FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Endogenous Social Networks:

An agent-based model of the formation of the loyalty in markets.

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Michael Kratochvila
Matrikelnummer 0600673

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuerin: Univ.Prof. Dipl.-Ing. Dr.techn. Alexia Fürnkranz-Prskawetz
Mitwirkung: Univ.Ass. Mag.rer.soc.oec. Dr.rer.soc.oec. Bernhard Rengs

Wien, 08.09.2011

(Unterschrift Verfasser/in) (Unterschrift Betreuer/in)

Eidesstattliche Erklärung

Michael Kratochvila, 1100 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 08.09.2011 ____

Danksagung

An dieser Stelle möchte ich mich bei meiner Familie bedanken, die mir das Studium der Wirtschaftsinformatik ermöglichte.

Des Weiteren möchte ich mich auch ganz besonders bei Univ.Prof. Dipl.-Ing. Dr. Alexia Fürnkranz-Prskawetz und Univ.Ass. Mag. Dr. Bernhard Rengs für die wissenschaftliche Betreuung bedanken.

Abstract

In various markets customers behave loyal to specific sellers. The purpose of this thesis is to replicate an agent-based model that explains the emergence of loyalty between buyers and sellers, where buyers learn to become loyal and sellers learn to offer advantages to loyal buyers. Both, sellers and buyers, use reinforcement learning to adapt their behaviour towards an optimal one for them. Furthermore, different network structures as well as mechanisms for modelling endogenous interactions are described and an overview of agent-based modelling is provided. Finally, this thesis examines how information spread between buyers affects the formation of loyalty. First, the agent-based model about loyalty was implemented in NetLogo to verify its results with the findings of the original model. Afterwards, the model was extended by allowing the buyers to spread and receive information about sellers, which influenced the seller-choosing process of the buyers. Therefore buyers were endowed with 'temporal spatial' social networks, which were formed by their actual neighbours of the sellers queue. Furthermore, buyers had the possibility to learn the importance of received information to incorporate them accordingly into their sellers-choosing process. The replicated model successfully reproduced the outcomes about loyalty. The results of the extended model showed that positive information about other sellers reduced the loyalty, whereas negative information about other sellers had no effects on the emerged level of loyalty. Moreover, buyers learned to put high attention to received positive information.

Kurzfassung

In den unterschiedlichsten Märkten ist loyales Kundenverhalten gegenüber Verkäufern anzutreffen. Das erste Ziel dieser Diplomarbeit ist es ein agenten-basiertes Modell zu replizieren, welches die Entstehung der Loyalität von Käufern gegenüber Verkäufern erklärt. Mit Hilfe von Reinforcement Learning lernen einerseits sich Käufer loyal zu verhalten und andererseits Verkäufer loyale Kunden bevorzugt zu behandeln. Des Weiteren werden unterschiedlichen Netzwerkstrukturen, sowie Mechanismen zur Modellierung endogener Interaktionen beschrieben und eine Übersicht über agenten-basierte Modellierung gebeben. Als zweites Ziel dieser Arbeit wird anschließend untersucht, wie sich Informationsaustausch zwischen den Käufern auf die Entstehung von Loyalität auswirkt. Zuerst wurde das agenten-basierte Modell über Loyalität in NetLogo implementiert, um die Ergebnisse mit jenen des originalen Modells zu verifizieren. Danach wurde das Modell dahingehend erweitert, dass Käufer nun die Möglichkeit hatten, Information über andere Verkäufer zu verbreiten. Diese Informationen beeinflussen den Prozess der Verkäufer-Auswahl der Käufer. Für die Informationsverbreitung wurden ‚temporäre räumliche' soziale Netzwerke eingeführt, welche von den Nachbarn des Käufers innerhalb einer Verkäufer-Warteschlange gebildet werden. Außerdem haben Käufer die Möglichkeit die Wichtigkeit der erhaltenen Informationen zu lernen und entsprechend in ihrem Prozess der Verkäufer-Auswahl zu berücksichtigen. Die Ergebnisse des originalen Modells bezüglich Loyalität konnten erfolgreich repliziert werden. Des Weiteren zeigen die Ergebnisse der Modellerweiterung, dass positive Informationen über andere Verkäufer sich negativ auf die Kundenloyalität auswirken, während negative Informationen keinen Einfluss auf diese haben. Zudem lernten Käufer, dass erhaltene positive Informationen sehr wichtig für den Prozess der Verkäufer-Auswahl sind.

Contents

Overview

This thesis replicates an agent-based model about loyalty and price dispersion of Kirman and Vriend (2001) and analyses the effects on the emergence of loyalty by introducing information spread as an extension of the original model.

In chapter 1 I give an overview of the different networks structures, explain the concept of simulation in social science and elucidate the main features of agent-based modelling. Furthermore, I review frequently used simulation tools by social scientists and point out the importance of model replication. At the end of chapter 1 I summarize models with endogenous interactions to explain mechanisms that are used to model endogenous interactions, since the key issue of the replicated model of this thesis is about loyalty that emerges as a result of endogenous interactions.

In the second chapter I explain the model of Kirman and Vriend (2001) and describe the implementation of my model replication. Moreover, I give an overview of the concept of reinforcement learning, since it is used by the agents in the model. At the end of this chapter I present the results of the replicated model.

In chapter 3 I give an overview of current agent-based models about loyalty, diffusion processes and I mention causes for the formation of rumours to incorporate these concepts afterwards in the extension of the model from Kirman and Vriend (2001). Then I explain the extension of the original model where I introduce information spread which affects the seller choosing process of the buyers. Finally, I present the simulation results of the extended model.

1. Dynamic Networks & Agent-based Modelling

1.1 Network Structures

A network consists of nodes (vertices) and connections between them, called links or edges and can be represented as a graph G = (V, E). V represents the set of nodes and E the set of edges. The node is referred by its index i in the set V. The set E contains pairs of nodes (i, j) of V (González, 2006).

A network is undirected, if (i,j) ϵ G ↔ (j,i) ϵ G. It is called simple, if no edge connects a node itself and only one edge connects any linked pair. A network is called connected if any node can be reached by any other node. Edges are unweighted, if all edges have the same intrinsic value (e.g. length) (Tesfatsion and Kenneth, 2006).

There exists various types of networks, each type of them has special characteristics and a specific structure.

1.1.1 Types of Networks

The following section about network – types together with its figures is mainly based on the book from Tesfatsion and Kenneth (2006, chap. 20).

Complete

A complete network is characterized by having a link from each node to each other node:

$$G = (V, E), \forall i \neq j : (i, j) \in E$$

There exit only few examples of this type of network in nature (e.g. telephone system). Because of its redundantly wired structure, reliability concerns (the probability that a piece of information send over the network, will reach its destination) often lead to this kind of network.

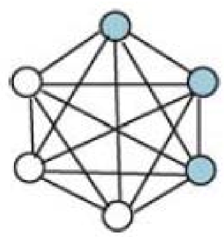


Figure 1: Complete Network.

Star

There exists one (randomly) selected node that is connected to each other node:

$$G = (V, E), \exists s \forall j, s \neq j : (s, j) \in E$$

Network decay (information degrades when travelling over more links) often leads to this efficient and minimal network structure.

It is an often used network topology in computer networks: The central node represent a hub or switch and the surrounding nodes the clients that are connected to the hub. Another application of this network is a mainframe system where the middle node represents the host and the other nodes are the terminals.

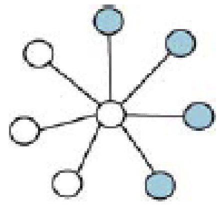


Figure 2: Star Network.

Ring

The nodes are organized in form of a ring. Each node i is connected with k neighbours. k/2 are located on the left side and k/2 to its right side.

$$G = (V, E), (i, j) \in E \text{ if } \begin{cases} j = \{(i - m) + n\} \bmod n \\ j = \{i + m\} \bmod n \end{cases}, \quad m \in \left\{1,2,\ldots,\frac{k}{2}\right\}, \text{ k is an even number.}$$

Every node of the network is only linked to a fraction of the whole node population.

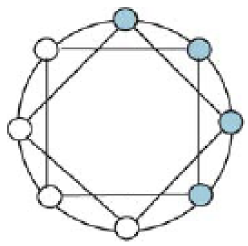


Figure 3: Ring Network.

One application is Token Ring: It is a computer network where each node is only connected to 2 neighbours. A token is passed through the network and only the node which possesses the token is allowed to send data to other nodes.

Grid

It has the structure of a chessboard. In each square a node is placed and connected with its surrounding neighbours nodes on the chessboard. The nodes that are located on the edges are additionally connected to the nodes which are located on the opposite (antipode) edges of the chessboard.

Sample applications are the layout of city blocks or rooms on the floor of a building.

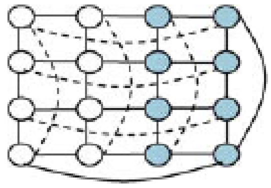


Figure 4: Grid Network.

Tree

In this kind of network, each node branches off to b other nodes. A field of application is the representation of hierarchical social systems.

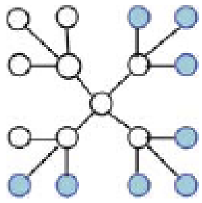


Figure 5: Tree Network.

Small-world network

The main characteristics of a small-world network is a high clustering (a high tendency of nodes to clump together[1]) and short path-length (average number of edges needed to be traversed to reach from a node any other node).

One way to construct these networks is to start with a k-ring network. Then a rewiring of the edges occurs with a probability p: A randomly selected edge (i,j) gets removed from the node j and rewired to a random selected other node b. So a new edge (i,b) is created. For the probability of p=0 the k-ring network stays unchanged. If p=1 every nodes gets randomly rewired. But only for small probabilities, small-world networks begin to emerge.

Two examples of applications are the distribution of co-authors in journals and electric power grids.

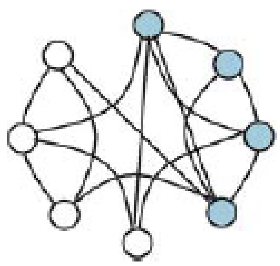


Figure 6: Small-world Network.

[1] *When two nodes i and j are connected to node k, then in a cluster node i and node j are also connected together.*

Figure 6 was originally a k-ring network with k = 4. Then randomly rewiring of the edges occurred. As a result of the randomly rewiring the edges are rewired to randomly selected other nodes (in this case to nodes that are relatively “far” away). So suddenly some nodes are able to reach other far-away nodes directly.

Power network

Each node in a network has a certain degree (number of edges to other nodes). Therefore a degree distribution is the probability distribution of these degrees over the whole network. In other words this means the degree distribution P(k) gives the probability that a node of the network has k edges. (Réka and Barabási, 2002)

According to (Tesfatsion and Kenneth, 2006) the degree distribution in power networks follows a power law: $P(k) \sim k^{-\gamma}$

Only a few nodes have a high degree (also called “hubs”) and many nodes have only a very low degree.

These networks are created by starting with a simple network and adding each time a new node that creates k edges to other already existing nodes according to a preferential weighing: $P((i,j) \in G \mid i) = \frac{j_d}{n_d}$. G represents the network, (i,j) is one of the k edges that are created by the new added node i, j_d is the number of edges reaching node j and n_d is the total number of edges in the network.

An example for this type of network is the world wide web, making it very resistant against random attacks (randomly selecting nodes that get disabled / attacked) and vulnerable to direct attacks (explicitly choosing the node to be attacked, e.g. a hub which would decrease the performance of the web dramatically).

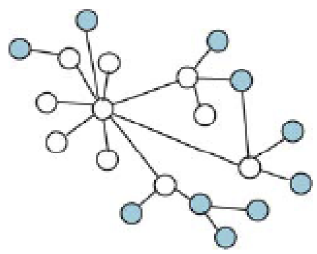


Figure 7: Power Network.

Figure 7 represents a power network. The nodes with many connections are hubs. It can be seen that only few hubs exist (3 big hubs with more then 2 connections) and many single connected nodes (13 nodes).

Summing up the above mentioned network types it can be seen that there exist two main categories of networks:

The first category are *artificial networks*. These kind of networks are created man-made to achieve a certain explicit goal like building redundancies in computer networks (complete network) or to implement a clear distribution of competencies (trees) being able to react highly efficient as needed in sectors like defence or emergency management. This category includes the complete, star, ring, grid and tree network.

The second category of networks are *naturally grown networks*. Small-world and power networks belong to this network category and are summarized under the term 'scale – free networks'. For example social networks and the World Wide Web belong to this kind of networks.

Social Networks

In contrast to the above mentioned networks types, a social network is one instance of such an abstract network. Wasserman and Faust (1994, p. 20) define a social network as a network that

> "*…consists of a finite set or sets of actors and the relation or relations defined on them. The presence of relational information is a critical and defining feature of a social network.*"

According to Knoke and Yang (2008), actors may be single persons, groups, organisations or states. Relations defined on them influence their decisions, actions, perceptions and beliefs. A relation occurs always as a joint property between two actors and exists as long as both actors maintain this relation. There has to be made a distinction between directed and non-directed relations. A directed relation consists of a sender (e.g. teacher) which initiates the relation and a receiver (e.g. student). In a non-directed relation, each sender is also a receiver, meaning mutuality occurs (e.g. a conversation, discussion).

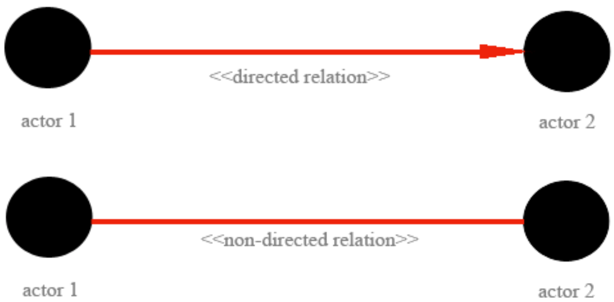


Figure 8: Directed and non-directed relation.

The first picture of figure 8 shows a directed relation where actor 1 is the initiator, actor 2 is the receiver and the red arrow represents the directed relation (e.g. actor 1 <<trusts>> actor 2 or actor 1 <<advices>> actor 2) between them. Similarly the second picture shows a non-directed relation with the two differences that now actor 1 and actor 2 are both initiators and receivers and the relation has no arrow to represent the mutuality between the two actors (e.g. for the non-directed relations like <<work together>> or <<fight with>>).

The relations are not just present or inexistent between actors, moreover they also could be valued to represent their strenght or intensity. Such valueted relations could be for example the dollar amount of trade between nations or the frequency of interaction between people (Wasserman and Faust, 1994).

A social network consists of the same type (e.g. only <<work together>> relations) of relations among the actors. So if there are multiple relation types, even among the same subset of actors, each relation type represents a social network of its own (Knoke and Yang, 2008).

A method that measures, represents, explains why these relations occur and even their consequences is called social network analysis (SNA). SNA is based on the assumptions that the relations between the actors can describe their behaviour better than their attributes, like age or gender, do. It is also assumed that these relations affect the actor's perceptions, beliefs and actions. For example direct contacts between actors give them the possibility to influence/being influenced stronger the other one/by the other one. Relations also could support and constrain rumour, gossip or the diffusion of knowledge through the whole network. The third assumption is that a social network is a dynamic construct. It underlies dynamic processes because for example actors influence / are influenced by others and have the possibility to learn, which changes the pattern of interaction over time. Therefore it can be seen as a dynamic network, having their edges (relations) continuously altering (Knoke and Yang, 2008).

It is important to point out that on the one hand the actors within the social network are influenced by the network structure and on the other hand they influence the structure of the network by their actions, beliefs and perceptions.

1.2 Agent-based Modelling

In the following, I describe the logic of simulation, the main characteristics of agent-based modelling and I give an overview of some frequently used simulation tools by social scientists.

1.2.1 Simulation in Social Science

Computer simulations have a wide field of applications. They are used in various branches of science like social sciences, natural sciences or engineering sciences. In social sciences simulations help us to get important insights of complex social systems.

A simulation has inputs, which are the attributes of the model to make it suitable for a desired social setting, and outputs which are the behaviours of the model through time (Gilbert and Troitzsch, 2005).

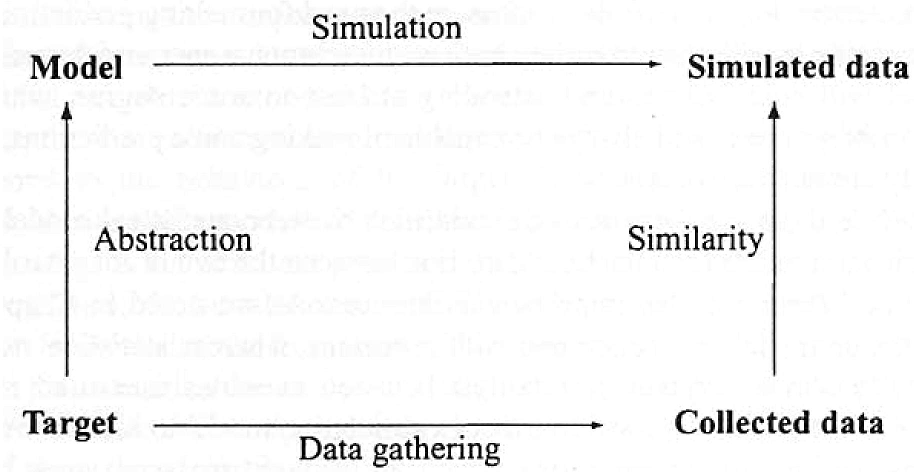


Figure 9: The logic of simulation (Gilbert and Troitzsch, 2005).

Figure 9 represents the logic of simulation: First a model is developed in form of a computer program which is an abstraction of the presumed social processes in the target (the real world phenomenon). Once this is done the simulation is executed (run) and its behaviour is recorded. This recorded behaviour is called ‘Simulated data’ and once it is generated, it can be compared with the collected data from the target system to check whether the outcomes of the initially presumed social processes are similar to the real social processes of the target (Gilbert and Troitzsch, 2005). The target represents a social phenomenon like unemployment or loyalty.

So the purpose of a simulation is to put accuracy either in the understanding of a social phenomena (called explanatory model) or on the predictability. In fact, independently from the desired emphasis of the modeller, each simulation model is both an explanatory model (to some degree) and a model that is usable (to a certain degree) for predictions (Gilbert and Troitzsch, 2005).

1.2.2 Introduction to Agent-based Modelling

Agent-based modelling (ABM) is a computational method. The main components are agents. Agents interact within a given environment. The environment is a virtual space where the interaction takes place. It could be neutral (has no influence to the agents) or has influence on the behaviour of the agents (e.g. as geographical space, knowledge space or linkage of agents to build a social network).

Agents are computer programs (or parts of a program) and could represent social actors like firms and individuals or bodies like states. They react to their environment and are able to interact with other agents by exchanging information messages. These messages can be direct by an information flow (e.g. spoken dialogue) or indirect by perceiving the actions of other agents (Gilbert, 2007).

The main characteristics of agents are autonomy, the ability to learn, react, communicate, cooperate and to perceive their environment, the degree of rationality and mobility. This characteristics can exist in various combinations and degrees, representing the different classes of agents and therefore heterogeneity among them (Gilbert, 2007).

ABM is used for analysing the individual behaviour that leads to aggregated behaviour (e.g. emergent phenomena) and to study how changes in the system-level influences the individual behaviour of the agents (Gilbert and Terna, 2000).

With ABM it is possible to capture emergent phenomena from the bottom up which result from the interactions of the individual agents. Another advantage is to be able to observe how small changes in the configuration can lead to unexpected outcomes. ABM is flexible because it is easy to add more agents to the model, to tune the complexity of the agents and to change the levels of aggregations of agents (subgroups of agents) (Bonabeau, 2002). Due to the fact that ABM is a computational method the modeller is forced to be precise and exact by specifying a computer program to be able to run it instead of using a natural language (Gilbert, 2007).

