Decision making for two learning agents acting like human agents*

*A proof of concept for the application of a Learning Classifier Systems

1st Tobias Jordan
School of Computing
University of Kent
Canterbury, United Kingdom
tj202@kent.ac.uk

2nd Philippe de Wilde
School of Computing
University of Kent
Canterbury, United Kingdom
P.Dewilde@kent.ac.uk

3rd Fernando Buarque de Lima Neto
School of Computer Sciences
Universidade de Pernambuco
Recife, Brazil
fbln@ecomp.poli.br

***Abstract*—The paper investigates the suitability of a Learning Classifier System (LCS) implementation for mimicking human decision making in agent based social simulations incorporating network effects. Model behavior is studied for three distinct scenario settings. We provide proof of concept for the adequacy of LCS to tackle the task at hand. Specifically, it is found that the LCS provides the agents within the simulation model with the ability to learn and to react to environmental changes while accounting for bounded rational decision making and the presence of imperfect information, as well as network effects. Moreover, it can be shown that the LCS-agents exhibit a habit like behavioural pattern.**

***Index Terms*—Agent Based Social Simulation, Learning Classifier Systems**

I. Introduction

Currently, General Equilibrium Models [1] represent the most popular paradigm for macroeconomic simulation and thereby the most popular measure for political decision support. However, those models are based on strong neo-classical assumptions like rational decision making, perfect market behavior and perfect information for all actors. These assumptions do obviously not hold in the real world and lead to a stereotype average consumer, that is the rational individual or Homo Oeoconomicus. Critics on Homo Oeconomicus became louder during the last decade due to the unrealistic assumptions of the underlying model and the recent failure of rational individual based models [2]. These assumptions also suppose that our highly heterogeneous societies can be understood by investigating the behavior of rational average individuals and their communication and group behavior. We argue against that irrationality does not exist, or at least not affect crowd behavior [3]. In order to better understand and predict human behavior, the concept of Agent Based Modeling came up as an alternative for economists. Agent Based Models (ABM) use autonomous acting, communicating computer programs, the so called agents that are able to decide in a bounded rational way [4]. Agents within these models may resemble individual, consumers or juristic persons like companies. ABM thereby are enabled, to better model human heterogeneity and thus create a more sophisticated image of reality. Complementary, the research area of Social Network Science and Complex Networks suggests that human decisions are not entirely autonomous, but influenced by peers, siblings or parents [5]. This influence may occur through spread of information or contagion of behavior via social networks. The former foils the assumption of perfect information, the latter challenges fully rational decisions. This motivates the attempt to join findings from Social Network Science and Agent Based Modeling in order to create models that better represent reality, facilitating simulation of societies and prediction of policy effects. In order to set-up a simulation model that addresses the stated shortcomings of state of the art General Equilibrium Models and copes with opinion dynamics in social networks, the agents within the models need to be equipped with an adequate decision making mechanism. Such a mechanism may approximate human decision making in the situation under investigation, enhancing the credibility and accuracy of the model. Moreover, the mechanism must be capable of coping with a dynamic environment. The research at hand proposes such a decision mechanism for ABM, incorporating network diffusion processes. In an early work, Holland proposes Learning Classifier Systems (LCS) as a good option to mimic human decision making in ABM. Principally he argues in favor of LCS because they enable the agent to allocate environmental situations to broad categories which are progressively refined by the experience made. This in turn enables the agent to build internal models of the world, while none of the models is immutable, but always provisional and subject to change [6]. Further, Classifier Systems have been shown to be able to learn to play nash-markov equilibria both with and without the presence of imitation [7] [8]. LCS also tend to the formation of niches within the environment, i.e.different rules within the population can match different parts of the search space. This is not only advantageous for the purpose of searching but also represents a certain level of abstraction typical for the human mind. Therefore, a niche- and strength based LCS is implemented in order to make allowance for the often posited characterization of the human mind as a system to classify things and situations. This work

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

shall serve as a proof of concept for the utilization of LCS as an agent learning representation in ABM. The presented paper is part of a broader study, dealing with the simulation of schooling decisions of children. Therefore the decision weather to dedicate or not at school serves as a use case for this work. An important determinant of schooling success is the dedication of children to their studies. As a motivation for this dedication serves the question if education pays off or not (expected utility). As schooling success depends on a large number of influence factors, such as socio-economic status, peer influence and current economic activity, we assume that children cannot assess that expected utility but rather base their decision on experience and peer information. Moreover, subjective perception, limited processing capacities and incomplete information may influence expected utility calculation of individuals. Due to the need of the considered individuals to map their own utility function and particularly assess the magnitude of payoff that an action provides, the applied LCS is strength-based and not accuracy-based as other recent LCS implementations.

II. Background

This Section gives a general overview on recent advances in the fields important to the presented research, namely Agent Based Computational Economics and Learning Classifier Systems.

