DECISION-MAKING USING RANDOM BOOLEAN NETWORKS IN THE *EL FAROL* BAR PROBLEM

Daniel Epstein and Ana L. C. Bazzan
Instituto de Informática, UFRGS
C.P. 15064, 91501-970, P.Alegre, RS, Brazil
Email: {depstein,bazzan}@inf.ufrgs.br

KEYWORDS

Multiagent Systems, Minority Games, El Farol Bar Problem, Random Boolean Networks

ABSTRACT

In our daily lives we often face binary decisions where we seek to take either the minority or the majority side. One of these binary decision scenarios is the *El Farol* Bar Problem, which has been used to study how agents achieve coordination. Previous works have shown that agents may reach appropriate levels of coordination, mostly by looking at their own individual strategies that consider the complete history of the bar attendance. No structure of the network of players has been explicitly considered. Here we use the formalism of random boolean networks to help agents to make decisions considering a network of other decision-makers. This is especially useful because random boolean networks allow the mapping of actions of K other agents (hence not based on complete history) to the decision-making of each single individual. Therefore a contribution of this work is the fact that we consider agents as participants of a social network. In the original proposition for this problem, strategies would change within time and eventually would lead agents to, collectively, decide on a efficient attendance, at each time step. Hence there was no explicit modeling of such a social network. Our results using random boolean networks show a similar pattern of convergence to an efficient attendance, provided agents do experimentation with the number of boolean functions, have a good update strategy, and a certain number of neighbors is considered.

INTRODUCTION

A common situation people face is when one has gone to his/her favorite pub only to find out that it happened to be overcrowded that night, leading to one regretting not to have stayed home. That is the motivation for minority games (MG) — a game where the decision that is made by the minority of the players is rewarded. One instance of such a game is the *El Farol* Bar Problem (EFBP) by B. Arthur (1994), which has also been the subject of studies in statistical physics, e.g., Challet and Zhang (1997) and Johnson *et al.* (1998).

The idea of rewarding the decision that is made by the minority of the players (or agents) is interesting in many scenarios. For instance, in agent-based simulation of traffic, a minority game is clearly useful: choosing routes that are least busy is thoroughly rewarding.

Without being too specific, here we use the basic scenario of B. Arthur, i.e., the original EFBP. In his setting, the basic idea is that the bar is enjoyable only if up to a certain threshold ρ of customers visit the bar on a single night. These visitors (players or agents) can make predictions about the attendance for the next time based on the results of the previous m weeks. This is the basis for figuring out strategies for playing the game. Each agent having a set of such strategies, Arthur's main point was the study of how well these strategies perform. By means of computer simulations, it was realized that the mean attendance converges to an ideal threshold (in his case it was set to $\rho = 60$).

In the present paper we depart from Arthur's approach that has based individuals' strategies on the complete history of the attendance. For instance, one of Arthur's strategies was "stay at home if the bar was crowded in the last m weeks". Instead of considering this scenario, which requires that every agent is informed about the correct attendance in the past, we let agents decide whether or not to go to the bar based on a boolean function that maps the inputs of K other agents (e.g. his/her friends, a social network, etc.) to one's output. To do so, we use a well-known formalism introduced by Kauffman (1969): the random boolean networks (RBN's). An example of such a boolean function for $K = 2$ would be the AND function: "I go to the bar only if A and B have gone in the last week". This has also the implication that agents are not considered isolated decision-makers but, rather, as a social network.

We have organized this paper as follows: The next two sections review some works on the EFBP and its more general version, the minority game, and explain how the RBN formalism works. Following these, the approach is presented. Simulations' results and their analysis then follow. Finally, the last section discusses several aspects of this work and its future directions, and provide concluding remarks.

Proceedings 25th European Conference on Modelling and Simulation ©ECMS Tadeusz Burczynski, Joanna Kolodziej Aleksander Byrski, Marco Carvalho (Editors)
ISBN: 978-0-9564944-2-9 / ISBN: 978-0-9564944-3-6 (CD)

EL FAROL BAR PROBLEM AND MINORITY GAMES

Microeconomics and game-theory assume human behavior to be rational and deductive – deriving a conclusion by perfect logical processes from well-defined premises. But this assumption does not hold especially in interactive situations like the coordination of many agents. There is no *a priori* best strategy since the outcome of a game depends on the other players. Therefore, bounded and inductive rationality (i.e. making a decision based on past experience) is supposed to be a more realistic description.