As mentioned above agents are able to interact with other agents with the consequence that networks, underlying their interaction, can emerge or are already exogenously given by the modeller. Therefore special notice on networks should be taken. (see chapter 'Network Structures')

1.2.3 Simulation Tools

Due to the fact that ABM had become more and more popular in many different disciplines during the last years, also a lot of different toolkits and frameworks had been developed to create and being able to run agent-based models. Each of them has its strengths and weaknesses. Nikolai and Madey (2009) examine the available toolkits by characterizing them according to the programming language, the required operating system, the primary domain for which the toolkit is intended, the available support for the users of these toolkits and the type of licence (free available, proprietary, …). They categorized 53 toolkits where most of them are freely available (76%).

I will now focus on some frequently used toolkits / frameworks by social scientists. In fact there exists a huge variety of toolkits and frameworks. The choice which toolkit is the best one for a certain project depends not only on its characteristics, moreover it depends on the own past experiences in using them and the publicity of the toolkit (Nikolai and Madey, 2009). And of course it is important that there exists an active, perhaps big, community to gain specific and fast support for particular questions when having an ABM project.

Some common used toolkits are SWARM, MASON, Repast and NetLogo. SWARM and Mason are general purpose toolkits and Repast was designed for social sciences. Only NetLogo was designed for educational purposes (Nikolai and Madey, 2009). All of them are available for free and, except NetLogo whose source code is not released, are published under an open source license.

SWARM[2] was developed at the Santa Fe Institute in 1999 and is available for free. It consists of a set of code libraries to create agent-based models and is available on many operating system platforms (Windows, Linux and MacOS X). The used programming languages are Objective-C and Java.

MASON [3]stands for 'Multi-Agent Simulator Of Neighbourhoods (or Networks)' and was developed from the George Mason University. The main focus of this Java library is to have a fast execution speed and to be very small. Every model created with MASON is completely decoupled from its visualizations. So each model can be attached to an optional visualization toolkit. A simulation can also be serialized to disk (in MASON a checkpoint is created) and perhaps moved to a different workstation / operating system where it can be continued to run.

There exist three implementations of Repast[4]. One for Java (called Repast J), one for .NET (Repast .NET) and one for Python Scripting (Repast Py). Apart from these three implementations of Repast, also a java-based modelling system, called Repast Symphony, for the most common operating systems exists. It supports Java, ReLogo and Groovy as programming languages (which also can be interleaved) and the import of NetLogo models. Repast was originally developed at the University of Chicago and stands for 'Recursive Porous Agent Simulation Toolkit'.

[2] http://www.swarm.org
[3] http://cs.gmu.edu/~eclab/projects/mason
[4] http://repast.sourceforge.net

NetLogo[5] was designed at the Tufts University, moved later to Northwestern University and has a Logo-like syntax. It is implemented in Java and its programming language is a procedural language meaning there exists no object-oriented programming possibility. NetLogo is not just a programming library, moreover it is a whole modelling environment that provides a programming interface with code highlighting and an interface for visualizations. In contrast to MASON and Co it also supports the visually creation of System Dynamics models.

NetLogo is used as simulation tool in this master thesis. Therefore the characteristics of this toolkit are described more in detail in table 1:

positive aspects	negative aspects
easy to learn, intuitive syntax	not object-oriented
fast developing (powerful commands, easy creation of diagrams)	bad IDE (only syntax checking, syntax highlighting)
many existing sample models	slow execution speed → interpreted language
platform independent (Java)	one simulation couldn't take advantage of multiple processors
building agent-based models, System Dynamics models	

Table 1: Characteristics of NetLogo.

1.2.4 Model Replication

Although there exist many ABM models about endogenous networks in the literature, not every one provides a description in detail about its concrete model architecture. So in many cases it becomes hard to replicate a given model.

But model replication is an important part in science: Replication needs to be done to ensure that the published description of the model is detailed enough and independent on any local conditions from the scientist who originally performed the experiment (Wilensky and Rand, 2007).

Only when someone is able to reproduce a model from scratch, the derived results of a simulation are reliable. This confirmation step ensures that results are not just a consequence of programming errors. It also eliminates misinterpretations of the output, errors in analysing or reporting the results (Axelrod, 2003).

Wilensky and Rand (2007) recommend to post certain important information when doing model replication. This should ensure that the reader is able to get an overview of the replication approach:

[5] http://ccl.northwestern.edu/netlogo

Details about the model replication	
Categories of Replication Standards	[Numerical Identity, Relational Alignment, Distributional Equivalence]
Focal Measures	measures used to meet the goal
Level of Communication	[none, rich discussion, personal meetings, brief email contact]
Familiarity with Language / Toolkit	[none, surface understanding, have built other models in this language]
Examination of Source Code	[none, studied in depth, referred to for particular questions]
Exposure to Original Implemented Model	[none, run, re-ran original experiments, ran experiments other than original ones]
Exploration of Parameter Space	[only examined results from original paper, examined other areas of the parameter space]

Table 2: Important information to be included when doing model replication.

The replication standard is the criteria for valuing the output from the replicated model as a successful or an unsuccessful replication. The options are: 'numerical identity' where the exact same numerical results are needed to be reproduced, 'relational alignment' means that the results of the original and replicated model show similar relationships between the input and output variables and 'distributional equivalence' where the produced results are statistically similar to the ones of the original model.

The focal measures are these kind of measures that are compared (as specified under 'Categories of Replication Standards'; for example in form of relational alignment) with the measures from the original model. Whereas the level of communication shows to what extend communication between the model replicator and the model developer occurred. This can give a hint if the published information about the original model is detailed enough to preserve the results for the future.

The familiarity with the language / toolkit should indicate how experienced the replicator is with the language / toolkit in which the original model was implemented. A higher familiarity of the language also helps to understand the original model better.

The examination of the source code can eliminate differences between the replication and the original model when the written description is not sufficient enough. Wilensky and Rand (2007) note that if comparison between the replicated code and the original code occurs in an too early phase this may result in groupthink and thus the replication becomes too depended on the original model developer.

The exposure to the original implemented model specifies if the replicator has executed the original implementation (by executing the original source code) of the model to re-run the original experiments to get a certain feeling about how the simulation looks like or perhaps to explore the parameter space by running experiments other than the original ones.

The last mentioned point in table 2 is the exploration of the parameter space. There the replicator notes whether he has validated only the described results of parameter space of the original published model or even other areas of the parameter space.

1.3 Endogenous Networks

According to Tesfatsion and Kenneth (2006, chap. 21), endogenous networks in agent-based computational economics (ACE; ABM applied to an economic domain) models are reflected in endogenously determined relationships. There agents do not just play a specific game, they also decide with whom they want to play the game.

Social interactions in reality are characterized by being endogenous. And so it is important to turn the attention to the study of agent-based models with endogenous interactions to analyse the way how connections/relationships are formed (e.g. by being a neighbour or sending some communication signals) and the way how connections are evaluated and established (e.g. by taking into account some learning process) by the single agents.

Therefore it is fundamental to know the different already existing mechanism that are used to model endogenous interactions:

- residential pattern
- resource gradient
- predictors
- advertising / patronage
- threshold / expected payoff
- arbitrary tags
- trust
- expected payoff / familiarity
- past success rate
- directed random search

On the following pages I will shortly explain these different kinds of mechanisms.

In all these mechanisms, the agents decide if they want to establish, maintain or close a connection with some other agent(s). These decisions are normally based on the perceived success of their interactions.

It is important to know that random interactions where agents are interacting only randomly, and local interactions, for example where agents interact with their nearest neighbours, are not endogenous! These interactions are determined by an exogenous process like exogenously determining the locations of the agents.

The following section, which gives an overview of the different mechanisms for modelling endogenous interactions and their applications in specific models, is based on Tesfatsion and Kenneth (2006, chap. 21):

1.3.1 Residential pattern

The segregation model of Schelling (1971) uses the residential pattern: Two different kind of agents are placed randomly on a lattice structure (like a chessboard). So each agent has a neighbourhood (Moore-neighbourhood where 8 neighbours surrounding one agent; see figure 10) and faces a certain ratio of unwanted agents (that are agents of the other type).

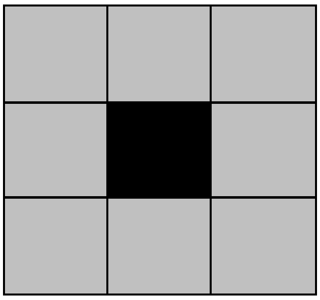


Figure 10: Moore neighbourhood.

Furthermore every agent is endowed with a specific desired ratio of unwanted agents. If this ratio is exceeded, by having to many neighbours of the different agent-type in the neighbourhood (= unwanted residential pattern), the agent becomes unsatisfied and wants to change his residence.

At each iteration of the model, all unsatisfied agents move to the nearest free satisfactory position. This process continues until all agents are satisfied and the usual outcome of this process is a segregated state.

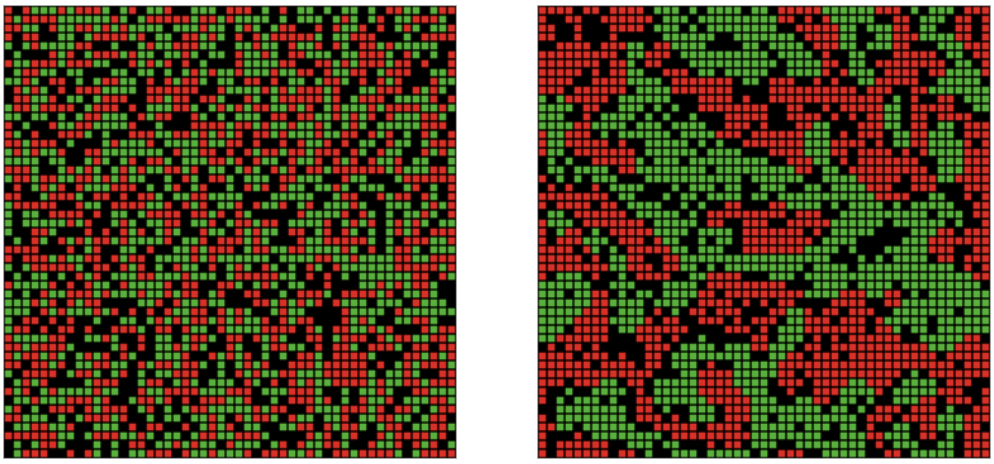


Figure 11*:* Segregation[6].

The left picture of figure 11 shows the initial state where 2000 agents are placed randomly on the chessboard. Whereas the right picture shows the emerged segregation state at the end of the simulation. The desired ratio of unwanted agents was set to 50% and the two agent-types are represented by the green and red colours.

An *endogenous interaction* occurs due to two different externalities that are created when an unsatisfied agent changes his residence:

1. If an agent leaves his old "home", this has an impact on the ratio of his old neighbours.
2. Likewise the move to a new position has an impact on the ratio of the new neighbours.

1.3.2 Resource gradient

Epstein and Axtell (1996) use the resource gradient in the so-called 'sugarscape', which is a space with a lattice structure that is formed to a torus. At each cell sugar can grow at a specific rate. The amount of sugar an agent needs to survive in one period, is specified as a given metabolic rate (=consumption rate of the agent). Each agent also has a maximum age, a vision to find free sites, a neighbourhood consisting of max. 4 neighbours in form of the von Neumann neighbourhood and optional a utility function.

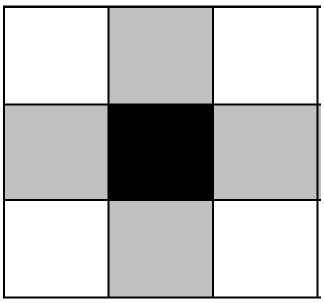


Figure 12: von Neumann neighbourhood.

[6] http://ccl.northwestern.edu/netlogo/models/Segregation

Figure 12 shows the von Neumann neighbourhood. The black field represents an agent and the grey fields are the places the agent is able to see. Another often used neighbourhood is the Moore neighbourhood as mentioned in the residential pattern model.

At the beginning sugar grows up to an upper limit at each cell. Then each agent determines the best site he can see (according to his vision), moves to it and recognizes his new neighbours there. After doing so, agents start to collect the sugar and therefore increasing their wealth. The metabolic rate decreases their wealth and agents with a negative wealth or a reached maximum age die. In another variant of the model, the agents are also able to trade sugar for spice and vice versa with their neighbours.

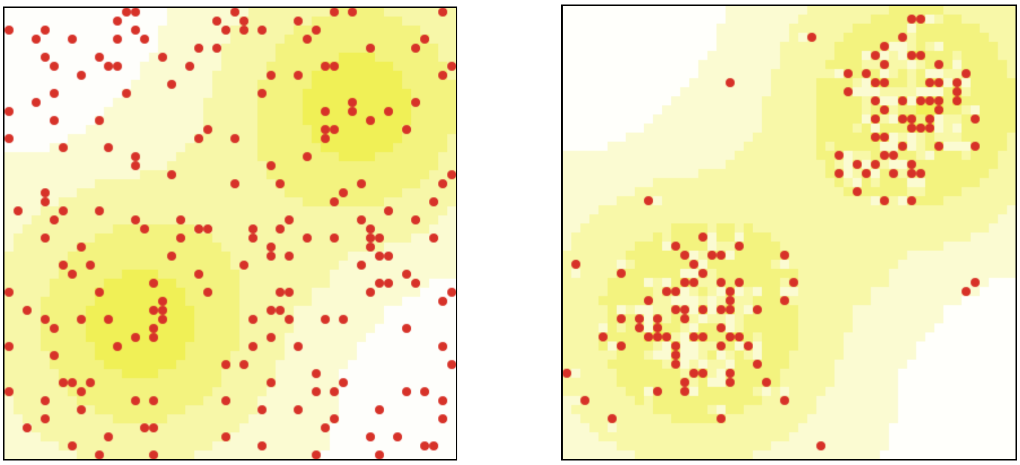


Figure 13: Agents on the sugarscape[7].

In figure 13 the left picture shows the agents on the sugarscape landscape at the beginning of the simulation and the picture to the right shows the agents after 30 time-steps. The different graduations of the yellow colour indicate the amount of sugar available at the specific positions. White areas are without any sugar. The red points represent the agents. It can be seen that the agents move to the richest endowed locations to survive and increase their welfare.

Endogenous (indirect) interactions occur due to the fact that all interactions depend on the choices made by the agents where to move. And these location choices are controlled by the availability of resources (resource gradients) on the ‘sugarscape’. But when agents move through the space to collect sugar, they also manipulate the pattern of resource availability in the landscape.

[7] http://ccl.northwestern.edu/netlogo/models/Sugarscape2ConstantGrowback

1.3.3 Predictors

Arthur (1994) uses predictors in the 'EL Farol' bar problem: Agents like to spend time together in the bar if there are fewer than 60 % of them present in the bar. If more than 60 % are present, an agent does not want to go to the bar and instead stays at home.

Each agent has a specific set of predictors (e.g. the average number of attendances of the last month, the same number of attendances as last week, …) that determine the expected number of agents in the bar, based on the attendance figures of the last weeks. The agents also trace the accuracy of each predictor by comparing the made prediction with the real actual attendance figure.

At the beginning each agent is initialised with a specific set of predictors. In each period, each agent selects the predictor with the highest accuracy and uses the made prediction of this predictor to decide if he will stay at home or will go to the bar. At the end of the period, agents update the accuracy of their predictors by comparing the realized attendance figure with the made predictions.

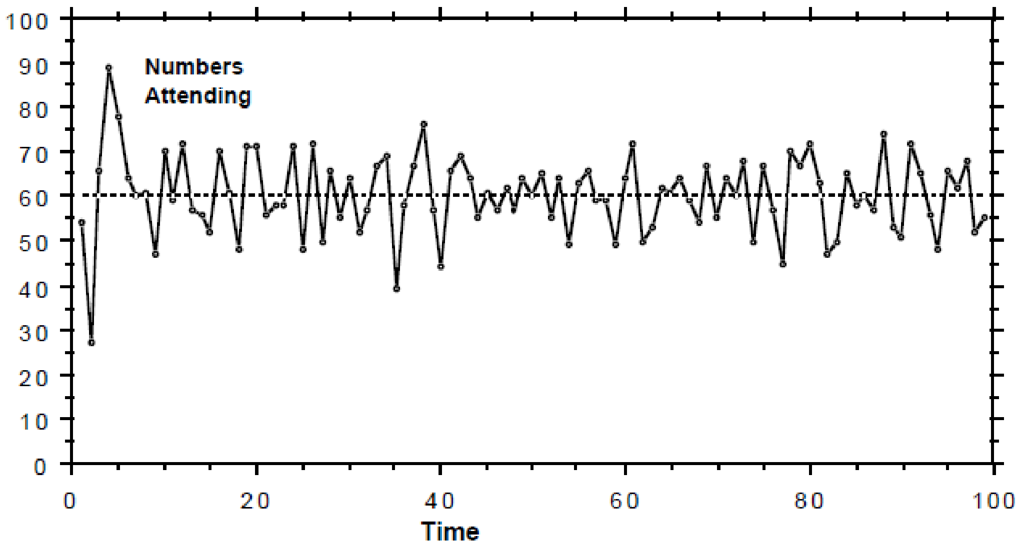


Figure 14: Attendance figures of the last 100 weeks (Arthur, 1994).

Figure 14 shows the number of persons attending the bar for the last 100 weeks. It can be seen that this number fluctuates around the 60% level.

Endogenous interactions exist because the single interaction decision of an agent (to go to the bar or to stay at home) depends on the past pattern of interactions of all agents. These single interaction decisions lead to an evolution of the pattern of interactions itself by becoming a part of the pattern of interactions on which future interaction will rely on.

1.3.4 Advertising / Patronage

Vriend (1995) used advertising and patronage in his model to implement endogenous interactions: In his model firms and customers exist. Each firm produces goods in advance without knowing the demand in the current period. They attract customers by sending information signals randomly to the customer population. The production and the information signals are costly for them.

Customers buy products by shopping around randomly, being loyal to the previous firm (=patronage) or following one received information signal.

Each firm has a set of alternative rules: A rule specifies a specific production and advertising level. The fitness of a rule depends on the received payoff by using this rule. Rules with a higher fitness are more likely to be selected by the firm. This behaviour is called reinforcement learning. After 50 periods a genetic algorithm (elimination and reproduction with mutation) is applied to the rules based on the fitness of these rules.

A genetic algorithm is based upon the biological evolution: It has the ability to adapt (find the best solution) to a given or even to a changing environment. At the beginning a start-population has to be created (e.g. the initial rules from a firm). The first step of these kind of algorithms is to calculate with a given fitness-function the fitness for each possible individual / solution (in the case of the firm the fitness of each rule would be calculated). Second, selection occurs by selecting the fittest individuals. Next, crossover starts by building couples out of the selected individuals and crossing them. Each individual can be represented as a binary string and new individuals are created by combining the sub strings of their parents. The last step is called mutation and is done by inverting single bits from the new individuals at specific rates (Gerdes et al., 2004).

Each customer has 15 condition based rules („if <<condition>> then <<action>>“ rules). The condition takes into account the shopping experience (being satisfied, did not find any products, ...) of the previous day and the information state (received any information signal or not). Possible actions are patronization, visiting the corresponding firm of a received information signal or choosing a firm at random. The fitness of a rule depends on the generated payoff and the higher the fitness of a rule the higher is the possibility of selecting this rule again (=reinforcement learning).

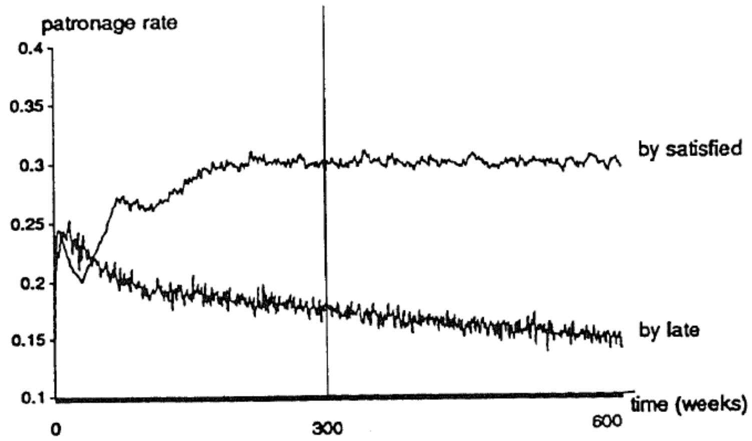


Figure 15: Patronage rate of informed customers (Vriend, 1995).

