Combined Proceedings of the Social Network Analysis and Norms for MAS Symposium - SN-MAS2010 Section,
Giulia Andrighetto, Guido Boella, Ugo Pagallo, and Serena Villata (Ed.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

Agent-based economic modeling with finite state machines

Ivan Gnilomedov (St. Petersburg State University of Information Technologies, Mechanics and Optics. Email: zilsmail@gmail.com)
Sergey Nikolenko (Steklov Mathematical Institute, St. Petersburg. Email: sergey@logic.pdmi.ras.ru)

Abstract. We offer a novel approach to agent-based economic modeling. Previous work has modeled learning agents as neural networks, sets of fuzzy rules and other learning algorithms. In this paper, we present an approach based on representing an agent's production cycle as a finite state machine. We show that the finite state machine model offers natural representations for basic economic features such as complementary and substitute commodities. Our experimental results show that the model behaves in perfect accordance with basic economic laws that shows the model's validity.

1. INTRODUCTION

Agent-based modeling has played an increasingly important role in understanding market behaviour. Existing research has been primarily centered on modeling financial markets with agents either following predefined strategies or learning the optimal strategy via different learning algorithms.

We aim to create a general equilibrium model representing a market of companies producing different commodities with different manufacturing processes, a model that would accurately reflect the behaviour of supply, demand, and prices in a real economy. A classical model of this market is the well-known Walrasian model for N different commodities produced with M different resources (production factors), where general equilibrium is achieved by maximizing each agent's profit; see Black [1] for a detailed exposition. The general equilibrium model leads to nice mathematical properties (Walras' Law and others). However, this model assumes each agent is perfectly rational and computationally unbounded and the general equilibrium is static, while the real world market tries to achieve equilibrium in a constantly changing world. Thus, the need arises to offer agent-based models with limited and learning agents.

Zimmermann, Neuneier, and Grothmann [2] offer an agent-based model of the *FX-Market* where agents base their decisions on incomplete information coming from a limited number of error-prone sources of information. The agents are modeled as error-correcting neural networks. Kooths [3] and Kooths, Mitze, and Ringhut [4] create a macroeconomic model where agents are modeled as neural networks with additional fuzzy rules representing knowledge about the economy. Other approaches have also been tested; for a detailed survey of agent-based economy models see LeBaron, [5], Tesfatsion [6] and references therein.

Agent-based models with agents represented as finite state machines (FSMs) are widely used in computer animation (see, for example, Rudomin, Millan, and Hernandez [7]), while learning FSMs has been already applied to automata-based programming developed by Shalyto and Tukkel [8].

Developing systems of reactive agents with finite automata was suggested in Naumov, Shalyto [9]. The authors assert that their approach can allow to create systems of self-learning adaptive agents.

However, as far as we know this is the first agent-based economy simulation where agents would be represented and trained as FSMs. In this paper, we fill the gap by constructing an economic model with adaptive agents based on finite state machines. The underlying FSM of an agent represents its production cycle. It gives us an opportunity to model a production process with more detail than previous works do and to naturally interpret such relations between commodities as substitutes and complementary goods. Moreover, finite state machines are well-studied mathematical objects with many known properties, which allows us to approach the problem of devising (or learning) a suitable program for the agents.

We have implemented a model economy with these agents. Results of our experiments are in good accordance with basic economic laws. We have attempted to make the strategy that agents use to make economic decisions both easy to implement but efficient in achieving the proper suboptimal behaviour of the agents. The agents' programming (artificial intelligence) is based on the ant colony optimization algorithm which will be described below.

The paper is organized as follows. Section 2 describes the model itself: in 2.1, we show how to model a company with finite state machines, 2.2 describes the market model in our approach and shows a concrete example of a production cycle, 2.3 deals with the agents' intelligence, and 2.4 outlines the benefits of the proposed model. Section 3 lists experimental results; we show that the model behaves just as the economic laws predict, thus establishing the model's validity.