In this spirit Arthur introduced in 1994 a coordination game called the *El Farol* Bar Problem. (The name is inspired by a bar in Santa Fe.) Every week n players wish to visit the bar *El Farol* attracted by live music. Up to a certain threshold ρ of customers the bar is very comfortable and enjoyable. If it is too crowed, it is better to stay home. The payoff of the game is clear: if the number of visitors is less than the threshold ρ, these visitors are rewarded, otherwise those who stayed home are better off. In the original work, $n = 100$ and $\rho = 60$ were used, but arbitrary values of n and ρ have also been studied, as e.g., Johnson *et al.* (1998).

The players can only make predictions about the attendance for the next time based on the results of the previous m weeks; this is the basis for strategies for playing the game. Now, there is a huge set of possible strategies and every player possesses a fraction of them. For the decision whether to go or to stay home, the player always selects the strategy which predicts the outcome of the last weeks most accurately. After every week the player learns the new attendance and thus evaluates his strategies. By means of computer simulations, it was realized that the mean attendance converges to the threshold $\rho = 60$.

Later, the EFBP was generalized to a binary game by Challet and Zhang (1997), the so-called Minority Game (MG). An odd number n of players has to choose between two alternatives (e.g., yes or no, or simply 0 or 1). With a memory size m there are 2^{2^m} possible strategies. Each player has a set S of them. These are chosen randomly out of the whole set. In the simplest version of the game, players are rewarded one point if they are in the minority group. Other functions that favor, for instance, smaller minorities were studied by several authors as, e.g., Challet and Zhang (1998); Johnson *et al.* (1998).

The MG and the EFBP are gradually becoming a paradigm for complex systems and have been recently studied in detail[1]. We will refer briefly to some of the basic results.

In their original work, Challet and Zhang have systematically studied the influence of the memory size m and number of strategies S on the performance of the game. They concluded that the mean attendance always converges to $n/2$ but for larger m there are less fluctuations, which means the game is more efficient.

Additionally, different temporal evolution processes were studied. It has been shown that such evolutionary processes lead to stable, self-organized states. Some analytical studies showed that there is a phase transition to an efficient game if $2^m/n$ is approximately one.

B. Edmonds (1999) has investigated the emergence of heterogeneity among agents in a simulation. His paper tackles evolutionary learning as well as communication among agents, which might lead to a differentiation of roles at the end of the run.

The work in Bazzan *et al.* (2000) has introduced *personalities* that model certain types of human behavior. Different populations of these personalities have been simulated. It was found that there is one personality that performs better than the average: the wayward personality (do the opposite of what has proved a good action in the last m steps).

In Galstyan *et al.* (2003) the authors study the EFBP under the light of resource allocation: agents using particular resources are rewarded or punished according to the availability of these resources.

Also, there has been an interesting line of research connecting minority games and collective intelligence such as Wolpert *et al.* (2000). For a discussion see Tumer and Wolpert (2004).

Finally, this problem is closely related to learning in a many-agent system. In Hercog and Fogarty (2002) the authors investigate the utility of using learning classifier systems (in particular, the ZCS is used) as a learning engine. Both the macro and micro levels are addressed, thus not only overall attendance is investigated but also agents' behaviour, reward, and action selection. In Rand (2006) the author also uses the EFBP to illustrate the use of machine learning techniques in agent-based modeling.

RANDOM BOOLEAN NETWORKS

Boolean networks have been used to explain self-organization and adaptation in complex systems. The study of the behavior of regulatory systems by means of networks of boolean functions was introduced by Kauffman in 1969 Kauffman (1969). Examples of the use of this approach in biology, genomics, and other complex systems can be found in Kauffman (1993).

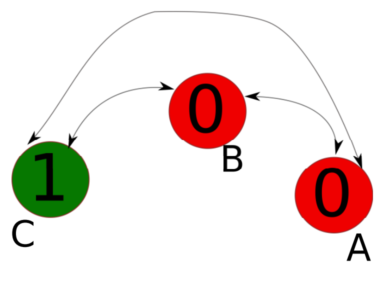

Figure 1: $N = 3$ connected agents ($K = 2$) in an RBN.

[1] For a collection of papers and pre-prints on the Minority Game, see the web page at `http://www.unifr.ch/econphysics/`.

Table 1: Boolean functions for agents C, A and B.

