour, has been studied extensively both in theory and by means of simulation. However, these have not been studied in a homogeneous setting. As a result, due to differences in information, market structure, the presence of mutations or various other essential aspects, the results are frequently incomparable.

Cooperation between firms is of particular interest because the resulting increase in the firms' welfare comes at a cost of social welfare. Having a better understanding of the incentives and conditions for cooperative behaviour would be relevant for antitrust authorities who are in charge of preventing forms of cooperation between firms which are detrimental to consumers.

Cooperation in this setting refers to firms setting a price above their Nash Equilibrium price but the notion of cooperation also has strong implications in the field of evolutionary game theory. In this context agents often have two choices; to cooperate or to defect and winning in a game is equivalent to surviving in an evolutionary setting. The research in this field served as a starting point for the extensive analysis of cooperation, in particular the agents incentives to cooperate. An important underlying concept is that of repeated games which induce agents to delay short term gratification in order to benefit in later stages. Bergstrom and Stark (1993) used this concept to show that altruism, the equivalent of cooperating in a Prisoners Dilemma game, can prevail between siblings in an evolutionary setting.

Eshel et al. (1998) also proved that altruistic behaviour may survive, and in doing so define a model to help illustrate the nature of the agents steady state behaviour. In their work they consider two types of players; Altruists and Egoists, all of whom are arranged in a circle. They abandon two important assumptions, namely that of rational, utility maximising agents and equal interaction amongst players. Instead they consider learning rules in which players imitate the most successful strategy in their neighbourhood and they limit the scope of interaction amongst players. They derived all possible absorbing sets and find that this limited, local interaction is in fact essential for the existence of altruism and it has been an important factor of analysis since.

These researchers' theoretical findings on altruism have cleared the way for more complex forms of cooperation and have since been applied to various contexts. This concept of altruism and egoism, for example, runs parallel to cooperation and defection in the Prisoner's Dilemma. In this set-up both players would be better off cooperating, however, profit-maximising behaviour will often lead them to defect. How the players cooperate or defect is subject to the situation at hand and with respect to competition between firms it often refers to price-setting behaviour. This set-up between firms has been examined extensively in previous literature and in various settings. The simpler case of a duopoly has been used to investigate, among other things, the effect of different pricing policies, the conditions under which collusion is credible and the optimal locations for firms in the market (Gupta et al., 1997; Gross and Holahan, 2003; Thisse and Vives, 1988).

Although theoretical results have provided noteworthy insights into cooperative behaviour, the use of simulation has allowed for the analysis of increasingly complex game structures. Rather than considering duopolies, previous research has studied the steady state behaviour in markets with numerous firms and it has been shown that this behaviour is highly dependent on the market structure (Greenhut and Greenhut, 1975; Eaton and Lipsey, 1975). In particular, Pal (1998) has demonstrated the tendency for firms located in a circle to place themselves at equidistance of each other, justifying the extensive line of research in spatial price competition.

The steady state behaviour between homogeneous firms competing in this manner has been studied both by means of derivation and by simulation. These firms, arranged in either a circle or a torus, may select their level of cooperation from a set of strategies and have the opportunity to adjust this strategy according to a specified learning rule. Several variations of the learning rule have been provided, mainly involving forms of imitation or optimisation. Commonly used rules involve copying either the most successful strategy or the most successful player. These strategies may lead to some form of cooperative pricing, given that the firms only interact on a local level (Kirchkamp, 1999, 2000; Tieman et al., 2000). The importance of local interaction is emphasised by the work of Pinkse et al. (2002), who analyse spatial price competition using semiparametric methods and find that being nearest neighbours is the most important factor of rivalry.

In order to investigate the robustness of cooperative outcomes, several researchers have introduced a stochastic element to their models, namely mutations (Eshel et al., 1998; Waltman et al., 2013). Following this introduction, it has been proven that, under specific conditions, collusive behaviour may be interrupted by temporary price wars before returning to the same state of cooperation (Tieman et al., 2001). This reinforces the notion that the extent and duration of cooperation is highly-dependent on the exact setting in which firms compete. Furthermore, this dependence is what makes the results of previous research incomparable to one another.

For this reason, this paper will focus on the comparison of various learning rules in a homogeneous setting. The situation to be examined is competition between spatially distributed firms whose level of cooperation is depicted by their price-setting behaviour. Whereas theoretical derivations are essential for predicting steady state behaviour, the final results are obtained by means of simulation.

The learning rules used are such as presented by Waltman et al. (2013) and Tieman et al. (2001). These learning rules are based on imitation and reciprocity respectively and hence, cover a large section of the previous research done on cooperative behaviour. These rules are then adjusted in order to fully incorporate a players own profits and to enforce the use of up-to-date profits. In addition, the effect of explicit cooperation is analysed by introducing price agreements to the markets. Previous research tests for stability by considering unilateral deviation, but these price agreements are based on the profitability of bilateral deviation between two neighbouring firms.

The aim of this research is therefore to investigate under which circumstances cooperation can result between spatially distributed firms and how the degree of cooperation can be influenced by altering the structure of the simulation, in particular the learning rule of the individual firms.

The starting point for the investigation is the model as presented by Waltman et al. (2013). Their work involves simulations of price competition between spatially distributed firms on either a circle or torus. The corresponding learning rule of a firm involves imitating the strategy which has proven to be most successful in terms of average profit, given a certain level of noise, amongst the firms in its neighbourhood. The learning rule from Tieman et al. (2001), however, involves reciprocating cooperative or uncooperative behaviour by increasing or decreasing its price respectively. In addition, firms may increase or decrease their price at random based on the experimentation probability.

This paper finds that both categories of learning rules result in cooperation between firms, albeit it of a different nature. The rule based on imitation results in cooperation on a global level where firms set prices which are very close to or equal to the continuous Nash Equilibrium. The rule based on reciprocity on the other hand, results in a pattern of extremity where cooperation is of a more local scale. In this context groups of firms set either the maximum or minimum price and are interrupted by firms who set prices which tend to be in between the Nash Equilibrium and the maximum. The results are robust with respect to noise, experimentation and even to the presence of a price agreement. In fact, in many scenarios the price agreement is virtually undetectable in the market. The main factor of influence, however is the level of information available to firms. Similarly to the results found by Waltman et al. (2013), having more information may in fact diminish the extent of cooperation between firms.

The paper is structured as follows. Section 2 presents the model and its exact structure. Section 3 provides an overview of the implemented learning rules along with the expected outcomes of these learning rules. Section 4 discusses the price agreement and Section 5 goes on to present the results followed by the conclusion and suggestions for further research.

2 The Model

The model simulated in this paper is as follows. Four hundred homogeneous firms are spatially distributed over three different market types; a circle, a torus where consumers are located only on the line segments (Torus A) and a torus where consumers are located throughout the market (Torus B). A torus is a grid-like structure where the opposite ends are connected. This implies that the firms on the top row are neighbours with the firms on the bottom row and similarly, that firms on the left column are neighbours with firms on the right column. As a result, all firms have the same number of neighbours and there are no structural differences with firms in the centre of the market and those on the boundaries.

The firms are assumed to produce identical goods for which they may demand any price from their strategy set. The strategy set has the following structure

$$[0.5p_n, 0.55p_n, ..., 1.45p_n, 1.5p_n] \tag{1}$$

where p_n is the Nash equilibrium price. The Nash Equilibrium price equals 1 for the Circle and Torus A market structures, and equals 0.5 for Torus B. Firms also have a learning neighbourhood size, which is provided as input. The learning neighbourhood consists of the neighbours of which a firm knows the price and profit. A learning neighbourhood of $\rho = 2$ in the circular market, for example, implies that every firm knows the profit and price of its left and right neighbour. A learning neighbourhood of $\rho = 4$ consists of the two nearest left neighbours and the two nearest right neighbours and so on for ρ equal to 6, 8, 10 and 20. In a torus a firm has either a learning neighbourhood of $\rho = 4$, consisting of the direct horizontal and vertical neighbours, or $\rho = 8$, consisting of the eight neighbours surrounding the firm. Furthermore, firms have unlimited production capacity and a marginal cost of 0.

Consumers, which are uniformly distributed throughout the market, purchase exactly one unit of the product. The price of this good is the sum of the price set by the firm and the consumer's transportation cost. These transportation costs are equal to the distance travelled and hence consumers will always purchase from one of the firms in their direct vicinity.

The simulation then plays out as follows. First, initial strategies are chosen at random for all of the 400 firms. This is done at random in order to determine the steady state behaviour of the firms irrespective of the initial state. Then one million rounds are played during each of which one firm is selected to potentially adjust its price. After a firm is selected, the information required by the learning rule, the firm's profit and strategy as well as that of its neighbours, is computed. Based on this information and the nature of the pre-specified learning rule, the firm may adjust its price and the simulation moves on to the next round by selecting another firm. After one million rounds, the effect of the initial strategies is negligible and the behaviour as a result of the learning rule can be examined. This entire process is referred to as a run, and for each learning rule 500 runs and hence 500 different sets of initial strategies are simulated.

In addition, the robustness of the results is tested by allowing for noise and experimentation. Noise biases the information a firm has about its neighbours' profits. The noise level is indicated by the variable σ and the profits are increased or decreased by $\sigma * p_n * z$ where z is a draw from the standard Normal distribution. Experimentation on the other hand, tests for the robustness of the results with respect to mutations. At the end of each round, the firm selected may, with the specified experimentation probability, adjust its price either upwards or downwards with equal probability. The results are then considered stochastically stable if they are robust with respect to the experimentation probability.

The learning rules examined in this paper are as follows:

- Imitate Best Strategy: firms copy the strategy which earns the highest average profit (given a pre-specified level of noise) within their learning neighbourhood.
- Imitate Best Strategy Adjusted: firms copy the most successful strategy within their learning neighbourhood if the average profit of that strategy exceeds their current profit.
- Win Cooperate, Lose Defect(WCLD): firms increase their price by one step if the their neighbours' average profit exceeds their own and vice versa. In accordance with the work of Tieman et al. (2000), profit is defined as the profit resulting

from competing with all direct neighbours when they were last selected by the learning rule.

- Win Cooperate, Lose Defect Adjusted: firms adjust their prices according to the Win Cooperate, Lose Defect rule but using the most recent profit of their neighbours as input, rather than the profits of neighbours when it was last their turn.

In addition to these learning rules, this paper also considers the effect of explicit cooperation in the form of a price agreement. In this setting two neighbouring firms maintain their collusion price throughout the simulation whilst the remaining firms in the market abide by the learning rule. The two learning rules considered are Imitate Best Strategy Adjusted and Win Cooperate, Lose Defect Adjusted. Only the adjusted versions of these rules are considered due to the original rules' failure to fully incorporate own profits in the one and the use of outdated profits in the other. More details on the nature of the price agreement can be found in Section 4.

Of interest is the level of cooperation between firms as a result of the learning rules examined. Cooperation is defined as setting a price above the continuous Nash Equilibrium which is computed as follows. A Nash Equilibrium is a set of strategies such that, given the strategies of the other players, no player has an incentive to deviate (Nash, 1951). The Nash Equilibrium prices are 1 in the circular and toroidal A markets and 0.5 in the toroidal B market. To confirm this, if we were to maximise the unilateral deviation profit, given that the remaining firms in the market maintain the Nash Equilibrium price, this deviating firm should maximise its profit by setting the Nash Equilibrium, and it does. For the precise computation of the Nash Equilibrium for each market see Appendix A.4.

Although the structure of the model is conform to that presented by Waltman et al. (2013), some adjustments have been made. In the original model for example, there was a similar possibility to experiment after each round. With a pre-specified probability, firms would adjust their price upwards or downwards, each direction being equally likely. This experimentation possibility is still present in the models presented in this paper, however with the difference that only the firm chosen in that round has the ability to experiment whilst in the original model all firms could experiment during each round. The reasoning behind this adjustment is that in the original model the experimentation probabilities of 0.00001, 0.0001 and 0.001 would, in probability, result in 4000, 40,000 and 400,000 adjustments in 1,000,000 rounds. With such frequent adjustments, it is questionable to what extent the outcomes were a result of the learning rule rather than the experimentation

Moreover, due to the relatively robust results with respect to the noise levels and experimentation probability, the range of these parameters was limited. The noise levels considered were levels of 0 and 0.2 and the experimentation probabilities chosen were equal to 0 and 0.0001, the latter resulting in 100 mutations on average. For more details on the implementation of the model, see Appendix A.

3 Learning Rules

This section presents the learning rules in greater detail. Imitate Best Strategy is the learning rule presented by Waltman et al. (2013) and is one based on imitation. This learning rule is adjusted in order to fully incorporate a player's own profit when considering a price adjustment. Win Cooperate, Lose Defect is the learning rule presented by Tieman et al. (2001) and is one based on reciprocity. Given that this rule only considers outdated profits, it is also adjusted such that firms use only up-to-date information as input for their decision making process.

3.1 Imitate Best Strategy

This paper replicates and enhances the results derived by Waltman et al. (2013), but in addition, examines the effect of alternative learning rules on the level of cooperation in the set-up they presented. The learning rule they considered was such that firms, chosen at random, would play the strategy which earned the highest profit in the neighbourhood, subject to a certain degree of noise. Moreover, after potentially changing their strategy, firms are subject to possible mutations as a result of which they might increase or decrease their price. The profit for each firm is defined as follows: $\pi = \sum\limits_{ne \in N} \frac{p}{2}(p_{ne} - p + 1)$ where N is the set of direct neighbours, p_{ne} is the neighbour's price and p is the firm's price.

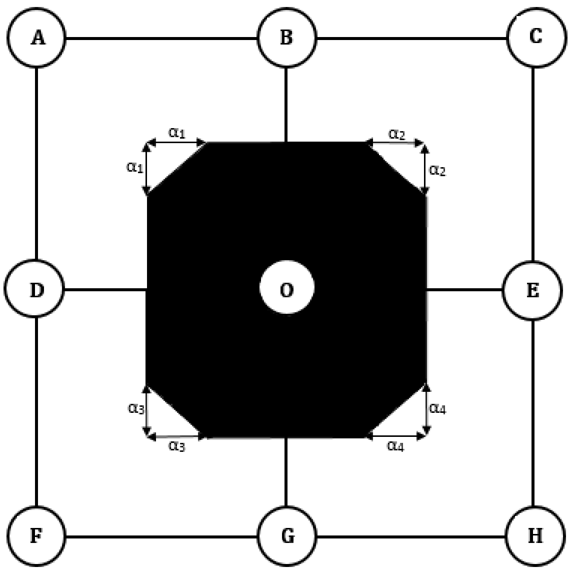


Figure 1: A segment of the Torus B market where firms play according to the Imitate Best Strategy learning rule and firm O's market share is indicated by the shaded region.

In the Torus B market, where consumers are located throughout the market rather than only on the line segments, indirect neighbours may also impact a firm's market share. For each indirect neighbour, firms A, C, F and H depicted in Figure 1, if they are significantly competitive, the firm's market share is trimmed by a value dependent on α, as can be seen in Figure 1. For indirect neighbour A, the formula for the value α is as follows:

$$\alpha = \frac{p_B - p_O + 1}{2} + \frac{p_D - p_O + 1}{2} - \frac{p_A - p_O + 2}{2} \tag{2}$$

If this value α is positive, then firm O's market share is reduced by $\frac{\alpha^2}{2}$. This is done for each indirect neighbour. Figure 1 shows an example where all of the indirect neighbours are competitive enough to reduce the market share of firm O, whose market share is indicated by the shaded area.

The results in this scenario should be virtually identical to those found by Waltman et al. (2013), namely that firms do act cooperatively and more so when the scope of interaction is limited. Similarly, the results should not be significantly impacted by noise, experimentation, nor the market structure. Moreover, as the frequency with which experimentation takes place has been decreased substantially, the effect of experimentation should become almost negligible.

3.2 Imitate Best Strategy Adjusted

A critique of the Imitate Best Strategy learning strategy, as incorporated by Waltman et al. (2013), is that the firm at hand does not fully take into account its own profit. The firm merely chooses the strategy with the highest average payoff, in its learning neighbourhood, and hence may switch to a strategy with an average payoff which is lower than its current payoff. Examples can be found in all settings considered in this paper such that according to the current 'Imitate the Best Strategy Rule', firms would switch to a strategy with an average payoff which is less than their current payoff.

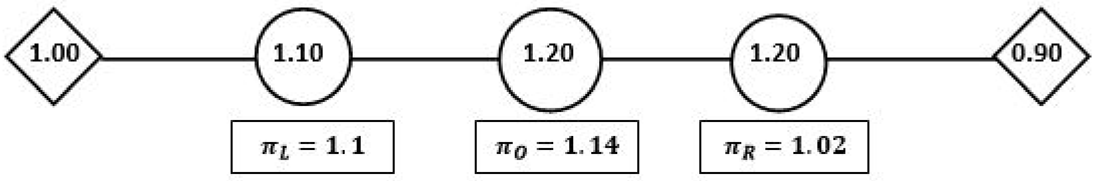


Figure 2: An example of where a firm (firm O depicted in the centre of the graph) would switch to a lower strategy with an average payoff which is less that its own current payoff in a circular market with a learning neighbourhood of $\rho = 2$.

Figure 2 shows such an example in the circular market structure with a learning neighbourhood $\rho = 2$. The firm in the middle has a current profit of 1.14, but due to the low profit of its right neighbour (who plays the same strategy), the learning rule dictates that it should decrease its price to that of its left neighbour. In a similar fashion, Figure 3 presents such a case for the toroidal market structure with $\rho = 8$. In this situation the firm in the centre will switch to a price of 1.05 despite the fact that his current profit exceeds the average profit of this strategy.