2. DESCRIPTION OF THE MODEL

2.1. Modeling production with finite state machines

A *finite state machine* is a directed graph with captions on edges $FSM = \langle V, E, \Sigma \rangle$ where V is the set of vertices, Σ is the FSM's *alphabet* (the set of input events), and $E = \left\{ e = \langle v_1, v_2, \sigma \rangle \,/\, v_1, v_2 \in V, \sigma \in \Sigma \right\}$ is the set of edges (possible *transitions*). An edge $e = \langle v_1, v_2, \sigma \rangle$ means that transition from v_1 to v_2 with input event σ is possible. A finite state machine has two kinds of special vertices: a unique $s \in V$ is labeled as the *initial vertex* and some $\left\{ t_0, \ldots, t_K \right\} \subset V$ represent the set of *terminal* vertices. Sometimes it is reasonable to associate both income and outcome events with each edge. Then $Tr = \langle V, E, \Sigma, \Pi \rangle$, $E = \left\{ e = \langle v_1, v_2, \sigma, \pi \rangle \,/\, v_1, v_2 \in V, \sigma \in \Sigma, \pi \in \Pi \right\}$ (these FSMs are usually called *transducers*). A *path* in a FSM is an ordered set

$$P = \langle v_0, e_0, v_1, e_1, \ldots, v_L \rangle \,/\, e_0 = \langle v_0, v_1, \sigma_0, \pi_0 \rangle,$$
$$e_l = \langle v_1, v_2, \sigma_1, \pi_1 \rangle, \ldots, e_{L-1} = \langle v_0, v_1, \sigma_{L-1}, \pi_{L-1} \rangle$$

We call the word $ILabel(P) = \sigma_0 \sigma_1 ... \sigma_{L-1}$ the *input label* of the path *P*, and by the *output label* of the path *P* we mean the word $OLabel(P) = \pi_0 \pi_1 ... \pi_{L-1}$. A *cycle* is a path with identical start and finish: $v_0 = v_L$. We describe a complete working example of a production cycle FSM below; for more information on FSMs we refer to Lothaire [10].

We model the production cycle of each agent as a finite state machine. The state of the FSM corresponds to a certain stage of the production process. The initial state corresponds to the zero-stage of the production; the product then travels from one production department to another, as the FSM travels from one state to another.

Each state transition corresponds to performing a certain production stage and, as such, requires a certain amount of resources. We add an internal resource pool for each agent and consider the resources necessary to complete a certain production stage as the input event for the corresponding transition. The transition may take place only if enough resources are available.

A product travels from one production stage to another until it is ready. A complete product corresponds to a terminal state of the FSM. Then the product goes to a "warehouse" where it awaits deployment to the market. In the FSM terms, we model a certain production stage as a transition from a terminal state to the initial state; to this transition we associate the resources necessary to store and transport the complete product to the market.

Each production stage produces something; thus, we add to each edge a set of *output* resources that get produced during this transition. It is natural for finite state machines to have both input and output (entry and exit) actions; in our model, the entry action consumes resources, while the exit action produces new resources.

Besides a natural representation of a real-life company, the FSM formalism has several formal advantages. First, it easily allows an agent to be flexible. In the real world, an agent can reorganize his or her company to produce different commodities (at the very least, different kinds of a commodity). In the FSM model, we allow the FSM to be non-linear (to have forks). Depending on the path an agent takes, different commodities will be consumed and produced.

Second, while in the real world agents may adapt their strategies to changing market conditions, it takes time and resources to complete the adaptation. In the FSM model, this "transition cost" is modeled naturally: if a FSM is currently in a non-terminal state, it will take time and resources to reach the terminal state before the agent can choose a different path starting from the initial state.

Finally, in the real world the agents' behaviour is only suboptimal: agents do not possess complete information about the market and sometimes cannot compute the absolute best strategy. With this in mind, we model the agents as having rather simple strategies. An agent is modeled as a pair $\langle FSM, S \rangle$ consisting of a FSM and a strategy *S*.

2.2. Modeling the market

In this subsection, we describe how the market itself and interaction between agents are represented in our model. Note that [5] outlines two practical approaches to determining prices determination: *true order book* (where agents post offers to buy and sell goods) and agents' *random bump mechanism* (bumped agents trade if it benefits them both). The technique proposed in this paper combines the two approaches introduced in [5]. We model the market as a virtual "bulletin board" where each agent may post an offer indicating that he is willing to sell a certain amount of a certain product for a certain price. When an agent needs to buy a certain product, he queries the market for the selling offers on this particular product. To model insufficient information, we have the market to return only a certain random subset of the offers (the offers that this agent "knows of"). The agent then reviews these offers, chooses the most profitable, and satisfies them (buys the necessary product).