In figure 15 the different developed patronage levels of informed customers (customers which received an information signal) can be seen. Due to the fact that rules containing a "did not find any products" in their condition-block and a "patronage"-action in their action-block generate a lower payoff than other rules, their fitness decreases and as a consequence also their reapplication.

Endogenous interactions occur due to the following fact: On the one hand firms decide the desired number of interactions by sending advertising signals and selecting the output level. The output level determines how many of these desired interactions are successful (by being able to satisfy the demand of the attracted potential customer). These decisions (production level and number of advertising signals) are influenced by their profitability. But the profitability depends on the decisions (condition based rules) made by the customers. The customers decisions depend on the success of the previous period and whether having received an advertising signal or not. On the other hand these two variables (success and advertising signal received) are affected by the made decisions by the firms and other customers.

1.3.5 Threshold / Expected Payoff

Ashlock et al. (1996) uses a two person iterated Prisoner's Dilemma game with the following payoffs: (cooperation, cooperation)=(3,3); (defection, defection)=(1,1), (defection, cooperation)=(5,0) and (cooperation, defection)=(0,5).

Each agent is represented by a binary string, consisting of two parts. The first part specifies the dynamic game strategy and the second part represents with whom the agent wants to play the game (and therefore the endogenous interactions of the agent). The model consists of 2000 generations, each generation has 150 rounds where in each round (some of) the agents play a Prisoner's Dilemma game. In each round each agent specifies one opponent with whom he wants to play the PD game by choosing the agent with the highest expected payoff. An agent keeps in mind the weighted (by putting more importance on the last received payoffs) average received payoff of each other agent and accepts all offers that are above his personal payoff threshold. Then each accepted pair of agents plays a PD game. In another step all agents update their expected payoff from their opponent and take part in a genetic algorithm with elimination, reproduction with crossover and mutation of the agents. The fitness of an agent is represented by the average received payoffs from all played games.

Endogenous interactions exists because agents are able to make a proposal to the "best" agent from their point of view by memorizing the payoffs realized with each other agent. Another important aspect is that agents are able to refuse received proposals to protect themselves from defectors. This refusal leads to an endogenous exclusion of defectors and therefore occurs to all agents that have no expected payoff above the agents threshold. The threshold itself is part of the agents binary string and evolves through the genetic algorithm.

1.3.6 Arbitrary Tags

Riolo (1997) uses arbitrary tags as a partner-matching mechanism in an iterative Prisoner's Dilemma. In his model, the being of an agent is given by a 5-tuple: The first three parameters of the tuple specify the dynamic game strategy and the last two ones the endogenous interactions. The arbitrary tag $\tau_i \in [0,1]$ from agent i is an external label to the other agents. The idea of arbitrary tags is that agents choose more likely agents with similar tags.

Each agent has to find an opponent 10-times. For each opponent-search there exist search costs. Once a pair of players has successfully matched, they play a four-round iterative Prisoner's Dilemma. After this step all existing agents evolve via a genetic algorithm that considers the realized payoff as the fitness of each agent. The above mentioned steps are repeated for 5000 times.

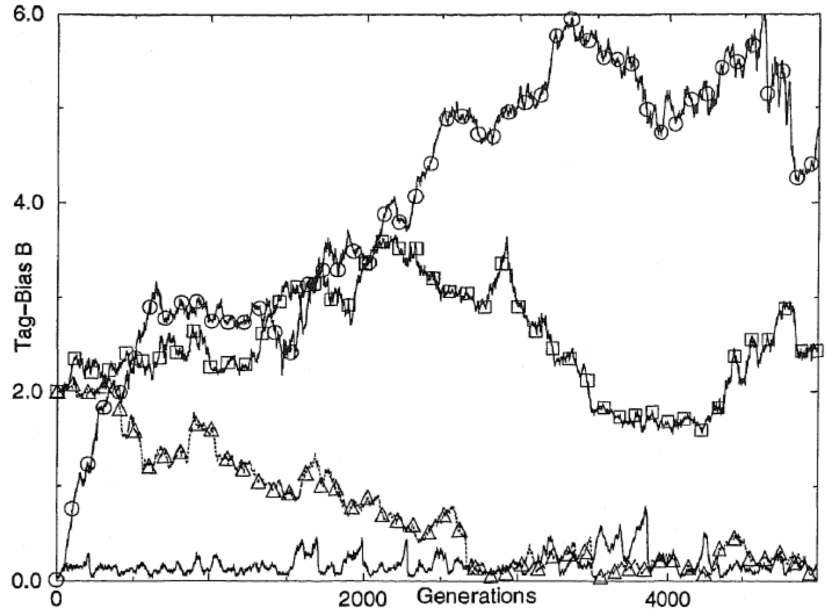


Figure 16: Average pickiness-parameter b (Riolo, 1997).

Figure 16 shows the evolution of the pickiness parameter b with different simulation settings: If the population starts with attaching importance on the distance between both tags (b_0=0.01) and there exists no search costs (D=0) then the population remains in the attitude to see the distance between both tags as important (see '—' line). But if there are search costs introduced into the model (D=0.02), then the population becomes indifferent with respect to tags (see 'o' line). Whereas there is no clear picture if b_0=2 (meaning that the population is not very picky) and there are no search costs (D=0). In one run the pickiness-parameter remains at its starting level (see '□' line) and in a second run b moves to zero (see '△' line) implying the population cares about tags.

The *interactions are endogenous* when an agent tries to find an opponent: First an agent selects an opponent randomly and then he compares the similarities of their tags. The probability that agent i chooses / agrees to an agent j is $1-|\tau_i - \tau_j|^{b(i)}$. $|\tau_i - \tau_j|$ represents the similarity of both tags and $b(i) \in [0,100]$ represents the 'pickiness' of an agent. A low b(i) means that agent i puts special importance on the distance between both tags and therefore is very picky. A high b(i) implies that the agent is not very picky and the distance does not matter. Both agents have to carry out this evaluation and play together only when both agree with their opponent. τ and b are the last two parameters of the 5-tupel of an agent and therefore evolve in the genetic algorithm.

1.3.7 Trust

Hanaki et al. (2004) used trust as a method to model endogenously determined interactions in an repeatedly played one-shot Prisoner's Dilemma game.

At the beginning, all agents are endowed with a game strategy and a trust level (which is the subjective probability of an agent that a new partner will cooperate) at random.

In each period each agent plays one Prisoner's Dilemma with all his partners. Then with exogenously given probabilities some agents can update their game strategy (to cooperate or to defect) and some agents can update their connections by either creating one new link or choosing an already existing link again to a game partner. When an agent updates his game strategy he chooses the most successful game strategy from his connected partners (he also considers his own most recent game strategy). The success is measured by the sum of all received payoffs from the last period.

E*ndogenous interactions* occur between the agents when creating a new link:

An agent has two possibilities for choosing a new partner (creating a new link). In both cases he only chooses the new partner if the net benefit seems to be positive. The first one is to choose a new partner from the partners of his already existing partners (friend of friend). In this case the partner informs the agent about the recent played game strategy of this potential new partner. The second one is to choose a new partner at random out of the whole population. In this case the expected payoff depends on the trust level. This level is calculated as a weighted average of the already experienced amount of cooperation of partners with whom they have already interacted, putting more weight on more recent experiences.

So the endogeneity of the interactions is modelled as a level of trust (subjective probability of facing cooperation from the opponent) which evolves over time.

1.3.8 Expected Payoff / Familiarity

Kirman and Vriend (2001) modelled the wholesale fish market of Marseille by using the loyalty (=familiarity) of customers and the expected payoff as methods to implement endogenous interactions.

Before the fish market opens, the sellers determine the amount of fish they want to supply for the day without knowing the demand. Then the market opens and each buyer chooses a queue of a seller depending on the average realized payoff. Every buyer wants to buy one unit of fish each day. Next the sellers handle their queues by serving sequentially the buyers in any order they want. In doing so the sellers could offer each buyer an individual price on basis of the familiarity of their faces in their queue (=loyalty).

After having handled all sellers queues, the market closes and reopens as a market in the afternoon. There all unsatisfied buyers choose a queue of a seller, the sellers handle their queues and finally all unsold fishes perish.

Then reinforcement learning starts: Each seller has to make four decisions meaning there exist four classifier systems (decision boxes). A decision box is a set of alternative rules. There is one decision box for one of the following decisions to make: The quantity to supply, how to handle the queue and the function for determining the different prices for the single buyers in the morning and in the afternoon. Fitter rules, which return a higher realized payoff, are more likely to be used again. In contrast to the seller, the buyer's alternative rules specify the price level for accepting an offer and the choice of a seller (depending on the average realized payoff with him) once for the morning and once for the afternoon. So the buyer also has four decisions to make implying there exist four classifier systems for each buyer.

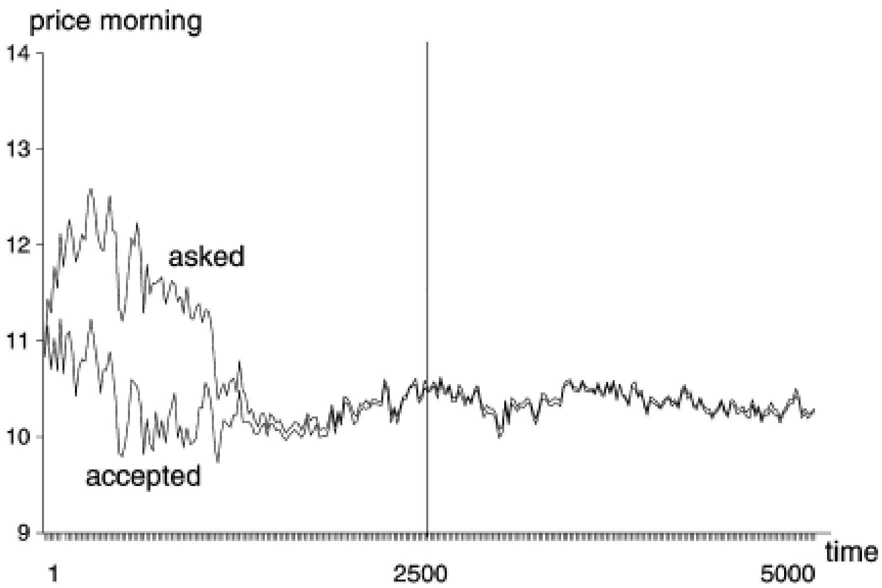


Figure 17: Average prices asked / accepted during the morning sessions (Kirman and Vriend, 2001).

Figure 17 shows the average asked and accepted prices during the first 5000 morning sessions. At the beginning of the simulation the asked and accepted prices diverge (the asked prices increase whereas the accepted prices decrease). After some time the asked prices are dragged down by the accepted prices. Both prices are stable at a price level of 10.3 after around 2000 ticks.

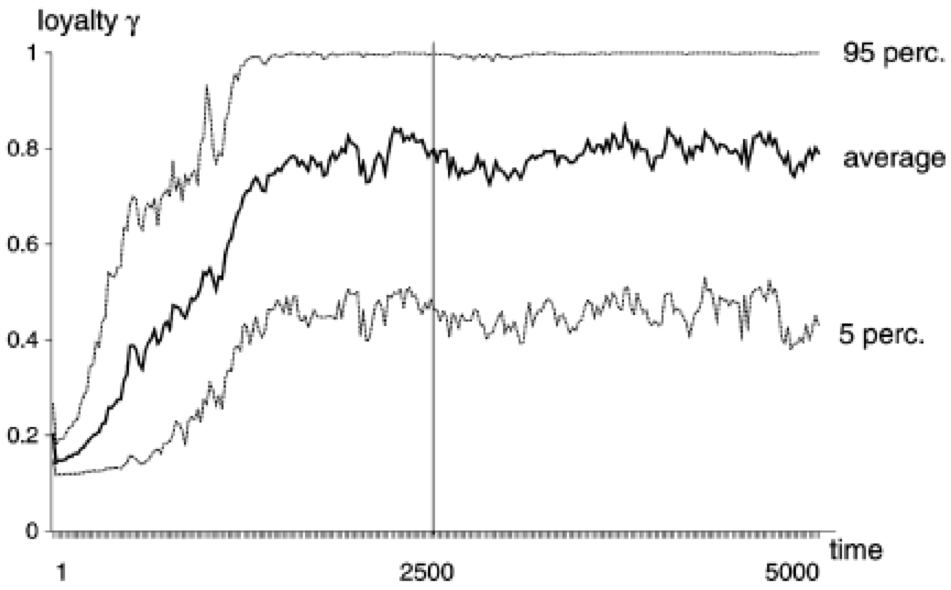


Figure 18: Average loyalty during morning sessions (Kirman and Vriend, 2001).

Figure 18 shows that loyalty emerges despite the buyers do not even know for what loyalty stands for. At first loyalty emerges only slowly but at some point it starts to grow fast up to an average level of ~0.8. The variance among the buyers is represented with the 5 - percentile and the 95 - percentile measure of loyalty over all buyers. With increasing loyalty also the variance increases.

The *interactions are endogenous* because a buyer chooses a seller's queue by considering the expected payoff (=average realized payoff with the seller). Therefore the higher the expected payoff the higher the possibility of selecting the specific seller. Likewise sellers can favour some buyers over other buyers by considering the familiarity of the faces of the buyers, meaning they prefer loyal buyers. Also the individual price asked from a buyer depends on the familiarity. The familiarity is a weighted average of the past presence of a buyer in a seller's queue. It is important to point out that the sellers have to learn whether to give advantage (e.g. when handling their queues, determining the individual price) or disadvantage to loyal buyers. At the beginning of the simulation they are absolutely indifferent to this decision.

1.3.9 Past success rate

Chang and Harrington (2005) use the past success rate (modelled as weights) as a method to influence the decision process of agents whether to interact or to stay alone:

In their model all agents have to solve the same exogenously given number of tasks. The method to solve a single task is described as a binary string. At the same time there exists a target vector, describing the optimal way to solve all tasks. This vector can change over

time. An agent only adopts a new method if this method is better for solving the task than the already used one by comparing the amount of different bits (Hamming distance)[8] from this method with the target vector.
Furthermore an agent has to decide whether to innovate (randomly choosing a binary string) or to imitate another agent by copying his method.

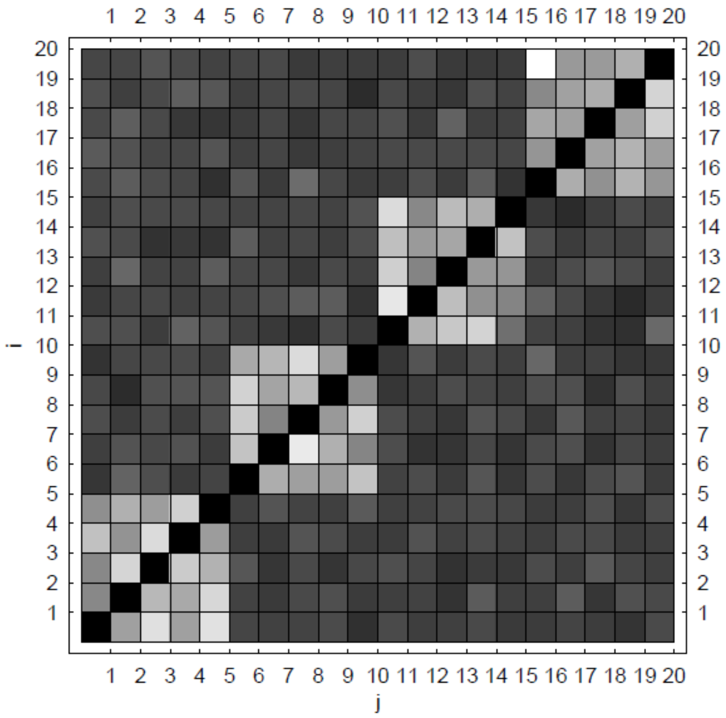


Figure 19: Probability of agent i to imitate agent j (Chang and Harrington, 2005).

Chang and Harrington (2005) formed four groups, each group consisting of five agents. Group one consists of agent 1 to agent 5, group 2 consists of agent 6 to agent 10 and so on. The target vector of each group follows a separate stochastic process (the vector changes over time). Hence, the target vectors of the groups differ over time. But the agents do not know this fact. Figure 19 shows the endogenous emerged probabilities of agent i imitating agent j. The lighter the areas the higher is the probability that agent i imitates agent j. It can be seen that the agents have learned to imitate more with agents within their group. This can be seen by the four lighter 5x5 boxes.

The *interactions are endogenous* because interaction (in their model addressed as imitating) depends on the past success of choosing to imitate. The decision process of an agent is a probabilistic one, only influenced by weights depending on the past success of the according chosen option (imitation / innovation). So the first decision step determines whether to interact at all. If an agent has chosen to interact (to imitate) then the question arises with

[8] Hamming (1950)

whom he should interact (from whom he should copy the method). In this case another decision step occurs, where the probability of choosing an agent increases (his weighting gets increased) if the past imitation of this agent was successful. If an agent is not chosen in one period, then his weight decreases automatically.
Reinforcement learning (by considering the past success rate) is used at each decision step to ensure to choose the optimal decisions.

1.3.10 Directed random search

In the model of Jackson and Rogers (2004) at each time step one agent is added to a network. Before he is added to the network he chooses an uniform randomly sample out of all existing agents within the network. Then he chooses an uniform randomly second sample of agents that are directly linked to the chosen agents of the first sample. After doing so an agent creates out of the first and second sample of chosen agents myopically links with those agents that generate a positive net utility for him. In the simplest version of the model the net utility of a link is independently and identically distributed across all pairs of agents. The sizes of these two samples are exogenously given.

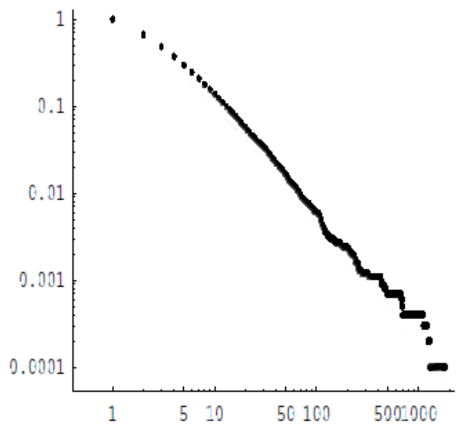


Figure 20: Distribution of node degrees (Jackson and Rogers, 2004).

Figure 20 represents the results from a simulation with 10 000 periods in a diagram with log-scaled axes. In this simulation the two samples had both a size of two agents and each agent in the sample had a positive net utility implying that all sampled agents had been linked. The upper tail of the graph is linear which indicates that there is a scale-free distribution (distribution of the degrees of the nodes follows a power law). But the lower tail is not linear, meaning no scale-free distribution exists there.

Forming links is endogenous because the second sample chosen by an agent depends on the network structure. If a new agent creates links, he automatically influences the structure of the network and therefore he also influences the second samples chosen by new agents in the future.

Finally there is to say that *most endogenous interactions follow a certain schema*:

1. *Interactions between agents are guided by a certain variable x*

   This means that the agents put attention to this variable x. x could be for example the average expected payoff, the loyalty of an agent or the residential pattern which an agent faces. Of course agents could consider multiple variables and not just one. (e.g. by extending the model of Kirman and Vriend (2001) by letting the buyer considering not just the expected payoff with a seller, but also the length of the queue or the familiarity of the seller.)

2. *Variable x evolves as a consequent of the interactions between the agents.*

   E.g. In the 'El Farol' – bar problem the predicted attendance value by a certain predictor changes as a consequent of the single made decision of all agents to move to the bar or to stay at home. Another example is the expected payoff where the expected payoff changes due to the last made interactions (to join a sellers queue and got served well/badly or not).