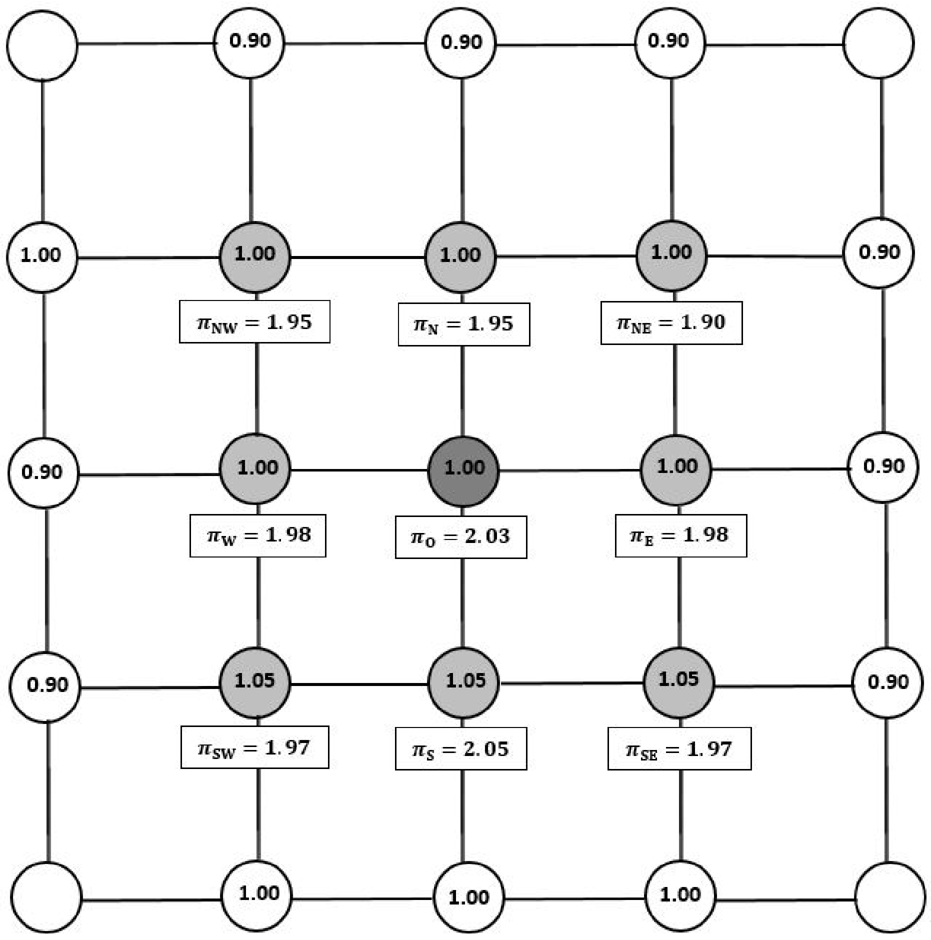


Figure 3: An example of where a firm (firm O depicted in the centre of the graph) would switch to a higher strategy with an average payoff which is less that its own current payoff in a Torus A market with a learning neighbourhood of $\rho = 8$.

	Circle $\rho = 2$		**Torus $\rho = 8$**	
	Original Price	**New Price**	**Original Price**	**New Price**
Profit Before Switch	1.08	1.1	1.96	1.99
Profit After Switch	0.96	1.1	1.96	2.01

Table 1: Payoff per strategy before and after firm O switches from a price of 1.20 to 1.10 as depicted in Figure 2, and before and after firm O switches from a price of 1.00 to 1.05 as depicted in Figure 3.

Various scenarios can be found such that firms will either increase or decrease their price to one with an average profit which is less than their current profit. Given that firms may only choose from strategies that are used in their neighbourhood (not taking into account the possibility to experiment), incorporating that firms will only switch if the average payoff exceeds their current payoff may not alter the outcomes significantly. However, as a result of the Imitate the Best Strategy rule the strategy which the firm (illogically) switches to becomes relatively more attractive. This can be seen in Table 1 which presents the average profits corresponding to the firm's original strategy and that which it switched to. In both cases, and in all cases presented in the appendix (see Appendix B), the profit corresponding to the chosen strategy increases more or decreases less than the profit corresponding to the original strategy. This induces a tendency for firms to cluster in terms of price and therefore adjusting this rule may result in a more scattered distribution of prices. The effect should be most significant in the absence of mutations which would interrupt clusters in the original setting.

Although this alteration may only have a minor impact on the results, it has important implications for the realism of the game structure. The information available to players is a key factor in game theoretical settings and many assumptions are made about the extent of the players' knowledge. In this paper firms only know the profits and strategies of firms in a given neighbourhood and profits are subject to noise. However, it should be a natural assumption that the firms would have perfect information about their own profits. This paper therefore also examines an adjusted version of the Imitate the Best Strategy rule so that firms only switch if the average profit of a strategy exceeds its current profit. The results are discussed in Section 5.

3.3 Win Cooperate, Lose Defect

Research on evolutionary game theory has studied various forms of behaviour, one category of which involving reciprocity; players reward cooperative behaviour of the other players by acting more cooperatively themselves and punish uncooperative behaviour by acting less cooperative. Tieman et al. (2001) propose an interesting variation of this rule where profit is measured as a result of competing against the most competitive neighbour, i.e. the neighbour with the lowest price. Moreover, cooperative behaviour, defined as when the average profit of a firm's neighbours is less than its own, is reciprocated by that firm increasing its price. Vice versa uncooperative behaviour is reciprocated by the firm acting more competitively by decreasing its price. Based on these conditions they find that states of cooperation are interrupted by temporary price wars, defined as a downward spiral of firms decreasing their prices, before returning to the same state of cooperation. Their findings are noteworthy due to the inherent instability of the firms' steady state behaviour.

Although there are various similarities in the structure of their work and that of Waltman et al. (2013), there are also several distinct differences. For one, Tieman et al. (2001) focused only on one market structure, namely the torus where firms have a learning neighbourhood of size eight. Moreover, they vary the range of prices from 2 to 20, finding that price wars exist when firms may choose from at least 12 prices. However, they do not incorporate noise nor experimentation which would provide an indication of the robustness of their results. Lastly, their toroidal market is of a larger dimension, containing 900 rather than 400 firms.

The reason for defining profit with respect to the most competitive neighbour is related to the nature of the consumers in their market. Consumers are located at the firms and for this reason they will either purchase at the firm at which they are located or, due to the linear transaction costs, to that firm's cheapest neighbour. In the market considered in this paper, consumers are located uniformly in the market, either on the line segments or throughout, and hence, a firm's profit is not merely determined by it's most competitive neighbour but is a function of all of its neighbours' prices.

Nevertheless, in correspondence with their work, a firm's profit is only updated once it is selected to apply the learning rule. The information a firm receives about the profits of its neighbours is therefore not a reflection of its neighbours current state of affairs but rather how those firms were off when they were last selected.

In the absence of noise and experimentation, the toroidal structures of this paper should result in similar price wars. However, it is questionable whether the triggers inducing price wars can occur when the learning neighbourhood is limited. This is because price wars are a result of one firm acting competitively, incentivising its neighbours to also lower their prices. The less firms in the direct neighbourhood, the less impact a competitive firm has and therefore the less likely that a price war will occur.

3.4 Win Cooperate, Lose Defect Adjusted

Albeit that the inventive learning rule presented by Tieman et al. (2001) leads to original results, it contains an unrealistic aspect. The learning rule which they associate with price wars involves using outdated information, namely the profits of firms when it was last their turn. Their model consists of a torus with 900 firms and due to firms being chosen at random, a firm's profit may have changed significantly between the time it last adjusted its strategy and the time that one of its neighbours uses this profit as input. For this reason, it would seem more realistic to consider the case where current profits are used as input and to investigate how the results would change.

A price war is a situation where firms act more competitively with one another and hence set lower prices. According to the Win Cooperate, Lose Defect rule this could only follow from a firm's profit being less than the average profit of its neighbours. This could be a result of the lagged information firms have because it does not account for the gradual transition towards equilibrium. Starting from randomly generated initial strategies, firms tend to gradually reach an equilibrium state during which their prices converge to an evermore limited price range and the difference in their profits decreases. The use of outdated information implies that the learning rule does not fully incorporate this conversion of profits due to which firms may be triggered to act more competitively than they would had they known the current profits. I would therefore expect that, when using recent information, firms will no longer deviate from an equilibrium once that state is reached, i.e. no price wars, and that the steady state behaviour of the firms will be cooperative.

4 Price Agreements

Rather than using simulations, various authors have approached the problem of cooperation between agents by using theoretical derivations. These methods can be used to predict the steady-state behaviour of firms and to determine the Nash Equilibria of these games. The Nash Equilibrium for the circle and torus of variant A are such that all firms set a price of 1. The equilibrium for the torus of variant B is one where all prices are set to 0.5. Note that a Nash Equilibrium is defined as the set of strategies such that, given the strategies of other players, no player has an incentive to deviate.

The key in this definition, and the steady state outcomes derived by other authors, is that they only consider unilateral deviation. This form of deviation does not take into account commonly known practices of unfair competition such as price agreements, cartels or collusion where more than one agent deviates from the equilibrium price. Whereas there is no incentive to unilaterally deviate, there is incentive to jointly deviate. This incentive is derived from the nature of consumers. By assumption consumers will always purchase from one of the firms in their direct neighbourhood and therefore colluding, neighbouring firms could set the maximum price and still serve the customers between them.

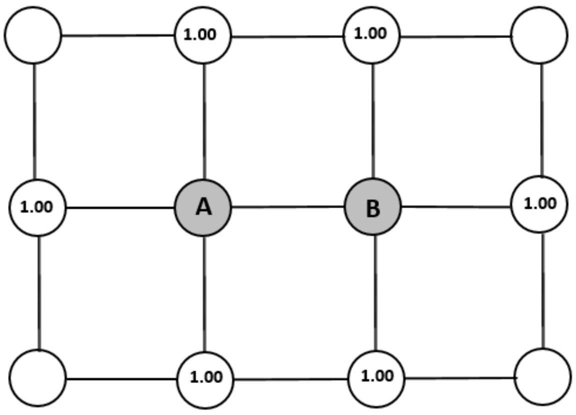


Figure 4: Illustration of the setting for the price agreement in the Torus A market. Firms A and B engage in a price agreement and all neighbours are assumed to set the Nash Equilibrium price of 1. The firms who have a price listed are those who determine the joint profit of firms A and B.

Consider the case as depicted by Figure 4. The structure is conform to that of a torus of variant A so that consumers are located on the line segments and a firm's profit is only affected by its direct horizontal and vertical neighbours. The Nash Equilibrium involves all firms setting their prices to p = 1. Now assume that firms A and B want to jointly deviate from this strategy. Due to symmetry, the profit of both A and B is equal to

$$\pi_A = \pi_B = (\frac{7 - 3p^*}{2})p^* \tag{3}$$

Optimising this profit with respect to their jointly chosen price p* leads to an optimal price of 1.16. This is not an option in our market structure, however Table 2 presents the profits corresponding to the closest alternatives as well as the Nash Equilibrium price of 1. It can be seen that jointly deviating is in fact profitable and that both firms would be better off doing so as long as their neighbours stick to the equilibrium price.

	Price = 1	**Price = 1.15**	**Price = 1.2**
Profit	2	2.041	2.040

Table 2: Profits for each of two neighbouring firms who take part in a price agreement setting either the Nash Equilibrium price of 1 or two possible collusion prices, given that the remaining firms in the market set the Nash Equilibrium price.

In order to test the stability of their cooperation, it is key to compute the payoffs A or B would receive from breaking their price agreement. Given that A and B set a price of

1.15, it would be optimal for either of the two firms to return to the Nash Equilibrium, earning them a profit of 2.075. However, this would only be temporary as the other firm would then return to the Nash Equilibrium as well. Assuming that the firms compete forever (i.e. an infinite game) and that future payoffs are discounted, we can compute the discount factor for which A and B will be willing to continue cooperating. In order to cooperate, the infinite sum of payoffs associated with cooperating needs to exceed the one-off payoff from deviating and the infinite sum of Nash Equilibrium payoffs which follow. The discount factor δ should be such that the following equation holds

$$2.075 + \sum_{i=1}^{\infty} 2 * \delta^i < \sum_{i=0}^{\infty} 2.041 * \delta^i \tag{4}$$

It follows that δ should be at least 0.45. This assumes however that firms can change prices immediately, which is not the case in the simulations. If there are just three rounds between one firm breaking the agreement and the other returning to the Nash Equilibrium then the discount factor should be at least 0.82. For the purpose of our investigation we will however assume that the discount factor is sufficiently large and that the firms trust each other.

Now let's consider the behaviour of the colluding firms' neighbours. Given that these firms are in the neighbourhood of only either A or B, their profit is the same as A or B's profit for deviating from the price agreement, namely 2.075. Their optimal strategy, disregarding the possibility of them joining the price agreement, is therefore to continue playing the Nash Equilibrium. In theory we would therefore expect a situation where all firms play the Nash Equilibrium price but firms A and B set a price of 1.15.

The corresponding collusion prices for the circular market and Torus B market can be found in a similar manner. For the circle we would optimise the prices of two neighbouring colluding firms given that all remaining firms play the Nash Equilibrium price of 1. This leads to a collusion price of 1.5 and a profit of 1.125 versus the Nash Equilibrium profit (where all firms set a price of 1) of 1. The Nash Equilibrium price in the Torus B market is 0.5 and given that all firms maintain this strategy, two neighbouring firms would maximise their profits by committing to a price of 0.6 earning them profits of 0.5115 rather than profits of 0.5.

In order to determine to the effect of bilateral cooperation in the form of price agreements, these agreements are incorporated explicitly in the models. For each market structure, two neighbouring firms are chosen to collude and to set the corresponding collusion prices (1.5 in the Circle, 1.15 in Torus A and 0.6 in Torus B). These firms will not apply the learning rule and their price agreement is therefore assumed to be stable, implying that they do not at any point in the simulation default on their agreement. The remaining firms will act as dictated by the learning rule. Given the objections of the original versions of Imitate Best Strategy and Win Cooperate, Lose Defect, only the adjusted versions of these learning rules will be applied.

It is of particular interest to investigate whether, upon inspection of the market, it is evident that there is a price agreement and which firms take part in this agreement. Seeing that cooperative behaviour tends to induce more cooperative behaviour (Bergstrom and Stark, 1993), it is expected that the remaining firms in the markets will also behave more cooperatively, especially those which find themselves in the direct neighbourhood of the collusive firms. However, it is of interest to know the exact impact on the market as a whole in terms of average prices, price dispersion and overall cooperation.

5 Results

For each of the learning rules, including or excluding a price agreement, 500 runs of the simulation are analysed. Of interest is the level of cooperation between firms in the final round of the model, as well as the stability of the outcomes with respect to the varying parameters. The stability is tested by examining the effect of the experimentation probability as well as the effect of noise. In addition, the final 1000 rounds are examined in order to determine if a steady state has been reached.

With respect to the price agreements, it is of interest to examine their impact on the overall level of cooperation in the market, as well as their impact on the behaviour of the colluding firms' direct neighbours. Moreover, it is examined whether or not it is obvious that there is a price agreement and if the partaking firms can be identified.

5.1 Imitate the Best Strategy

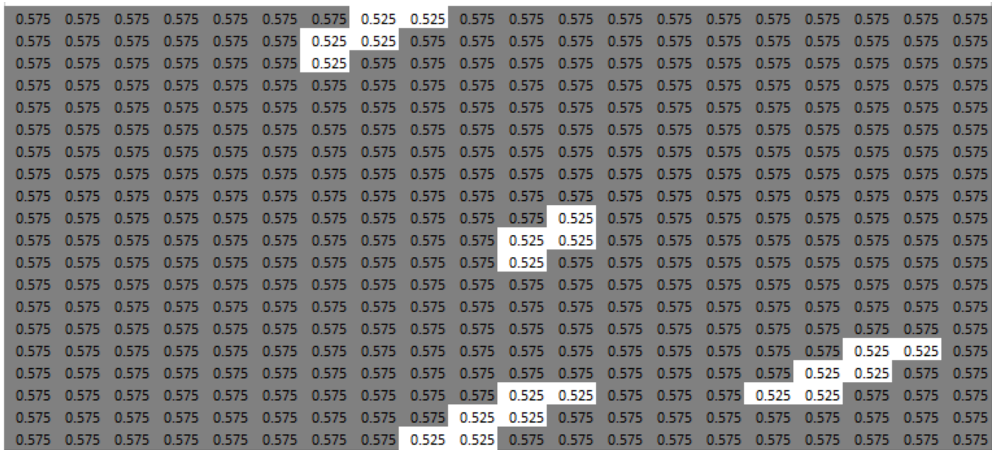
The mean prices resulting from the learning rule Imitate the Best Strategy are presented in Table 3. Given that cooperation is defined as setting a price above the continuous Nash Equilibrium p_n (equal to 1 for the Circle and Torus A market structures and 0.5 for Torus B), cooperation is not a frequent occurrence. In fact, the mean prices are only substantially above the Nash Equilibria in situations where the entirety of a firm's learning neighbourhood directly impacts its market share. The results in this sense reflect the idea that cooperation requires a limited, local interaction amongst agents (Eshel et al., 1998). This is the case in the circular market with two neighbours, the toroidal A market with 4 neighbours and in the toroidal B market. The mean prices in these cases range from 1.281 to 1.317 in the Circle, 1.035 to 1.164 in Torus A and from 0.505 to 0.583 in Torus B.