In economics, two goods are said to be *substitutes* if they can be used instead if each other. An increase of demand for the first substitute implies a reduction of demand for the second. On the other hand, *complementaries* are goods that are frequently consumed with each other, and raising the demand for the first one would also raise the demand for the second. The *cross-elasticity*:

$$E_c = \frac{Q_1^B - Q_0^B}{Q_1^B + Q_0^B} \frac{P_1^A + P_0^A}{P_1^A - P_0^A}$$

helps to describe substitutes and complementaries mathematically. Here Q_0^B, Q_1^B are demand values for good *B* before and after the price of commodity *A* changed, P_1^A, P_0^A are prices on the commodity *A*. The cross-elasticity is positive for complementary goods and negative for substitutes. In this model, each agent is both a customer and a producer (people are considered as suppliers of labor), and complementary and substitute goods can be interpreted with FSMs. In the model, complementary goods will often appear on consecutive edges of the production FSM, while substitute goods will appear on edges of parallel paths of the FSM. This results in that whenever an agent buys a certain commodity, his demand for the complementaries increases (he is halfway through the production, so he needs to complete it), while his demand for the substitutes decreases (he has already come through a certain path, and he will not return to a parallel edge in this cycle).

Finally, let us present a complete example of a production cycle in our model. The FSM belonging to an agent *D* is depicted on Fig. 1. The initial state is marked as 0, while 3 and 5 are terminal states. Labels with – and + denote resource costs and resource outcome, respectively; the example has four different resources, from `R1` to `R4`. Here is a sample production cycle.

1. *Registering offers*. Suppose that other agents F_1 ... F_N have registered the following offers:

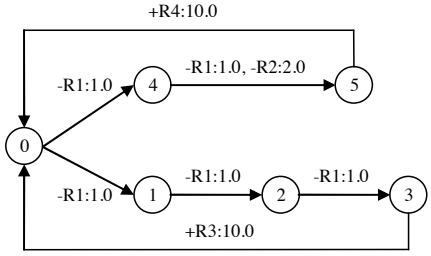

Fig 1. A sample production FSM

```
<R1, price=1.0, amount=12.0>,
<R1, price=2.0, amount=34.0>,
<R1, price=3.0, amount=45.0>,
<R2, price=10.0, amount=67.0>,
<R3, price=4.0, amount=89.0>.
```

2. *Deciding upon the production path*. Agent *D* analyses situation on the market and decides that the path {0, 1, 2, 3} is expected to be better than the path {0, 4, 5}.

3. *Buying resources*. To begin his production cycle, *A* needs to buy 1.0 of the resource `R1`. He sends a request for purchasing `R1`. The market randomly selects a subset of registered propositions, say,

```
<R1, price=1.0, amount=12.0>,
<R1, price=3.0, amount=45.0>,
```

and shows this set to *D*. Agent *D* chooses the most profitable proposition, with the price of 1.0, and makes his purchase.

4. *Transition*. Agent *D* now has the necessary resources to complete the first step of his strategy and move to state 1. Note that after these steps agent *D* is completely committed to his chosen production path; he cannot switch to producing `R4` before completing this round of `R3`, even if in state 2 he suddenly discovers that the market is desperately lacking `R4`, and the prices could be exorbitant. We now skip a few steps as *D* gathers all necessary resources and completes his production cycle. Suppose that *A* has just made the transition from state 3 back to state 0 and obtained 10.0 units of `R3`.

5. *Registering a selling offer*. As *D* now has a surplus of `R3`, he will be willing to register an offer to sell `R3`. Again, we do not precisely specify the decision rule, but as input *D* should consider both his production cost and the market situation which he is able to peek by viewing the (random subset of) offers on `R3`.

After this step, the production cycle repeats. *D* is now able to choose another path.

2.3. Strategies of the agents

As we have already noted, agents use an ant colony optimization (ACO) algorithm choose what and how to produce (what path of the FSM to choose). ACO is a kind of swarm intelligence algorithms based simulating an ants colony behaviour; see Dorigo and Stützle [11] for more details.