(AND)			(OR)			(OR)		
A	B	C	B	C	A	A	C	B
0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	1	1
1	0	0	1	0	1	1	0	1
1	1	0	1	1	1	1	1	1

RBN's are made up of binary variables. In the setting investigated here, a network is composed of N agents that must decide whether or not to make a binary decision (in this case going to a bar). Each agent is represented by one of these binary variables. These in turn are, each, regulated by some other variables, which serve as inputs. The dynamical behavior of each agent, namely which action it will execute at the next time step (whether or not it goes to the bar), is governed by a logical rule based on a boolean function. These functions specify, for each possible combination of K input values, the status of the regulated variable. Thus, being K the number of input variables regulating a given agent, since each of these inputs can be either on or off (1 or 0), the number of combinations of states of the K inputs is 2^K. For each of these combinations, a specific boolean function must output either 1 or 0, thus the total number of boolean functions over K inputs is 2^{2^K}. When $K = 2$, some of these functions are well-known (AND, OR, XOR, NAND, etc.) but in the general case functions have no obvious semantics.

To illustrate the regulation process, Figure 1 depicts a simple example of a network of $N = 3$ agents where each was assigned a boolean function randomly, and $K = 2$. The boolean functions for these 3 agents are then depicted in Table 1 (adapted from Kauffman (1993)): agents A and B are regulated by function OR, while agent C is regulated by an AND. In this table, one can see all possibilities for C (3rd column) to make a decision, where 1 means for instance go to the bar and 0 means do not go. Similarly, A's output is determined by the inputs from both B and C, and B's output depends on inputs from A and C.

Given the 3 boolean function from Table 1, Table 2 shows, for all 2^3 states at a given time T, the action taken by each agent at time $T + 1$, i.e. the successor state of each state. Further, from this table, it is possible to determine the state transition graph of the network, which appears in Figure 2. One sees that there are 3 state cycles (attractors).

If we assign each of the N agents randomly to one of the 2^{2^K} boolean functions, the dynamics of the decision-making is deterministic and the system ends up in one of the state cycles. It is then a matter of "luck" that only a certain number ρ of agents end up all going to the bar. For instance in the case depicted in Figure 2, in both cycles 1 (`000`) and 3 (`111`), either none go to the bar (state 1) or all go (state 3).

However, if the network depicted in Figure 1 evolves,

Table 2: States' transition for Table 1.

(T)			(T+1)		
A	B	C	A	B	C
0	0	0	0	0	0
0	0	1	1	1	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	1	1	1

	↶					
	000					state cycle 1
010	⇆	100				state cycle 2
		001				
		↓		↶		state
011	→	110	→	111		cycle 3
		↑				
		101				

Figure 2: States' transition graph for Table 1 (3 state cycles and attractors).

Table 3: Mutated version of Table 1.

(NAND)			(OR)			(OR)		
A	B	C	B	C	A	A	C	B
0	0	1	0	0	0	0	0	0
0	1	1	0	1	1	0	1	1
1	0	1	1	0	1	1	0	1
1	1	0	1	1	1	1	1	1

Table 4: States' transition for Table 3.

(T)			(T+1)		
A	B	C	A	B	C
0	0	0	0	0	1
0	0	1	1	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	1	1	0

		000	→	001				
				↓		↶		state
010	→	101	→	111	→	110		cycle 1
				↑				
		100	→	011				

Figure 3: States' transition graph for Table 3 (single state cycle and attractor).

the system may scape a bad attractor (which is a state in which either too many or too few agents go to the bar). This evolution of the network may happen in several ways: agents get reconnect to others, the boolean functions change, etc.

Let us consider an example in which the boolean function of agent C changes from AND to NAND. The boolean functions are now depicted in Table 3 while Table 4 shows the successor state of each state and Figure 3 depicts the state transition graph.

The dynamics of the regulation changes as seen in Figure 3. Now only one state or attractor exists (110), namely one that has the property that agents A and B always go but agent C never does.

The extent of such a change – whether or not the system will be attract to another attraction basin – obviously depends on the extent of the changes in the network and/or functions. In Kauffman (1993) the author extensively discusses many of these factors, as well as the properties of RBN's, including the issue of stability and cycle length. In the present paper, because the logic of the functions and the structure of the network changes along time, properties such as periodic behavior cannot be observed.