			Experimentation	
Market	**Noise**	**Neighbours**	**0**	**0.0001**
Circle	**0**	**2**	1.283 (0.029)	1.283 (0.023)
		4	0.901 (0.005)	0.898 (0.000)
		6	0.899 (0.008)	0.900 (0.001)
		8	0.901 (0.007)	0.900 (0.000)
		10	0.930 (0.014)	0.931 (0.013)
		20	0.981 (0.009)	0.991 (0.011)
	0.2	**2**	1.316 (0.034)	1.314 (0.032)
		4	0.928 (0.017)	0.927 (0.014)
		6	0.944 (0.014)	0.942 (0.013)
		8	0.953 (0.021)	0.952 (0.016)
		10	0.960 (0.016)	0.960 (0.015)
		20	0.979 (0.016)	0.978 (0.014)
Torus A	**0**	**4**	1.154 (0.014)	1.156 (0.010)
		8	1.036 (0.007)	1.035 (0.004)
	0.2	**4**	1.060 (0.013)	1.060 (0.011)
		8	0.996 (0.011)	0.995 (0.008)
Torus B	**0**	**4**	0.577 (0.006)	0.0579 (0.005)
		8	0.519 (0.004)	0.518 (0.003)
	0.1	**4**	0.534 (0.009)	0.535 (0.008)
		8	0.505 (0.008)	0.507 (0.007)

Table 3: The average mean prices of 500 simulation runs where firms abide by the Imitate Best Strategy learning rule. The standard deviations are presented in parentheses. The shaded entries indicate the combinations of parameters for which one of the Nash Equilibria was obtained.

Conversely, in the presence of external neighbours, who are in the firm's learning neighbourhood but whose strategies do not directly influence the firm's payoff, firms tend to act more competitively. In the circular structure this results in average prices which are two steps below the Nash Equilibrium. However, as the learning neighbourhood increases, the distance between the mean price and the Nash Equilibrium price converges to 0. In Torus A the mean prices stay close to the Nash Equilibrium price.

Despite the slight dispersion of outcomes, as indicated by the standard deviations, in several cases the continuous Nash Equilibrium or one of the discrete Nash Equilibria is reached. This is especially true in the absence of noise even though noise tends to bring the mean price closer to p_n. The settings for which one or more of the Nash Equilibria is reached are highlighted in Table 3. These are outcomes when the entire market sets the same price. The stability of these equilibria, as well as non-equilibrium outcomes, has been analysed by examining the development of the firms' prices in the final 1000 rounds. When a Nash Equilibrium is reached, the firms do not deviate from this state, however, non-equilibrium outcomes are less stable. In circular markets with a learning neighbourhood of $\rho = 10$ or higher, and in both toroidal markets significant dispersion of prices is observed. Moreover, in the final 1000 rounds of these outcomes, firms tend to fluctuate between a selected range of prices. This often occurs in clusters. However, in both tori with a learning neighbourhood of 4, there are also several fixed clusters where small subgroups deviate from the Nash Equilibrium price but no firm changes strategy during the final 1000 rounds. An example of this can be seen in Figure 5 where the Nash Equilibrium price is 0.575. Firms do not deviate from their strategies because the average profits in their respective learning neighbourhoods are highest for their current strategy.



0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.525	0.525	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.525	0.525	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.525	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.525	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.525	0.525	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.525	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.525	0.525	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.525	0.525	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.525	0.525	0.575	0.575	0.575	0.525	0.525	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.525	0.525	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575
0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.525	0.525	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575	0.575

Figure 5: Example output based on Imitate Best Strategy in Torus B with $\rho = 4$, $\sigma = 0$ and $\mu = 0$.

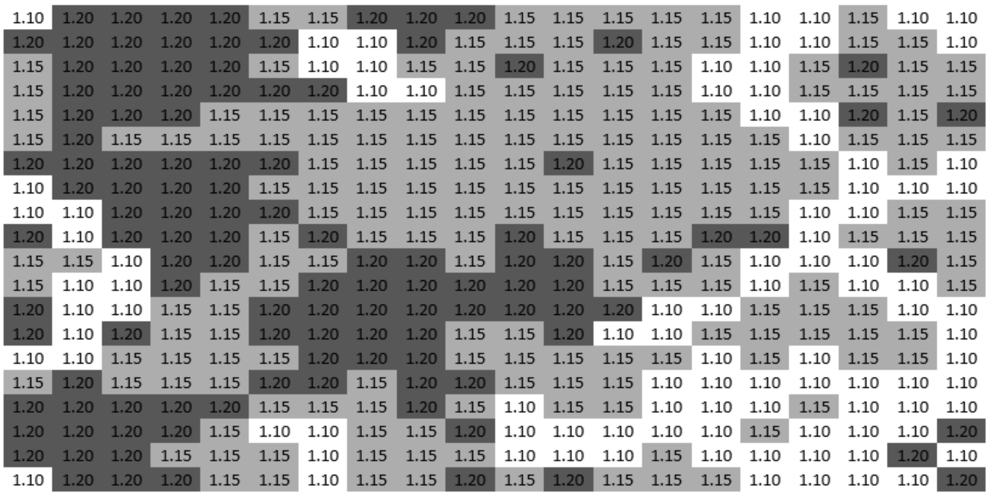
Another interesting outcome is one in the circular market with four neighbours. Here the set of prices switches from one where all firms set a price of 0.9 which the exception of one firm who sets a price of 0.8. Then for 390 rounds, its right neighbour also sets a price of 0.8 before returning to a price of 0.9 for the remaining 239 rounds.

5.2 Imitate Best Strategy Adjusted

As the learning rules Imitate Best Strategy and Imitate Best Strategy Adjusted do not differ substantially, it comes as no surprise that their resulting mean prices also do not differ substantially. The main difference however, is that the mean prices resulting from Imitate Best Strategy Adjusted are all closer to the continuous Nash Equilibria, on average one step (0.05) closer. The single exception to this is the Torus A market with $\rho = 8$ and $\sigma = 0.2$. However, the difference between the mean prices of this structure is less than 0.01. An overview of the mean prices along with their standard deviations is presented in Appendix C.

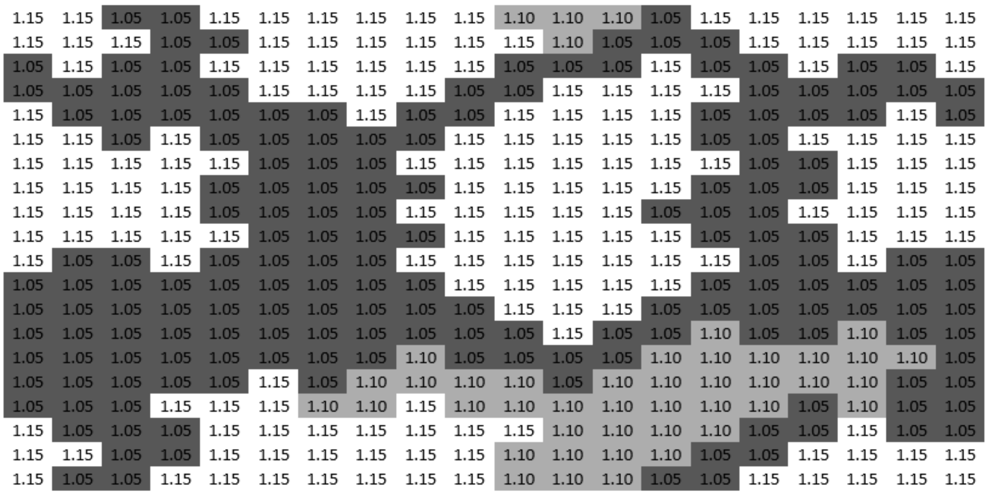
Another noteworthy difference between the learning rules is the frequency with which the Nash Equilibria are obtained. Whereas Nash Equilibria occurred rather sporadically in Imitate Best Strategy (only a few, if any, of the 500 runs reached an equilibrium), they occur much more frequently as a result of the Imitate Best Strategy Adjusted rule. To illustrate, for the circle with a learning neighbourhood of $\rho = 4$ and in absence of noise and experimentation, the continuous Nash Equilibrium was reached 40.6% of the time. For both tori with $\rho = 8$ this was even more substantial with an occurrence of 83.4% and 80.4% for Torus A and Torus B respectively.

It was predicted that the steady states would contain less clustering in prices, based on the findings in Section 3.2 where if firms switch to a strategy with an average profit less than their own, then this strategy becomes relatively more attractive. It turns out that the opposite is true. This is confirmed in the increased frequency with which the equilibrium states are reached, but also in the nature of clustering when the equilibrium is not reached. In Imitate Best Strategy clustering often takes place between small groups of firms, which mostly consist of only one or two members. In Imitate Best Strategy Adjusted however, the clusters tend to be significantly larger and sometimes even involve a 50/50 division where firms continually fluctuate between two prices. These differences can be illustrated by considering example output resulting from the Torus A market structure with a learning neighbourhood of $\rho = 4$ (excluding noise and experimentation).



1.10	1.20	1.20	1.20	1.20	1.15	1.15	1.20	1.20	1.20	1.15	1.15	1.15	1.15	1.15	1.10	1.10	1.15	1.10	1.10
1.20	1.20	1.20	1.20	1.20	1.20	1.10	1.10	1.20	1.15	1.15	1.15	1.20	1.15	1.15	1.10	1.10	1.15	1.15	1.10
1.15	1.20	1.20	1.20	1.20	1.15	1.10	1.10	1.15	1.15	1.20	1.15	1.15	1.15	1.10	1.10	1.15	1.20	1.15	1.15
1.15	1.20	1.20	1.20	1.20	1.20	1.20	1.10	1.10	1.15	1.15	1.15	1.15	1.15	1.10	1.10	1.15	1.15	1.15	1.15
1.15	1.20	1.20	1.20	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.10	1.10	1.20	1.15	1.20
1.15	1.20	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.10	1.15	1.15	1.15
1.20	1.20	1.20	1.20	1.20	1.20	1.15	1.15	1.15	1.15	1.15	1.20	1.15	1.15	1.15	1.15	1.15	1.10	1.15	1.10
1.10	1.20	1.20	1.20	1.20	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.10	1.10	1.10
1.10	1.10	1.20	1.20	1.20	1.20	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.10	1.10	1.15	1.15
1.20	1.10	1.20	1.20	1.20	1.15	1.20	1.15	1.15	1.15	1.20	1.15	1.15	1.15	1.20	1.20	1.10	1.15	1.15	1.15
1.15	1.15	1.10	1.20	1.20	1.15	1.15	1.20	1.20	1.15	1.20	1.20	1.15	1.20	1.15	1.10	1.10	1.10	1.20	1.15
1.15	1.10	1.10	1.20	1.15	1.15	1.20	1.20	1.20	1.20	1.20	1.20	1.15	1.15	1.15	1.10	1.15	1.10	1.10	1.15
1.20	1.10	1.10	1.15	1.15	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.10	1.10	1.15	1.15	1.15	1.10	1.10
1.20	1.10	1.20	1.15	1.15	1.20	1.20	1.20	1.20	1.15	1.15	1.20	1.10	1.10	1.15	1.15	1.15	1.15	1.15	1.10
1.10	1.10	1.15	1.15	1.15	1.15	1.20	1.20	1.20	1.15	1.15	1.15	1.15	1.15	1.10	1.15	1.10	1.15	1.15	1.10
1.15	1.20	1.15	1.15	1.15	1.20	1.20	1.15	1.20	1.20	1.15	1.15	1.15	1.10	1.10	1.10	1.10	1.10	1.10	1.10
1.20	1.20	1.20	1.20	1.20	1.15	1.15	1.15	1.20	1.15	1.10	1.15	1.15	1.10	1.10	1.10	1.15	1.10	1.10	1.10
1.20	1.20	1.20	1.20	1.15	1.10	1.10	1.15	1.15	1.20	1.10	1.10	1.10	1.10	1.10	1.15	1.10	1.10	1.10	1.20
1.20	1.20	1.20	1.15	1.15	1.15	1.10	1.15	1.15	1.15	1.10	1.10	1.10	1.15	1.10	1.10	1.10	1.10	1.20	1.10
1.10	1.20	1.20	1.20	1.15	1.15	1.10	1.15	1.15	1.20	1.15	1.20	1.15	1.15	1.15	1.10	1.10	1.10	1.10	1.20

Figure 6: Example of prices set in the Torus A market with a learning neighbourhood of $\rho = 4$, no noise $\sigma = 0$ and no experimentation $\mu = 0$ as a result of the Imitate Best Strategy learning rule.



1.15	1.15	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.15	1.10	1.10	1.10	1.05	1.15	1.15	1.15	1.15	1.15	1.15
1.15	1.15	1.15	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.15	1.10	1.05	1.05	1.05	1.15	1.15	1.15	1.15	1.15
1.05	1.15	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.15	1.05	1.05	1.05	1.15	1.05	1.05	1.15	1.05	1.05	1.15
1.05	1.05	1.05	1.05	1.05	1.15	1.15	1.15	1.15	1.05	1.05	1.15	1.15	1.15	1.15	1.05	1.05	1.05	1.05	1.05
1.15	1.05	1.05	1.05	1.05	1.05	1.05	1.15	1.05	1.05	1.15	1.15	1.15	1.15	1.05	1.05	1.05	1.05	1.15	1.05
1.15	1.15	1.05	1.15	1.05	1.05	1.05	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.05	1.05	1.15	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.15	1.05	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.05	1.05	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.05	1.05	1.05	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.05	1.05	1.05	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.05	1.05	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.05	1.05	1.05	1.15	1.15	1.15	1.15
1.15	1.15	1.15	1.15	1.15	1.05	1.05	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.05	1.05	1.05	1.15	1.15	1.15
1.15	1.05	1.05	1.15	1.05	1.05	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.05	1.05	1.15	1.05	1.05
1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.05	1.05	1.05	1.05	1.05	1.05
1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.15	1.15	1.15	1.05	1.05	1.05	1.05	1.05	1.05	1.05
1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.15	1.05	1.05	1.10	1.05	1.05	1.10	1.05	1.05
1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.10	1.05	1.05	1.05	1.05	1.10	1.10	1.10	1.10	1.10	1.10	1.05
1.05	1.05	1.05	1.05	1.05	1.15	1.05	1.10	1.10	1.10	1.10	1.05	1.10	1.10	1.10	1.10	1.10	1.10	1.05	1.05
1.05	1.05	1.05	1.15	1.15	1.15	1.10	1.10	1.15	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.05	1.10	1.05	1.05
1.15	1.05	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.10	1.10	1.10	1.10	1.05	1.05	1.15	1.05	1.05
1.15	1.15	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.15	1.10	1.10	1.10	1.10	1.05	1.05	1.15	1.15	1.15	1.15
1.15	1.05	1.05	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.10	1.10	1.10	1.05	1.05	1.15	1.15	1.15	1.15	1.15

Figure 7: Example of prices set in the Torus A market with a learning neighbourhood of $\rho = 4$, no noise $\sigma = 0$ and no experimentation $\mu = 0$ as a result of the Imitate Best Strategy Adjusted learning rule.

Given that, in this structure, a firm's neighbourhood consists only of its direct horizontal and vertical neighbours, Imitate Best Strategy leads to substantially more price clusters than Imitate Best Strategy Adjusted does. In fact, in Figure 6 we can find 22 clusters with a price of 1.20, 13 clusters with a price of 1.15 and 6 clusters who set a price of 1.10. This is in stark contrast to Figure 7 which contains a mere 6 clusters in total.

The reason for this unexpected outcome could be due to the nature of the examples considered. In those examples, the prices were not significantly dispersed and suggested that there was already significant clustering present. This would not be the case in the randomly generated initial strategies however and hence as a result of behaviour in the earlier rounds, firms would not tend towards the same price as commonly. The results do however support the existence of Nash Equilibria in the Imitate Best Strategy structure because as firms tend towards uniform behaviour, they follow this tendency up to the equilibrium. Another reason for the increased stability in the adjusted version of the learning rule is that firms are also less likely to switch prices. This is because with respect to the original learning rule, they are met with an additional requirement, namely, that the average profit of a strategy must exceed their own.

In line with the previous section, the stability of the outcomes has been examined by considering the final 1000 rounds. Again, once an equilibrium is reached, firms do not deviate. The large clusters, which are most common in toroidal markets, are relatively stable, even more so when only two prices are involved, and maintain the same approximate structure despite fluctuations. The circular market also experiences large clusters, as can be seen in Figure 8 but in other situations may switch between unilateral and bilateral deviation.

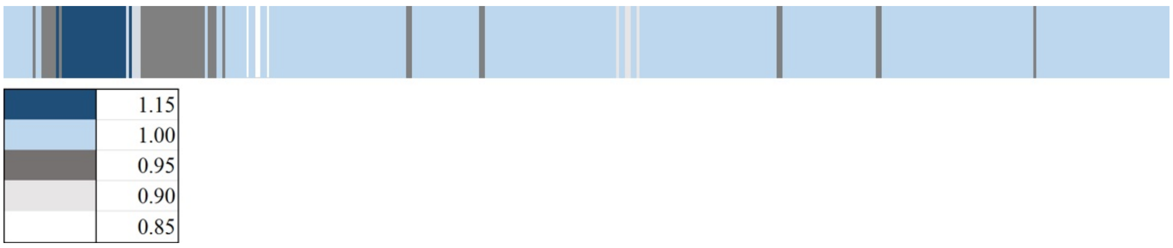


Figure 8: Example of prices set in the circular market with a learning neighbourhood of $\rho = 8$, no noise $\sigma = 0$ and no experimentation $\mu = 0$ as a result of the Imitate Best Strategy Adjusted learning rule.

5.3 Win Cooperate, Lose Defect

In Tieman et al.'s (2001) original work, they found that states of cooperation were temporarily disrupted by price wars, during which firms would set lower prices before returning to the same state of cooperation they were in before. This was only the case in their toroidal market structure with a substantial price range from which firms could choose. Profit was defined as a firm's profit whilst competing with the most competitive neighbour when they were last called to apply the learning rule. The reason for defining profit in this manner is that it follows naturally from the structure of their market. In this case, however, a firm's market share is determined by the prices set by all neighbours rather than just the most competitive one. By applying the learning rule to another market structure we test the robustness of their results. It turns out that the price wars found by Tieman et al. (2001) are not obtained in this context.