Adapting ACO for use with FSM that models production process, we obtain the following algorithm. Each agent owns a number of ants. Each ant emulates its owner's random behaviour. It means that an ant has the same production FSM as its owner, but it makes random decisions when choosing a path. If an ant gets profit, it puts pheromones on his last path, the amount of pheromone being proportional to the profit. As a result, more profitable paths contain more pheromones, and in the end the only thing the real agent has to do is to choose a path with maximum amount of pheromones. We also make the pheromones to evaporate with time in order for the agent to be able to update its behaviour if environment changes.

Here is the pseudo-code of an agent's behaviour:

```
out = maximizeFeramon(cs.getOutcomes())
market.purchase(out.getRequirements())
cs = out.getDestinationState()
for r, v in out.getProduced():
    p = choosePrice(r, market)
    market.registerSellProp(r, v, p)
```

where `cs` is a current state of agent's FSM, `market` is an implementation of virtual market described in 2.2. Implementation of method `choosePrice(…)` is guided by the formula:

$$P = \alpha \cdot Gr \cdot AC + (1 - \alpha) \cdot \max(AC, P_{alt})$$

where $Gr \geq 1$ measures an agent's greediness, P_{alt} is an expected price of the resource in the market, AC is the average cost of produced resource and $\alpha \in [0..1]$ measures how much the agent wishes to consider other agents' costs. We use $\max(C, P_{alt})$ to forbid making price less than cost (being unprofitable).

2.4. Motivation of the FSM approach

Using finite state machines allows for modeling the agents' production process more accurately. It provides an opportunity to naturally model such relations between commodities as substitutes and complementary goods. It also makes agents to think a few steps forward when they make decisions.

An agent whose production is modeled with a FSM cannot switch from one production cycle to another in the middle of production because it has to wait until a terminal state is achieved. This is another real world property that gets naturally modeled in our approach.

On the other hand, FSMs are a well known and deeply studied mathematical object, and there exist many highly developed mechanisms of analyzing their behaviour. For example, the ACO algorithm is a natural fit for learning optimal strategies.

3. EXPERIMENTS

We have implemented the FSM market model and performed experiments that were aimed to check the basic economic facts about the model; they should serve as "sanity checks" for our modeling method. As a general result, the model turned out to be extremely viable, in clear-cut cases always performing just as the economic theory predicts. Thus, we expect it to have some predictive power as well, but this should be supported by further practical experiments. In this section, we describe the experiments we have carried out.

A common scenario of an experiment consists of the following steps: generating FSMs, assigning FSMs to agents, distributing initial cache among agents, introducing the *initial resources merchant* (this is an agent which sells resources necessary for all other agents to perform theirs first steps), running a number of steps, and collecting results data.

Generated automata look much like the one on Fig. 1 (but more complicated); they consist of a number of production cycles starting with an initial state and finishing with a terminal state. Some pairs of production cycles have a common prefix of length more than one. For example, if we add vertices 6, 7 and the cycle {0, 1, 2, 6, 7, 0} to the automata on Fig. 1, then cycles {0, 1, 2, 3, 0} and {0, 1, 2, 6, 7, 0} will have a common beginning with length 3, so the agent will have to make a choice not only in the initial state. Almost all edges have nonzero income resources and zero outcome