On the other hand, a central question raised by Kauffman, and which is relevant to our work, relates to the problem of adaptation in a system with many interacting parts. The key question is whether an adaptive process which is constrained to altering the input connections between elements in a network and the logic governing them can hill climb to networks with desired attractors.

RBN-BASED APPROACH

As mentioned, in Arthur (1994), the EFBP was proposed as a proxy to investigate how to deal with and to model inductive reasoning. The author has used a series of strategy in order to verify whether there would be a convergence to a certain state or the behavior would be chaotic.

In the present paper we use RBN's to equip the agents with a decision-making framework. This is appropriate for binary (i.e. boolean) decision-making, which aim at considering inputs from other agents in this decision-making process. To do so, we replace the possible strategies described by Arthur (1994) by the possible boolean functions a node may have. Hence, instead of having a random strategy, each node has random boolean functions and uses them to determinate whether or not to go to the bar. Contrarily to the work in Arthur (1994) (and many others that have followed), we consider that the agents are organized in a kind of network. Arthur has based the decision-making of his agents solely on the agent itself, i.e. the agent would look at its own history to decide about going to the bar.

In our case, each agent ($a_i \in A$) is a node in a random boolean network; such node has to choose whether or not to go to the bar. Also, each agent has K neighbors randomly chosen. Therefore, each of them has K entries in its boolean function, which then outputs either 0 or 1, depending on the values of these K inputs and on the function the agent possesses. Such an output will be denoted here as fr. The inputs are the actions of the K neighbors in the last time step (in the initial time step, all agents start with a random action). We will assume that if a boolean function returns 0, this means the agent stays home; when the function returns 1 the decision is to go to the bar.

At each time step, all agents (nodes) make their decision simultaneously. This decision is considered the right one if it decides to go to the bar and less than ρ % of all agents have made the same decision. Similarly, the decision is also considered correct if the agent has decided to stay home and ρ % or more of all agents have decided to go to the bar. The decision is considered wrong otherwise.

This way, at each time step the quality of every function will be measured and its score (s) will be updated. We denote the number of agents attending the bar as ϕ. If a function led the agent to the right decision about going to the bar, then the score s of this function is increased. Otherwise, it is decreased. This is formalized as follows:

$$
\begin{array}{ll}
s \leftarrow s+1 & \text{if } (fr = 1 \text{ and } \phi < \rho) \\
 & \text{or } (fr = 0 \text{ and } \phi \geq \rho)) \\
s \leftarrow s-1 & \text{otherwise}
\end{array}
$$

Finding the best functions that each agent should use would allow us to create an efficient network. Efficiency here means that the bar has exactly $\phi = \rho$ of the agents. In this case, we would have ρ% of total number of agents attending the bar and hence no one would have made the wrong choice. Thus the equilibrium would have been found.

Of course, one cannot be sure that a function that works well for an agent a_1 will also work well for another agent a_2. That is because the outcome of the function depends on its inputs (in this case, the agent's neighbors).

Our approach for adaptation of the functions that are used at local level has two different variants: *CopyBest* and *Epsilon-greedy*. Next we give more details about these two variants.

1. *CopyBest*: each agent has only one boolean function and at each time step it investigates if it leads to a correct choice. As mentioned before, this sole boolean function is assigned randomly to each agent. If this function causes the agent to make θ consecutive mistakes, it is replaced by another one that has yielded a better result (made the right choices) for some other node. This way we intend to reduce the number of functions that result on a consecutive wrong choice by the agent, keeping only those that provided good results for some node. *CopyBest* on its turn has further two variants that are related to the source of functions to be copied:

- *LocalCopy*: agents are allowed to copy functions only from their neighbors. Here, nodes have at most K functions to choose from. Thus, a "good" function would propagate in a gradual way, giving all function a chance to be tested before disappearing from the network (e.g. when they are replaced).
- *GlobalCopy*: allows the nodes to choose a function from any other node in the network. This will propagate good functions much faster, but will also rapidly reduce the variability of functions in the network.

2. *Epsilon-Greedy*: each node has a set of ten possible functions to choose from. This choice may be made at random or greedily. Here, at every τ time steps, the node chooses one among these functions. In ϵ times, the node chooses the function that has yielded the highest score so far. Otherwise it is random. ϵ starts at a low value but every τ time steps, the value of ϵ is increased by Δ (annealing). The score s of a function is updated as previously mentioned.