A. Agent Based Models (ABM) - Agent Based Economics

According to Holland [9], ABM describes the study of systems consisting of autonomous computational agents. The agents may be designed heterogeneously and are able to interact, which enables the ABMs to reproduce macro phenomena that emerge from micro level behavior. Examples for the use of ABM are models of racial segregation [10], political opinion building [11] or consumer behavior [12]. Agent Based Social Simulation (ABSS)a [13] [14] and Agent Based Computational Economics (ABC) [15] are among the sub-fields of this area, where applications reach from demography [16] to tax compliance [17] or school effectiveness [18]. Using ABM to simulate social or economic contexts forces the researcher to debug and understand macro phenomena better, while large experimental studies may be conducted without numerical or ethical concerns arising in real world experimental setups. Contrary to traditional economic models, ABM enables the researcher to incorporate the imperfection of human rationality as well as limited information availability to the model. In addition, the iterative interaction of agents triggers insights that may be overseen in general equilibrium approaches. Literature on ABC suggest very distinct approaches to model agent decision making.

B. Learning Classifier Systems (LCS)

LCS are rule based programs. They usually contain a Genetic Algorithm to manipulate the set of rules they operate on and a Reinforcement Learning part that aims at choosing the best performing rules [19]. Holland proposed LCS first as a model of the emergence of cognition [20]. Classifier Systems are regarded as an approximation to human decision making, given a perceived situation [21] although they are not belief based, which means that agents are not conscious about the existence of other agents within their environment [22]. According to Brenner [21], Classifier Systems consist of a set of condition-action rules, where the conditions $\overline{c}$ describing the perceived state and the actions $\overline{a}$, representing the respective action to be taken are stored as feature strings of the form $\{c_1, c_2, ..., c_n\}$ or respectively $\{a_1, a_2, ..., a_n\}$. The set of condition - action rules $R_i i = 1, 2, ..., n$ combines then respectively condition strings with action strings. Whereas c_{ij} or a_{ij} may be represented as a wild-card $\#$ indicating that the respective feature applies independently from the given situation. For each iteration, the current signal $E = \{s_1, s_2, ..., s_n\}$ is compared to the condition strings of the available condition - action rules. The most adequate of those rules with corresponding condition strings is being chosen for execution. Literature discriminates between Strength-based and accuracy-based LCS [23]. In Strength-based LCS, for the purpose of choice, each rule is being assigned a *Specificity* value and a *Strength* value. The *Specificity* is determined by the number of wild-cards within the rule, while the *Strength* is defined by the pay-off, the rule generated in preceding iterations. The value $B(R_i)$ is calculated according to Equation 1, where α, β and γ are parameters. Accordingly, the corresponding rule with the maximum value of $B(R_i)$ is regarded the most adequate rule.

$$B(R_i) = \alpha(\beta + \gamma \cdot Specificity(R_i)) * Strength(t, R_i) \quad (1)$$

The *Strength* of each Rule R_i at time t is hereby calculated according to Equation 2.

$$Strength(t+1, R_i) = Strength(t, R_i) + Payoff(t) - B(R_i) \quad (2)$$

Subsequently, the Classifier System employs a genetic operator that allows for creating new rules from the existing best performing rules and forgetting rules that did not perform well in the past. Here the system may either employ panmictic- or niche-based Rule Discovery [23].

III. Problem

The agents within the presented simulation model are embedded in an environment consisting of their peers[1] and an individual socio-economic environment represented by individual variables. We aim at modeling the behavior "dedication at school" which cannot be observed easily. Hence we employ the mark in mathematics of the respective pupil as a proxy for the engagement at school. The agents within the model iteratively decide what mark to achieve in the next iteration. It is assumed that agents benefit from aligning their behavior with peer behavior. Thus, an agent's utility is affected by the behavior her peers exhibit. Both, individual socio-economic status and peer social-economic status hereby affect the utility. Moreover, the agents are unaware of their own

[1] for the use case of this work, peers are thought of as friends within the friendship network of pupils

utility function and hence have to learn which action pleases them most.Perceptions(or Signals) are represented as condition strings E of the form $\{s, p_1, p_2, ...p_n\}$, where s stands for the mark of the current individual and p_i stands for the mark of peer i. Subsequently, we explain, how those perceived condition strings are processed in the decision module set up as a Classifier System. In every case, the agent decides on a set of actions, that may include all possible marks within the range $[0, 100]$.