Albeit that the toroidal markets both show significant levels of cooperation in terms of mean prices, which lie several steps above the continuous Nash Equilibria; for Torus A the mean price lies between 1.117 and 1.155 and for Torus B the mean price lies between 0.552 and 0.581, there is no such inherent instability as was found by Tieman et al. (2001). Once a steady state is reached, there is no or limited deviation from these prices, deviation mainly occurring as a result of noise or experimentation. The outcomes in the circular market have a similar stability, albeit that the average behaviour is not as cooperative. In this market structure the mean prices remain close to the Nash Equilibrium price of 1, being 1.073 with a learning neighbourhood of just two and gradually decreasing to 0.964 as the learning neighbourhood increases.

Whereas the mean prices, being similar to those from Imitate Best Strategy, suggest that firms act according to the continuous Nash Equilibrium or a more cooperative equilibrium, this is not the case. In fact, upon inspection of the firms' price setting behaviour, an interesting pattern emerges. In the final output there are various groups of firms, mainly consisting of 2 to 6 members (clusters), setting either the maximum or minimum prices on an alternating basis. Interrupting these groups are individual or pairs of firms which take advantage of the extremity of their neighbours by setting prices which are most commonly between the maximum and the continuous Nash Equilibrium price.

Figure 9 illustrates one of the final outcomes resulting from the circular market with a learning neighbourhood of $\rho = 6$ and in absence of noise and experimentation. The mean price is very close to the Nash Equilibrium price of 1, but in fact the firms' pricing behaviour is far from unanimous. As can be seen there is an alternating sequence of firms setting the minimum price, indicated in white, followed by firms setting the maximum price, indicated in grey, with firms setting prices between 1 and 1.45, indicated in black, in between.

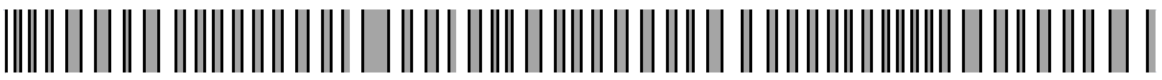


Figure 9: Example of prices set in the circular market with a learning neighbourhood of $\rho = 6$, no noise $\sigma = 0$ and no experimentation $\mu = 0$ as a result of the Win Cooperate, Lose Defect learning rule. The lighter shade indicates firms setting the minimum price of 0.5, the grey shaded areas indicate firms setting a price of 1.5 and the firms setting alternative prices are depicted in black.

This almost 50/50 like pattern is prominent in each of the market structures, the main difference being that the size of the clusters as well as the level of the interrupting prices increases with the learning neighbourhood. However, despite this diversity, the outcomes are relatively stable with minimal fluctuations. Win Cooperate, Lose Defect is a rule based on reciprocity. Firms reciprocate cooperative behaviour (the firm's profit exceeds the average profit of its neighbours) by increasing their price. Vice versa, they reciprocate uncooperative behaviour by lowering their price. The extremity in pricing is therefore a result of firms' profits being constantly above or below their neighbours' average payoffs.

There are however still slight fluctuations in pricing which take place even in the last of the one million rounds. These price changes occur when one of the firms in between the clusters is chosen by the learning rule. In order to illustrate the motivation for switching, let us consider an example from the structure with the most limited interaction amongst players, namely the circular market with a learning neighbourhood of $\rho = 2$.

Own Price	1.5	1.15	0.5	0.5	1.2	1.5	1.5	1.5	1.15	0.5
Own Profit	1.24	0.98	0.66	0.68	0.96	1.28	1.5	1.24	0.98	0.66
Avg. Profit Neighbours	1.15	0.95	0.83	0.81	0.98	1.23	1.26	1.24	0.95	0.82

Table 4: Prices and profits of a section of the circular market with $\rho = 2$, $\sigma = 0$ and $\mu = 0$ as a result of Win Cooperate, Lose Defect

One of the final outputs from this market is partially presented in Table 4. The firms' prices are presented in the top row, the firms' own profits in the middle row and the average profits corresponding to their neighbours are presented in the bottom row. Notice that the profits of the firms setting the maximum or minimum price are such that they would increase or decrease their prices respectively. However, as they cannot exceed the price range they continue to set their current prices. According to these current profits, the firms setting non-extreme prices also have an incentive to deviate. The second firm in the table, for example, setting a price of 1.25, has a profit of 0.938 whereas the average profit of its neighbours is 1. Therefore, according to the learning rule, this firm would decrease its price. Similarly, the fifth firm would decrease its price and the ninth firm would increase its price.

If we assume that the second firm indeed lowers its price by one step to 1.2, its profit will still be less than the average profit of its neighbours, and its neighbours incentives also do not change. If the second firm was once more selected to adjust its price it would again lower it but now, as a result of this further decrease, its own profit would exceed the average of its neighbours, whereas its neighbours incentives once more do not change. This implies that if the second firm was selected twice, and assuming its neighbours prices do not change, it would then increase its price back to 1.2. These incentives explain why these intermediate firms do not eventually set the maximum or minimum price like its neighbours but continue to act as barriers in the market.

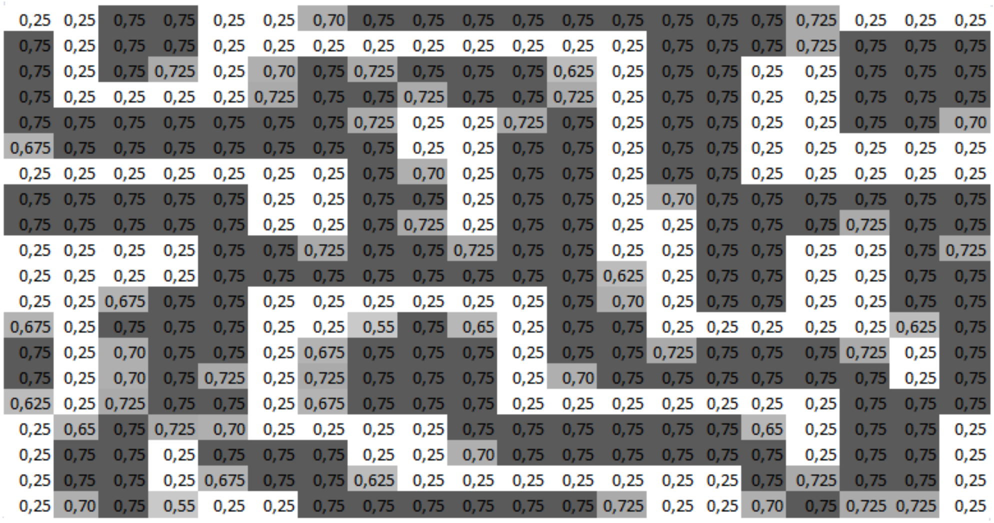
Note that given that profits are only updated once a firm is selected by the learning rule, the above description may not be entirely accurate. However, due to the relative stability of the final outcomes and the infrequency with which firms adjust their prices, profits will be for a large part up-to-date in the final rounds of the simulation. Furthermore, in the presence of noise or experimentation, such stability is compromised. This is because noise biases a firm's information about its neighbour' profits and experimentation forces them to adjust their strategy even when they have no incentive to do so.

The results presented in this section differ from those found by Tieman et al. (2001) in various ways. The outcomes presented here were fascinating in terms of their extremity yet those found by Tieman et al. (2001) were significant due to their inherent instability. The main cause for the difference in results is likely to be the effect of different cost structures and the nature of consumers. Whereas in this context firms are assumed to have zero marginal production costs and consumers are located uniformly either on the lines (Circle and Torus A) or throughout the market (Torus B), in the work by Tieman et al. (2001) marginal costs were assumed to be linear and consumers were located at the firms. Both of these dissimilarities result in an entirely different profit structure and hence different outcomes.

5.4 Win Cooperate, Lose Defect Adjusted

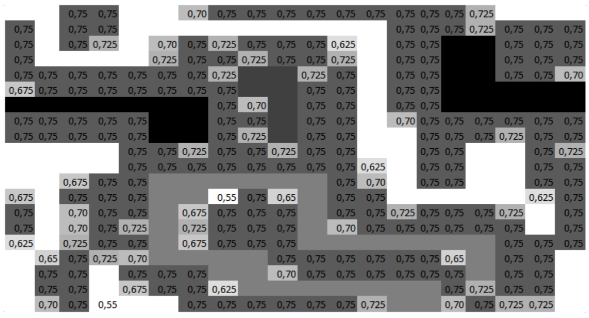
A critique of the Win Cooperate, Lose Defect(WCLD) rule is its use of outdated information with regards to the profits of firms. The learning rule WCLD Adjusted is an altered version of this rule such that firms use the most recent profits of their neighbours when considering potential price changes. Given that the outdated information could have been the cause for the price wars found by Tieman et al. (2001), updating the learning rule could remove the inherent instability of their steady states. However, the WCLD learning rule did not lead to price wars in the simulations presented in this paper. In fact, due to the relative stability in the general structure of the outcomes, in absence of noise and experimentation, firms do in fact use somewhat up-to-date profits in the final rounds. The main difference in output is then a result of what happens during earlier stages of the simulation.

As it happens, the results do not vary significantly in structure. Due to the nature of the profit functions and the lack of marginal costs, the reciprocity in the learning rule leads to the same form of extremity in pricing. Groups of firms set either the minimum price of 0.5 or 0.25 and the maximum price of 1.5 of 0.75 with firms between these groups setting a variety of prices and hence acting as barriers between the clusters. The main difference is that in the circular market firms act on average more cooperatively than they would in the original version of the learning rule, implying that the maximum price occurs more frequently than it did previously. This suggests that the use of outdated profits may indeed lead to more competitive behaviour amongst firms, however, the difference is small and in the toroidal markets even negligible.



0,25	0,25	0,75	0,75	0,25	0,25	0,70	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,725	0,25	0,25	0,25
0,75	0,25	0,75	0,75	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,75	0,75	0,75	0,725	0,75	0,75	0,75
0,75	0,25	0,75	0,725	0,25	0,70	0,75	0,725	0,75	0,75	0,75	0,625	0,25	0,75	0,75	0,25	0,25	0,75	0,75	0,75
0,75	0,25	0,25	0,25	0,25	0,725	0,75	0,75	0,725	0,75	0,75	0,725	0,25	0,75	0,75	0,25	0,25	0,75	0,75	0,75
0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,725	0,25	0,25	0,725	0,75	0,25	0,75	0,75	0,25	0,25	0,75	0,75	0,70
0,675	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,25	0,25	0,75	0,75	0,25	0,75	0,75	0,25	0,25	0,25	0,25	0,25
0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,75	0,70	0,25	0,75	0,75	0,25	0,75	0,75	0,25	0,25	0,25	0,25	0,25
0,75	0,75	0,75	0,75	0,75	0,25	0,25	0,75	0,75	0,25	0,75	0,75	0,25	0,70	0,75	0,75	0,75	0,75	0,75	0,75
0,75	0,75	0,75	0,75	0,75	0,25	0,25	0,75	0,725	0,25	0,75	0,75	0,25	0,25	0,75	0,75	0,75	0,725	0,75	0,75
0,25	0,25	0,25	0,25	0,75	0,75	0,725	0,75	0,75	0,725	0,75	0,75	0,25	0,25	0,75	0,75	0,25	0,25	0,75	0,725
0,25	0,25	0,25	0,25	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,625	0,25	0,75	0,75	0,25	0,25	0,75	0,75
0,25	0,25	0,675	0,75	0,75	0,25	0,25	0,25	0,25	0,25	0,25	0,75	0,70	0,25	0,75	0,75	0,25	0,25	0,75	0,75
0,675	0,25	0,75	0,75	0,75	0,25	0,25	0,55	0,75	0,65	0,25	0,75	0,75	0,25	0,25	0,25	0,25	0,25	0,625	0,75
0,75	0,25	0,70	0,75	0,75	0,25	0,675	0,75	0,75	0,75	0,25	0,75	0,75	0,725	0,75	0,75	0,75	0,725	0,25	0,75
0,75	0,25	0,70	0,75	0,725	0,25	0,725	0,75	0,75	0,75	0,25	0,70	0,75	0,75	0,75	0,75	0,75	0,75	0,25	0,75
0,625	0,25	0,725	0,75	0,75	0,25	0,675	0,75	0,75	0,75	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,75	0,75	0,75
0,25	0,65	0,75	0,725	0,70	0,25	0,25	0,25	0,25	0,75	0,75	0,75	0,75	0,75	0,75	0,65	0,25	0,75	0,75	0,25
0,25	0,75	0,75	0,25	0,75	0,75	0,75	0,25	0,25	0,70	0,75	0,75	0,75	0,75	0,75	0,75	0,25	0,75	0,75	0,25
0,25	0,75	0,75	0,25	0,675	0,75	0,75	0,625	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,75	0,725	0,75	0,75	0,25
0,25	0,70	0,75	0,55	0,25	0,25	0,75	0,75	0,75	0,75	0,75	0,75	0,725	0,25	0,25	0,70	0,75	0,725	0,725	0,25

Figure 10: Example of prices set in the Torus B market with a learning neighbourhood of $\rho = 8$, a noise level of $\sigma = 0.1$ and no experimentation $\mu = 0$ as a result of the Win Cooperate, Lose Defect Adjusted learning rule.



		0,75	0,75			0,70	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,725			
0,75		0,75	0,75										0,75	0,75	0,75	0,725	0,75	0,75	0,75
0,75		0,75	0,725		0,70	0,75	0,725	0,75	0,75	0,75	0,625		0,75	0,75			0,75	0,75	0,75
0,75					0,725	0,75	0,75	0,725	0,75	0,75	0,725		0,75	0,75			0,75	0,75	0,75
0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,725			0,725	0,75		0,75	0,75			0,75	0,75	0,70
0,675	0,75	0,75	0,75	0,75	0,75	0,75	0,75			0,75	0,75		0,75	0,75					
							0,75	0,70		0,75	0,75		0,75	0,75					
0,75	0,75	0,75	0,75	0,75			0,75	0,75		0,75	0,75		0,70	0,75	0,75	0,75	0,75	0,75	0,75
0,75	0,75	0,75	0,75	0,75			0,75	0,725		0,75	0,75			0,75	0,75	0,75	0,725	0,75	0,75
				0,75	0,75	0,725	0,75	0,75	0,725	0,75	0,75			0,75	0,75			0,75	0,725
				0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,625		0,75	0,75			0,75	0,75
		0,675	0,75	0,75							0,75	0,70		0,75	0,75			0,75	0,75
0,675		0,75	0,75	0,75			0,55	0,75	0,65		0,75	0,75						0,625	0,75
0,75		0,70	0,75	0,75		0,675	0,75	0,75	0,75		0,75	0,75	0,725	0,75	0,75	0,75	0,725		0,75
0,75		0,70	0,75	0,725		0,725	0,75	0,75	0,75		0,70	0,75	0,75	0,75	0,75	0,75	0,75		0,75
0,625		0,725	0,75	0,75		0,675	0,75	0,75	0,75								0,75	0,75	0,75
	0,65	0,75	0,725	0,70					0,75	0,75	0,75	0,75	0,75	0,75	0,65		0,75	0,75	
	0,75	0,75		0,75	0,75	0,75			0,70	0,75	0,75	0,75	0,75	0,75	0,75		0,75	0,75	
	0,75	0,75		0,675	0,75	0,75	0,625								0,75	0,725	0,75	0,75	
	0,70	0,75	0,55			0,75	0,75	0,75	0,75	0,75	0,75	0,725			0,70	0,75	0,725	0,725	

Figure 11: Example of prices set in the Torus B market with a learning neighbourhood of $\rho = 8$, a noise level of $\sigma = 0.1$ and no experimentation $\mu = 0$ as a result of the Win Cooperate, Lose Defect Adjusted learning rule.

Once again, noise may alter the outcomes slightly because it causes firms to believe that their neighbours' profits are slightly higher or lower than they really are. For this reason, the prices of firms between the clusters often only lies one step below the maximum price. An example is provided in Figure 10 which presents one of the final outcomes of the Torus B market with a learning neighbourhood of $\rho = 8$, a noise level of $\sigma = 0.1$ and no experimentation. In markets without noise, the firms in between clusters set a diverse range of prices between the continuous Nash Equilibrium of 0.5 and the maximum of 0.75. However, in the absence of noise, approximately half of these in between firms set the price which is one step removed from the maximum, namely 0.725. Of the 400 firms, 201 set the maximum price of 0.75 and 24 set a price of 0.725.

This figure also shows the interconnectedness of firms setting the maximum or minimum price, despite the influence of noise. Given that the learning neighbourhood is 8,there are only four clusters of firms setting a price of 0.5 and merely one large cluster setting a price of 0.75. The four clusters setting a price of 0.25 have been highlighted in Figure 11

5.5 Price Agreement in Imitate Best Strategy Adjusted

In this section, the results are discussed with respect to the situation where two neighbouring firms maintain a price agreement of the optimal collusion price of the corresponding market structure (a price of 1.5 in Circle, 1.15 in Torus A and 0.6 in Torus B), and the remaining firms follow the Imitate Best Strategy Adjusted learning rule. It was expected that the firms, especially those within the direct neighbourhood of the colluding firms, would set higher prices than in the absence of the price agreement and that firms would in general behave more cooperatively in this context. This is confirmed in terms of the mean prices which are on average 0.006 higher due to the existence of a price agreement. The difference is larger in the presence of noise, with an average difference of 0.010. In terms of mean prices, the effect is not substantial enough in order to suggest the existence of a price agreement, but this does not imply that upon further inspection of the market, the price agreement cannot be identified. For an overview of the mean prices for each combination of parameters see Appendix C.

Besides the effect of a price agreement on the average market price, it is also of interest to examine the effect of such an agreement on the structure of the market. In particular, whether upon inspection of the market, it can be determined that there is explicit cooperation between firms and if the two colluding firms can be easily identified. In this aspect there is quite a lot of variation even within the output of just one set of parameters.