EXPERIMENTS AND ANALYSIS OF THE RESULTS

Settings

The main metric to be analyzed is the amount of agents that go to the bar (as in Arthur (1994)). This information is collected at every time step. The objective is to verify if using RBN's in the EFBP also leads to an emerging behavior and, if so, what is the outcome. We thus compare our results with the ones from Arthur (1994). To do so we use $\rho = 60\%$.

Each experiment comprises 100 agents, which must choose whether or not to go to the bar for 1000 time steps. Each agent has a number K of neighbors. This K can assume two different values: $K = 2$ and $K = 3$. For all agents in a given simulation, K remains the same. At every time step the number of agents that chose to go to the bar is calculated. In order to remove as much randomness as possible, we repeat every experiment 100 times. The final result is the average number of agents at the bar every time step.

For the *CopyBest* we set the value of θ to 5. If we chose a lower value, the agents would change their function very often leading to excessive noise. If we chose a higher value, the agents would keep their bad functions for many time steps (once they only change functions upon *consecutive* wrong choices).

For the *Epsilon-greedy* experiments, Δ is set to different values in each case: 0.001, 0.005, 0.01, 0.05 and 0.1. These values were chosen because they allow us to study those cases when ϵ increases at different rates. This of course has an effect on how often agents explore different boolean functions before exploiting.

Experiments have been made using different values for τ ($\tau = 2$ to $\tau = 30$). In those experiments where τ was lower than 10, there were too many functions with the same score. τ higher than 10 provided no difference in the number of agents going to the bar. Hence, $\tau = 10$ was used.

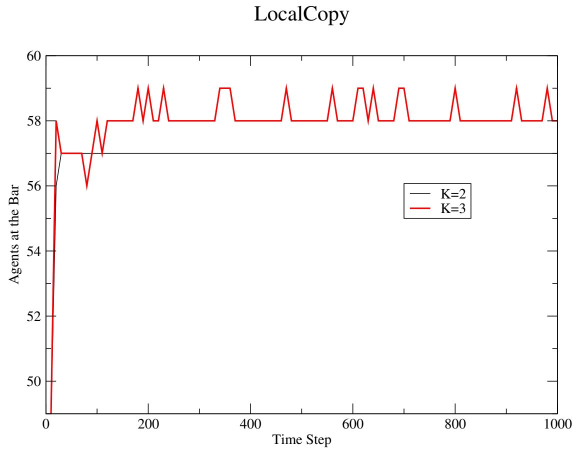

Figure 4: Bar attendance for *CopyBest/CopyLocal*.

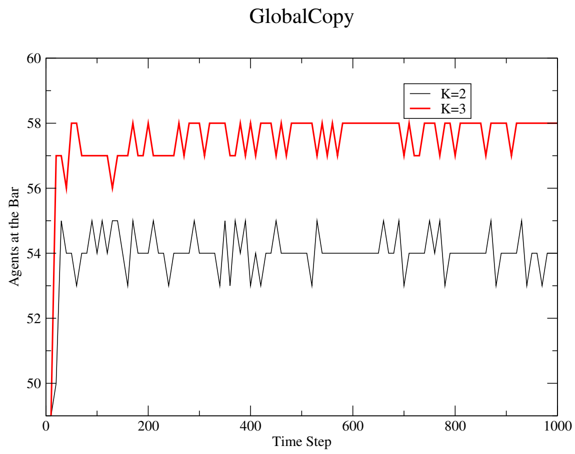

Figure 5: Bar attendance for *CopyBest/GlobalCopy*.

Results

We remark that when RBN's are assigned randomly to the agents and do not change along time, it is expected that 50% of the agents go to the bar. This is of course not satisfactory as $\rho = 60$. This figure of 50% can be seen as a kind of baseline, from which we want to improve the decision-making of the agents.

The experiments using *CopyBest* (both variants) performed relatively well, depending on the value of K. An improvement from the initial state of the system (where about 50% of the agents choose to go to the bar) could be observed, with the attendance reaching almost 60%.