IV. The LCS Decision Mechanism

The classifier is based on a set of condition-action-rules R of the form $\overline{c} \longrightarrow \overline{a}$, where each $\overline{c}$ represents a condition string $c_1, c_2, ...c_n$. Respectively, $\overline{a}$ represents the action to be taken if the rule is selected. In the given scenario $\overline{c}$ contains the mark in mathematics of the respective agent as well as the current mark of her peer. Accordingly, the action $\overline{a}$ may be any mark between 0 and 100 that the agent will achieve in the subsequent iteration. The length n of $\overline{c}$ is given by the formula $n = d+1$, where d denotes the degree of the respective agent. c_i stands for the interval $[x_i, y_i]$ with $x_i, y_i \in [0, 100], y_i \geq x_i$ but can adopt the # symbol also, indicating that this digit of the condition string matches all possible values of s or p_i respectively. The first digit of $\overline{c}$ narrows the mark of the respective agent, while the remaining digits narrow the mark of her peers. For example, one $\overline{c}$ may be $[0, 10], [80, 100]$. this condition would for instance match a situation where agent 1 achieves the mark 7 and agent 2 achieves the mark 90. with $\overline{a} = 56$, agent1 would change her mark for the next iteration to 56. At each time step, the algorithm creates the list of matching condition-action-strings M_i. M_i contains those strings for which the condition $\forall x \in E, x_i \in c_i$ holds. To setup the system, a number of condition-action-rules is created randomly. Here for each rule to be created, a random interval is set for each digit of the condition-string. The respective action of the condition-action-string is then drawn from a normal distribution with variance $VAR(x)_1$, while the mean is set to the initial mark of the respective agent. As posited before, we implement a Strength-based system. Calculation of $Strength$ and $B(R_i)$ occurs according to Equation 2 and Equation 1 respectively for all $R_i \in M_i$. Subsequently, a roulette wheel mechanism ensures that the action of that R_i with the highest $Strength$ is most likely to be taken, while the likelihood for the selection of $R_i \in M$ decreases with decreasing $Strength$. If R does not contain any rule that is compatible to the current perception string - meaning that $M_i = \emptyset$ -, that rule in R that is most similar to the current perception E mutates so that it matches E. Hereby the action of the mutated string is also drawn from a normal distribution where the mean is the currently performed mark of the agent and variance is $VAR(x)_3$. Furthermore,for the purpose of Rule Development an evolutionary process is implemented, aiming at continuous improvement of the solutions found. In order to avoid the generation of inadequate rules by the combination of rules from very different areas of the search space, we employ a niche-based approach. Hereby a fraction of the weakest rules $death - rate$ in M_i is being deleted from R and new rules are created, recombining the n strongest rules in M via a cross-over operator until the original number of rules in R is reached. In order to ensure diversity, an additional mutation operator is introduced: A random mutation process starts with a probability of $mutation - rate$, altering random characters of the condition string of a randomly chosen rule $R_i \in M_i$ that is not the currently best performing rule. The character that indicates the action of the condition-action-string to be mutated is drawn from a normal distribution with variance $VAR(x)_2$ while the mean is set to the currently adopted mark of the respective agent. Figure 1 illustrates this Classifier System for the simple case of an agent with degree 2.

A. Evaluate Action

The evaluation of the fitness or utility, an action taken by the agent causes, is being measured by a utility function. The utility function proposed in [24] is implemented as presented in Equation 3. In this case $\theta_i(y)$ is a component that introduces exogenous heterogeneity to the model and δ is the imitation-factor of the model, controlling the peer influence. Moreover, x_i represents the mark achieved by the respective agent i and g_i stands for the binary peer matrix of the agent. In that way we generate highly irregular utility functions, each agent incorporates a unique utility function defined by her individual network g_i and her individual variables y_a.

$$U_i(x_i, g_i) = [\mu g_i + \theta_i(y_i)]x_i - \frac{1}{2}x_i^2 + \delta \sum_{j=1}^{n} g_{ij} x_i x_j \qquad (3)$$

The exogenous heterogeneity component $\theta_i(y_a)$ is computed according to Equation 4. y_a is a vector of variables that resemble observable differences between individuals, such as race, age, and other socio-economic variables. σ and ϕ are parameter vectors. In that way we generate highly irregular utility functions, each agent incorporates a unique utility function defined by her individual network g_i and her individual variables y_a.

$$\theta_i(y) = \sum_{m=1}^{M} \sigma_m y_i^m + \frac{1}{g_i} \sum_{m=1}^{M} \sum_{j=1}^{n} \phi_m g_{ij} y_j^m \qquad (4)$$

This fitness function not only introduces wide individual heterogeneity, but also accounts for a strategic complementarity in efforts [24]. this means that if the peer of agent i, agent j increases her behavior level, then agent i will receive increasing marginal utility, if she also increases her behavior level. Table I summarizes the model parameters and contains a brief explanation for each parameter.

V. Experiments

Seeking to verify, if the implemented decision making algorithm is capable of mimicking human decision making in the situation of interest, we choose the most simple model set-up, containing two interconnected agents. The parameter vectors σ and ϕ of the utility function $U_i(x_i, g_i)$ are chosen so that clear strategies emerge for each agent. For the purpose of