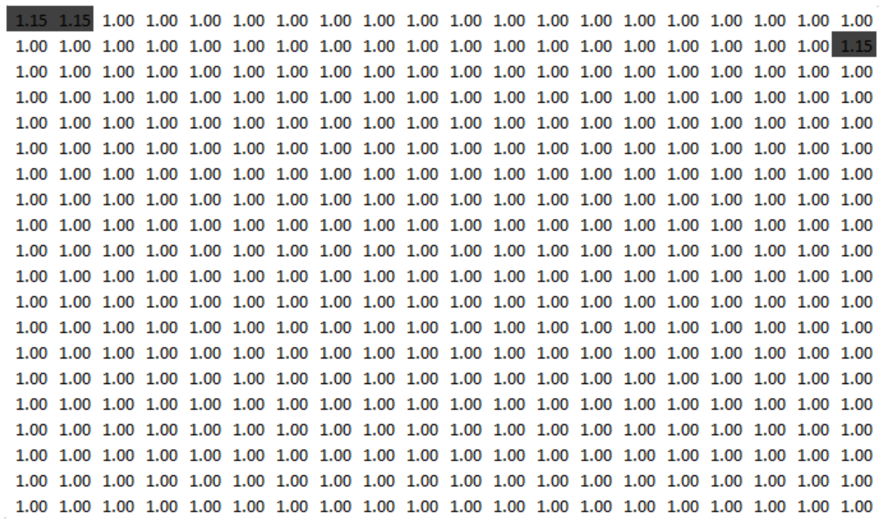
If we start with the simplest market structure; the circular market with only two neighbours, no noise and no experimentation, we see that the results are quite similar to that of Imitate Best Strategy Adjusted. There are large clusters of firms setting prices above the continuous Nash Equilibrium of 1, interrupted by one or two firms setting prices which are significantly below the Nash Equilibrium. In Imitate Best Strategy Adjusted, the collusion price of 1.5 in this market was a regular occurrence, being part of 185 of the 500 final outcomes. Now the price agreement forces this price to be present in all 500 outcomes but the remaining firms do not deviate significantly in behaviour.

Cooperative behaviour tends to induce more cooperative behaviour (Bergstrom and Stark, 1993), hence it is expected that the direct neighbours of the colluding firms would set higher prices. In 42.4% of the 500 cases, one or more of the direct neighbours also sets the collusion price of 1.5, making it impossible to identify which firms are partaking in the price agreement. Moreover, due to the variety of prices it isn't clear in this context that there is a price agreement. There are three main types of output; one where direct neighbours set the collusion price, one where direct neighbours set prices above the continuous Nash Equilibrium and one where direct neighbours take advantage of the high price set by colluding firms by setting prices which are significantly below the continuous Nash Equilibrium. Either the neighbours on each side act similarly or they take a combination of these types of behaviour. Some examples are presented below in Table 5.

	Direct Neighbour	**Colluding Firms**		**Direct Neighbour**	
0.65	0.65	1.5	1.5	0.85	0.85
0.55	1.5	1.5	1.5	1.5	0.85
0.65	0.65	1.5	1.5	1.4	1.4

Table 5: The prices set by the colluding firms and their direct neighbours as a result of the Imitate Best Strategy Adjusted in the circular market with a learning neighbourhood of $\rho = 2$, no noise and no experimentation

In all remaining market structures there is a similar disparity in the behaviour of the colluding firms' direct neighbours. The main difference, however, is the general behaviour of the remaining firms in the market. Whereas in the circle with a learning neighbourhood $\rho = 2$ there was significant dispersion in prices, often ranging from the minimum price of 0.5 to the maximum price of 1.5, the remaining market structures have much less variation. In fact, similarly to Imitate Best Strategy Adjusted, the majority of firms set prices which are equal to the continuous Nash Equilibrium prices of 1 (in the Circle or Torus A markets) or 0.5 (in the Torus B market) or prices which are merely one step away from these Nash Equilibria. In both tori with learning neighbourhoods of $\rho = 4$ however, the prices are often closer to or equal to the collusion prices (1.15 in Torus A and 0.6 in Torus B). In these settings it is therefore impossible to clearly identify the presence of a price agreement, whereas in the other market structures it may well be. An example of Torus A with $\rho = 8$ where the price agreement is quite evident is presented in Figure 12 where all firms set the continuous Nash Equilibrium price of 1 with the exception of the colluding firms and one of their direct neighbours.



1.15	1.15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.15
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Figure 12: Example of prices set in the Torus A market with a learning neighbourhood of $\rho = 8$, a noise level of $\sigma = 0$ and no experimentation $\mu = 0$ as a result of the Imitate Best Strategy Adjusted learning rule in the presence of a price agreement between the adjacent firms in the top left corner.

It follows that especially in the case of limited, local interaction it is often indeterminable which two firms are partaking in a price agreement and whether or not there is a price agreement at all. The larger the scope of interaction, the more uniform the firms' behaviour becomes and the easier it becomes to identify the price agreement.

5.6 Price Agreement in Win Cooperate, Lose Defect Adjusted

In this section we present the results of the various markets where two firms maintain a price agreement whilst the remaining firms act according to the Win Cooperate, Lose Defect Adjusted learning rule. In the absence of a price agreement we found a consistent pattern in the outcomes. Small groups of direct neighbours would set either the maximum or minimum market price and would be interrupted by one or two firms setting prices between the continuous Nash Equilibrium and the maximum price. In the presence of a price agreement, two firms are forced to maintain the optimal collusion price equal to 1.5 in the circular market, 1.15 in Torus A and 0.6 in Torus B. Given that 1.5 equals the maximum price in the circular market and hence was quite prominent in absence of the price agreement, the price agreement should not have a significant impact on the behaviour in the market as a whole. The toroidal markets may be more heavily influenced.

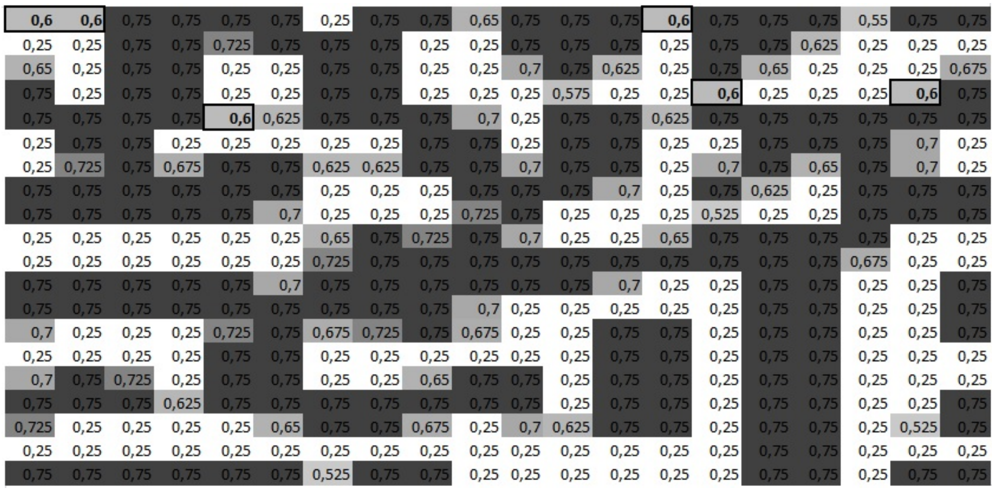
In terms of general cooperativeness, the price agreement hardly makes an impact. The mean prices are the slightest bit higher but the difference is not substantial enough to suggest the presence of a price agreement. What is more interesting then, is the impact the price agreement has on the structure of the market and whether the colluding firms can be identified. For this purpose, it is essential to examine the market more closely, and in particular the direct neighbours of the colluding firms.

Market Structure	**Direct Neighbours Set Collusion Price (%)**
Circle $\rho = 2$	52.2
Circle $\rho = 4$	69.8
Circle $\rho = 6$	68.8
Circle $\rho = 8$	81
Circle $\rho = 10$	86.6
Circle $\rho = 20$	92.4
Torus A $\rho = 4$	8.4
Torus A $\rho = 8$	7.4
Torus B $\rho = 4$	18
Torus B $\rho = 8$	10.2

Table 6: The percentage of the 500 simulations run in which one of the direct neighbours of the colluding firms sets the collusion price.

Table 6 presents the percentages corresponding to the amount of times one of the direct neighbours; defined as the neighbours whose behaviour has a direct impact on the market share of the firm, sets the collusion price. In the circular market this is especially high, ranging from 52.2% to a substantial 92.4%, increasing as the learning neighbourhood of the firms increases. This implies that as the scope of the interaction amongst firms increases, it becomes exceedingly difficult to identify the two colluding firms.

In the toroidal market, however, it is often quite easy to distinguish that two firms are deviating from the general behaviour in the market due to the limited occurrence of the collusion prices. However, as these prices (1.15 in Torus A and 0.6 in Torus B), lie below the maximum market prices, it would not raise suspicion by anti-trust authorities. Let us consider an example from the Torus B market with a learning neighbourhood of $\rho = 8$ as presented in Figure 13. The deviating firms are the two adjacent firms in the top left corner of the torus, as indicated in bold. None of their direct neighbours also set the collusion price, however, four firms spread out over the market, indicated in bold, do. Moreover, the firms setting neither the maximum or minimum price set a range of prices starting from 0.525 to 0.725, the collusion price being almost exactly in the middle.



0,6	**0,6**	0,75	0,75	0,75	0,75	0,25	0,75	0,75	0,65	0,75	0,75	0,75	**0,6**	0,75	0,75	0,75	0,55	0,75	0,75
0,25	0,25	0,75	0,75	0,725	0,75	0,75	0,75	0,25	0,25	0,75	0,75	0,75	0,25	0,75	0,75	0,625	0,25	0,25	0,25
0,65	0,25	0,75	0,75	0,25	0,25	0,75	0,75	0,25	0,25	0,7	0,75	0,625	0,25	0,75	0,65	0,25	0,25	0,25	0,675
0,75	0,25	0,75	0,75	0,25	0,25	0,75	0,75	0,25	0,25	0,25	0,575	0,25	0,25	**0,6**	0,25	0,25	0,25	**0,6**	0,75
0,75	0,75	0,75	0,75	**0,6**	0,625	0,75	0,75	0,75	0,7	0,25	0,75	0,75	0,625	0,75	0,75	0,75	0,75	0,75	0,75
0,25	0,75	0,75	0,25	0,25	0,25	0,25	0,25	0,75	0,75	0,25	0,75	0,75	0,25	0,25	0,75	0,75	0,75	0,7	0,25
0,25	0,725	0,75	0,675	0,75	0,75	0,625	0,625	0,75	0,75	0,7	0,75	0,75	0,25	0,7	0,75	0,65	0,75	0,7	0,25
0,75	0,75	0,75	0,75	0,75	0,75	0,25	0,25	0,25	0,75	0,75	0,75	0,7	0,25	0,75	0,625	0,25	0,75	0,75	0,75
0,75	0,75	0,75	0,75	0,75	0,7	0,25	0,25	0,25	0,725	0,75	0,25	0,25	0,25	0,525	0,25	0,25	0,75	0,75	0,75
0,25	0,25	0,25	0,25	0,25	0,25	0,65	0,75	0,725	0,75	0,7	0,25	0,25	0,65	0,75	0,75	0,75	0,75	0,25	0,25
0,25	0,25	0,25	0,25	0,25	0,25	0,725	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,675	0,25	0,25
0,75	0,75	0,75	0,75	0,75	0,7	0,75	0,75	0,75	0,75	0,75	0,75	0,7	0,25	0,25	0,75	0,75	0,25	0,25	0,75
0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,7	0,25	0,25	0,25	0,25	0,25	0,75	0,75	0,25	0,25	0,75
0,7	0,25	0,25	0,25	0,725	0,75	0,675	0,725	0,75	0,675	0,25	0,25	0,75	0,75	0,25	0,75	0,75	0,25	0,25	0,75
0,25	0,25	0,25	0,25	0,75	0,75	0,25	0,25	0,25	0,25	0,25	0,25	0,75	0,75	0,25	0,75	0,75	0,25	0,25	0,25
0,7	0,75	0,725	0,25	0,75	0,75	0,25	0,25	0,65	0,75	0,75	0,25	0,75	0,75	0,25	0,75	0,75	0,25	0,25	0,25
0,75	0,75	0,75	0,625	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,25	0,75	0,75	0,25	0,75	0,75	0,25	0,25	0,75
0,725	0,25	0,25	0,25	0,25	0,65	0,75	0,75	0,675	0,25	0,7	0,625	0,75	0,75	0,25	0,75	0,75	0,25	0,525	0,75
0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,75	0,75	0,25	0,25	0,25
0,75	0,75	0,75	0,75	0,75	0,75	0,525	0,75	0,75	0,25	0,25	0,25	0,25	0,25	0,25	0,75	0,75	0,25	0,75	0,75

Figure 13: Example of prices set in the Torus B market with a learning neighbourhood of $\rho = 8$, a noise level of $\sigma = 0.1$ and no experimentation $\mu = 0$ as a result of the Win Cooperate, Lose Defect Adjusted learning rule in the presence of a price agreement between the adjacent firms in the top left corner.

To conclude, two firms may set a price agreement in a market which abides by the Win Cooperate, Lose Defect Adjusted learning rule without raising any suspicion. The overall market behaviour does not change significantly, the deviating firms cannot be pinpointed, and their behaviour is not distinct enough to make any real impact on the level of cooperation amongst firms.

6 Conclusion

The aim of this paper has been to examine the effect of various learning rules on the level of cooperation between spatially distributed firms. In addition, to investigate the impact of explicit cooperation between two neighbouring firms who maintain a price agreement. The results were obtained by simulating circular and toroidal markets where homogeneous firms were set at equidistance of each other and consumers were located uniformly, either on the line segments or throughout. For 500 distinct sets of initial strategies, one million rounds were simulated. During each round, one firm was chosen at random in order to apply the learning rule and hence potentially adjust its price.

The first learning rule examined was the Imitate Best Strategy rule, where firms copy the strategy which proves most successful in their learning neighbourhood. In the case of limited interaction amongst firms, the market shows significant cooperation, the average price being several steps above the continuous Nash Equilibrium price. Moreover, the continuous Nash Equilibrium as well as one of the discrete Nash Equilibria occurs for various structures.

The Imitate Best Strategy rule was then adjusted so that firms only adjust their prices if the mean price of a neighbour's strategy exceeds their own current profit. In this scenario the overall level of cooperation is similar to that resulting from the original learning rule, however, whereas the Nash Equilibria were previously a sporadic occurrence, in this setting they occur more often than not. Adjusting the learning rule therefore results in more stable behaviour amongst firms and a more uniform market in all settings.

After adjusting the learning rule, the final step was to examine the effect of explicit cooperation in the form of a price agreement. Two firms would maintain a price agreement of 1.5 in the Circle, 1.15 in Torus A and 0.6 in Torus B whilst the remaining firms followed the Imitate Best Strategy Adjusted rule. The main matter of importance was whether, upon inspection of the market, it could be determined that a price agreement was present. In terms of mean prices this was not the case, however in terms of pricing behaviour it often was. In markets where all firms in the learning neighbourhood directly impact a firm's market share, one of the direct neighbours of the colluding firms often sets the collusion price. This makes it impossible to exactly pinpoint the two colluding firms, at least without accusing the direct neighbour. When the scope of interaction increases, however, the price agreement becomes easier to identify. The reason for this is that the remaining firms approach the continuous Nash Equilibrium whilst the collusion prices lie above this equilibrium.

The next category of learning rules was one based on reciprocity, referred to as Win Cooperate, Lose Defect. This learning rule dictates that if the average profit of a firm's neighbours exceeds its own profit, the firm will reciprocate this uncooperative behaviour by decreasing its price. Vice versa, firms reciprocate cooperative behaviour by increasing their price. The original version of this rule involves the use of outdated profits which are only updated once a firm is selected by the learning rule. The adjusted version corrects this by allowing firms to use the most up-to-date profits of their neighbours as input for the decision making process.

Both versions of the learning rule result in a distinct pattern in the markets. Small groups of firms (clusters) set either the minimum or maximum market price and are interrupted by individual or pairs of firms setting alternative prices. These prices tend to lie between the continuous Nash Equilibrium price and the maximum and may be adjusted even in the final rounds of the simulation. Increasing the learning neighbourhood of firms does not impact the general pattern, however, it does increase the size of the clusters.

Due to the prominence of the maximum price in all market structures, and the differentiation amongst the non-extreme firms, the price agreement did not have a significant impact on the general behaviour in the market. Once more, the presence of a price agreement does not raise suspicion in terms of mean prices nor in terms of market structure.

The two main learning rules, based on imitation and reciprocity, have lead to almost polar opposites in terms of market structures despite being extremely similar in terms of average market prices. Whereas the Imitate Best Strategy rule led to prices which centred around the continuous Nash Equilibrium, the Win Cooperate, Lose Defect rule conversely led to prices on opposite ends of the equilibrium. All results have proven to be stochastically stable due to their robustness with respect to the experimentation probability. Noise also did not have a significant impact on the structure of the outcomes, but often led to firms being one step above or below the prices they would have otherwise set due to the biased information. The main factor of influence was, just as in the work of Waltman et al. (2013), the learning neighbourhood. Increasing the scope of interaction of firms tends to decrease the level of cooperation. Moreover, the level of cooperation tends to be highest when a firm's neighbourhood only consists of the firms which directly impact its market share.

To conclude, both markets found forms of cooperation, albeit of completely different nature, and were robust to external influences on the firm's behaviour. Cooperation as a result of imitation is of a more global form, resulting in uniform behaviour across the market. Cooperation as a result of reciprocity is of a more local nature resulting in clusters of firms taking advantage of their neighbours' cooperation. Even in the presence of explicit cooperation, this implicit cooperation between firms remains robust.

7 Limitations and Suggestions for Further Research

The main contribution of this research has been to compare and adjust previously examined learning rules in a homogeneous setting and thereby to test the robustness of the results found by Waltman et al. (2013) and Tieman et al. (2001). The setting in which these results were obtained was based on that presented by Waltman et al. (2013) with the only exception being the frequency with which the experimentation possibility is applied.