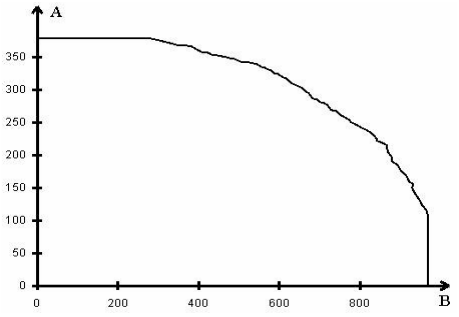

Fig 2. The production-possibility frontier

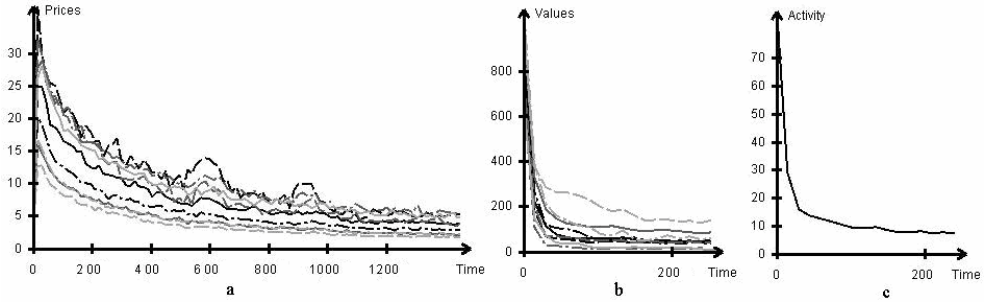

Fig 3. Overstocking: a – prices; b – gross sales; c – active agents

resources; nonzero outcome is allowed only for edges from a terminal state to the initial state. The precise way to assign income and outcome (I/O) resources depends on the purpose of an experiment.

3.1. Production-possibility frontier

The production-possibility frontier is a curve that shows the maximal volume of producing a certain commodity dependent on the level of production of another commodity. This function should be decreasing, as increasing the level of production of a certain commodity should lower the amount of resources the market is ready to assign to producing other commodities. Besides, it should be convex, as in shifting from commodity *A* to commodity *B* the market would first shift the resources that are most useful for *B* and least useful for *A*.

Our experimental scenario included two commodities: *A* (automobiles) and *B* (blankets). Each agent is able to produce both commodities, and agent's FSM has several paths corresponding to different output ratio (automobiles)/(blankets) of the commodities that this agent produces during a production cycle. Moreover, each agent has a personalized production FSM that has its own (randomly varied) production costs and *A*/*B* ratios. A certain "command center" orders some agents to switch production from *A* to *B*, and the agents (with some delay, as described above) switch production; the remaining agents act in their own interest. Fig. 2 shows the results of this experiment: the production-possibility frontier is almost convex, with small fluctuations that can be related to suboptimal behaviour of the agents.

3.2. Overstocking

In this experiment we randomly assign I/O resources to edges but the expected output of all production cycle of all agents covers expected input and has some surplus. In other words, if all agents sequentially use all their production cycles, they will not only cover all their costs of all resources, but get a profit. In the real world, after an initial surge the overproducing warehouses would be full, and prices would experience a steady monotone decrease as the overproduction continues. As prices drop, some agents should become unprofitable and should quit the market (switch to the strategy of producing and consuming nothing), which gradually compensates overproduction.

Fig. 3 depicts the results of the experiment; the graph on the left shows how prices behave with time, the graph in the middle shows how many agents are active (produce something) at this time, and the graph on the right shows the gross sales volume. Both curves behave just as predicted reach an equilibrium.

3.3. Equilibrium

In the real world, overproduction is natural, and it is naturally compensated by consumers who buy end products. To model this situation, we introduced a special agent (representing the consumers) that buys surplus resources. Fig. 4 shows that in this case, equilibrium is reached faster, and both sales and prices stabilize at higher levels.