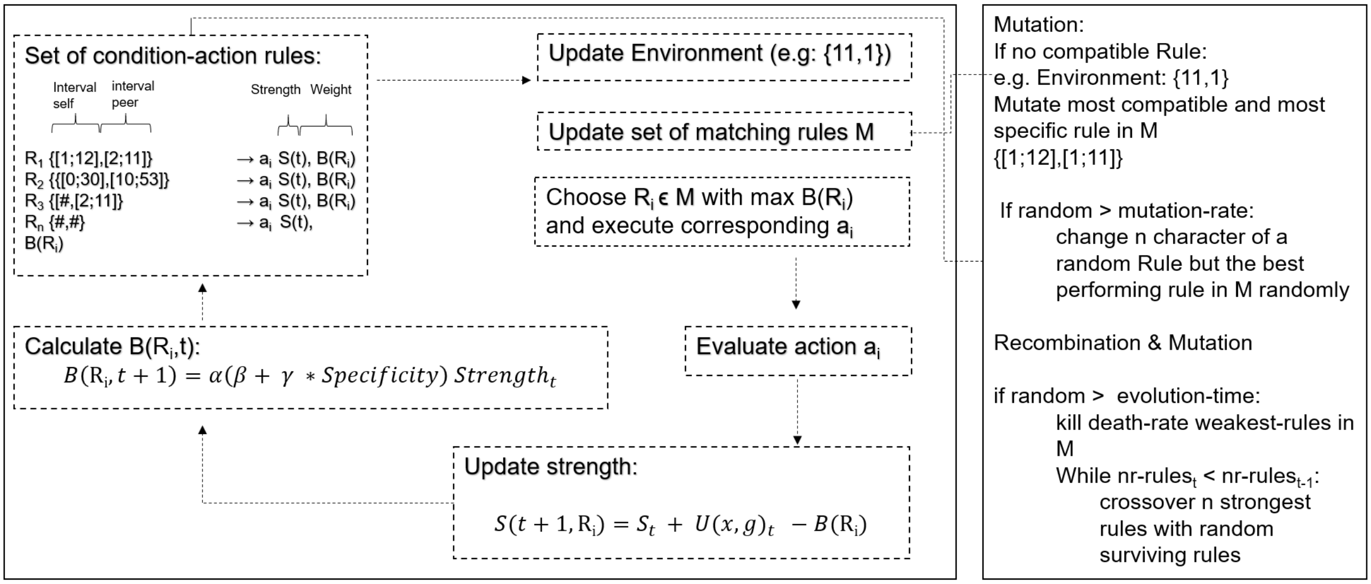


Fig. 1: LCS - Decision

TABLE I: model parameters

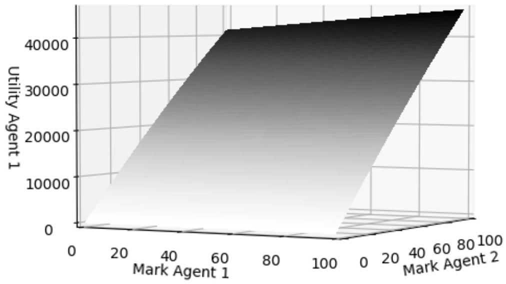
Model modules	Parameters	Explanation
Strength Calculation	α	controls the importance of past performance for the selection of a Rule $R_i \in M_i$
	β	controls the importance of past performance for the selection of a Rule $R_i \in M_i$
	γ	controls the importance of specificity of rules in the LCS
Genetic Operators	$mutation-rate$	controls how frequently rules within the LCS are replaced by randomly created rules
	$death-rate$	controls which share of the population of rules within the LCS is replaced by newly created rules (cross-over recombination)
	$evolution-time$	controls how often an evolutionary process is triggered for all agents
LCS	$nr-action-rules$	controls how many condition-action-rules an agent possesses
	$VAR(x)_1$, $VAR(x)_2$, $VAR(x)_3$	Variance of the normal distributions in the generation of action rules and mutation. Control the maximum step-size for the increasing or respectively decreasing of marks at each iteration.
Utility Function	δ	Imitation Factor, controls the weight of peer behavior within the utility function
	σ	Parameter Vector, assigns weights to the individual variables of each agent
	ϕ	Parameter Vector, assigns weights to the individual variables of peers

experimentation, we define the three distinct strategy settings listed below. (i) ”Good mark”: both agents may always prefer to achieve the better mark, this is achieved by setting σ and δ so that $\frac{du}{dx} > 0$. (ii) ”Bad mark”: both agents may always prefer to achieve the worse mark, this is achieved by setting σ and δ so that $\frac{du}{dx} < 0$. (iii) ”Good mark imitation”: achieving a good mark is a dominant strategy for both agents. However, peer behavior heavily influences the utility outcome. The parameter vectors are set as in (i) and the imitation factor γ is set to 20. For each scenario, the vector of variables resembling observable differences between individuals, y_a is set randomly in order to create two random agents. Figure 2 illustrates the respective utility for agent 1 as a function of her achieved mark $mark1$ and the achieved mark of her peer $mark2$. We set the model parameters as presented in Table II. The model parameters have been chosen manually, analyzing the model behavior. As this paper shall serve as a proof of concept, it is not the purpose to find the best performing parameter setting, but merely one that performs sufficiently well. If more elaborated methods for parameter search where applied, measures should be taken to make sure the parameters are not over fitted. In order to assess, if the model behaviour fulfills our expectations, we measure, if the algorithm is capable of finding good solutions for each scenario. As we seek to mimic human behaviour, we do explicitly not expect fully accurate

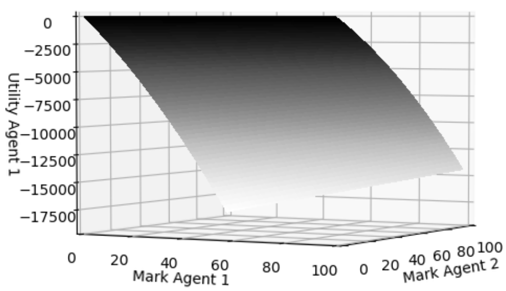
TABLE II: model parameters for experiments

Model modules	Parameters	Values
Strength Calculation	g_1	0.74
	g_2	0.83
	g_3	0.42
Genetic Operators	$mutation-rate$	0.3
	$death-rate$	0.75
	$evolution-time$	5
LCS	$nr-action-rules$	200
	$VAR(x)_1$	4
	$VAR(x)_2)$	40
	$VAR(x)_3)$	10
Utility Function	δ	$(i)(ii) : 0.5; (iii) : 20$
	σ	*
	ϕ	*

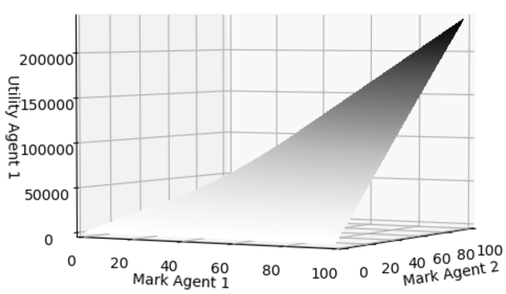
*set to create the respective strategy (i), (ii) or (iii).