In order to obtain a more global picture on how learning rules influence the level of cooperation in a market, alternative learning rules, for example one where players imitate their most successful neighbour rather than the most successful strategy, could also be examined. In terms of explicit cooperation, it may also be of interest to consider when more than two firms collude and to determine how many firms should partake in a price agreement in order to significantly impact the market as a whole.

Whereas this paper has focused on the effect of various learning rules, it may also be of interest to determine the sensitivity of these results to factors related to the market structure rather than the way firms make decisions. Examples are the size of the market, the use of continuous prices, the nature of the profit function or the distribution of consumers. Evolutionary game theory is an immensely broad topic focusing on various players, diverse contexts and the analysis of which has been approached both in theory and by means of simulation. It is a topic which allows for significant generalisation as well as extreme specificity which makes the comparability of previous research limited. This paper has allowed for this comparability whilst maintaining a reasonable scope, outside of which there are still various factors of influence to be examined.

A Appendix - The Model

A.1 Implementation

The model used for the simulations involves three different market structures; a circle, a torus where consumers are located on the line segments and a torus where consumers are located throughout. Each round a firm is chosen at random to potentially adjust its price according to a pre-specified learning rule and the steady state behaviour is then studied. The details of the model is such as presented by Waltman et al. (2013), however their model has been adjusted and moved to another platform. The simulations performed by Waltman et al. (2013) were for a large part programmed in the language C and thereafter implemented in MATLAB. This was done by converting the C code to C-MEX files which are compatible with MATLAB. Albeit that C has advantages in terms of efficiency with respect to MATLAB, there are several issues related to adjusting C-MEX files, particularly with debugging (Gamma, 1995). Java, on the other hand, is a high-level object-orientated programming language which is efficient in terms of speed and should have no such troubles when adjusting code for alternative learning rules (Guzdial and Ericson, 2007).

For this reason, the simulation models for both the circle and torus (amounting to approximately 600 lines of code) were translated to Java and adjusted accordingly to compare the outcomes for various learning rules. The model makes use of Java's object orientated programming with both Market and Firm interfaces to depict both market structures and the set of firms corresponding to the different learning rules. The classes "RunCircle", "RunTorusA" and "RunTorusB" (the general code for these and all other classes can be found in Appendix E) initialise the markets and play one million rounds for 500 different initial strategy sets, chosen at random. Due to the nature of the random number generators, all results are reproducible. The amount of code has been decreased to approximately 300 lines and the running time has been halved (see Appendix A.3 for the comparison of running times), whilst producing identical output to those presented in the original model.

A.2 Random Number Generators

Given that the results of these simulations are highly dependent on stochastic elements such as the randomly generated initial strategies, the choice of firms for each round and the noise level, it is essential to have a sophisticated random number generator. Java's standard generators Math.random and java.util.Random are fine in some contexts, but due to their limited period of 2^{48} the frequency with which these methods would be called in the simulations may result in serial correlation (L'Ecuyer and Simard, 2007). For this reason, it was decided to use one of the random number generators developed by L'Ecuyer et al. (2002). The authors provide an array of random number generators and tools for stochastic simulation, and from this list the generator LFSR113 was chosen. The reasoning behind this decision was that LFSR113 had the best combination of speed and diversity for the context of this paper. With a period of 2^{113} serial correlation should not be a problem. What is more, the use of one of the other generators with a larger period would reduce simulation time and would be somewhat exaggerated for simulations where the maximum range generated integers lies between 0 and 399.

A.3 Speed of Model

Market	**Neighbourhood**	**Wendelien**	**Waltman**
Circle	**2**	70	415
	4	123	415
	6	159	415
	8	197	415
	10	233	415
	20	412	415
Torus A	**4**	125	628
	8	232	598
Torus B	**4**	155	795
	8	222	873

Table 7: Speed of the original (Waltman) vs. the new model (Wendelien) in computing mean prices for 16 combinations of noise (0, 0.1, 0.2 and 0.5) and experimentation probability (0, 0.00001, 0.0001 and 0.001) measured in minutes.

A.4 Computing Nash Equilibria

A Nash Equilibrium is defined as the set of strategies for which, given the strategies of the remaining players, no player has an incentive to deviate (Nash, 1951). In these models, every price is a discrete price Nash Equilibrium because, given that all players play the same strategy, each player maximises its profit by continuing to play that strategy. Of more interest is the continuous Nash Equilibrium.

If we consider the circular market structure, then a firm's market share and hence profit depends only on it's own price and the prices set by its two direct neighbours. To show that the Nash Equilibrium price p_N equals 1, consider the situation where all firms set a price of 1 and only one firm deviates. If all firms set the same price, consumers will simply purchase from the firm which minimises their transportation costs. However, if a firm deviates, consumers will purchase from the firm for which the sum of the price and distance travelled is minimised. The market profit function for the deviating firm is thus equal to:

$$\Pi = p * \frac{(p_N - p + 1) + (p_N - p + 1)}{2} \tag{5}$$

We know that p_N equals 1 so this simplifies down to $\Pi = p * (2 - p)$. Maximising this expression with respect to the firm's collusion price p gives us $p = 1 = p_N$. A similar computation can be done for the Torus A market structure where the profit function is determined by a firm's direct horizontal and vertical neighbours and equals:

$$\Pi = p * \frac{(p_N - p + 1) + (p_N - p + 1) + (p_N - p + 1) + (p_N - p + 1)}{2} \tag{6}$$

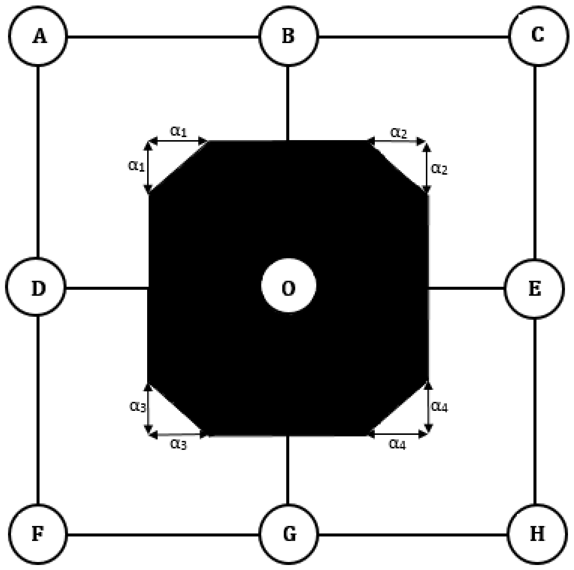


Figure 14: Illustration of the market share of a firm in the Torus B market

The Torus B market is slightly more complicated due to the effect of the indirect neighbours, namely firms A, C, F and H in Figure 14. For this reason we need to distinguish between two cases: one where the deviating firm sets a price above the Nash Equilibrium of 0.5, and one where he sets a price below 0.5. The α as indicated in the figure, is computed as follows, taking α_1 as an example:

$$\alpha_1 = \frac{p_B - p_O + 1}{2} + \frac{p_D - p_O + 1}{2} - \frac{p_A - p_O + 2}{2} \tag{7}$$

Each α is computed, and if it is positive $\alpha^2/2$ is subtracted from Firm O's market share. Assuming that all firms set the Nash Equilibrium price of 0.5, each α simplifies down to $\alpha = \frac{1}{4} - \frac{p_O}{2}$. This is only positive for $p_O < \frac{1}{2}$, hence the need to distinguish between a price increase and decrease.

If Firm O were to deviate from the Nash Equilibrium by increasing its price, its profit function would be $\Pi = p_O * (\frac{3}{2} - p_O)^2$. Maximising with respect to p_O gives an optimum price of $p_O = 0.5 = p_N$. If Firm O were to deviate by decreasing its price, its profit function would be $\Pi = p_O * ((\frac{3}{2} - p_O)^2 - 2 * (\frac{1}{4} - \frac{p_O}{2})^2)$. This too is maximised for $p_O = 0.5 = p_N$, proving that in Torus B, the Nash Equilibrium price is equal to 0.5.

B Appendix: Examples Where Firm's Profit Exceeds Average Profit

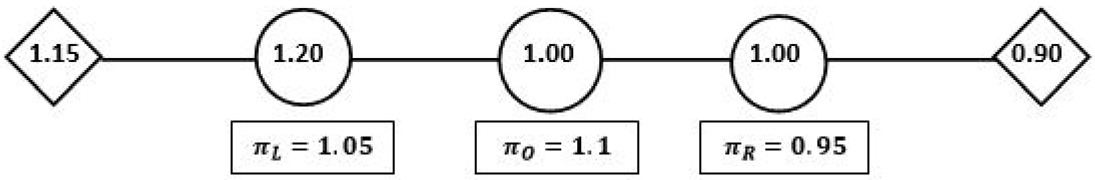


Figure 15: An example of where a firm (firm O depicted in the center of the graph) would switch to a higher strategy with an average payoff which is less that its own current payoff in a circular market with a learning neighbourhood of $\rho = 2$.

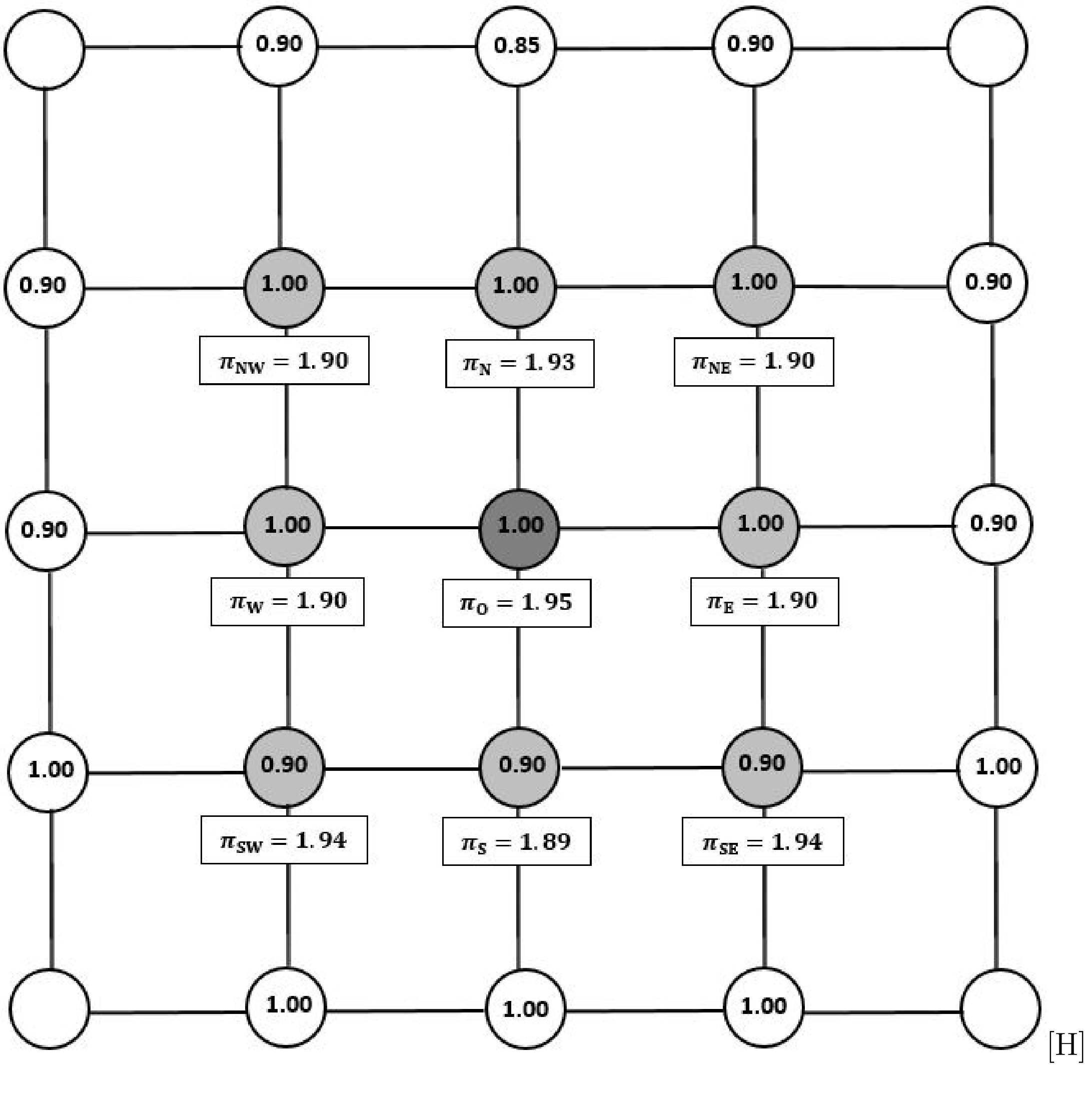


Figure 16: An example of where a firm (firm O depicted in the center of the graph) would switch to a lower strategy with an average payoff which is less that its own current payoff in a Torus A market with a learning neighbourhood of $\rho = 8$.

C Overview of Mean Prices - Imitate Best Strategy

			Imitate Best Strategy(IBS)		**IBS Adjusted**		**IBS Adjusted & Price War**	
Market	σ	ρ	$\mu = 0$	$\mu = 0.0001$	$\mu = 0$	$\mu = 0.0001$	$\mu = 0$	$\mu = 0.0001$
Circle	**0**	**2**	1.283 (0.029)	1.283 (0.023)	1.008 (0.018)	0.998 (0.000)	1.180 (0.036)	1.181 (0.028)
		4	0.901 (0.005)	0.898 (0.000)	0.901 (0.005)	0.898 (0.000)	1.009 (0.012)	1.004 (0.004)
		6	0.899 (0.008)	0.900 (0.001)	0.993 (0.007)	0.997 (0.003)	1.002 (0.009)	1.004 (0.003)
		8	0.901 (0.007)	0.900 (0.000)	1.002 (0.009)	0.998 (0.003)	1.010 (0.009)	1.005 (0.003)
		10	0.930 (0.014)	0.931 (0.013)	0.998 (0.004)	0.997 (0.001)	1.006 (0.004)	1.006 (0.003)
		20	0.981 (0.009)	0.991 (0.011)	0.993 (0.012)	0.997 (0.011)	1.007 (0.011)	1.008 (0.006)
	0.2	**2**	1.316 (0.034)	1.314 (0.032)	1.107 (0.024)	1.103 (0.022)	1.115 (0.025)	1.107 (0.024)
		4	0.928 (0.017)	0.927 (0.014)	0.900 (0.018)	0.899 (0.013)	1.003 (0.083)	0.974 (0.041)
		6	0.944 (0.014)	0.942 (0.013)	0.933 (0.016)	0.930 (0.015)	0.940 (0.019)	0.937 (0.014)
		8	0.953 (0.021)	0.952 (0.016)	0.940 (0.020)	0.942 (0.016)	0.950 (0.019)	0.947 (0.017)
		10	0.960 (0.016)	0.960 (0.015)	0.958 (0.014)	0.956 (0.013)	0.965 (0.015)	0.961 (0.012)
		20	0.979 (0.016)	0.978 (0.014)	0.985 (0.015)	0.981 (0.014)	0.995 (0.015)	0.990 (0.015)
Torus A	**0**	**4**	1.154 (0.014)	1.156 (0.010)	1.106 (0.022)	1.093 (0.011)	1.100 (0.018)	1.093 (0.011)
		8	1.036 (0.007)	1.035 (0.004)	1.001 (0.008)	1.000 (0.000)	1.002 (0.007)	1.002 (0.003)
	0.2	**4**	1.060 (0.013)	1.060 (0.011)	1.012 (0.013)	1.013 (0.011)	1.026 (0.016)	1.028 (0.011)
		8	0.996 (0.011)	0.995 (0.008)	0.988 (0.013)	0.988 (0.010)	1.005 (0.015)	1.003 (0.011)
Torus B	**0**	**4**	0.577 (0.006)	0.0579 (0.005)	0.551 (0.007)	0.0548 (0.005)	0.551 (0.007)	0.0550 (0.005)
		8	0.519 (0.004)	0.518 (0.003)	0.501 (0.004)	0.500 (0.000)	0.506 (0.003)	0.508 (0.004)
	0.1	**4**	0.534 (0.009)	0.535 (0.008)	0.513 (0.008)	0.515 (0.007)	0.523 (0.009)	0.522 (0.009)
		8	0.505 (0.008)	0.507 (0.007)	0.501 (0.008)	0.502 (0.009)	0.514 (0.008)	0.514 (0.008)

Table 8: The average mean prices after 500 simulation runs where firms act according to the learning rule Imitate Best Strategy, Imitate Best Strategy Adjusted and Imitate Best Strategy Adjusted where two firms maintain a price agreement. The standard deviations of the 500 mean prices are presented in brackets. The learning neighbourhood is indicated by the symbol ρ, noise by σ and the experimentation probability by μ.