3. *The importance of variable x for a single agent may evolves by a learning process.*

   In most models there exists multiple instances of the variable x (e.g. there exist multiple predictors in the 'El Farol' – bar problem like the average number of attendances of the last month, the same number of attendances as last week, …) that could be considered. Before an agent decides to interact (or with whom to interact) or not, he first has to decide which instance he should take to make this decision. Therefore he must evaluate the fitness of each instance (which can be calculated by experiences of past made interactions). Many models use a classifier system to implement this learning process.

2. Implementation of the model of Kirman and Vriend (2001)

2.1 Motivation

Kirman and Vriend (2001) analysed the Marseille fish market and found two *stylised facts* about this market: The first one is the existence of a high *loyalty* of buyers to sellers and the second one is the presence of *price dispersion*. Due to the fact that these two phenomena (stylised facts) are existent in other markets too, the derived insights about them, from studies made on the Marseille fish market, might be transferable to other markets as well.

Their reason for analysing this fish market is that there exists a rich data set about the single transactions that had taken place in this market over some years. At the same time this market is simple and well-structured, which makes an analysis easier. Both reasons support the construction of a suitable model.

To understand how loyalty and price dispersion could emerge they created an agent-based model to provide a mechanism that could explain these phenomena from its lowest level: the interaction between buyers and sellers.

2.2 The Model

The following sections describe, the model set-up of Kirman and Vriend (2001).

2.2.1 The Market

The *real fish market in Marseille* consists of a fixed population of 40 sellers and around 400 buyers. Before the market opens sellers buy their supply of fish outside the market. The buyers are either retailers, implying they get some specific price for reselling the bought fish afterward outside the market, or restaurants. During the opening hours from 2 a.m. to 6 a.m., buyers visit sellers to buy fish. Once a buyer has selected a seller, they stand face to face and the buyer tells the seller the type and quantity of fish he wants to have. Then the seller decides the individual price, which is not posted to the other buyers. The individuality of the price is characterized by the possibilities of the seller to decide his own price and to offer different buyers different prices. Another feature of the individual price setting is that a seller may ask a different price for the same buyer who wants to have the same type and quantity of fish at different times. The asked price are ultimate prices meaning there exists no bargaining between buyers and sellers. If the market closes, unsold fish perish and get disposed. Some types of fishes can be sold the next day but with quality losses which can be recognized by the buyers (Kirman and Vriend, 2001).

Kirman and Vriend (2001) model an abstract version of the fish market in Marseille with 10 identical sellers and 100 identical buyers, both with adaptive behaviour by using classifier systems. Additionally, the authors introduced a morning session and a afternoon session, which means that the market opens and closes two times a day. The structure of the simulation can be seen in figure 21:

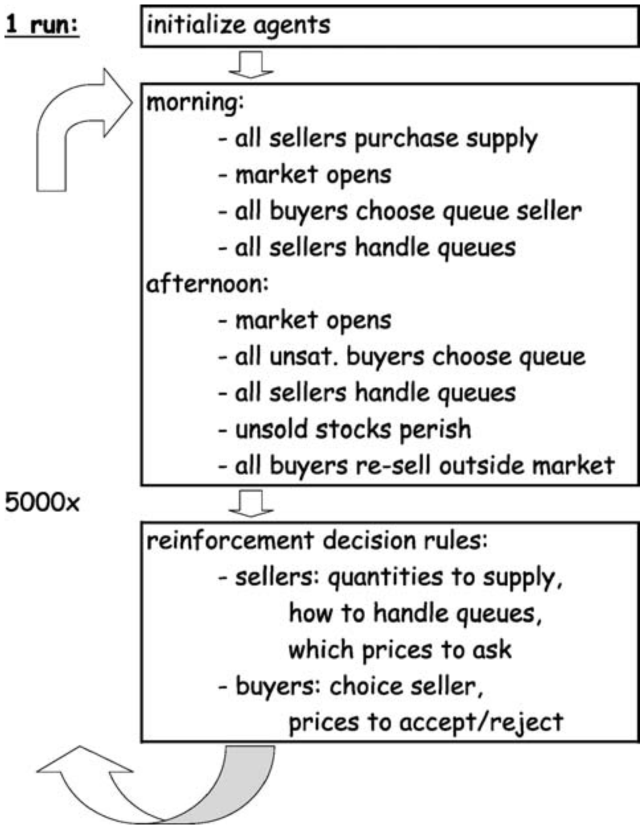


Figure 21: Overview of the simulation of Kirman and Vriend (2001)[9].

In the morning session, all sellers buy their stock of fish for an exogenously given price $p^{in} = 9$ for each unit of fish. Then the market opens and each buyer chooses a seller because every buyer wants to have one unit of fish per day. Therefore a buyer adds himself to the queue of his desired seller. Next all sellers handle their queues by serving the buyers in any order they want until either the stock of fish is empty or there are not any further buyers in the queue. When the seller serves a buyer, he chooses an individual price for the specific buyer. Then the buyer ether accepts his offer or rejects it. Once this is done the market reopens in the afternoon (afternoon session). There all unsatisfied buyers that haven't got any fish in the morning, ether they had rejected the asked price or the sellers stock was empty when it was their turn, choose again a seller that has open (implying that the seller still has any fish) and add themselves to the desired sellers queues. Then, like in the morning session, the sellers

[9] Tesfatsion & Kenneth (2006, chap. 21)

handle their queues. If all queues are handled, all unsold fish perish. The buyers resell their fish for a fixed given price of $p^{out} = 15$ outside the market..

Once this is done, reinforcement learning starts to optimise their decision behaviour: During the day, each agent (whether being a buyer or a seller) made some decisions. Each *buyer* had to make the following four *decisions*:

1. the choice of seller in the morning and
2. the choice of seller in the afternoon he wants to visit.
3. prices to accept or reject in the morning
4. prices to accept or reject in the afternoon

Whereas each *seller* had to *decide* upon:

1. the quantity of fish he wants to supply
2. how to handle his queue
3. prices to ask in the morning
4. prices to ask in the afternoon

The whole procedure is repeated 5000 times, starting in the morning session with the buying of the sellers stock and ends after the afternoon session with the reinforcement learning of all agents.

2.2.2 Reinforcement Learning

Kirman and Vriend (2001) endowed in their model buyers and sellers with an adaptive behaviour. This means both agent types are able to learn to optimise their decision making by adaptively adjusting their behaviour to the actual situation. Therefore they used in their model classifier systems, which are a special form of reinforcement learning.

Kaelbling et al. (1996) describe *reinforcement learning* as the problem of an agent to learn its behaviour through trial and error interactions with a dynamic environment. When an agent interacts with his environment, first he perceives his environment (as an input) to know the current state $s \in S$ of it, where S specifies the space of possible states. Then the agent generates an output by choosing an action $a \in A$ that changes the actual state of the environment. A, the action space, represents all possible actions of the agent. In another step the value of this transition from the old state to the new state is communicated to the agent as a so called reinforcement signal. The goal of each agent is to find a behaviour B that maximizes the sum of values of the reinforcement signals over time. (see figure 22)

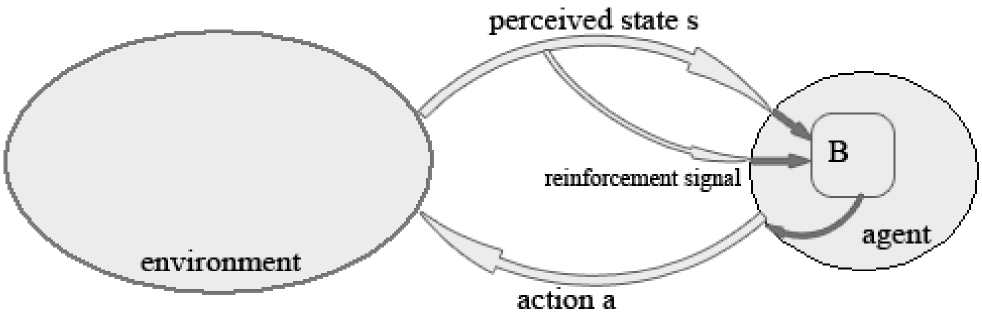


Figure 22: Standard reinforcement learning model.

A very simple version of a *classifier system*, which uses reinforcement learning, is used by Kirman and Vriend (2001) to implement the adaptive behaviour of the agents. A classifier system CS (see *table 3*) consists of a set of condition based rules (IF <<condition>> THEN <<action>>), each rule is a classifier, and the according strength $s_i(t)$ for each rule.

condition (IF)	**action** (THEN)	**strength**	
…	…	…	} classifier 1
…	…	…	} classifier n

Table 3: Classifier system.

Each time t the classifier system chooses the active rules (whose condition block is fulfilled / true) and calculates for each active rule i according to

$$b_i(t) = s_i(t) + \varepsilon \qquad , \varepsilon \sim N(0, \sigma)$$

a bid $b_i(t)$. The rule with the highest bid is the winner and gets selected by the CS. The error term ε ensures that other rules get the chance to get selected even if they are not the strongest currently. Once the action of the active rule i has been executed, the CS updates the strength of this rule by taking into account the reinforcement signal (reward π) received from the environment by:

$$s_i(t+1) = s_i(t) - c^*s_i(t) + c^*\pi(t) \leftrightarrow \Delta s_i(t+1) = c^* (\pi(t) - s_i(t)) \qquad , 0 < c < 1$$

This implies, that the strength of the rule will increase as long as the received reinforcement signal π is greater than $s_i(t)$ (the actual strength of the activated rule i) (Kirman and Vriend, 2001).

2.2.3 Agents & Behavioural Rules

There exists two classes of agents, buyers and sellers. As mentioned before, each agent has to make four decisions. Kirman and Vriend (2001) modelled each decision problem as a single classifier system. If there exists 10 sellers and 100 buyers as in the initial setup of their model, they had to model 440 classifier systems.

A **seller** first has to decide the *quantity of fish* he wants to *supply* for the day. In this case the single rules have no condition part and just consist of an action and strength part:

action	strength
supply 0 units	…
…	…
supply 30 units	…

Table 4: CS for the quantity of fish to supply.

The strength is calculated by using the achieved net profit.

Second, the seller has to decide *how* he wants *to handle his queue*. This decision is influenced by the familiarity of the face of the single buyer. So the seller is able to differentiate between the single buyers. He looks in his queue, sees a crowd of buyers and associates the different faces to different degrees of familiarity. This degree of familiarity of buyer i to seller j on day t is calculated as:

$$L_{ij}(t) = \sum_{x=1}^{t} \frac{r_{ij}(t-x)}{(1+\alpha)^{t-x}}$$

Where $r_{ij}(t\text{-}x) = \alpha$ indicates that buyer i visited seller j on day x and $r_{ij}(t\text{-}x) = 0$, if buyer i did not visit seller j on day x. α is an exogenous given parameter between 0 and 1. L_{ij} is just a weighted average of the past attendances of the buyer in the sellers queue, putting more weight to the last visits. If $L_{ij} = 0$, the buyer is not loyal and if $L_{ij} = 1$, he is perfect loyal. Once the sellers knows the familiarity of a buyer, he has to decide whether to put advantage, disadvantage or being indifferent to a familiar face. Therefore Kirman and Vriend (2001) introduced a roulette wheel, where each buyer of the queue gets a slot of a specific size $(1 + L_{ij})^b$. If b = 0, the slot size for all buyers is equal and therefore the seller is indifferent in setting the serving order whether he should prefer loyal or occasional buyers. If b > 0, the seller puts advantage in serving loyal customers, because their slot size in the roulette wheel increases and therefore their probability of getting selected increases as well. Otherwise, if b < 0, a familiar buyer gets disadvantaged because his slot size gets decreased.

action	strength
b = -25	…
…	…
b = 25	…

Table 5: CS for queue handling.

The strength of the classifiers are daily updated by calculating the payoff as:

$$payoff = \frac{gross\ revenue\ of\ day}{highest\ achieved\ price\ of\ day * supply\ of\ fish}$$

Once more, it is important to point out that the seller first has to learn if it is more profitable for him to put advantages or disadvantages to loyal buyers!

The third and the fourth decision a seller has to make is about the *price to ask*, once for the morning and once for the afternoon. The CS for the morning session and the afternoon session look both equal (see table 6). The condition part of the rules consists of two variables: The first one is the loyalty class, which is calculated by categorizing the loyalty L_{ij} into three classes (IF L_{ij} <= 0.2 THEN loyaltyClass ='low'; IF L_{ij} <= 0.80 THEN loyaltyClass = 'middle' and otherwise loyaltyClass = 'high'). The second variable of the condition part is called ratio and is calculated by categorizing the ratio of the remaining stock of fish and the remaining length of the queue of buyers into three classes, ranging from low to high (IF stockQueueRatio <= 0.75 THEN ratio = 'low'; IF stockQueueRatio <= 1.75 THEN ratio = 'middle'; otherwise ratio = 'high'). The possible prices to ask in the action block ranging from 0 to 20. By building the combination of all these three variables (3 classes of loyalty * 3 ratio classes * 21 prices to ask = 189 rules) 189 rules can be derived. Table 6 shows the CS for the prices to ask. It should be noted, that two CS are need, one for the morning session and one for the afternoon session. One reason therefore is that it could be more profitable to ask higher prices in the afternoon than in the morning.

condition	action	strength
loyaltyClass = 'low' & ratio = 'low'	$price^{ask} = 0$	…
…	…	…
loyaltyClass = 'high' & ratio = 'high'	$price^{ask} = 20$	…

Table 6: CS for prices to ask in morning / afternoon.

The strengths of the rules are updated daily by using the payoff:

$$payoff = \frac{times\ accepted\ this\ rule\ from\ buyer * price\ asked}{times\ activated\ this\ rule * price\ asked}$$

A **buyer** has to decide *which seller* he wants *to choose* in the morning / afternoon.
If a transaction takes place or the price is rejected by the buyer, the actual realized payoff is calculated as the max$\{p^{out} - p^{found}, 0\}$. Otherwise, the payoff is 0. p^{out} is the price at which the buyers can resell the fish outside the market and p^{found} is the price the buyer has to pay to buy one unit of fish from the seller. He records the weighted average realized payoff with every seller, by calculating 0.95 * old average payoff + 0.05 * actual payoff and uses this payoff for the strength of the rule. Table 7 shows how the CS for the choice of seller looks like. A

second CS exists for the afternoon session, because choosing a specific seller could not be profitable in the morning but profitable in the afternoon.

action	strength
choose seller 1	…
…	…
choose seller 10	…

Table 7: CS for choosing a seller in the morning.

The third and fourth decision a buyer has to make is the *price to accept or reject* in the morning and in the afternoon. There are 21 possible price the sellers are able to ask for, reaching from 0 to 20. The buyer has the choice to accept or reject an asked price. Therefore 42 classifiers exist (21 possible prices to ask for * 2 {accept, reject}). The strength are updated daily. $p^{morning}$ is the price accepted in the morning session and $p^{afternoon}$ is the price accepted in the afternoon session. In the case of the morning session, if the price is accepted the payoff = $p^{out} - p^{morning}$ is used as strength. Otherwise, if the price in the morning is not accepted but a transaction occurred in the afternoon, the payoff = $\max\{p^{out} - p^{afternoon}, 0\}$ is used. The payoff is zero, if neither a transaction occurred in the morning session nor in the afternoon session.

The strength for the CS for the afternoon session is updated by using the payoff = $p^{out} - p^{afternoon}$ if a transaction occurred in the afternoon. Otherwise the payoff is zero. Table 8 shows the structure of the classifier system for the price to accept or reject in the morning / afternoon session.

condition	action	strength
$price^{asked} = 0$	reject	…
$price^{asked} = 0$	accept	
…	…	…
$price^{asked} = 20$	reject	…

Table 8: CS for price to accept in the morning / afternoon.

2.2.4 Loyalty Index

Kirman and Vriend (2001) introduce a *loyalty index* to measure the loyalty of a buyer i at time t:

$$\gamma_i(t) = \frac{\sum_{seller\_j} \left(L_{ij}(t)\right)^2}{\left(\sum_{seller\_j} L_{ij}(t)\right)^2}$$

A buyer who is perfect loyal to one seller has a $\gamma_i(t)$=1, whereas $\gamma_i(t) = \dfrac{1}{number\ of\ sellers}$ if the buyer is not loyal to all sellers meaning he visits periodically all sellers. The total number of sellers the market consist of is called 'number of sellers'. L_{ij} is the degree of familiarity (see chapter 2.2.3 Agents & Behavioural Rules).

The average loyalty level among all n seller is simply calculated by:

$$avg\ loyalty\ level = \frac{1}{n} * \sum_{i=1}^{n} \gamma_i(t)$$

2.3 Implementation

The model about evolving market structures of Kirman and Vriend (2001) represents how an endogenous network arises by using the mechanisms of expected payoff and familiarity to model endogenous interactions.

On the following pages I try to reproduce this model by implementing it as a NetLogo-program. Furthermore I focused on the most important findings of Kirman and Vriend (2001) about loyalty which are:

- emergence of loyalty
- increase of loyalty with rising p^{out} (price for reselling the fish outside the market)

2.3.1 Details About The Model Replication

As mentioned in chapter 1.2.4 Model Replication, Wilensky and Rand (2007) recommend to post certain important information when doing model replication:

Details about the model replication	
Categories of Replication Standards	relational alignment
Focal Measures	loyalty measure, price accepted, price asked
Level of Communication	none
Familiarity with Language / Toolkit	unknown
Examination of Source Code	none, used pseudo-code for reinforcement learning step
Exposure to Original Implemented Model	none
Exploration of Parameter Space	only examined results from original paper

Table 9: Details about the model replication.

The chosen replication standard is 'relational alignment'. This means that the results of the original and replicated model show similar relationships between the input and output variables. For example if the type of buyer increases (buyer 15 → buyer 18) then also the loyalty increases.

Loyalty measure, price accepted and price asked, chosen as focal measures, are these kind of measures that are compared (see relational alignment) with the measures from the original model. No communication between the original model developer and the replicator has taken place.

Because of not having the original source code of the model and also not knowing the used simulation language / toolkit, I can not specify whether I am familiar with the language or not.

Due to not having the source code the examination of the source code was not possible. But the pseudo-code was available and so it was possible to compare my source code to a certain extend with the pseudo-code (especially for the reinforcement learning part and the configuration of the normal distribution for certain error terms).