(a) Utility Function for Dominant Strategy agent 1: Good mark (i)



(b) Utility Function for Dominant Strategy agent 1: Bad mark (ii



(c) Utility Function for Dominant Strategy agent 1: Good mark & factor imitation = 20 (iii)

Fig. 2: Utility functions for the three strategy settings (i) "Good mark", (ii) "Bad mark" and (iii) "Good mark imitation"

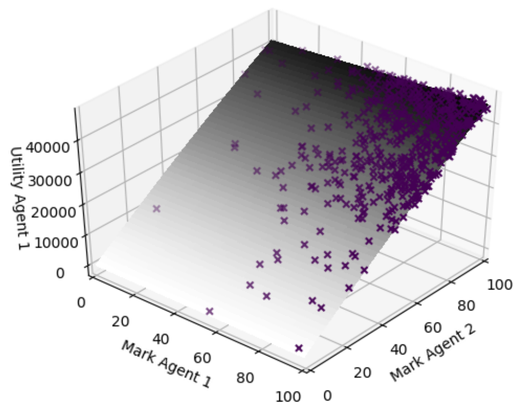
and rational decision making. The agents are expected to demonstrate a tendency towards the optimal solution while sporadic not optimal solutions are tolerated. Moreover, a learning process should be observable throughout run-time. Ultimately a human-like agent is expected to react on changes in her environment, namely the change of behavior of her peers and the alteration of her own situation. We measure this examining the probability for an agent to change the current action subject to recent alterations of the environmental variables, peer behavior and self-behavior. Although utility functions are heterogeneous, we selected scenarios where both agents under observation are expected to react in a similar way. Therefore and for the sake of illustration, we only present the results for the search of agent 1 within this paper. The global conclusions of this work remain unchanged when observing the utility of agent 2. The models are run 500 times with a run-time of 500 iterations.

A. *Overall Performance - Learning Process*

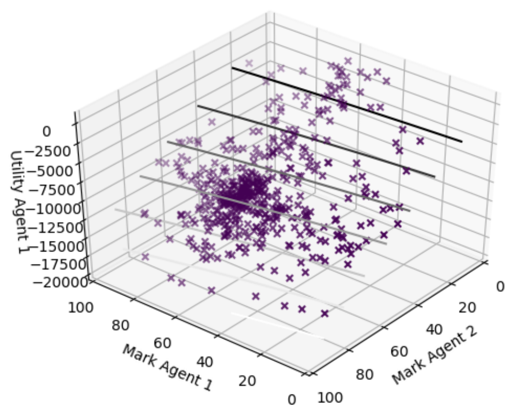
The finally achieved mark of the agents after each run may be revised in Figure 3 for each scenario. Here each cross indicates the final mark of agent1 and agent2 and the respective utility derived by agent1 after 500 iterations. One may observe that for scenarios (i) and (iii) both agents achieved final marks close to the function optimum. Also, for the majority of simulations, marks for both agents can be found in the upper half of the scale. The best possible solution in scenario (ii) would be a mark of 0 for both agents. however, as Figure 3c reveals, the agents did not achieve this optimal solution frequently. Nevertheless, a tendency towards lower marks is observable.

B. *Run-time Performance*

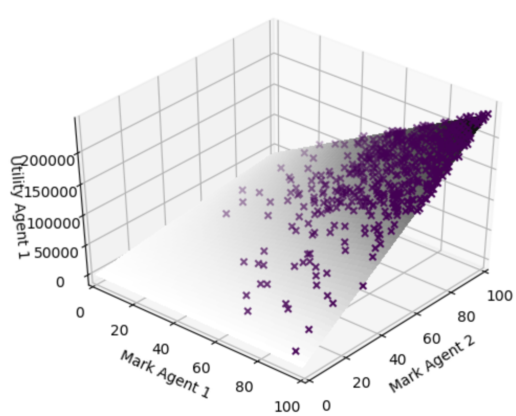
In order to investigate the model behaviour for each iteration, we analyzed the marks achieved by both agents, as well as the utility for agent1. Figure 4 illustrates the average outcome for each iteration in 500 simulations. The solid green line indicates the averagely achieved utility of agent 1 for each iteration, while the dashed red line and the dashed blue line indicate the averagely achieved mark of agent1 and agent2 respectively. The plot for scenario (i) reveals that all indicators develop positively until the end of the run-time. While an average final mark just below 80 is achieved. Plotting the average outcomes for scenario (ii) indicates a negative development of marks throughout the run-time and respectively increasing average utility values. Finally achieved average mark for both agents lies below 60 while the achieved average utility amounts above -8800. Recall that the best possible decision for this scenario for both agents would be a final mark of 0 and respectively a utility of 0. Also, we understand utility as an abstract value for the comparison of decisions. Hence, a negative Utility values does not have a special meaning. Scenario (iii) yields average mark and utility development comparable to scenario (i). Moreover, the run-time analysis encompasses examination of agent behavior over time. In order to observe, how repeatedly