D Overview of Mean Prices - Win Cooperate, Lose Defect

			WCLD		**WCLD Adjusted**		**WCLD Adjusted & Price War**	
Market	σ	ρ	$\mu = 0$	$\mu = 0.0001$	$\mu = 0$	$\mu = 0.0001$	$\mu = 0$	$\mu = 0.0001$
Circle	**0**	**2**	1.073 (0.01)	1.072 (0.01)	1.076 (0.009)	1.076 (0.009)	1.077 (0.009)	1.078 (0.01)
		4	1.018 (0.009)	1.018 (0.009)	1.030 (0.009)	1.029 (0.009)	1.033 (0.01)	1.031 (0.01)
		6	1.003 (0.008)	1.003 (0.007)	1.014 (0.007)	1.013 (0.007)	1.014 (0.008)	1.015 (0.008)
		8	0.993 (0.007)	0.993 (0.007)	1.000 (0.007)	1.001 (0.007)	1.000 (0.007)	1.001 (0.006)
		10	0.986 (0.006)	0.986 (0.006)	0.991 (0.007)	0.991 (0.006)	0.992 (0.007)	0.991 (0.007)
		20	0.964 (0.007)	0.964 (0.007)	0.967 (0.007)	0.966 (0.007)	0.967 (0.007)	0.967 (0.006)
	0.2	**2**	1.064 (0.005)	1.064 (0.005)	1.064 (0.005)	1.065 (0.004)	1.064 (0.004)	1.063 (0.005)
		4	1.031 (0.005)	1.003 (0.005)	1.032 (0.006)	1.033 (0.005)	1.035 (0.005)	1.035 (0.005)
		6	0.999 (0.007)	0.998 (0.009)	1.003 (0.007)	1.001 (0.007)	1.002 (0.007)	1.001 (0.008)
		8	1.001 (0.008)	1.000 (0.007)	1.004 (0.007)	1.002 (0.007)	1.004 (0.007)	1.002 (0.006)
		10	0.995 (0.006)	0.996 (0.006)	0.997 (0.006)	0.996 (0.005)	0.996 (0.006)	0.997 (0.006)
		20	0.965 (0.005)	0.964 (0.005)	0.965 (0.005)	0.965 (0.005)	0.967 (0.005)	0.966 (0.005)
Torus A	**0**	**4**	1.147 (0.013)	1.149 (0.015)	1.15 (0.011)	1.152 (0.014)	1.15 (0.012)	1.149 (0.012)
		8	1.129 (0.013)	1.128 (0.014)	1.127 (0.013)	1.129 (0.013)	1.129 (0.013)	1.127 (0.012)
	0.2	**4**	1.155 (0.011)	1.155 (0.01)	1.154 (0.011)	1.154 (0.01)	1.156 (0.01)	1.156 (0.01)
		8	1.117 (0.013)	1.119 (0.012)	1.116 (0.011)	1.118 (0.013)	1.118 (0.012)	1.117 (0.012)
Torus B	**0**	**4**	0.562 (0.006)	0.563 (0.006)	0.566 (0.006)	0.566 (0.006)	0.565 (0.006)	0.564 (0.006)
		8	0.552 (0.007)	0.553 (0.007)	0.553 (0.006)	0.552 (0.007)	0.554 (0.006)	0.553 (0.006)
	0.1	**4**	0.579 (0.004)	0.581 (0.005)	0.580 (0.005)	0.581 (0.005)	0.582 (0.005)	0.582 (0.004)
		8	0.554 (0.007)	0.554 (0.006)	0.555 (0.006)	0.555 (0.005)	0.555 (0.006)	0.555 (0.006)

Table 9: The average mean prices after 500 simulation runs where firms act according to the learning rule Win Cooperate, Lose Defect (WCLD), WCLD Adjusted and WCLD Adjusted where two firms maintain a price agreement. The standard deviations of the 500 mean prices are presented in brackets. The learning neighbourhood is indicated by the symbol ρ, noise by σ and the experimentation probability by μ.

E Code

E.1 Firms

```
import java.util.HashMap;
import java.util.Map;
import java.util.Random;
import umontreal.iro.lecuyer.rng.LFSR113;

/** This class represents the homogeneous firms in the market who compete
   by price and adjust their strategies
 *  based on the Imitate Best Strategy learning rule */
public class FirmIBS
{
   private int indexStrategy; // the firm's current strategy
   private double[] setOfPrices; // the strategy set consisting of all
      possible prices
   private double profit; // the firm's current profit
   private int numStrategies; // the total number of strategies in the
      strategy set
   private double noise; // the noise parameter
   private double expProb; // the experimentation probability
   private LFSR113 rand; // the random number generator used to generate
      integers and doubles
   private Random gaussian; // the random number generator used to
      generate draws from a Gaussian distribution
   private Map<FirmIBS, Double> directNeighbours; // consists of the
      firm's neighbours who directly impact its market share
   private Map<FirmIBS,Double> indirectNeighbours; // consists of the
      firm's neighbours who do not directly impact its market share


   /** Initialises the FirmIBS */
   public FirmIBS(int initialStrategy, int numberOfStrategies, double
      noiseLevel, double experimentProb, int[] randomSeed, double[]
      allPrices)
   {
      indexStrategy = initialStrategy;
      numStrategies = numberOfStrategies;
      noise = noiseLevel;
      expProb = experimentProb;
      LFSR113.setPackageSeed(randomSeed);
      rand = new LFSR113();
      gaussian = new Random(randomSeed[0]);
      setOfPrices = allPrices;
      indirectNeighbours = new HashMap<FirmIBS,Double>();
      directNeighbours = new HashMap<FirmIBS, Double>();
   }

   /** Calculates and returns the current profit of the firm */
```

```
public double getProfit()
{
    double marketShare = 0.0;
    for(FirmIBS neighbour : directNeighbours.keySet())
    {
        if(neighbour == null)
        {
            throw new IllegalArgumentException();
        }
        else
        {
            marketShare += (neighbour.getPrice() -
                setOfPrices[indexStrategy] + 1)/2;
        }
    }
    profit = marketShare*setOfPrices[indexStrategy];
    return profit;
}

  /** The firm determines the average profit of each strategy in its
      learning neighbourhood (with a certain level of noise added to the
      neighbours' profits)
 *  The firm then imitates the strategy earning the highest average
     profit.
 *  After copying the price, the firm may increase or decrease its price
     dependent on the draws experiment and upDown provided as input.
 *  @param experiment: random number between 0 and 1 which determines
     whether the firm will experiment
 *  @param upDown: random number between 0 and 1 which determines, if
     the firm experiments, if it will increase or decrease its price*/
public void changeStrategy(double experiment, double upDown)
{
    double[] totalProfitsPerStrategy = new double[numStrategies];
    int[] firmsPerStrategy = new int[numStrategies];
    firmsPerStrategy[indexStrategy]++;
    totalProfitsPerStrategy[indexStrategy] += (this.getProfit());
    for(FirmIBS neighbour : directNeighbours.keySet())
    {
        if(neighbour == null)
        {
            throw new IllegalArgumentException();
        }
        else
        {
            firmsPerStrategy[neighbour.getStrategy()]++;
            totalProfitsPerStrategy[neighbour.getStrategy()] +=
                (neighbour.getProfit() + noise*gaussian.nextGaussian());
        }
    }
    for(FirmIBS neighbour : indirectNeighbours.keySet())
    {
        if(neighbour == null)
        {
            throw new IllegalArgumentException();
        }
        else
        {
            firmsPerStrategy[neighbour.getStrategy()]++;
            totalProfitsPerStrategy[neighbour.getStrategy()] +=
                (neighbour.getProfit() + noise*gaussian.nextGaussian());
        }
    }
    double[] meanProfitsPerStrategy = new double[numStrategies];
    double maxProfit = Double.NEGATIVE_INFINITY;
    int newStrategy = 0;
    for(int i = 0; i < numStrategies; i++)
    {
        if(firmsPerStrategy[i] > 0)
        {
            meanProfitsPerStrategy[i] =
                totalProfitsPerStrategy[i]/firmsPerStrategy[i] +
                1E-10*rand.nextDouble();
        }
        if (meanProfitsPerStrategy[i] > maxProfit)
        {
            maxProfit = meanProfitsPerStrategy[i];
            newStrategy = i;
        }
    }
    if(experiment <= expProb)
  {
     if(upDown < 0.5 && newStrategy < (numStrategies - 1))
     {
        newStrategy++;
     }
     if(upDown >= 0.5 && newStrategy > 0)
     {
        newStrategy--;
     }
  }
 indexStrategy = newStrategy;
}


/** Returns the firm's current price */
public double getPrice()
{
    return setOfPrices[indexStrategy];
}

/** Returns the firm's current strategy ranging from 0 to numStrategies
    - 1 */
    public int getStrategy()
    {
        return indexStrategy;
    }

    /** Adds a neighbour who influences the firm's market share directly to
        the map of direct neighbours */
    public void addDirectNeighbour(FirmIBS neighbour)
    {
        if(neighbour == null || neighbour.getPrice() == 0)
        {
            throw new IllegalArgumentException();
        }
        else
        {
            directNeighbours.put(neighbour, neighbour.getPrice());
        }
    }

    /** Adds a neighbour which the firm doesn't share the market with, but
        does have information about to the map of indirect neighbours */
    public void addIndirectNeighbour(FirmIBS neighbour)
    {
        if(neighbour == null || neighbour.getPrice() == 0)
        {
            throw new IllegalArgumentException();
        }
        else
        {
            indirectNeighbours.put(neighbour, neighbour.getPrice());
        }
    }
}
```

```
/** The implementation of the Imitate Best Strategy Adjusted learning rule
    */
    public void changeStrategy(double experiment, double upDown)
    {
        double[] totalProfitsPerStrategy = new double[numStrategies];
        int[] firmsPerStrategy = new int[numStrategies];
        firmsPerStrategy[indexStrategy]++;
        double ownProfit = this.getProfit();
        totalProfitsPerStrategy[indexStrategy] += ownProfit;
        for(FirmWAdjusted neighbour : directNeighbours.keySet())
        {
            if(neighbour == null)
            {
                throw new IllegalArgumentException();
            }
            else
            {
                firmsPerStrategy[neighbour.getStrategy()]++;
                totalProfitsPerStrategy[neighbour.getStrategy()] +=
                    (neighbour.getProfit() + noise*gaussian.nextGaussian());
            }
        }
        for(FirmWAdjusted neighbour : indirectNeighbours.keySet())
        {
            if(neighbour == null)
            {
                throw new IllegalArgumentException();
            }
            else
            {
                firmsPerStrategy[neighbour.getStrategy()]++;
                totalProfitsPerStrategy[neighbour.getStrategy()] +=
                    (neighbour.getProfit() + noise*gaussian.nextGaussian());
            }
        }
        double[] meanProfitsPerStrategy = new double[numStrategies];
        double maxProfit = ownProfit; //key adjustment with original Waltman
            model

        for(int i = 0; i < numStrategies; i++)
        {
            if(firmsPerStrategy[i] > 0)
            {
                meanProfitsPerStrategy[i] =
                    totalProfitsPerStrategy[i]/firmsPerStrategy[i] +
                    1E-10*rand.nextDouble();
            }
            if (meanProfitsPerStrategy[i] > maxProfit)
            {
                maxProfit = meanProfitsPerStrategy[i];
                indexStrategy = i;
            }
        }

        if(experiment <= expProb)
        {
            if(upDown < 0.5 && indexStrategy < (numStrategies - 1))
            {
                indexStrategy++;
            }
            if(upDown >= 0.5 && indexStrategy > 0)
            {
                indexStrategy--;
            }
        }
    }
```

```
import java.util.HashMap;
import java.util.Map;
import java.util.Random;

/** This class represents the homogeneous firms in the market who compete
    by price and adjust their strategies
 *  based on the Win Cooperate, Lose Defect learning rule */
public class FirmWCLD //implements Firm
{
   private int indexStrategy; // the firm's current strategy
   private double[] setOfPrices; // the strategy set consisting of all
       possible prices
   private double profit; // the firm's current profit
   private int numStrategies; // the total number of strategies in the
       strategy set
   private double noise; // the noise parameter
    private double expProb; // the experimentation probability
    private Random gaussian; // the random number generator used to
       generate draws from a Gaussian distribution
    private Map<FirmWCLD, Integer> directNeighbours; // consists of the
       firm's neighbours who directly impact its market share
    private Map<FirmWCLD, Integer> indirectNeighbours; // consists of the
       firm's neighbours who do not directly impact its market share

    /** Initialises the FirmWCLD */
    public FirmWCLD(int initialStrategy, int numberOfStrategies, double
       noiseLevel, double experimentProb, int[] randomSeed, double[]
       allPrices)
    {
       indexStrategy = initialStrategy;
       numStrategies = numberOfStrategies;
       noise = noiseLevel;
       expProb = experimentProb;
       gaussian = new Random(randomSeed[0]);
       setOfPrices = allPrices;
       indirectNeighbours = new HashMap<FirmWCLD, Integer>();
       directNeighbours = new HashMap<FirmWCLD, Integer>();
       profit = Double.NEGATIVE_INFINITY;
    }

    /** Calculates and returns the current profit of the firm */
    public double updateProfit()
    {
      double marketShare = 0.0;
      for(FirmWCLD neighbour : directNeighbours.keySet())
      {
         if(neighbour == null)
         {
            throw new IllegalArgumentException();
         }
            else
            {
```

```
        marketShare += (neighbour.getPrice() -
            setOfPrices[indexStrategy] + 1)/2;
      }
  }
  profit = marketShare*setOfPrices[indexStrategy];
    return profit;
}

/** Returns the firm's profit when last updated. If no such profit has
    been stored yet, it computes the firm's current profit */
public double getProfit()
{
  if(profit < 0)
  {
     profit = this.updateProfit();
  }
    return profit;
}

/** The firm determines the average profit of its neighbours when they
    were last selected by the learning rule.
 * Based on the Win Cooperate, Lose Defect learning rule, if the
     average profit of its neighbours exceeds its own profit, the firm
     will decrease its price and vice versa.
 * It may then still experiment by increasing or decreasing its price.
 * @param experiment: random number between 0 and 1 which determines
     whether the firm will experiment
 * @param upDown: random number between 0 and 1 which determines, if
     the firm experiments, if it will increase or decrease its price
 */
public void changeStrategy(double experiment, double upDown)
{
  double ownProfit = this.updateProfit();
    double totalProfits = 0;
    int count = 0;
    for(FirmWCLD neighbour : directNeighbours.keySet())
    {
        if(neighbour == null)
        {
            throw new IllegalArgumentException();
        }
        else
        {
            totalProfits += (neighbour.getProfit() +
                noise*gaussian.nextGaussian());
            count++;
        }
    }
    for(FirmWCLD neighbour : indirectNeighbours.keySet())
    {
        if(neighbour == null)
        {
            throw new IllegalArgumentException();
        }
        else
        {
            totalProfits += (neighbour.getProfit() +
                noise*gaussian.nextGaussian());
            count++;
        }
    }
    double averageProfit = totalProfits/count;
    if(averageProfit > ownProfit && indexStrategy > 0)
    {
     indexStrategy--;
    }
    else if(averageProfit < ownProfit && indexStrategy < (numStrategies
        - 1))
    {
     indexStrategy++;
    }
    if(experiment <= expProb)
  {
     if(upDown < 0.5 && indexStrategy < (numStrategies - 1))
     {
        indexStrategy++;
     }
     if(upDown >= 0.5 && indexStrategy > 0)
     {
        indexStrategy--;
     }
  }
}


/** Returns the firm's current price */
public double getPrice()
{
    return setOfPrices[indexStrategy];
}

/** Returns the firm's current strategy ranging from 0 to numStrategies
    - 1 */
public int getStrategy()
{
    return indexStrategy;
}

/** Adds a neighbour who influences the firm's market share directly to
    the map of direct neighbours */
public void addDirectNeighbour(FirmWCLD neighbour, int index)
{
        if(neighbour == null)
        {
            throw new IllegalArgumentException();
        }
        else
        {
            directNeighbours.put(neighbour, index);
        }
    }

    /** Adds a neighbour which the firm doesn't share the market with, but
        does have information about to the map of indirect neighbours */
    public void addIndirectNeighbour(FirmWCLD neighbour, int index)
    {
        if(neighbour == null)
        {
            throw new IllegalArgumentException();
        }
        else
        {
            indirectNeighbours.put(neighbour, index);
        }
    }
}
```

```
/** The implementation of the Win Cooperate, Lose Defect learning rule. The
    firm determines the average current profit of its neighbours
    (getProfit() now returns current profit).
     * Based on the Win Cooperate, Lose Defect learning rule, if the
       average profit of its neighbours exceeds its own profit, the firm
       will decrease its price and vice versa.
     * It may then still experiment by increasing or decreasing its price.
     * @param experiment: random number between 0 and 1 which determines
       whether the firm will experiment
     * @param upDown: random number between 0 and 1 which determines, if
       the firm experiments, if it will increase or decrease its price
     */
    public void changeStrategy(double experiment, double upDown)
    {
      double ownProfit = this.getProfit();
        double totalProfits = 0;
        int count = 0;
        for(FirmWCLDAdj neighbour : directNeighbours.keySet())
        {
            if(neighbour == null)
            {
                throw new IllegalArgumentException();
            }
            else
            {
                totalProfits += (neighbour.getProfit() +
                    noise*gaussian.nextGaussian());
                count++;
            }
        }
        for(FirmWCLDAdj neighbour : indirectNeighbours.keySet())
        {
            if(neighbour == null)
            {
                throw new IllegalArgumentException();
            }
            else
            {
                totalProfits += (neighbour.getProfit() +
                    noise*gaussian.nextGaussian());
                count++;
            }
        }
        double averageProfit = totalProfits/count;
        if(averageProfit > ownProfit && indexStrategy > 0)
        {
       indexStrategy--;
        }
        else if(averageProfit < ownProfit && indexStrategy < (numStrategies
            - 1))
        {
       indexStrategy++;
        }
        if(experiment <= expProb)
      {
       if(upDown < 0.5 && indexStrategy < (numStrategies - 1))
       {
            indexStrategy++;
       }
       if(upDown >= 0.5 && indexStrategy > 0)
       {
            indexStrategy--;
       }
      }
    }
```