2.3.2 Activity diagram

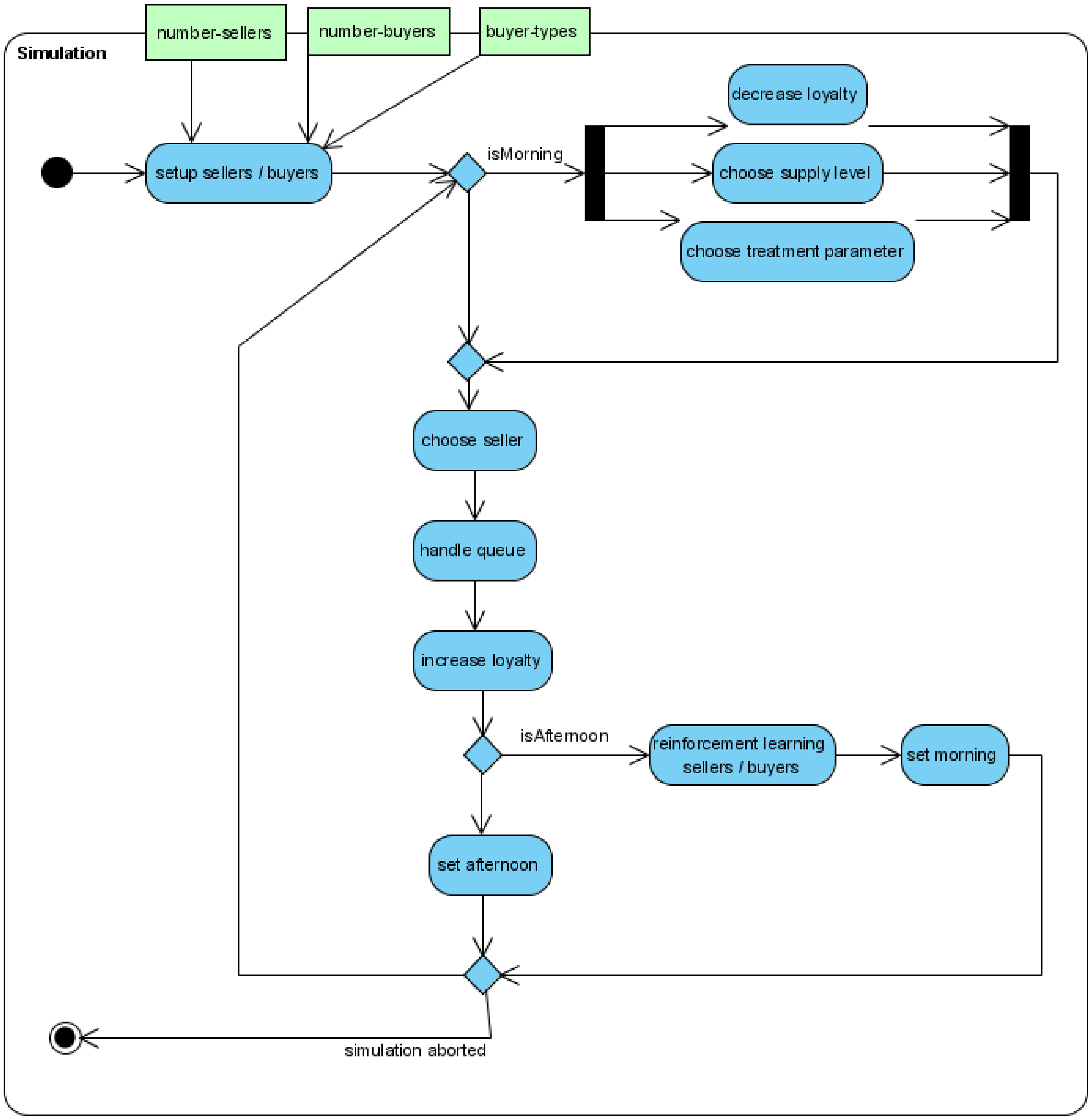


Figure 23: Activity Diagram of the model.

Figure 23 represents the activity diagram of the reproduced model.

On the basis of the description of the model and the given pseudo code of Kirman and Vriend (2001), it was possible to derive a semantic equal activity diagram of the model.

The three main parameters for the simulation are the number of sellers, the number of buyers and the type of buyers. The last parameter defines whether all buyers receive the same price for reselling the fish outside the market (p^{out} is the same for all buyers) or if there exist three types of buyers, each type receiving another p^{out}.

At the beginning of the simulation the whole system is set into the morning session. The setup – activity initialises the buyers and sellers by setting the fitness (=strengths) for all rules of all classifier systems to 1. Then each seller decrease the loyalty for all buyers, chooses a supply level of fish for the whole day and the treatment parameter b that defines the degree of giving advantage or disadvantage to loyal buyers. These three activities can theoretically be parallelized due to not being conditioned among each other. The two decision steps (choose supply level & choose treatment parameter) are done by determining the fittest rule for each decision.

Then each buyer chooses a seller by taking into account the fitness of each seller of the specific time session (considering whether it is morning or afternoon). After doing so the sellers start to handle their queues: First they determine the next buyer to serve out of their queues by calculating for each buyer in their queues a weight $(1 + L_{ij})^b$ (by using the loyalty of the specific buyer and the treatment parameter b; see also chapter 2.2.3 Agents & Behavioural Rules). Next they compute the price to ask by using the fittest suitable price rule. Once the buyer has accepted / rejected the price, the buyer is removed from the queue, the fish stock is decreased by one (if he has accepted the price) and the same procedure starts again until the fish stock is zero or all buyers are served.

After the queue handling – activity follows an increase in the loyalty of all buyers having been in the queue. If the system is in the morning state then it is set to the afternoon state. Otherwise it is already in the afternoon state which implies that reinforcement learning for all classifier systems starts by adapting the fitness of the single rules. The new fitness of a rule is calculated by adding accordingly to the specific rule a weighted utility value (e.g. ratio of profit to revenue) to the old fitness value. Next the system is set to the morning state.

2.3.3 Assumptions

I found two errors in the paper that lead to conflicts when I tried to reproduce the model: First, I was only able to reproduce similar results as Kirman and Vriend (2001) by setting the random error term $\varepsilon \sim N(0, 0.01)$. When using a standard deviation of 0.1 as used in their paper, I was not able to reproduce their findings of the model. So it is very likely to interpret this high standard deviation (0.1) of the normal distribution as a typo.

Second, their pseudo-code is not consistent with their verbal description of the model. According to page 470, the buyer makes reinforcement of rules only once a day. In their described main procedure of the pseudo-code, the buyers do reinforcement both in the morning and in the afternoon session.

According to these two found errors, I assumed that the standard deviation of the normal distribution of the random error term ε is 0.01 and that reinforcement learning of the buyers occurs only at the end of the day.

Furthermore I did not implement the case that every valid rule (a rule whose condition block is fulfilled) in every decision step of an agent (e.g. choice of seller in the morning, which price to ask in the afternoon, …) is ignored with the probability of 0.025.

2.3.4 Agents

Although NetLogo is not an object-oriented programming language, it provides a rich set of functions to query and manipulate subsets of agents out of the agents population. Unlike to an object-oriented programming language, NetLogo does not allow to encapsulate the data (attributes) and the functions (methods) to one class. It only offers the possibility to represent agents in form of structs like in the programming language C. At the lowest level this just means that different data types can be subsumed to one complex data type, called breed in NetLogo. So the data an agent consists of, is subsumed to one breed and the programmed functions of the program represent the behaviour of an agent.

The model consists of two types of agents. As mentioned before, an agent type can be represented as a 'breed'. To represent the two types of agents, two breeds, called buyers and sellers, are created. If an agent of the type buyer / seller is created, he belongs to the breed 'buyers' / 'sellers'.

```
BUYER
        id;
        satisfied;
        p_out;
        fitness-sellerMorning [];
        fitness-sellerAfternoon [];
        fitness-priceRulesBuyerMorning [];
        fitness-priceRulesBuyerAfternoon [];

SELLER
        id;
        fishStock;
        queue list;
        loyalties [];
        fitness-quantities [];
        fitness-treatmentParameters [];
        fitness-priceRuleSellerMorning [];
        fitness-priceRuleSellerAfternoon [];
```

Pseudo Code 1: The two agent types.

Both agent types have in common that they are able to identify one another by their id. The buyers have additionally the variables p_out (p^{out}), the price for reselling the fish outside the market, a status variable satisfied indicating if they have received any fish and four fitness arrays, which store the strengths of the single rules of the four classifier systems for the decision making. A variable followed by brackets [] indicates that this variable represents an array.

Apart from the four fitness arrays for storing the strengths for the single CS, sellers have the variables fishStock, which represents the daily stock of fish, a list called queue to represent the queue of buyers wanting to buy some fish from the seller and an array named loyalties, to store the degree of loyalty about every single buyer.

The fitness arrays are initialised before the simulation starts with the value one at every position, indicating that all rules are equal good.

2.3.5 Rules

The simulation consists of two classes of classifier systems CS: The first class is a CS that consists only of rules with no condition-part (e.g. choice of seller, quantity of fish to supply, how to handle the queue), contrary to the second class of CS (e.g. price to ask, price to accept).

Rules of a CS with no condition part are represented as an array. Each position of the array contains one action block. The according strength of the rule is saved on the same position in the particular fitness-array (see chapter 2.3.4 Agents). Figure 24 shows how a CS for the choice of seller in the morning is modelled. The array rules-seller-choice contains the ids of the available sellers in the market.

rules-seller-choice:

1	2	3	4	5	6	7	8	9	10

fitness-seller-morning:

1	0.6	0	0.5	0.5	1	08.	0.2	0.3	0.4

Figure 24: Implementation of a CS with no condition-part (CS for choosing a seller in the morning).

Whereas CS whose rules contain a condition-part are modelled as an extra agent type in NetLogo. This made it very convenient to query the actual valid rules because NetLogo provides the appropriate commands for querying agents / subset of agents according to their internal variables (e.g. id, loyaltyClass, stockQueueRatioClass,…). In fact I modelled two pseudo agent types to represent the rules for two CS: One type is modelled for the prices to ask in the morning / afternoon (see PRICERULE seller) and one type for the prices to accept in the morning / afternoon (PRICERULE Buyer). Both price-rule types consist of an id. Again, the according strength of each rule is saved in the particular fitness-array and gets associated by the value of the id variable of the rule. Pseudo Code 2 shows the two rule types of the classifier systems for prices to ask in the morning / afternoon and for the prices to accept / reject in the morning / afternoon.

```
PRICERULE Seller
        id;
        loyaltyClass;
        stockQueueRatioClass;
        priceToAsk;                         // action-block

PRICERULE Buyer
        id;
        price;
        accept;                             // action-block
```

Pseudo Code 2: Implementation of CS with a condition part.

As mentioned before, the NetLogo-language facilitates to query easily subsets of agents. For example, if a seller wants to calculate the specific price to ask in the morning for a specific buyer, this can be implemented in the NetLogo-Syntax as:

```
ask pricerulesSeller with
[loyaltyClass = BuyersLoyaltyClass AND stockQueueRatioClass = actualStockQueueRatio]
```

According to chapter '2.2.2 Reinforcement Learning' a rule gets selected by calculating a bid for each valid rule (whose condition-part is fulfilled) and then choosing the rule with the highest bid. A bid for rule i is calculated by $b_i(t) = s_i(t) + \varepsilon$, whereas the error term ε is normal distributed as $\varepsilon \sim N(0, 0.01)$.

2.3.6 Reinforcement Learning

The reinforcement learning part could be easily reproduced in NetLogo by making use of the provided pseudo code of Kirman and Vriend (2001, Appendix). Reinforcement learning is executed at the end of each day. Sellers have to update their strength of the activated supply level rule, the strength of the chosen treatment parameter b and the strengths of all morning-price and afternoon-price rules. Buyers have to distinguish between seven possible scenarios:

1. transaction occurred in the morning
2. offer was rejected (by the buyer) in the morning and transaction occurred in the afternoon
3. offer was rejected in the morning and in the afternoon
4. offer was rejected in the morning and buyer was late (did not get any offer) in the afternoon
5. buyer was late in the morning and transaction occurred in the afternoon
6. buyer was late in the morning and offer was rejected in afternoon
7. buyer was late in the morning and in the afternoon

Depending on the entered scenario, different classifier systems are involved in the reinforcement step. For example, if a buyer faces the second scenario, then he has to update the strength of the seller-choice in the morning, the morning buyer-pricerule, the seller-choice in the afternoon and the afternoon buyer-pricerule (for details see Appendix).

2.4 Results

In the following, I present the results of the simulations, once for the one buyer-type (all buyer receive the same price (p^{out}) for reselling the fish outside the market) and once for the three buyer-type (three classes of buyers, each of them receive a different p^{out}) with focus on the average emerged loyalty and the average prices asked / accepted.

2.4.1 One Buyer-Type

Configuration of the simulation

The simulation consists of 10 sellers and 100 buyers (number-sellers = 10, number-buyers = 100). All buyers receive the same $p^{out} = 15$ (buyer-types = 1). Each simulation runs for 4000 time units (ticks). To generate sufficient data for further analysis I have made 40 simulations.

Results

With the reproduction of the model I tried to reproduce the main findings about the emergence of loyalty:

In the first version of the simulation I tried to come to the same results about the appearance of loyalty in the morning session as Kirman and Vriend (2001) did.

Furthermore I tried to reproduce the fact, that when having three types of buyers (p^{out} = 12, p^{out} = 15, p^{out} = 18) the loyalty increases with the increase of p^{out}.

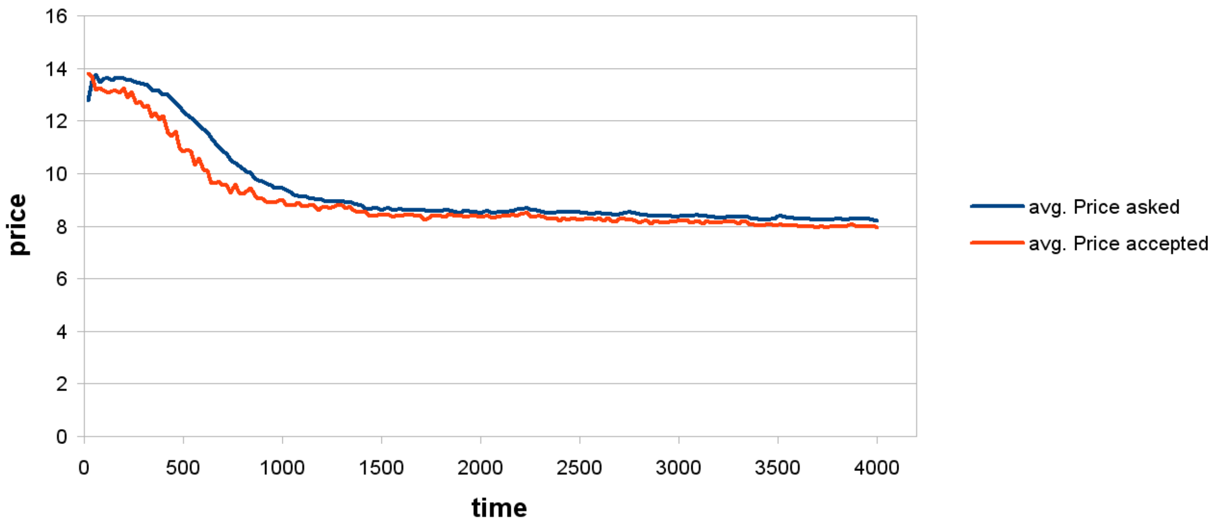


Figure 25: Average prices asked / accepted during the morning session.

Figure 25 represents the average price asked and the average price accepted during the morning session. At first, the two prices start to diverge to the same extend like in the simulation of Kirman and Vriend (2001). Also after the same time (~1250 ticks) the asked and accepted prices have converged.

But two main differences between my simulation and the simulation of Kirman and Vriend (2001) remain: First, the two prices never come so close together like in their figure (Kirman and Vriend, 2001, fig. 3). There is always a difference between the asked and accepted price in the later phases of around 0.2. Second, in the long run a price level of 8 is reached, whereas they reach a price level of 10.

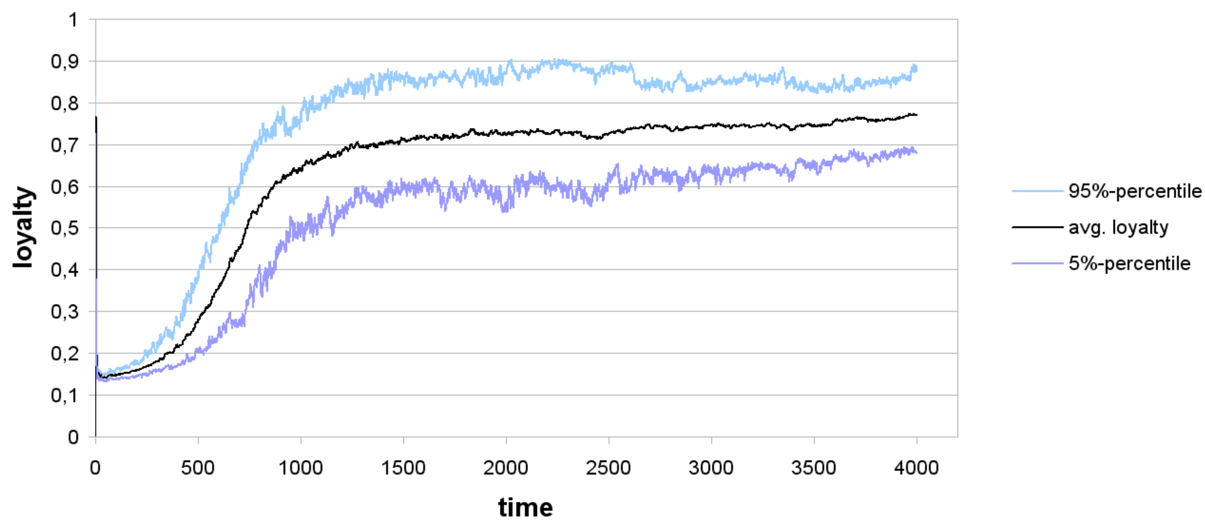


Figure 26: Average loyalty.

Figure 26 shows the average, the 5 - percentile and the 95 - percentile measure of loyalty over all buyers. In the last 1000 days the average loyalty was 0.75, the 5 – percentile was 0.63 and the 95 – percentile was 0.86. The diagram also shows that there would be a small further increase in the loyalty if the simulation would run for a longer time.

A comparison of the results from my simulation with the findings of Kirman and Vriend (2001) shows that the speed of the emergence of loyalty occurred nearly to the same extent. The average loyalty level with 0.75 in the last days equals the results from their model. Only the variance of the loyalty among the buyers is much lower. In my simulation the 5 – percentile is at each point of time at a higher level. At the end of the simulation run it reaches a level of 0.63 instead of 0.45. Contrary to the 5 – percentile, the 95 – percentile is always at a lower level of 0.86 instead of 1.

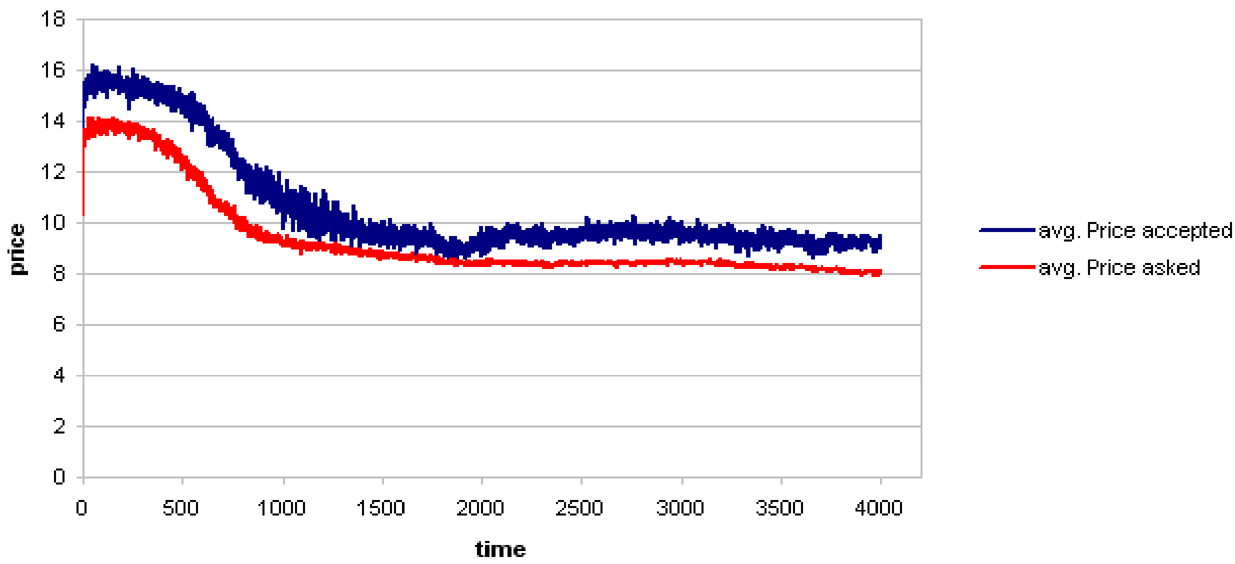


Figure 27: Average prices asked / accepted during the afternoon session.

Figure 27 shows the average asked and accepted prices during the afternoon session. It can be seen that the asked price is lower than the accepted price and remains for the last 2000 time units at a price level slightly above 8. However the accepted price is much more fluctuating and is located at a higher price level (for the last 2000 time units at a price level of 9.45).