(a) Simulation results for 500 simulations after 200 iterations for Dominant Strategy agent 1: Good mark (i)

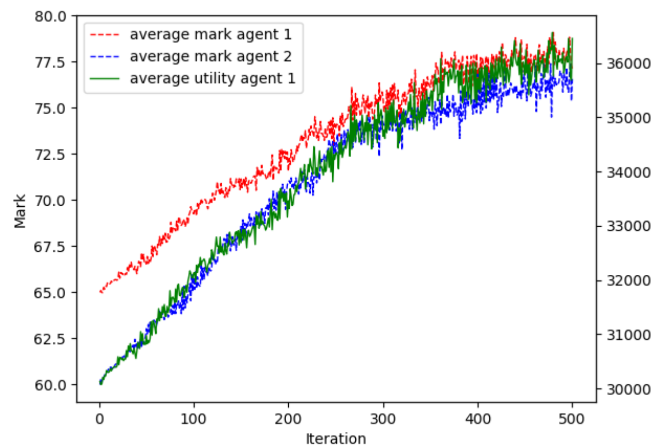


(b) Simulation results for 500 simulations after 200 iterations for Dominant Strategy agent 1: Bad mark (ii

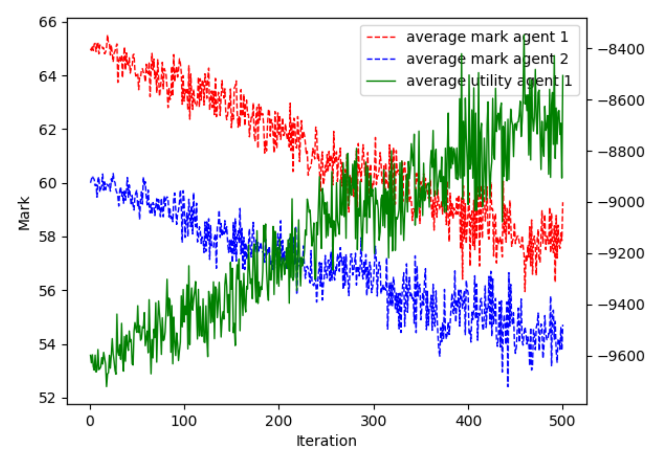


(c) Simulation results for 500 simulations after 200 iterations for Dominant Strategy agent 1: Good mark & factor imitation = 20 (iii)

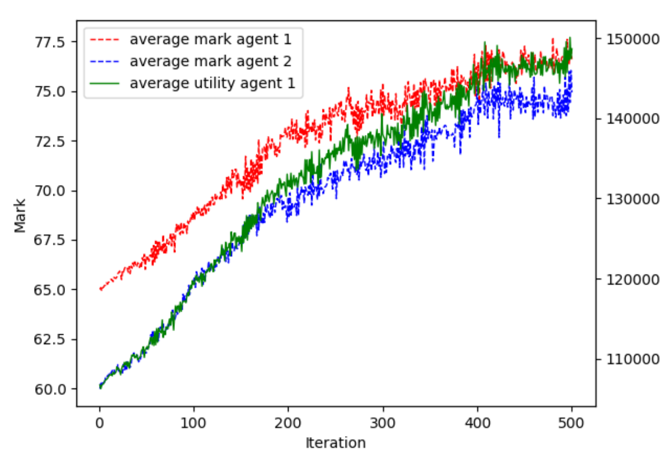
Fig. 3: results obtained after 200 iterations



(a) agent 1: Good mark (i)



(b) agent 1: Bad mark (ii)



(c) agent 1: Good mark & factor imitation = 20 (iii))

Fig. 4: Average Results per Iteration for 500 model runs for three scenarios

chosen actions affect the disposition of agents to try out different behavioral patterns, the frequency of occurrences of behavioral change have been related to the number of iterations with unchanged behavior preceding that alteration. Figure 5 illustrates the respective outcomes. Here the green dashed line indicates how often a change of behavior was observed throughout all experiments after x iterations. The red dashed line represents the probability density function of the distribution of x. It becomes clear that the vast majority of action changes occurs after few repetitions of the same behavior. very low frequencies are observed for more than 10 iterations. In order to ensure the validity of the calculated frequencies, x that occurred less than 20 times have not been considered for this analysis.