As we can see in Figure 4, when the agents can only copy functions from their neighbors (*CopyBest/LocalCopy* experiment), the average number of agents going to the bar has quickly converged to 57%

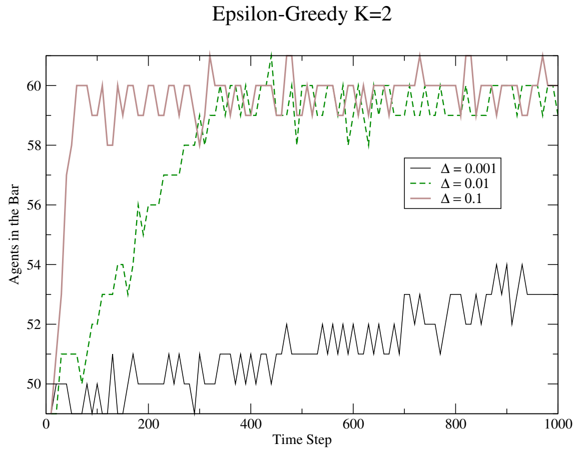

Figure 6: Bar attendance for *Epsilon-greedy*, K=2.

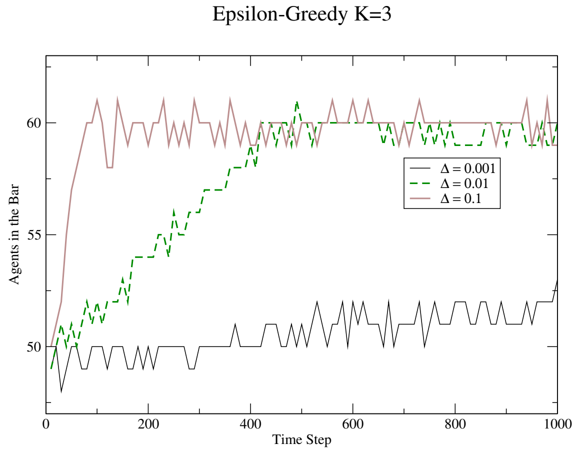

Figure 7: Bar attendance for *Epsilon-greedy*, K=3.

($K = 3$). When agents can copy functions from any other agent (*CopyBest*/*GlobalCopy* experiment, Figure 5) an oscillatory behavior is observed. This is explained by the fact that in the latter there are far more possible functions that can be tried.

The fact that the simulations in both cases do not fully converge to attendance of ρ is explained by the fact that at some points there is a lack of options regarding functions to select. If the functions can only be copied from neighbors, each agent may eventually run out of options and it will have to keep a specific function, even if it is not the best one for it. If the functions can be copied from any node, as explained before, functions with worse scores are replaced. This means that they disappear from the selection pool and again, lead to lack of options after some time steps.

Also, in both cases, the results for simulations where each agent had $K = 3$ are better than the one for $K = 2$. This can be seen in both figures 4 and 5. Having an extra neighbor allows the agent to choose from more functions and to propagate his own function to more nodes too, resulting in more functions to select from. However, the number of possible functions increase exponentially with K (2^{2^K} as mentioned before). Thus values higher than 3 for K do not bring significant improvements. Rather, there is a degradation in performance.

Regarding the *Epsilon-greedy* experiments, these have depicted better results than the *CopyBest*: in almost all cases a convergence towards $\phi = 60\%$ was verified. The time taken for this convergence of course strongly depends on the value of Δ, i.e., how fast ϵ increases. When the increase is fast, the convergence is achieved earlier, as seen in Figure 6 and in Figure 7.

In some experiments, in which Δ was high, convergence was not achieved. The reason is that agents start acting greedy too early, thus making impossible for them to test a reasonable number of functions. Whenever the agents had the time to test their functions before exploiting, the network converged to an efficient state.

CONCLUSION

B. Arthur's EFBP has been used as a metaphor to study inductive adaptation of agents that have to act in a coordinated way. In the original setting, because the bar is enjoyable only if up to a certain threshold of customers ρ visit it, these visitors have to use a set of individual strategies to make predictions about whether or not to go. These strategies are normally based on the complete knowledge about the attendance at the previous m weeks.

Instead of assuming complete knowledge, in the present paper we assume that agents interact in a kind of social network. We then let agents decide whether or not to go to the bar based on a boolean function that maps the inputs of K acquaintances to one's output. To do so, we use the random boolean networks formalism. This way, agents are not considered isolated decision-makers but, rather, as a collective.

We have found that even though agents make decisions using less information, each agent was able to select an action that would bring a good result for itself and for other agents as well.

Our approach admits some variants which were explained and tested. These variants relate to how to copy functions from other agents, and whether or not to act greedily in what regards one's own known functions. Such variants were called here *CopyBest* and *Epsilon-greedy* respectively. Both were able to show similar results to the one reported in Arthur (1994) but *Epsilon-greedy* was superior to *CopyBest*.