E.2 Markets

```
import umontreal.iro.lecuyer.rng.LFSR113;

/** This class represents the Circle market. The firms are initialised and
    stored after which the firms are assigned their neighbours.
 * Circle also has the function runSimulation() which applies the learning
     rule numberOfRounds times. */
public class Circle
```

```
{
   private final int numFirms; // the number of firms in the market
   private FirmIBS[] listOfFirms; // stores the firms in the market
   private final int numberOfStrategies; // the total number of strategies
       in the strategy set
   private final int learningNeighbourhood; // the size of the firms'
       learning neighbourhood
   private final int numberOfRounds; // the number of times the learning
       rule is applied
   private final int[] randomSeed; // the seed of the LFSR113 random number
       generator
   private final double noiseLevel; // the noise level which perturbs the
       firms' information with respect to their neighbours' profits
   private final double experimentationProb; // the experimentation
       probability
   private final double[] setOfPrices; // the strategy set
   private LFSR113 rand; // the random number generator

   public Circle(int dimension, int[] initialStrategies, int
       numStrategies,int neighbourhood,int numRounds,int[] seed,double
       noise, double experiment, double[] prices)
   {
      numFirms = dimension;
      listOfFirms = new FirmIBS[numFirms];
      numberOfStrategies = numStrategies;
      learningNeighbourhood = neighbourhood;
      numberOfRounds = numRounds;
      randomSeed = seed;
      noiseLevel = noise;
      experimentationProb = experiment;
      LFSR113.setPackageSeed(randomSeed);
      rand = new LFSR113();
      setOfPrices = prices;
      for(int i = 0; i < numFirms; i++)
      {
         /** Initialise firms with initial strategies */
         FirmIBS newFirm = new FirmIBS(initialStrategies[i],
             numberOfStrategies, noiseLevel, experimentationProb,
             randomSeed, setOfPrices);
         listOfFirms[i] = newFirm;
      }

      int i = 0;
      for(FirmIBS firm : listOfFirms)
      {
         for(int j = 1; j <= learningNeighbourhood; j++)
         {
            /** add neighbours to firms */
            if(j == 1)
            {
```

```
            int locationRight = ((i+j) >= numFirms) ? (i+j-numFirms) :
               (i+j);
            int locationLeft = ((i-j) < 0) ? (numFirms + (i-j)) : (i-j);
            firm.addDirectNeighbour(listOfFirms[locationRight]);
            firm.addDirectNeighbour(listOfFirms[locationLeft]);
          }
          if (j > 1)
          {
            int locationRight = ((i+j) >= numFirms) ? (i+j-numFirms) :
               (i+j);
            int locationLeft = ((i-j) < 0) ? (numFirms + (i-j)) : (i-j);
            firm.addIndirectNeighbour(listOfFirms[locationRight]);
            firm.addIndirectNeighbour(listOfFirms[locationLeft]);
          }
        }
        i++;
      }
    }


    /** Implements the learning rule numberOfRounds times */
    public void runSimulation()
    {
      for(int i = 0; i < numberOfRounds; i++)
      {
        int index = rand.nextInt(0,numFirms-1); // picks a firm at random
        FirmIBS currentFirm = listOfFirms[index];
        currentFirm.changeStrategy(rand.nextDouble(), rand.nextDouble());
           //the firm obtains the relevant information and applies the
           Imitate Best Strategy learning rule
      }
    }

    /** Returns the mean market price */
    public double getMean()
    {
      double mean = 0.0;
      for(FirmIBS firm : listOfFirms)
      {
        mean += firm.getPrice();
      }
      return mean/numFirms;
    }
}
```

```
import umontreal.iro.lecuyer.rng.LFSR113;

/** This class represents the Torus A market. The firms are initialised and
    stored after which the firms are assigned their neighbours.
 *  Torus A also has the function runSimulation() which applies the
     learning rule numberOfRounds times. */
```

```java
public class TorusA
{
   private final int numFirms; // the number of firms in the market
   private FirmIBS[] listOfFirms; // stores the firms in the market
   private final int numberOfStrategies; // the total number of strategies
       in the strategy set
   private final int learningNeighbourhood; // the size of the firms'
       learning neighbourhood
   private final int numberOfRounds; // the number of times the learning
       rule is applied
   private final int[] randomSeed; // the seed of the LFSR113 random number
       generator
   private final double noiseLevel; // the noise level which perturbs the
       firms' information with respect to their neighbours' profits
   private final double experimentationProb; // the experimentation
       probability
   private final double[] setOfPrices; // the strategy set
   private final FirmIBS[][] torusFirms; // stores the firms in the
       location corresponding to the torus
   private LFSR113 rand; // the random number generator

   public TorusA(int dimension, int[] initialStrategies, int
       numStrategies,int neighbourhood,int numRounds,int[] seed,double
       noise, double experiment, double[] allPrices)
   {
      numFirms = dimension*dimension; /** this is circle! for torus use
          dimension*dimension */
      listOfFirms = new FirmIBS[numFirms];
      numberOfStrategies = numStrategies;
      learningNeighbourhood = neighbourhood;
      numberOfRounds = numRounds;
      randomSeed = seed;
      noiseLevel = noise;
      experimentationProb = experiment;
      setOfPrices = allPrices;
      LFSR113.setPackageSeed(randomSeed);
      rand = new LFSR113();

      torusFirms= new FirmIBS[dimension][dimension];
      int count = 0;
      for(int i = 0; i < dimension; i++)
      {
         for(int j = 0; j < dimension; j++)
         {
         /** initialise firms with initial strategies */
         FirmIBS newFirm = new FirmIBS(initialStrategies[count],
             numberOfStrategies, noiseLevel, experimentationProb,
             randomSeed, setOfPrices);
         torusFirms[i][j] = newFirm;
         listOfFirms[count] = newFirm;
         count++;
```

```
        }
      }
      if(learningNeighbourhood == 4)
      {
        for(int i = 0; i < dimension; i++)
        {
          for(int j = 0; j < dimension; j++)
          {
            /** add neighbours to firms */
            FirmIBS firm = torusFirms[i][j];
            int locationUp = ((i-1) < 0) ? (dimension-1) : (i-1);
            int locationDown = ((i+1) >= dimension) ? (0) : (i+1);
            int locationLeft = ((j-1) < 0) ? (dimension-1) : (j-1);
            int locationRight = ((j+1) >= dimension) ? (0) : (j+1);
            firm.addDirectNeighbour(torusFirms[locationUp][j]);
            firm.addDirectNeighbour(torusFirms[locationDown][j]);
            firm.addDirectNeighbour(torusFirms[i][locationLeft]);
            firm.addDirectNeighbour(torusFirms[i][locationRight]);
          }
        }
      }
      else
      {
        for(int i = 0; i < dimension; i++)
        {
          for(int j = 0; j < dimension; j++)
          {
            FirmIBS firm = torusFirms[i][j];
            /** add neighbours to firms */
            int locationUp = ((i-1) < 0) ? (dimension-1) : (i-1);
            int locationDown = ((i+1) >= dimension) ? (0) : (i+1);
            int locationLeft = ((j-1) < 0) ? (dimension-1) : (j-1);
            int locationRight = ((j+1) >= dimension) ? (0) : (j+1);
            firm.addDirectNeighbour(torusFirms[locationUp][j]);
            firm.addDirectNeighbour(torusFirms[locationDown][j]);
            firm.addDirectNeighbour(torusFirms[i][locationRight]);
            firm.addDirectNeighbour(torusFirms[i][locationLeft]);
            firm.addIndirectNeighbour(torusFirms[locationUp][locationRight]);
            firm.addIndirectNeighbour(torusFirms[locationUp][locationLeft]);
            firm.addIndirectNeighbour(torusFirms[locationDown][locationRight]);
            firm.addIndirectNeighbour(torusFirms[locationDown][locationLeft]);
          }
        }
      }
    }


    /** Implements the learning rule numberOfRounds times */
    public void runSimulation()
    {
      for(int i = 0; i < numberOfRounds; i++)
```

```
        {
            int index = rand.nextInt(0, numFirms-1); // picks a firm at random
            FirmIBS currentFirm = listOfFirms[index];
            currentFirm.changeStrategy(rand.nextDouble(), rand.nextDouble());
                //adjusts the strategy and updates neighbour information
        }
    }

    /** Returns the mean market price */
    public double getMean()
    {
        double mean = 0.0;
        for(FirmIBS firm : listOfFirms)
        {
            mean += firm.getPrice();
        }
        return mean/numFirms;
    }
}
```

```
import umontreal.iro.lecuyer.rng.LFSR113;

/** This class represents the Torus B market. The firms are initialised and
    stored after which the firms are assigned their neighbours.
 *  Torus B also has the function runSimulation() which applies the
     learning rule numberOfRounds times. */
public class TorusB
{
    private final int numFirms; // the number of firms in the market
    private FirmIBS[] listOfFirms; // stores the firms in the market
    private final int numberOfStrategies; // the total number of strategies
        in the strategy set
    private final int learningNeighbourhood; // the size of the firms'
        learning neighbourhood
    private final int numberOfRounds; // the number of times the learning
        rule is applied
    private final int[] randomSeed; // the seed of the LFSR113 random number
        generator
    private final double noiseLevel; // the noise level which perturbs the
        firms' information with respect to their neighbours' profits
    private final double experimentationProb; // the experimentation
        probability
    private final double[] setOfPrices; // the strategy set
    private final FirmVarB[][] torusFirms; // stores the firms in the
        location corresponding to the torus
    private LFSR113 rand; // the random number generator

    public TorusB(int dimension, int[] initialStrategies, int
        numStrategies,int neighbourhood,int numRounds,int[] seed,double
        noise, double experiment, double[] allPrices)
    {
```

```
numFirms = dimension*dimension; /** this is circle! for torus use
    dimension*dimension */
listOfFirms = new FirmVarB[numFirms];
numberOfStrategies = numStrategies;
learningNeighbourhood = neighbourhood;
numberOfRounds = numRounds;
randomSeed = seed;
noiseLevel = noise;
experimentationProb = experiment;
setOfPrices = allPrices;
LFSR113.setPackageSeed(randomSeed);
rand = new LFSR113();

torusFirms= new FirmVarB[dimension][dimension];
int count = 0;
for(int i = 0; i < dimension; i++)
{
   for(int j = 0; j < dimension; j++)
   {
   /** initialise firms with initial strategies */
   FirmVarB newFirm = new FirmVarB(initialStrategies[count],
       numberOfStrategies, noiseLevel, experimentationProb,
       randomSeed, setOfPrices, learningNeighbourhood);
   torusFirms[i][j] = newFirm;
   listOfFirms[count] = newFirm;
   count++;
   }
}

for(int i = 0; i < dimension; i++)
{
   for(int j = 0; j < dimension; j++)
   {
      FirmVarB firm = torusFirms[i][j];
      /** add neighbours to firms */
      int locationUp = ((i-1) < 0) ? (dimension-1) : (i-1);
      int locationDown = ((i+1) >= dimension) ? (0) : (i+1);
      int locationLeft = ((j-1) < 0) ? (dimension-1) : (j-1);
      int locationRight = ((j+1) >= dimension) ? (0) : (j+1);
      firm.addDirectNeighbour(torusFirms[locationUp][j],0); //North
      firm.addDirectNeighbour(torusFirms[locationDown][j],2); //South
      firm.addDirectNeighbour(torusFirms[i][locationRight],1); //East
      firm.addDirectNeighbour(torusFirms[i][locationLeft],3); //West
      firm.addIndirectNeighbour(torusFirms[locationUp][locationRight],0);
          //NorthEast
      firm.addIndirectNeighbour(torusFirms[locationUp][locationLeft],3);
          //NorthWest
      firm.addIndirectNeighbour(torusFirms[locationDown][locationRight],1);
          //SouthEast
      firm.addIndirectNeighbour(torusFirms[locationDown][locationLeft],2);
          //SouthWest
```

```
            }
        }
    }


    /** Implements the learning rule numberOfRounds times */
    public void runSimulation()
    {
        for(int i = 0; i < numberOfRounds; i++)
        {
            int index = rand.nextInt(0, numFirms-1); // picks a firm at random
            FirmVarB currentFirm = listOfFirms[index];
            currentFirm.changeStrategy(rand.nextDouble(), rand.nextDouble());
                //adjusts the strategy and updates neighbour information
        }
    }

    /** Returns the mean market price */
    public double getMean()
    {
        double mean = 0.0;
        for(FirmVarB firm : listOfFirms)
        {
            mean += firm.getPrice();
        }
        return mean/numFirms;
    }

}
```

E.3 Run Model

```
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import umontreal.iro.lecuyer.rng.LFSR113;

/** This class provides the input for the Circle market and runs the
    simulation for all combinations of noise and experimentation */
public class RunCircle
{
    public static void main(String[] args) throws FileNotFoundException
    {
        int n = 400;
        int n_strategies = 21;
        int learning_neighbourhood = 1;
        int n_rounds = 1000000;
        int n_simulation_runs = 500;
        int[] randomSeed = new int[4];
        randomSeed[0] = 987654321;
        randomSeed[1] = 987654321;
        randomSeed[2] = 987654321;
        randomSeed[3] = 987654321;
        double[] noise_levels = {0, 0.1, 0.2, 0.5};
        double[] experimentation_probs = {0, 0.00001, 0.0001, 0.001};
        double[] prices = {0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9,
           0.95, 1.0, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45, 1.5};

        LFSR113.setPackageSeed(randomSeed);
        LFSR113 rand = new LFSR113();
        int[] initialStrategies = new int[n];

        long startTime = System.currentTimeMillis();
        PrintWriter writerMean = new PrintWriter("CircleL1.txt");
        for(double noise : noise_levels)
        {
           for(double exp : experimentation_probs)
           {
              for(int i = 0; i < n_simulation_runs; i++)
              {
                 for(int j = 0; j < n; j++)
                 {
                    initialStrategies[j] = rand.nextInt(0, n_strategies-1);
                 }
                 Circle circle = new Circle (n, initialStrategies,
                     n_strategies, learning_neighbourhood, n_rounds,
                     randomSeed, noise, exp, prices);
                 circle.runSimulation();
                 writerMean.println(circle.getMean());
              }
           }
        }
        long endTime = System.currentTimeMillis();
        long totalTime = endTime - startTime;
        writerMean.println("Speed Circle L1:" + totalTime);
        writerMean.close();
    }
}
```

References

Bergstrom, T. C., Stark, O., 1993. How altruism can prevail in an evolutionary environment. The American Economic Review 83 (2), 149–155.
URL http://www.jstor.org/stable/2117656

Eaton, B. C., Lipsey, R. G., Jan. 1975. The Principle of Minimum Differentiation Reconsidered: Some New Developments in the Theory of Spatial Competition. The Review of Economic Studies 42 (1), 27.
URL http://restud.oxfordjournals.org/lookup/doi/10.2307/2296817

Eshel, I., Samuelson, L., Shaked, A., 1998. Altruists, Egoists, and Hooligans in a Local Interaction Model. The American Economic Review 88 (1), 157–179.
URL http://www.jstor.org/stable/116823

Gamma, E., 1995. Design patterns: elements of reusable object-oriented software. Addison-Wesley professional computing series. Addison-Wesley, Reading, Mass.

Greenhut, J. G., Greenhut, M. L., Nov. 1975. Spatial Price Discrimination, Competition and Locational Effects. Economica 42 (168), 401.
URL http://www.jstor.org/stable/2553798?origin=crossref

Gross, J., Holahan, W. L., 2003. Credible Collusion in Spatially Separated Markets*. International Economic Review 44 (1), 299–312.
URL http://onlinelibrary.wiley.com/doi/10.1111/1468-2354.t01-1-00071/full

Gupta, B., Pal, D., Sarkar, J., Jun. 1997. Spatial Cournot competition and agglomeration in a model of location choice. Regional Science and Urban Economics 27 (3), 261–282.
URL http://www.sciencedirect.com/science/article/pii/S0166046297000021

Guzdial, M., Ericson, B., 2007. Introduction to computing & programming in Java: a multimedia approach. Pearson Prentice Hall.
URL http://coweb.cc.gatech.edu/mediaComp-plan/uploads/101/javaMediacomp-book--9-6-05-4chaps.pdf

Kirchkamp, O., Nov. 1999. Simultaneous evolution of learning rules and strategies. Journal of Economic Behavior & Organization 40 (3), 295–312.
URL http://www.sciencedirect.com/science/article/pii/S0167268199000694

Kirchkamp, O., Oct. 2000. Spatial evolution of automata in the prisoners dilemma. Journal of Economic Behavior & Organization 43 (2), 239–262.
URL http://www.sciencedirect.com/science/article/pii/S0167268100001189

L'Ecuyer, P., Meliani, L., Vaucher, J., 2002. Ssj: Ssj: A framework for stochastic simulation in java. In: Proceedings of the 34th Conference on Winter Simulation: Exploring New Frontiers. WSC '02. Winter Simulation Conference, pp. 234–242.
URL http://dl.acm.org/citation.cfm?id=1030453.1030488

L'Ecuyer, P., Simard, R., Aug. 2007. Testu01: A c library for empirical testing of random number generators. ACM Trans. Math. Softw. 33 (4).
URL http://doi.acm.org/10.1145/1268776.1268777

Nash, J., 1951. Non-cooperative games. Annals of Mathematics 54 (2), 286–295.

Pal, D., 1998. Does Cournot competition yield spatial agglomeration? Economics Letters 60 (1), 49–53.
URL http://www.sciencedirect.com/science/article/pii/S0165176598000743

Pinkse, J., Slade, M. E., Brett, C., 2002. Spatial price competition: a semiparametric approach. Econometrica 70 (3), 1111–1153.
URL http://onlinelibrary.wiley.com/doi/10.1111/1468-0262.00320/abstract

Thisse, J.-F., Vives, X., 1988. On The Strategic Choice of Spatial Price Policy. The American Economic Review 78 (1), 122–137.
URL http://www.jstor.org/stable/1814702

Tieman, A. F., Houba, H., van der Laan, G., 2000. On the level of cooperative behavior in a local-interaction model. Journal of economics 71 (1), 1–30.
URL http://link.springer.com/article/10.1007/BF01227494

Tieman, A. F., van der Laan, G., Houba, H., 2001. Bertrand price competition in a social environment. De Economist 149 (1), 13–31.
URL http://dx.doi.org/10.1023/A:1004103630727

Waltman, L., van Eck, N. J., Dekker, R., Kaymak, U., Dec. 2013. An Evolutionary Model of Price Competition Among Spatially Distributed Firms. Computational Economics 42 (4), 373–391.
URL http://link.springer.com/10.1007/s10614-012-9358-3