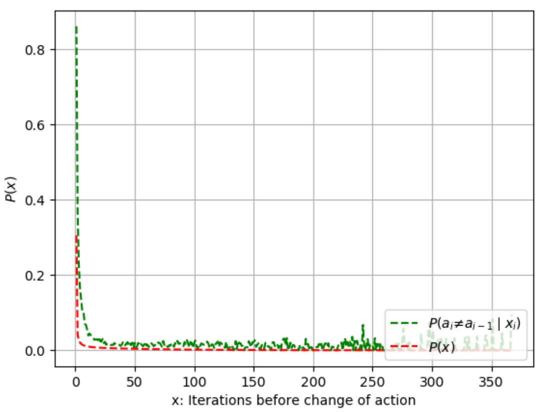


Fig. 5: Frequency of action change related to preceding number of repetitions of the same behavior

C. Reaction to variation of peer behavior

Finally we investigate how the agent responds to changes in peer behavior and in own behavior. To this purpose we calculate the variable Δ according to Equation 5, where a_k indicates the action of agent1 taken in iteration k, x_i indicates the mark of agent1 at iteration i and y_i the mark of agent2 at iteration i.

$$\Delta_i = \sqrt{(\sum_{i=k}^{j} (x_{i-1} - x_i^1))^2 + \sqrt{(\sum_{i=k}^{j} (y_{i-1} - y_i))^2}}, \quad (5)$$
$$a_k \neq a_{k-1}, a_j \neq a_{j+1}, a_j \geq a_k$$

In Figure 6 we plot the cumulative frequency of Δ in the 2.5×10^5 iterations of the 500 experiments as a red solid line. The green line however, indicates the cumulative frequency of Δ in the subset of iterations that actually triggered a change of action for the observed agent. As the relations presented in this Figure are very similar for all three scenarios, we demonstrate the outcomes for scenario (i). For $\Delta > 10$, the green line appears to grow much steeper than the red line. Also, the red plot appears to be much more concave than the green plot. The more concave shape of the red plot indicates that Δ is represented less than proportional within the set of Δ that actually triggered an action change for low Δ, while the opposite holds as Δ grows. Thus, it appears that the probability for an agent to change the current behaviour is substantially higher if the environment, respectively the peer behavior, changes.

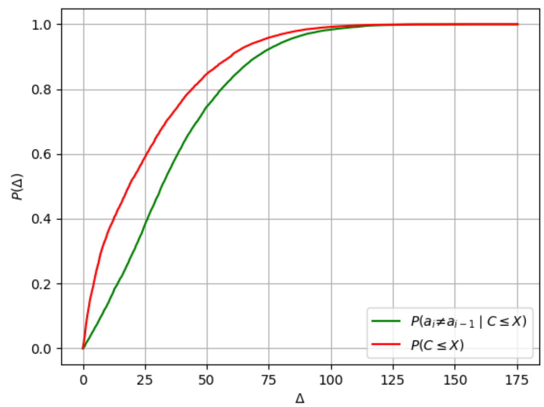


Fig. 6: Frequencies of cumulative environmental change

VI. Discussion

As stated above, this work seeks to present a solution for human alike agent decision making. Hence the decision making algorithm may account for bounded rational decisions that may not be optimal in all cases but demonstrate a tendency towards good decisions. The results presented in Section V-A indicate that the proposed LCS is capable of delivering good solutions for differently shaped utility functions. In the examined simple settings with only two interacting agents, solutions yielding high utility were encountered in the majority of simulations. However, the algorithm also exhibited miss judgment and biased decisions that may also be expected from human decision makers. Difficulties were particularly encountered in situations with negative pay-offs. It may be argued that humans particularly struggle with situations where the outcome is always negative. However there may be alternative parameter settings that help the agents to better perform in negative utility functions. Moreover, it is not clear yet, if the implemented LCS also performs well in more complicated settings with a larger number of heterogeneous peers and high imitation utility. Furthermore, the realistic agents are expected to exhibit the ability to learn from past experiences. Section V-B illustrates that on average, the agents decision improves with increasing run-time specifically for the scenarios (i) and (iii). The decisions in scenario (iii) also improve, yet on a rather low pace. This may indicate that the LCS implementation is more sensible to negative pay-offs. However, the continuously positive developing average utility is a strong signal that the agents exhibit learning behavior. Finally, it was posited that agents may react sensible to changes in peer behavior. In Section V-C we found that the probability for an agent to change her current action is significantly lower, when the cumulative difference of her mark and of the mark of her peer to the respective marks after the preceding action change is close to zero. This analysis also revealed that probability of action change increases with increasing cumulative difference of the environment. Hence, it can be argued that the agents do react on change in peer behavior and self behavior. The run-time analysis further revealed that agents are significantly less likely to change their course of action, once a certain action has been executed repeatedly. Most alterations in behavior have been observed in a short period after experimenting a new behavior. This may resemble habituation in human beings, a behavioural feature that frequently occurs in reality.