Comparing the afternoon session with the morning session the picture is inverted: In the morning session after some time the average price asked lies only slightly above the average price accepted and both prices do not fluctuate very much. Whereas in the afternoon session the average price accepted lies above the average price asked and fluctuates heavily.

Kirman and Vriend (2001) argue that the reason for the strong fluctuation of the prices is the smaller number of meetings in the afternoon session. This implies that each individual price has a bigger influence on the average price series and also a slower learning process occurs.

But three main differences between the findings of Kirman and Vriend (2001) and my results remain for the afternoon session: First, both price series are located at a lower price level in my simulation. Second, the average asked price lies below the average accepted price, whereas in their simulation it is the reverse case. And third, the average asked price does not fluctuate heavily in my simulation.

A possible explanation for the stable average price asked series could be that the number of sellers (10) faces a higher number of buyers (100) and therefore they have a faster learning process than the buyers. But in the simulation of Kirman and Vriend (2001) where they also used the same number of buyers and sellers, they did not produced such a stable average asked price series in the afternoon session.

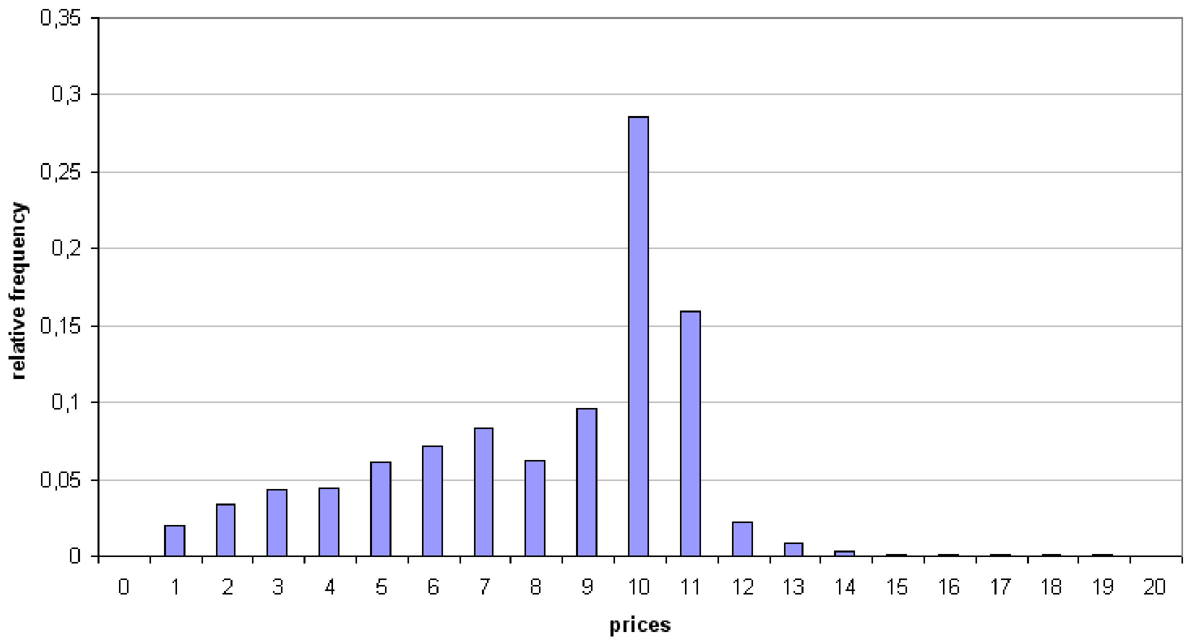


Figure 28: Price distribution during the last 2000 periods.

Figure 28 shows the relative frequency distribution of the paid prices during the last 2000 periods. The most common paid price is 10 (28.6%), the second and third common one is 11 (16%) and 9 (9.6%). Prices paid above a level of 12 and at the level 0 are paid with a frequency below 1%. The median of the paid prices is 8.5 and the average paid price is 8.19.

In contrast to the results of Kirman and Vriend (2001) the modal paid price is one level lower (10 instead of 11) and occurs less frequent (28.6% vs. 49.4%). Prices between 1 and 8 occur with a frequency between 2 and 8 percent in contrast to their findings, where the frequencies are below zero. Looking at prices reaching from 12 to 15 similarities are obtained:

Likewise in the results of Kirman and Vriend (2001), where these prices occur with a frequency of 3.7%, these prices are paid in 3.4% of the transactions in my simulation.

Furthermore it is interesting to point out that also prices above 15 are paid with a frequency of 0.4%, where the buyers achieve a loss (the p^{out} price is 15).

2.4.2 Three Buyer-Types

Configuration of the simulation

The simulation consists of 10 sellers and 100 buyers (number-sellers = 10, number-buyers = 100). There exist three types of buyers: Buyer-type one receives p^{out} = 12, buyer-type two p^{out} = 15 and buyer-type three gets p^{out} = 18. Each simulation runs for 4000 time units (ticks). I have made 40 simulations.

Results

	type of buyer		
	12	**15**	**18**
price accepted	5.59	7.8 (+ 39.6%)	9.19 (+ 64.5%)
price asked	5.91	8.03 (+ 35.9%)	9.04 (+ 52.9%)

Table 10: Prices accepted and received during morning sessions.

Table 10 represents the average accepted and received prices for the three types of buyers from the last 2000 morning sessions. It can be seen that the prices are at different levels depending on the type of buyer: The higher the p^{out}, the higher the accepted / asked price. The values in parentheses represent the percentage change to the buyer-type with a p^{out} = 12.

	type of buyer		
	12	**15**	**18**
avg. loyalty	0.66	0.75 (+ 13.8%)	0.83 (+ 25.8%)

Table 11: Average loyalty during morning sessions.

Table 11 shows the average loyalty during the last 2000 morning sessions for all three buyer-types. The higher the p^{out}, the higher the loyalty. Values in parentheses again represent the percentage change to the buyer-type with p^{out} = 12.

Comparing my results with the results of Kirman and Vriend (2001) it can be seen that the core findings are the same: The higher the p^{out} of a buyer, the higher the accepted / asked price and loyalty.

The results only differ in their levels:

	type of buyer		
	12	**15**	**18**
price accepted			
my results	5.59	7.8 (+ 39.6%)	9.19 (+ 64.5%)
results of Kirman and Vriend (2001)	9.34	9.61 (+ 2.9%)	9.78 (+ 4.8%)
price asked			
my results	5.91	8.03 (+ 35.9%)	9.04 (+ 52.9%)
results from (Kirman and Vriend, 2001	9.41	9.64 (+ 2.5%)	9.82 (+ 4.3%)
avg. loyalty			
my results	0.66	0.75 (+ 13.8%)	0.83 (+ 25.8%)
results of Kirman and Vriend (2001)	0.75	0.82	0.83

Table 12: Differences in the results.

Table 12 summarizes my results and the results of Kirman and Vriend (2001). Comparing both results it is observable that the higher the p^{out}, the lower the differences in the results. In the case of the average loyalty the results coincide at a p^{out} – level of 18.

3. Extension: Information Exchange Influencing Loyalty

3.1 Introduction

This introduction is used to give an overview of the different topics used in the extension of the model of Kirman and Vriend (2001) (see chapter 3.2 The Extension).

There exist many models about loyalty and even more models that describe diffusion processes. In the following, I give an overview of current agent-based models about loyalty (with exception of one analytical approach) and diffusion processes. Furthermore, I explain the term rumour and provide some reasons for the formation of rumours.

3.1.1 Models About Loyalty

There exists a wide set of models that addresses loyalty. In most cases agents have learned to be loyal or to put more importance on loyalty, to be able to increase their own utility.

Zhang and Tanniru (2005) build an agent-based model about virtual learning communities (VLC). They define VLC as a group of people who interactively learn and share about a specific topic among themselves with the help of networking technologies. Agents can ask questions, answer questions, express opinions and comment on opinions about a topic. Each agent consists of an expertise level that is increased if he gets enough satisfactory answers for the question he asked. They use loyalty to determine whether an agent will leave or remain as a member of the VLC. Loyalty is increased if an agent gets high quality answers for an asked question (=improvement in expertise about the topic, referred to as 'intellectual gain') or if he receives many replies for his posted opinion (=number of replying messages, referred to as 'social gain'). Thus, loyalty of an agent is endogenously determined through the behaviour of the community.

Klos and Nooteboom (1997) created an agent-based model in which relations between buyers and supplier are formed on an intermediate-good market. The greater the product differentiation of the created end-product of the buyer, the more profit can be generated by the buyer when reselling the product to his customers. Likewise the suppliers can increase profit by increasing the efficiency of their production process for creating the needed inputs for their buyers. The generated profit of such a buyer – supplier relation is split evenly between the two agents. An agent has to decide with whom he wants to form a relation by calculating for each potential partner the expected profit (expectedProfit = $\text{potentialProfit}^{a}$ * loyalty^{b}). Loyalty is used to calculate the expected profit by serving as the probability that the potential profit will be realized. The authors implemented adaptive agents to give them the possibility to learn (with a classifier system) how important loyalty is, by letting them adapt the value for a (b = 1 – a) to increase profit. The simulation results show that the agents decrease the weight of profitability and increase the weight of loyalty in their partner selection phase, meaning the agents have learned that loyalty is more important to gain high profits than just only considering the potential profitability.

Chang et al. (2008) present an agent-based model with adaptive agents (using classifier systems) about the Taiwanese lumber (wood) market to analyse relationships between customer value (e.g. price, quality of the good), loyalty and profits, especially to explore how customer value changes loyalty and profits. They observed positive correlations among customer value, loyalty and profits. Moreover, simulation results show that higher long-term profits are linked to higher loyalty. Furthermore, they found out that customer value is enough to explain the emergence of loyalty.

Nooteboom (2006) define loyalty as one basis for trustworthiness and likewise he says that trust can be expressed in terms of loyalty. Klose and Nooteboom (2001) created an agent-based model to check if trust in terms of loyalty is viable in markets (more specifically in transaction cost economics, where agents organizes to reduce transaction cost). In their model suppliers and buyers match on the basis of trust, which is based on observed loyalty of the partner, and potential profitability. Agents can act adaptively by putting more or less weight to trust instead of to profitability. This model shows, that trust / loyalty and opportunistic behaviour can be profitable. A further important finding is that trust-relations are a barrier to the evolution of the market towards its optimal configuration (in their model markets are efficient if economics of scale are fully exploited). Also in their model, loyalty emerges endogenously through the behaviour of the agent population by not selecting other partners.

De Francesco (2005) made a classical analytical approach to create a model of matching buyers and sellers: There exist m buyers with imperfect information about previous made choices by the other buyers, each of them want to have one unit of a homogenous good at each stage. Each of the n sellers supplies the same amount of good at an exogenously given price. De Francesco wanted to check if an efficient allocation will be reached (meaning each sellers has m/n buyers as customers) without any coordination. This could only be the case if a norm arises among the buyers, which instructs to remain loyal if a buyer was previously served by the seller and to switch to another seller if he did not get any good. He proves for the two sellers case that the m buyers created such a norm of conditional loyalty.

Vilà (2005) analysis the Bertrand duopoly model by creating an agent-based model that allows also the buyer to have a strategic behaviour when they have to decide from which of the two sellers they want to buy a good. There exists two competing firms, both offering the same homogenous good with the same cost function, and m buyers, which have to decide from which seller they will buy their good. There exists multiple periods, each period consists of R rounds, and for each period the sellers decide the strategy for the price setting. The Bertrand model has two important assumptions, namely, consumers choose the seller with the lowest price and there exists no switching cost. The classical Bertrand model also states that in equilibrium both firms will set their prices to earn zero profit (equal to marginal cost) therewith to gain market share. In his model he gives sellers and buyers the possibility to learn (implemented by a genetic algorithm) their best strategy. One important finding of this simulation is that if both sellers set the same price, a buyer stays loyal by selecting the previous seller again. Another one is that buyers are generally better off when developing a loyalty-strategy because in this case sellers set lower prices.

3.1.2 Agent Based Models About Diffusion Processes

The main characteristic of diffusion process models is that (social) networks act as the facility that allows spreading of entities like information, innovations or diseases. Further, diffusion process models are often agent-based models because ABM offers the possibility to explicitly model the behaviour of the agents and thus also their interaction.

On the one hand research has been conducted to study different kinds of diffusion processes like the diffusion of innovation or the diffusion of knowledge. On the other hand, various studies exist that aim to get to optain insights about the effects of different social networks structures on the diffusion processes.

Švarcová and Švarc (2009) created a model about the diffusion of innovation. In their model agents adapt innovations if the innovation offers them a certain minimum utility. The utility is calculated by a weighted sum of the willingness to use the new product (individual preference) and the social influence (agents neighbours that adopted the innovation). The social influence is determined by the structure of the social network. Furthermore, they analysed how different network structures (random, lattice, ring, small world and scale free network) influence the market penetration and the innovation diffusion process. Among other things they found out that random and scale-free social networks push faster innovation diffusion than other network structures.

Afshar and Asadpour (2010) made an agent-based model about the diffusion of opinions. More precisely, they extended the bounded confidence model from (Deffuant et al., 2001) by introducing informed agents. There exist two agent types. The first category are informed agents who pretend to have an opinion that is similar to the other agents with whom they interact. This should ensure the informed agents being able to influence others. Informed agents look like every other agent so they couldn't be distinguished from other agents, but act coordinated, meaning they try to change gradually the opinion of their neighbours towards a global desired opinion each time they interact with them. The second agent type are the majority. Their opinion is formed by their neighbours (social force) and a force towards their own opinion (self force). Furthermore, the authors analysed the effects of the structure (random, scale-free and small-world structure) of the social network, which defines the neighbourhood of all agents and thus their interaction possibilities. Their results show that an increasing of the inter-connectivity of the agents (increasing the average degree; increasing the average amount of neighbours) decreases the consensus time to establish a desired opinion in the population with the help of informed agents. But in small-world social networks, informed agents failed to establish a desired opinion.

Hui et al. (2010) modelled a general ABM about the diffusion of information (diffusion of warning messages for the evacuations of households) in dynamic networks. The social network is dynamic in terms of removing of agents from the network and thus also changing the degrees (their amount of neighbours) of the other agents during the diffusion process. Agents who believe a warning will evacuate and as a consequence leave the network. Moreover, Hui et al. analysed different network structures and introduced trust levels as weights on the edges between agents to form groups of agents. They also analysed different seeding strategies (assigning a warning message randomly to agents or selecting the nodes with the highest degrees as seeds for the message). Their results show that the proportion of evacuated agents is highest in scale-free networks and lowest in grid networks. Furthermore, the warning diffusion is increased by different trust levels based on social groups (population inhomogeneity).

Perez and Dragicevic (2009) created an ABM for the spread of diseases in an urban space by using geospatial data. Agents can have the health status susceptible, exposed, infected or immune to a certain disease. The model uses spatial information about the municipality Burnaby of Vancouver in Canada to represent urban space where the contact between the agents takes place. At beginning, agents are endowed with a certain home location. Each day, agents commute to work or study buildings (depending on whether they are workers or students). After work or university, some agents start to move to the nearest shopping wall for shopping. All agents uses the public transport to move from one location to another one. Furthermore, infected agents can only spread the disease at work, university or the shopping wall, meaning only at this places interactions between the agents occur. Of course in reality the public transport system is an important and quite effective disease spreading facility when just thinking at the flu season, but this fact is not considered in their model. Infected agents have a certain radius for spreading the disease to some of them when settled at a location like the university. The higher the population density of a certain area (university, shopping wall, …), the higher the probability of becoming infected if an susceptible agent is within the radius of an infected agent. Thus, this model uses demographic data (population density), the network of the public transport and the land use (location of the buildings). Results show that these dynamic spatial interaction between agents (at work, university or shopping wall) lead to a high exposure of healthy agents and thus ill agents are concentrated at places like schools and universities.

3.1.3 Definition Of Rumour

According to Merten (2009) there exists no common standardized definition of rumour in the literature but a wide set of different definitions. All of them have one part in common: the validity of the information content of a rumour is not trustworthy. Allport and Postman (1947, page ix) define a rumour as:

“A specific (or topical) proposition for belief, passed along from person to person, usually by word of mouth, without secure standards for evidence present.”

Furthermore, in their studies of message diffusion they found out that around 70% of the message content are lost if the message is transmitted six times from one person to another one (when reaching the sixth person in the transmission chain).

3.1.4 Origin Of Rumours

Psychological oriented research about rumours sees rumours as answers to individual or collective issues. Whereas the sociological explanation for the formation of rumours is that rumours act as ‘improvised news’ to substitute missing information (e.g. due to census) from conventional information channels like newspapers (Merten, 2009). Allport and Postman (1947) see the emergence of rumours as a collective stress relaxation to clarify an uncertain situation.

Merten (2009) states that the causes for the spreading of rumours are not yet been sufficient clarified. He lists two motives why people spread rumours: First, the possession of actual information increases the social status of a person. The second motive of a person is its ‘inner’ urge to communicate information to others.

3.2 The Extension

This extension is to some degree a combination of an opinion formation model (like Afshar and Asadpour (2010) but without informed agents) and a spatial epidemic spread model: The buyers opinions about sellers (whether a seller is ‘good’ or ‘bad’) are formed by their neighbours in the sellers queue and their own opinions. The spatial aspect of the extension is represented by looking at the location of the buyers in the sellers queue.

Dynamic ‘spatial’ networks are used as the medium to spread information. Each buyer has a dynamic spatial social network. His social network is dynamic, because each time he joins a sellers queue, different buyers are located within this queue which form his social network. Depending on the position of the buyer in the sellers queue and the position in the sellers queue of the buyer who spreads information about a seller, the buyer is more or less strongly influenced (he is strongly influenced, if he is a direct neighbour and he is decreasing strongly influences the greater the distance is to the buyer who spreads information).

The core concept of this extension is, that buyers located in the queue of a seller can spread information about other sellers (telling which one is good or bad in terms of being able to satisfy the needs of the buyer). Each member of a sellers queue can be both, an informant who spreads information and a receiver who receives information from informants of the current queue. The received information about a seller of the previous period influences the seller choosing process of the buyer who received information about sellers, in the actual period. Therefore the model of Kirman and Vriend (2001) is extended by the following points:

Now, *buyers* have the ability to spread positive and negative information about a seller. Therefore, each buyer is endowed by an exogenously given ‘probability to talk’ pT, which specifies the probability of the buyer that he will spread information about a seller. A buyer also consists of a ‘probability to receive’ pR. This value represents the receptivity of a buyer (i.e. how much attention he puts on a received information). Actually, pR is a weight between 0 and 1. If pR = 0, the buyer ignores the received information and if pR = 1, he takes the received information serious. Once a buyer has received information about sellers, he takes them into account the next time when he will have to select a seller. Due to having the possibility to receive positive and negative information about a seller, he might want to put different attentions to these two types of information (positive, negative information). Therefore two types of pR are introduced: pR+ shows how big the receptivity for positive information is and pR- for negative information, respectively.

When a buyer *spreads negative information* about a seller, he (called ‘informant’) looks at his classifier system for choosing sellers and selects the seller with the lowest fitness and spreads negative information (represented by the value -1) about him. To *spread positive information* about a seller, the informant determines his second best (fittest) seller and spreads positive information (represented by the value +1) about him. Because it is obvious for every buyer in the actual sellers’ queue that this actual seller is the ‘best’ one, information about an alternative to the current one might be useful for the other members of the queue.