In the *CopyBest* variant, those functions that did not had good results were rapidly replaced by others. This means that after some time steps the agents have few options to replace their bad functions. The *Epsilon-greedy* variant has shown results closer to those reported by B. Arthur, especially if the agents explore other functions with appropriate frequency, i.e., if they have a good balance between exploration and exploitation.

As a result, we manage to establish a scenario where the agents were capable of coordinating themselves with less information than that used in the classical approach to the EFBP.

In the current version of our approach, the network is somehow static, i.e., agents get connected to K others and this does not change. Thus, in a future work we want to investigate what happens if agents are allowed to disconnect themselves from acquaintances that have not act as providers of good functions, and conversely, to get connected to other, more successful agents.

AUTHOR BIOGRAPHIES

DANIEL EPSTEIN was born in Porto Alegre, Brazil and graduated in computer science from UFRGS in 2010. He began his master's degree in the field of Artificial Intelligence in 2011. His research interests include: use of Game-Theoretic Paradigms for Coordination of Agents, Agent-based Simulation, Complex Systems and Swarm Intelligence.

ANA L. C. BAZZAN was born in S. Paulo, Brazil and got her PhD degree from the University of Kalrsruhe, Germany, in 1997. From 1997 to 1998, she had a postdoc research associate position in the Multi-Agent Systems Laboratory at the University of Massachusetts in Amherst, under supervision of Prof. Victor Lesser. Since 1999 she is affiliated with UFRGS in P. Alegre, Brazil, now as an associate professor. Her research interests include: use of Game-Theoretic Paradigms for Coordination of Agents, Learning in MAS, Agent-based Simulation, Artificial Societies, Complex Systems, Bioinformatics, Traffic Simulation and Control, Pedestrian Simulation, and Swarm Intelligence. From April 2006 to March 2007 she had an appointment at the University of Würzburg (Germany) as a fellow of the Alexander von Humboldt Foundation. Her web site can be found at `http://www.inf.ufrgs.br/~bazzan/`.

REFERENCES

Arthur, B. (1994). Inductive reasoning, bounded rationality and the bar problem. Technical Report 94–03–014, Santa Fe Institute.

Bazzan, A. L. C., Bordini, R. H., Andriotti, G. K., Viccari, R., and Wahle, J. (2000). Wayward agents in a commuting scenario (personalities in the minority game). In *Proc. of the Int. Conf. on Multi-Agent Systems (ICMAS)*, pages 55–62. IEEE Computer Science.

Challet, D. and Zhang, Y. C. (1997). Emergence of cooperation and organization in an evolutionary game. *Physica A*, **246**, 407–418.

Challet, D. and Zhang, Y. C. (1998). On the minority game: Analytical and numerical studies. *Physica A*, **256**, 514–532.

Edmonds, B. (1999). Gossip, sexual recombination and the el farol bar: Modelling the emergence of heterogeneity. *Journal of Artificial Societies and Social Simulation 2(3)*, **2**(3). Avaiable on http://www.soc.surrey.ac.uk/JASSS/2/3/2.html.

Galstyan, A., Kolar, S., and Lerman, K. (2003). Resource allocation games with hanging resource capacities. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 145–152. ACM Press.

Hercog, L. M. and Fogarty, T. C. (2002). Co-evolutionary classifier systems for multi-agent simulation. In *Proceedings of the Congress on Evolutionary Computation*, pages 1798–1803, Honolulu, HI , USA.

Johnson, N. F., Jarvis, S., Jonson, R., Cheung, P., Kwong, Y. R., and Hui, P. M. (1998). Volatility and agent adaptability in a selforganizing market. *Physica A*, (258), 230–236.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol*, **22**(3), 437–467.

Kauffman, S. A. (1993). *The Origins of Order*. Oxford University Press, Oxford.

Rand, W. (2006). Machine learning meets agent-based modeling: when not to go to a bar. In *Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects*, pages 51–58, Chicago, IL, USA.

Tumer, K. and Wolpert, D. (2004). A survey of collectives. In K. Tumer and D. Wolpert, editors, *Collectives and the Design of Complex Systems*, pages 1–42. Springer.

Wolpert, D. H., Wheeler, K., and Tumer, K. (2000). Collective intelligence for control of distributed dynamical systems. *Europhysics Letters*, **49**(6).