VII. Conclusion

Within this paper we propose the implementation of a Learning Classifier System as a decision making module for agent based models that incorporate social influence and heterogeneous interconnected agents. We aim at developing a decision mechanism that resembles bounded rational human decision making (in the sense of H. A. Simon's approach to a more realistic theory of human economic decision making [25]) well and that incorporates imperfect information as a feature from real decision making situations. The use case of the simulation model is the decision about engagement at school of individuals, measured via the achieved mark of those individuals. Experiments with two interconnected agents are conducted in three distinct scenario settings: (i) Firstly, a scenario is set up, where the dominant strategy for both agents is to achieve the best possible mark. (ii) Secondly, the environment is set so that the best possible decision for both observed agents would be not to engage at school at all and consequently achieve the worst possible mark. Finally, we investigate a scenario with high utility derived from imitation of peer behavior. The simulation study shows that the proposed LCS performs well in achieving good solutions for both agents for the respective scenarios. Still, optimization is not accurate but biased by peer decisions and habit and thus well resembles human decision making. Moreover, a learning effect could be identified which is essential when mimicking human decision making. Finally it could be shown that the agents react to environmental change while exhibiting a tendency to create habits which are not changed even if the environment changes. Summarizing, it could be shown, that the application of LCS may in fact be an adequate approach to mimic human decision making in ABM. However, further study is required in order to verify if the LCS performs well also in more complicated settings, incorporating larger numbers of heterogeneous interconnected agents and settings incorporating exclusively negative pay-offs. When applied in a larger context, the actual performance of the LCS implementation should also be assessed in comparison with alternative approaches such as state of the art General Equilibrium Models. A promising field for the extension of this work is also the more careful comparison of the observed decision processes with models of human decision making. To this purpose also a deeper analysis of the acquired condition-action rules may be helpful.Finally, within this study, only one well performing calibration of the simulation model was tested. More detailed analysis of model behaviour under different parameter settings would most certainly contribute to further develop the decision module.

References

[1] O. J. Blanchard, "The state of macro," National Bureau of economic research, Tech. Rep., 2008.

[2] A. Kirman, "Can artificial economies help us understand real economies?" *Revue de l'OFCE*, vol. 124, no. 5, pp. 15–41, 2012.

[3] D. Colander, P. Howitt, A. Kirman, A. Leijonhufvud, and P. Mehrling, "Beyond dsge models: toward an empirically based macroeconomics," *The American Economic Review*, pp. 236–240, 2008.

[4] G. N. Gilbert, *Agent-based models*. Sage, 2008, no. 153.

[5] N. A. Christakis and J. H. Fowler, *Connected: The surprising power of our social networks and how they shape our lives*. Little, Brown, 2009.

[6] J. H. Holland and J. H. Miller, "Artificial adaptive agents in economic theory," *The American economic review*, vol. 81, no. 2, pp. 365–370, 1991.

[7] R. Marimon, E. McGrattan, and T. J. Sargent, "Money as a medium of exchange in an economy with artificially intelligent agents," *Journal of Economic Dynamics and Control*, vol. 14, no. 2, pp. 329–373, 1990.

[8] E. Ba, "Learning by imitation," *Journal of Economic dynamics and contro*, vol. 23, no. 9, pp. 1569–1585, 1999.

[9] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[10] T. C. Schelling, "Dynamic models of segregation," *Journal of mathematical sociology*, vol. 1, no. 2, pp. 143–186, 1971.

[11] G. Deffuant, F. Amblard, G. Weisbuch, and T. Faure, "How can extremism prevail? a study based on the relative agreement interaction model," *Journal of Artificial Societies and Social Simulation*, vol. 5, no. 4, 2002.

[12] M. Janssen and W. Jager, "An integrated approach to simulating behavioural processes: A case study of the lock-in of consumption patterns," *Journal of Artificial Societies and Social Simulation*, vol. 2, no. 2, pp. 21–35, 1999.

[13] P. Davidsson, "Agent based social simulation: A computer science view," *Journal of artificial societies and social simulation*, vol. 5, no. 1, 2002.

[14] M. Meyer, I. Lorscheid, and K. G. Troitzsch, "The development of social simulation as reflected in the first ten years of jasss: a citation and co-citation analysis," *Journal of Artificial Societies and Social Simulation*, vol. 12, no. 4, p. 12, 2009.

[15] L. Tesfatsion and K. L. Judd, *Handbook of computational economics: agent-based computational economics*. Elsevier, 2006, vol. 2.

[16] M. A. Barbosa Jr and F. B. de Lima Neto, "Distributed agent-based social simulations: An architecture to simulate complex social phenomena on highly parallel computational environments," in *Intelligent Agent (IA), 2011 IEEE Symposium on*. IEEE, 2011, pp. 1–8.

[17] F. J. Miguel, J. A. Noguera, T. Llacer, and E. Tapia, "Exploring tax compliance: An agent-based simulation." in *ECMS*, 2012, pp. 638–643.

[18] E. Marchione, M. Salgado, and N. Gilbert, "Analysing differential school effectiveness through multilevel and agent-based modelling," 2011.

[19] O. Sigaud and S. W. Wilson, "Learning classifier systems: a survey," *Soft Computing*, vol. 11, no. 11, pp. 1065–1078, 2007.

[20] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[21] T. Brenner, "Agent learning representation: advice on modelling economic learning," *Handbook of computational economics*, vol. 2, pp. 895–947, 2006.

[22] J. Duffy, "Agent-based models and human subject experiments," *Handbook of computational economics*, vol. 2, pp. 949–1011, 2006.

[23] R. J. Urbanowicz and W. N. Browne, *Introduction to learning classifier systems*. Springer, 2017.

[24] A. Calvó-Armengol, E. Patacchini, and Y. Zenou, "Peer effects and social networks in education," *The Review of Economic Studies*, vol. 76, no. 4, pp. 1239–1267, 2009.

[25] H. A. Simon, "Theories of bounded rationality," *Decision and organization*, vol. 1, no. 1, pp. 161–176, 1972.