Buyers also have the ability to *learn the importance to listen to received information*. If the received information in the previous period t-1 was useful (in terms of predicting correctly the shopping experience) and the buyer visited in period t the seller of whom the received information is about, then the according pR is increased , otherwise pR is decreased. The shopping experience specifies whether the buyer was satisfied by having received a good at an acceptable price or not:

```
IF received information about him in t-1 & visited seller in t THEN
{
        IF positive information & positive experience THEN
                increase pR+
        ELSE IF negative information & positive experience THEN
                decrease pR-
        ELSE IF positive information & negative experience THEN
```

decrease pR+

ELSE IF negative information & negative experience THEN

increase pR-

}

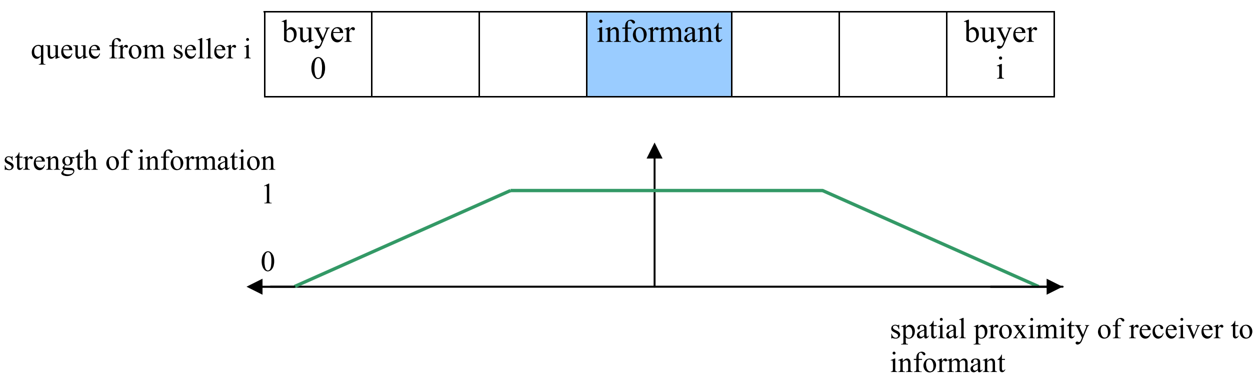
Pseudo Code 3: Learning the importance to listen to information.

The spreading and receiving of information occurs within a sellers queue. Once a buyer of the queue has received a information about a seller from another buyer, this received information influences the buyers' seller-choosing process the next period: The information changes the calculated bid for the according seller i when the buyer chooses the best seller from his point of view:

$$b_i(t) = s_i(t) + \varepsilon + \begin{Bmatrix} pR+ \\ pR- \end{Bmatrix} *\text{received information}_i\,(t) \qquad , \varepsilon \sim N(0, \sigma)$$

$$\text{received information}_i\,(t) = w(pos1,pos2) * \begin{Bmatrix} +1 \\ -1 \end{Bmatrix}$$

The fitness s_i of seller i is stored in the classifier system for choosing a seller in the morning/afternoon of the buyer (the receiver of the information). pR+ / pR- represents how much importance the receiver puts on a received positive / negative information. A spread information has ether a value of 1, if it is a positive information about a seller, or a value of –1 if it is a negative information. The received information is the discounted (by the information strength w) original spread information. w(p1,p2) is the information strength and depends on the spatial proximity of the receiver to the informant in the queue:



w(p1,p2) presents the circumstance that if a buyer starts talking he talks to his direct neighbours within the sellers queue. Other members, which are no direct neighbours of the informant also can listen to him but only catch parts of the told information. The more far away a buyer is in the queue to the informant, the less information he can 'hear'. Therefore, w(p1,p2) is a decreasing function of the strength of information ($w(pos1,pos2) \mapsto [0,1]$):

$$w(pos1, pos2) = \begin{cases} \dfrac{1}{|pos1 - pos2|} & , if\ |\ pos1 - pos2\ | \leq threshold \\ \\ 0 & , else \end{cases}$$

Pos1 is the position of the informant and pos2 the position of the receiver within the sellers queue. The threshold ($threshold \in [2,\infty]$) is exogenously given and determines the minimum strength of information which can be recognized by a buyer. E.g. if a receiver is located in the queue ten positions prior to or after the informant, then he will hardly understand any information of the informant. Once the information strength is too small (|positionOfInformant - positionOfReceiver| > threshold) the information couldn't be heard by the receiver.

The above mentioned spreading of information just considers the case that a buyer can only receive one information about a seller per period. This is done to give a simplified view on the mode of operation of the spreading of information. In fact, the extension also allows the buyers to *receive multiple information messages about the same seller* in one period: If multiple information of the same information-type (only positive or only negative information messages) about the same seller is received multiple times, the buyer uses just the information with the highest information strength. If positive and negative information is received about the same seller in one period, then the single received information are summed up. Formally this can be expressed by the equations below. χ is the evaluation function, which acts according the above mentioned verbal description of receiving multiple information messages about the same seller:

$$b_{i,t} = s_{i,t} + \varepsilon + \begin{Bmatrix} pR+ \\ pR- \end{Bmatrix} * \chi_N(receivedInfo_{i,N}, \chi_{N-1}) \quad , \varepsilon \sim N(0, \sigma) \quad (1)$$

$$\chi_j(receivedInfo_{i,j}, \chi_{j-1}) = \begin{cases} receivedInfo_{i,j}, if: j = 1 \\ \max\{receivedInfo_{i,j}, \chi_{j-1}\}, if: receivedInfo_{i,j} \geq 0 \wedge \chi_{j-1} \geq 0 \\ \min\{receivedInfo_{i,j}, \chi_{j-1}\}, if: receivedInfo_{i,j} < 0 \wedge \chi_{j-1} < 0 \\ receivedInfo_{i,j} + \chi_{j,i}, if: \left|receivedInfo_{i,j} + \chi_{j-1}\right| \leq 1 \\ 1, if: receivedInfo_{i,j} + \chi_{j-1} > 1 \\ -1, if: receivedInfo_{i,j} + \chi_{j-1} < -1 \end{cases} \quad (2)$$

$$receivedInfo_{i,j}(pos1, pos2) = w(pos1, pos2) * \begin{Bmatrix} +1 \\ -1 \end{Bmatrix} \qquad , j = 1,..,N \qquad (3)$$

j is the index for the single received information messags and ranges from 1 to N. $\chi_N(receivedInfo_{i,N}, \chi_{N-1})$ of equation (1) has all received information evaluated and is called rumor. The term $\begin{Bmatrix} pR & + \\ pR & - \end{Bmatrix}$ should represent a choice option, depending on the value of the rumour. If rumour < 0, pR- is chosen and if rumour >= 0, pR+ is chosen. The function receivedInfo(pos1, pos2) also consists of a decision term $\begin{Bmatrix} +1 \\ -1 \end{Bmatrix}$. If a buyer receives a positive message then +1, otherwise –1, is chosen.

The extension also takes into account of a morning and an afternoon session. To give the buyers the possibility to learn the importance of the two types of rumour (positive, negative rumour) once for the morning and once for the afternoon session, for each session a pR+ and a pR- variable exists. So there exist four variables that represent the different attentions a receiver puts on received rumour types, two for each session (called pR+morning, pR-morning, pR+afternoon, pR-afternoon). The received information about a seller are evaluated to create one rumour for the according seller for one session. The single rumours are stored in an array. The position i of the array contains the rumour for the seller i. Each buyer consist of two arrays, one that contains the rumours for the sellers in the morning and one for the afternoon session (rumours-morning, rumours-afternoon). Furthermore, if the buyers have chosen a seller, all rumours of the previous period are discarded.

In my extension of the model, rumours are only able to survive (or at least have the possibility to survive), if they turn out to be true: A buyer incorporates a rumour into the rating of a seller. If it turns out that a received rumour belongs to a seller who is either a seller with the lowest fitness among all sellers or the second fittest one, the next time he has to choose a seller, this rumour is spread again when the buyer is selected to spread positive or negative information (being a informant).

3.3 Results

In this chapter I analyse how information spread influences the loyalty of the buyers by comparing the simulation results of the extension with the original model.

3.3.1 Configuration of the simulations

Every simulation consists of 10 sellers and 100 buyers (number-sellers = 10, number-buyers = 100). All buyers receive the same p^{out} = 15 (buyer-types = 1). The configuration of the extension is summarized in table 13:

Parameter	Value
pT	0.1
pR+, pR-	0.1
threshold	5

Table 13: Default-Configuration of the simulation.

Each simulation runs for 4000 time units (ticks). To generate sufficient data for further analysis I have made 40 simulations.

3.3.2 Only Positive Information Spread

In this simulation, buyers are only allowed to spread positive information about sellers. The results of the simulation are compared to the one of the original model.

Results

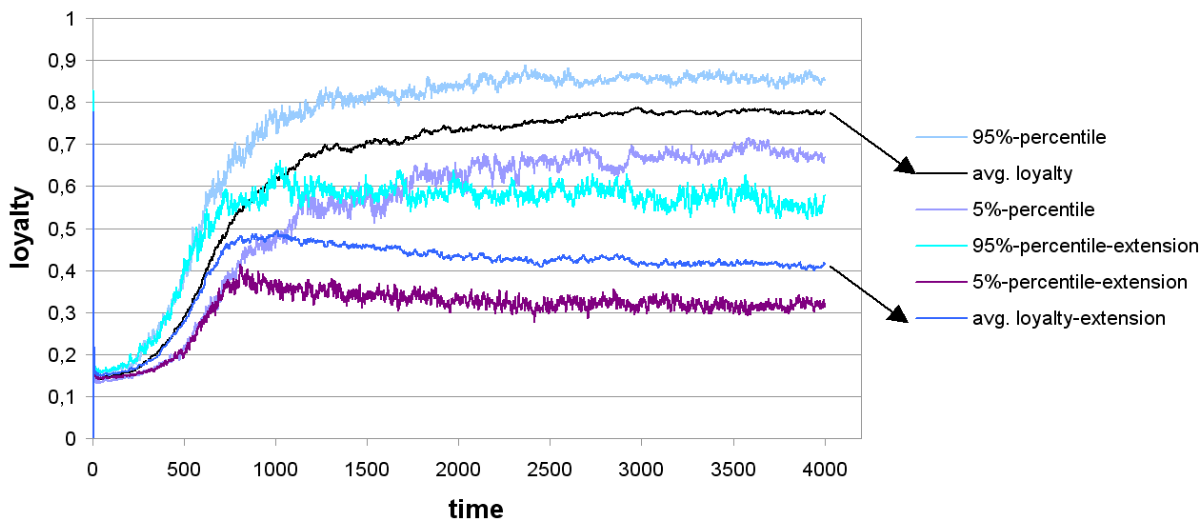


Figure 29: Comparison of the loyalties from the original model and the extended model.

Figure 29 shows the results of this simulation compared with the simulation results of the original model. Despite having set only a relative low probability to talk (pT=10%), differences in the loyalty can be seen: At first, the speed of the emergence of loyalty is the same compared to the original model. But after 600 periods, the loyalty levels begin to diverge. Then, while the loyalty of the original model further increases, the loyalty of the extension turns into a loyalty-level between 0.4 and 0.5. In the last 1000 periods an average loyalty of 0.41 is reached, which is nearly the half of the reached loyalty level of the original model (0.78).

Based on these observations, the effect of allowing positive information spread within the model, is a reduction of the loyalty. The reason therefore is, that buyers get alternative satisfying sellers recommended. And if for example looking at the case where a buyer has two sellers, one of them just being only slightly 'fitter' than the other one, a received recommendation of the less fit seller leads to a selection of him in the next period. This behaviour of course reduces the loyalty and never let establish a high loyalty level.

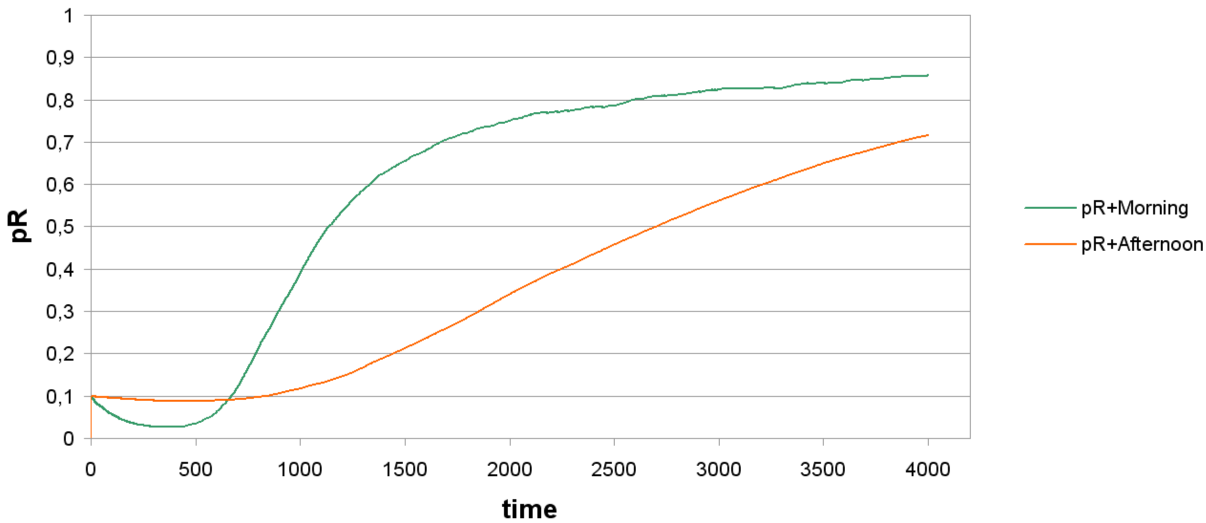


Figure 30: Average learned importance for received positive information.

Figure 30 shows how much attention buyers put on received positive information, in the morning session and in the afternoon session. Looking at the morning session (pR+Morning), it can be seen that in the first 400 periods a reduction of the importance of receiving positive information occurs. Then, pR+Morning increases up to a level of 0.86. In the afternoon session pR+Afternoon faces only a slight decrease in the first 600 periods, followed by an increased up to 0.72. The picture shows that buyers learn faster the importance of listen to positive information for the morning session.

3.3.3 Only Negative Information Spread

In this simulation, buyers are only allowed to spread negative information about sellers. The results of the simulation are compared to the ones of the original model.

Results

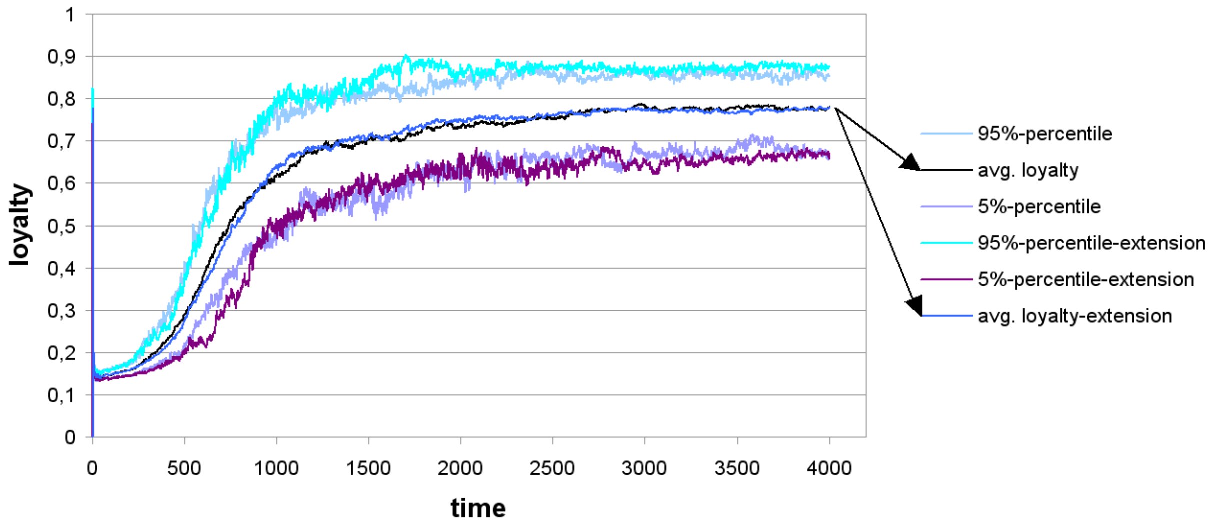


Figure 31: Comparison of the loyalties from the original model and the extended model.

Figure 31 shows the results of the simulation of the extended model compared with the simulation results of the original model. The results of both models coincides, which implies that the effect of negative information spread is zero. Allowing negative information spread neither changes the average loyalty of the buyers nor the speed of the emergence of loyalty.

Furthermore, figure 32 shows that buyers hardly change their attention to negative information, neither in the morning session nor in the afternoon session. The reason therefore is, that buyers only update their attention-level (pR-Morning, pR-Afternoon) if they visit the according seller (about whom the buyer has received a negative rumour) the next period. In doing so, they gain experience about the seller and can check the accuracy of the received negative rumour. But this happens (being able to verify a negative rumour by visiting the according seller the next period) within one example simulation run, less than 800 times among all buyers. Looking more closely at figure 32, it can be seen that buyers put slightly more importance on negative rumours in the morning and decrease the importance of received negative rumours in the afternoon (0.1036 vs. 0.095 in the last periods) (see figure 33).

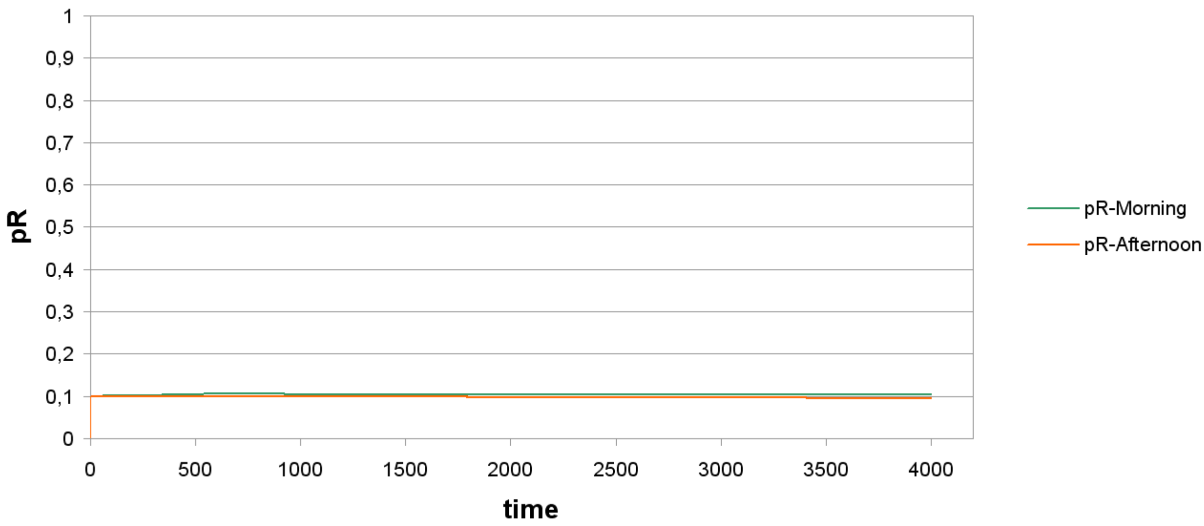


Figure 32: Average learned importance for received negative information.

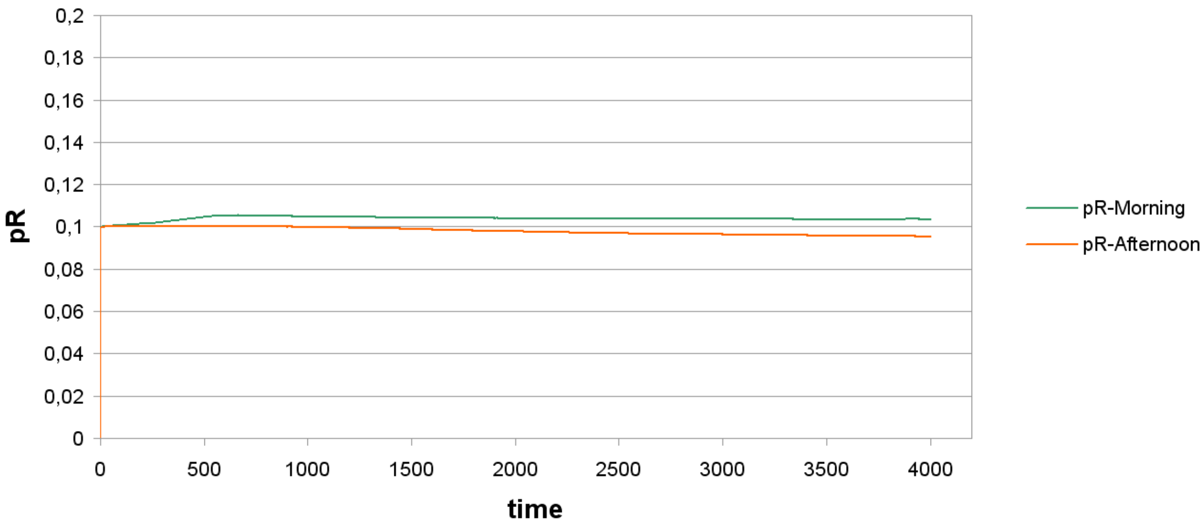


Figure 33: Average learned importance for received negative information.

3.3.4 Positive and Negative Information Spread

In the previous two chapters, I have analysed the single effect of spreading only positive information and the effect of spreading only negative information, respectively. In this chapter both types of information spreading are allowed and their overall effects are analysed. Again the probability for spreading information pT is set to 0.1, which means that each buyer has a probability of 10% to spread positive information and also has a probability of 10% to spread negative information.

Results

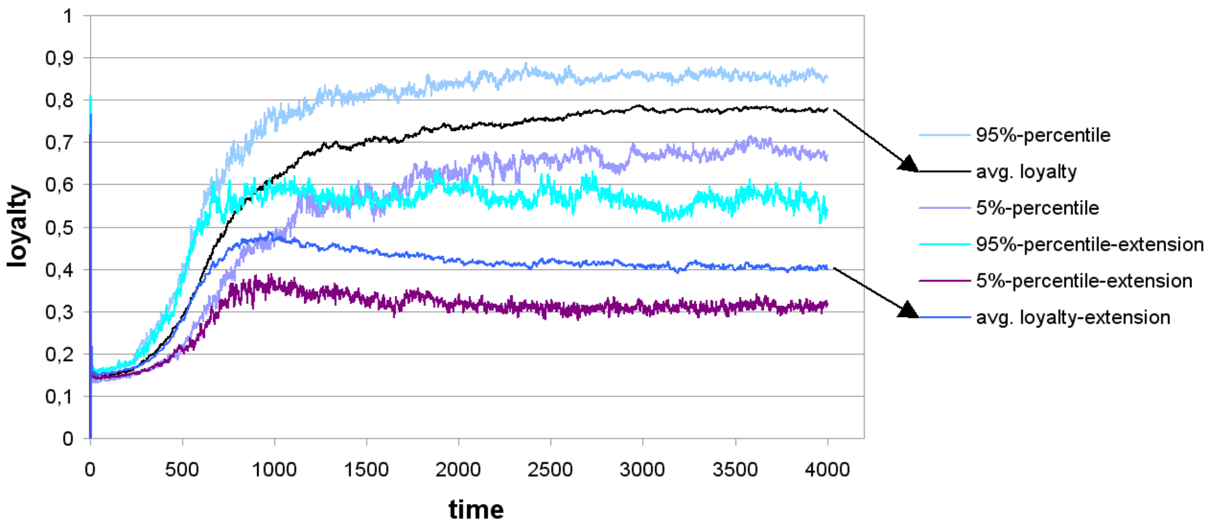


Figure 34: Comparison of the loyalties from the original model and the extended model.

Figure 34 shows the results with regard to the emerged loyalty of the simulation of the extended model, where each buyer has the possibility to spread positive and negative information and compares them with the simulation results of the original model. Allowing both types of information spread yields the same results as in the case where the buyers are allowed to spread only positive information (see chapter 3.3.2).

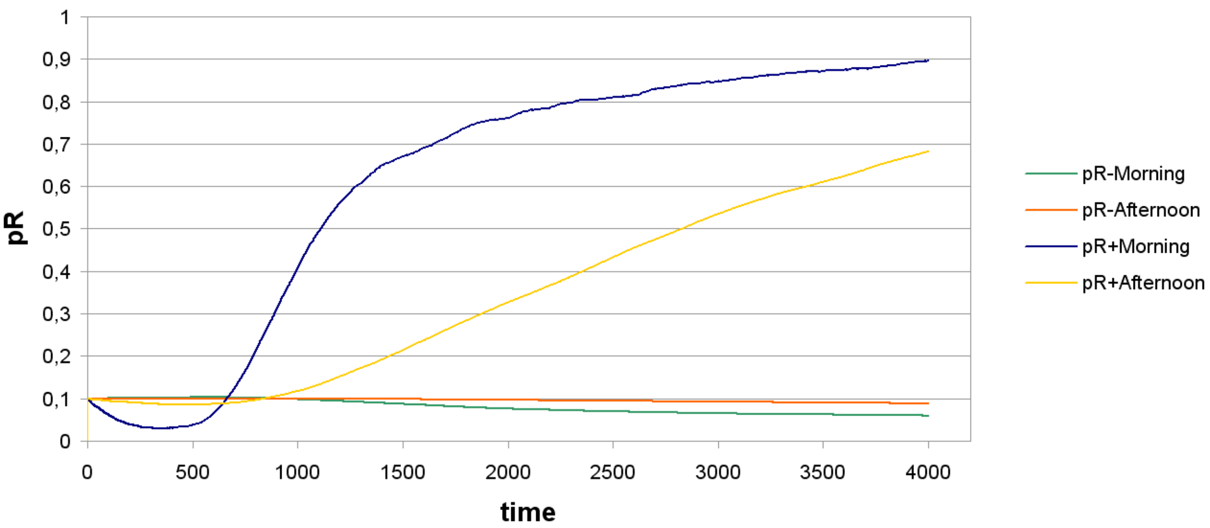


Figure 35: Average learned importance for received negative / positive information.

Figure 35 shows that buyers have increased the importance of listening to positive information with the same speed as in the case where only positive information spread is allowed. Small differences occur when looking at the reached level of pR at the end of the simulation. Values followed in brackets represent the according level of pR reached in the according simulation setup of the previous two chapters (only negative / only positive information spread): In the actual model setup, pR+Morning is 0.89 (0.86) and pR+Afternoon is 0.68 (0.72). Furthermore, pR-Morning is 0.06 (0.1036) and pR-Afternoon is 0.089 (0.095). This results show that the gap between the importance of positive rumour in the morning and afternoon as well as the gap between negative rumour in the morning and afternoon, has increased. Finally, I have to emphasis that there is no effect of the negative information spread on the emerged level of loyalty. The only factor influencing the loyalty-level are positive information spreads.

4. Conclusion

This thesis replicates an agent-based model about price dispersion and loyalty. In this model of Kirman and Vriend (2001), loyalty emerges as an endogenous social network, based on the single interactions between buyers and sellers. It shows that the loyal behaviour of the buyers is a result of a learning process. Furthermore, sellers learn to treat loyal buyers preferentially. Both, buyers and sellers, adapt their behaviour in such a way that establishes a high level of loyalty because they get a higher utility in doing so.

The published information about this model was detailed enough to replicate the model. Especially the published pseudo code turned out to be very useful when it comes to detailed questions about the concrete implementation of the model in NetLogo. Furthermore, two errors were found in their paper: A wrong specified random error term and an inconsistent description of how often reinforcement learning occurs per day when comparing their verbal description of the model with their pseudo code. Comparing my obtained results with the results of Kirman and Vriend (2001) 'relational alignment' occurred in most cases: Looking at the one-buyer case, the emergence of the loyalty occurred to the same extend like in their model. The average asked and accepted prices in the morning session hold relational alignments to the results of the original model. Only the prices asked and accepted during the afternoon session are different, because in my simulation the average asked price is lower than the average accepted price, whereas it is the reverse case in the simulation of Kirman and Vriend (2001). The three-buyers case shows similar relations of the accepted prices, the asked prices and the average loyalty as the results from Kirman and Vriend (2001): The higher the price for reselling the fish outside the market (p^{out}), the higher the accepted / asked prices and loyalty.

As an extension of the original model, information exchange between buyers has been introduced. Now, buyers have the possibility to spread positive and negative rumours about other sellers and are able to learn how important it is for them to listen to these rumours when incorporating them into their sellers choosing process. Simulation results show that the spreading of positive information about other sellers reduces the emerged level of loyalty dramatically, whereas negative information seems to have no effect. Furthermore, buyers put more importance to positive information. The importance of negative information remains nearly unchanged.

For further researches it would be interesting to study the effects on loyalty of different structures of social networks (e.g. small world, grid) which are used by the buyers as a medium for spreading rumours about sellers. Another interesting extension of the model could be to allow endogenous friendships between sellers. If for example a seller has not enough fish because he has too many buyers in his queue, he could recommend not served buyers to join the queue of a friend. Friendships could arise or become stronger the more utility is offered by this friendship-relation in terms of having buyers in the queue, whom are told from the friend of the seller to move to him.

Appendix

Reinforcement learning of buyers and sellers.

```
procedure: reinforcement-sellers {
for all sellers do
{
        scaledRevenue = revenueOfDay / (max_received_price_of_day * sold_supply)
        netProfit of the day = revenueOfDay – (sold_supply * p_in)
        normalizedNetProfit = normalize netProfit with last netProfits of 200 days to [0,1]

        // updating strength (=fitness) of activated supply level rule.
        activated supply level rule → fitness = 0.95*fitness + 0.05* normalizedNetProfit
        // updating strength of treatment parameter b.
        activated treatment parameter rule → fitness = 0.95*fitness + 0.05*scaledRevenue

        for all morning seller-pricerules do
        {
                // updating fitness of morning price rules.
                reward = (timesAccepted * priceToAsk) / (timesActive *
                        max_received_price_of_day)
                fitness = fitness*0.95 + 0.05*reward
        }
        for all afternoon seller-pricerules do
        {
                // updating fitness of afternoon price rules.
                reward = (timesAccepted * priceToAsk) / (timesActive *
                        max_received_price_of_day)
```

```
                fitness = fitness*0.95 + 0.05*reward
        }
}}

procedure: reinforcement-buyers {
IF transaction occurred in morning THEN
{
        utility = (p_out – priceReceived) / p_out
        reward = max(0, utility)
        // updating fitness of seller-choice in morning.
        fitness of visited seller in morning= 0.95*fitness + 0.05*reward
        // updating fitness of morning buyer pricerule.
        fitness of activated buyer-pricerule for morning = 0.95*fitness + 0.05*utility
}
IF rejected in morning & transaction occurred in afternoon THEN
{
        utility = (p_out – pricePaidAfternoon) / p_out
        utilityOffered = (p_out – priceReceivedMorning ) / p_out
        reward = max(0, utilityOffered)
        // updating fitness of seller-choice in morning.
        fitness of visited seller in morning = 0.95*fitness + 0.05*reward
        // updating fitness of morning buyer pricerule.
        reward = max(0, utility)
        fitness of activated buyer-pricerule for morning = 0.95*fitness + 0.05*reward
        // updating fitness of seller-choice in afternoon.
        fitness of visited seller in afternoon = 0.95*fitness + 0.05*reward
        // updating fitness of afternoon buyer pricerule
        fitness of activated buyer-pricerule for afternoon = 0.95*fitness + 0.05*utility
}
IF rejected in morning & rejected in afternoon THEN
{
        utilityOffered = (p_out – priceReceivedMorning ) / p_out
        reward = max(0, utilityOffered)
        // updating fitness of seller-choice in morning.
        fitness of visited seller in morning = 0.95*fitness + 0.05*reward
        // updating fitness of morning buyer pricerule.
        fitness of activated buyer-pricerule for morning = 0.95*fitness
        // updating fitness of seller-choice in afternoon.
        fitness of visited seller in afternoon = 0.95*fitness
        // updating fitness of afternoon buyer pricerule.
        fitness of activated buyer-pricerule for afternoon = 0.95*fitness
}
IF rejected in morning & late in afternoon THEN
{
        utilityOffered = (p_out – priceReceivedMorning) / p_out
        reward = max(0, utilityOffered)
        // updating fitness of seller-choice in morning.
        fitness of visited seller in morning = 0.95*fitness + 0.05*reward
        // updating fitness of morning buyer pricerule.
        fitness of activated buyer-pricerule for morning = 0.95*fitness
        // updating fitness of seller-choice in afternoon.
```

```
        fitness of visited seller in afternoon = 0.95*fitness
}
IF late in morning & transaction occurred in afternoon THEN
{
        utility = (p_out – priceReceivedAfternoon) / p_out
        reward = max(0, utility)
        // updating fitness of seller-choice in morning.
        fitness of visited seller in morning = 0.95*fitness
        // updating fitness of seller-choice in afternoon.
        fitness of visited seller in afternoon = 0.95*fitness + 0.05*reward
        // updating fitness of afternoon buyer pricerule.
        fitness of activated buyer-pricerule for afternoon = 0.95*fitness + 0.05*utility
}
IF late in morning & rejected in afternoon THEN
{
        utilityOffered = (p_out – priceReceivedAfternoon) / p_out
        reward = max(0, utilityOffered)
        // updating fitness of seller-choice in morning.
        fitness of visited seller in morning = 0.95*fitness
        // updating fitness of seller-choice in afternoon.
        fitness of visited seller in afternoon = 0.95*fitness + 0.05*reward
        // updating fitness of afternoon buyer pricerule.
        fitness of activated buyer-pricerule for afternoon = 0.95*fitness
}
IF late in morning & late in afternoon THEN
{
        // updating fitness of seller-choice in morning.
        fitness of visited seller in morning = 0.95*fitness
        // updating fitness of seller-choice in afternoon.
        fitness of visited seller in afternoon = 0.95*fitness
}}
```

References

Afshar, M. and Asadpour, M. (2010). *Opinion Formation by Informed Agents*. Journal of Artificial Societies and Social Simulation. Vol. 13. No. 4.

Allport, G. W. and Postman, L. J. (1947). *The psychology of rumor*. Henry Holt and Co. New York.

Arthur,W.B. (1994). *Inductive reasoning and bounded rationality*. American Economic Review, Papers and Proceedings 84, 406–411.

Ashlock, D., Smucker, M.D., Stanley, E.A. and Tesfatsion, L. (1996). *Preferential partner selection in an evolutionary study of prisoner's dilemma*. BioSystems 37, 99–125.

Axelrod, R. (2003). *Advancing the Art of Simulation in the Social Sciences*. Japanese Journal for Management Information System: Special Issue on Agent-Based Modeling. Vol. 12. No. 3.

Bonabeau, E. (2002). *Agent-based modeling: Methods and techniques for simulating human systems*. Proceedings of the National Academy of Sciences.

Chang, T., Lee, J. and Chen, R. (2008). *The Effects of Customer Value on Loyalty and Profits in a Dynamic Competitive Market*. Computational Economics. Springer. Vol. 32. No. 3. 317-339.

Chang, M.-H. and Harrington Jr., J.E. (2005). *Discovery and diffusion of knowledge in an endogenous social network*. American Journal of Sociology 110, 937–976.

Gerdes, I., Klawonn, F. and Kruse, R. (2004). *Evolutionäre Algorithmen – Genetische Algorithmen – Strategien und Optimierungsverfahren – Beispielanwendungen*. Vieweg Verlag. 1th Edition.

Deffuant, G., Neau, D., Amblard, F. and Weisbuch, G. (2001). *Mixing beliefs among interacting agents*. Advances in Complex Systems. Vol. 3. 87-98.

De Francesco, M. A. (2005). *Matching buyers and sellers*. Economic Bulletin. Vol. 3. No. 33. 1-10.

Epstein, J.M. and Axtell, R. (1996). *Growing Artificial Societies: Social Science from the Bottom Up*. Brookings/MIT Press, Washington, DC.

Gilbert, N. (2007). *Agent-Based Models*. Sage Publications. London.

Gilbert, N. and Terna, P. (2000). *How to build and use agent-based models in social science*. Mind & Society. Vol 1. 57-72.

Gilbert, N. and Troitzsch, K.G. (2005). *Simulation for the Social Scientist*. Open University Press. 2nd Edition.

González, M. C. (2006). *Contact Networks of Mobile Agents and Spreading Dynamics*. Dissertation. University Stuttgart.

Hamming, R. W. (1950). *Error-detecting and error-correcting codes*. The Bell System Technical Journal. Vol. 16. No 2. 147 – 160.

Hanaki, N., Peterhansl, A., Dodds, P.S. and Watts, D.J. (2004). *Cooperation in evolving social networks.* (mimeo).

Hui, C., Goldberg and M., Magdon-Ismail, M., Wallace, A. W. (2010). *Agent-based simulation of the diffusion of warnings*. Proceedings of the 2010 Spring Simulation Multiconference (SpringSim '10). Article 9.

Jackson, M.O. and Rogers, B.W. (2004). *Search and the strategic formation of large networks: When and why do we see power laws and small worlds?*. (mimeo).

Kaelbling, L. P., Littman, M.L. and Moore, A. W. (1996). *Reinforcement Learning: A Survey*. Journal of Artificial Intelligence Research. Vol. 4. 237-285.

Kirman, A. P. and Vriend, N. J. (2001). *Evolving market structure: An ACE model of price dispersion and loyalty*. Journal of Economic Dynamics & Control. Vol. 25. 459-502.

Klos, T. B. and Nooteboom, B. (1997). *Adaptive Governance: The Role of Loyalty*. University of Groningen. Research Institute SOM. Research Report 97B53.

Klos, T. B. and Nooteboom, B. (2001). *Agent-based computational transaction cost economics*. Journal of Economic Dynamics & Control. Vol. 25. 503 – 526.

Knoke, D. and Yang, S. (2008). Social Network Analysis. SAGE Publications. 2nd Edition.

Merten, K. (2009). *Zur Theorie des Gerüchts*. Publizistik. Vol. 54. No. 1. 15-42.

Nikolai, C. and Madey, G. (2009). *Tools of the Trade: A Survey of Various Agent Based Modeling Platforms*. Journal of Artificial Societies and Social Simulation. Vol. 12. No. 2.

Nooteboom, B. (2006). *Human Nature in the Adaption of Trust*. Discussion Paper 2006-37. Tilburg University. Center for Economic Research.

Perez, L. and Dragicevic, S. (2009). *An agent-based approach for modeling dynamics of contagious disease spread*. International Journal of Health Geographics. Vol. 8. No. 1.

Réka, A. and Barabási, A.-L. (2002). *Statistical Mechanics of Complex Networks*. Reviews of Modern Physics 74. 47-97.

Riolo, R.L. (1997). *The effects and evolution of tag-mediated selection of partners in populations playing the iterated prisoner's dilemma*. In: Bäck, Th. (Ed.), Proceedings of the 7th International Conference on Genetic Algorithms. Morgan Kaufmann, pp. 378–385.

Schelling, T.C. (1971). *Dynamic models of segregation*. Journal of Mathematical Sociology 1 (2), 143–186.

Švarcová, N. and Švarc P. (2009). *Diffusion Processes on Complex Networks*. Charles University Praque. Faculty of Social Sciences. Institute of Economic Studies. Working Papers IES 2009/27.

Tesfatsion, L. and Kenneth, L. J. (2006). *Handbook of Computational Economics. Agent-Based Computational Economic*. Vol 2.

Vilà, X. (2005). *Consumers' Behaviour and the Bertrand Paradox: An ACE approach*. Unitat de Fonaments de l'Anàlisi Econòmica (UAB) and Institut d'Anàlisi Econòmica (CSIC). UFAE and IAE Working Papers 654.05.

Vriend, N.J. (1995). *Self-organization of markets: an example of a computational approach*. Computational Economics 8 (3), 205–231.

Wasserman, S. and Faust, K. (1994). *Social Network Analysis. Methods and Applications.* Cambridge University Press.

Wilensky, U. and Rand, W. (2007). *Making Models Match: Replicating an Agent-Based Model*. Journal of Artificial Societies and Social Simulation. Vol. 10. No. 4 2.

Zhang, Y. and Tanniru, M. (2005). *An Agent-based Approach to Study Virtual Learning Communities*. Proceedings of the 38th Annual Hawaii International Conference on System Sciences. Track 1.