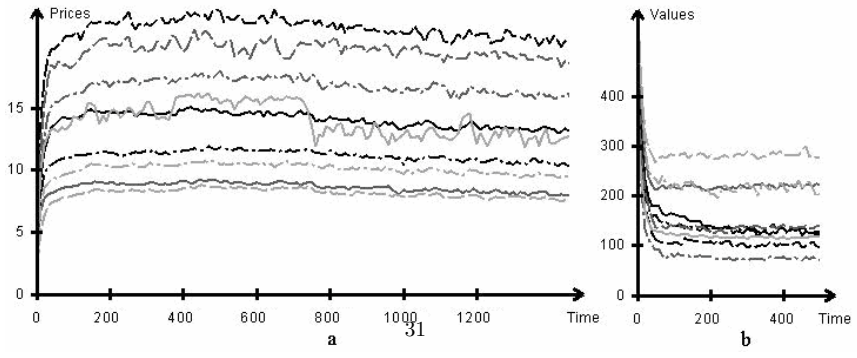

Fig. 4. Equilibrium: a – prices; b – gross sales volume

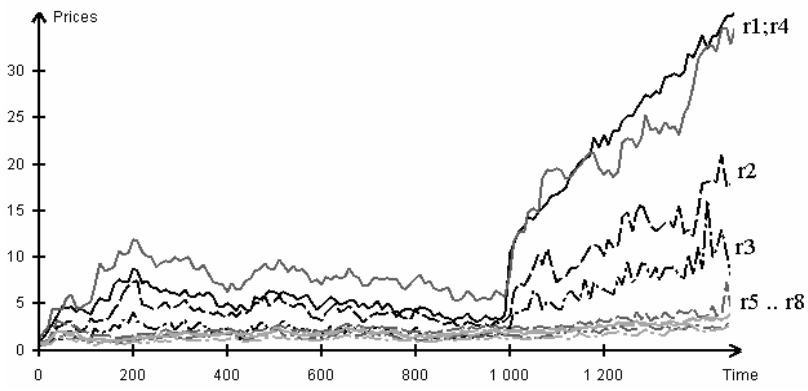

Fig. 5. Prices on a group of commodities: raw materials and end products

3.4. Raw materials and end products

In this experiment we divided nine commodities of the model into two groups: {r0 .. r3} and {r4 .. r8}. We assume that production of a commodity depends strongly on other commodities of its group and to a much lesser extent on the commodities of the other group. When we generate a production cycle for an agent's FSM in this experiment, we randomly choose an output resource for the cycle. When generating input resources for a certain cycle, we choose a resource of the same group as the output resource with much greater probability than a resource from another group.

At time 1000, we introduced an agent who buys lots of r0, thus raising prices. As a result, prices for r1 .. r3 also rise, while prices for r4 .. r8 stay virtually the same. Fig. 5 shows the price graph.

3.5. Complementaries and substitutes

In the first part of this experiment, `r0`, `r1`, and `r2` are complementaries, that is, we increase the probability that they appear on consecutive edges of the production FSMs. An increase in the price of `r0` should cause a decrease in demand on `r1` and `r2`.

Fig. 6 shows what happens if at time 1000 we introduce an agent who buys lots of `r0`. The market reacts by rising all prices except for r1 and `r2`: since producers buy less `r0`, they need less `r1` and `r2`, too. Fig. 7a shows the same graph with only r0 .. r2 visible. And Fig. 7b shows the results of the same experiment with three commodities left on the market, among which `r0` and `r1` are complementaries: an increase in the price of `r0` causes an increase only in the price of `r2`.

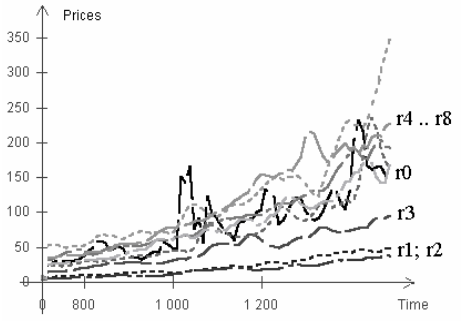

Fig. 6. Prices on complementary commodities

In the second part of this experiment, we introduce `r0`, `r1`, and `r2` as substitutes, that is, increase the probability that they appear on parallel paths in production FSMs. Here we use the following strategy for generating an agent's FSM. We begin with randomly assigning input resources for edges from the first production cycle of the FSM. When we generate an input resource for the edges from following cycles we are more likely to choose `r0 .. r2` if a similar edge from the first cycle has `r0 .. r2` as input resource. For example, if we choose r1 for edge `<1, 2>` from FSM from Fig. 1, then we will be more likely to choose `r0 .. r2` for edge `<4, 5>` than `r3 .. r8`. Doing so, we ensure that resources `r0 .. r2` are alternatives, substitutes from the agents' outlook.

At time 600, we introduced an agent who buys lots of `r0`, thus increasing its price; as a result, agents begin to use less `r0` and more `r1` and `r2`. Fig. 8 shows the demand volumes for experiment with substitute commodities.

4. CONCLUSION

In this paper, we introduced a novel approach to modeling agents that act as producers and consumers on a market. We model an agent's production as a finite state machine with spent and produced resources as entry and exit actions. We have shown that a finite state machine is a natural way to model a company's production, and have shown how to model the basic properties of an economic agent and basic types of market commodities via finite state machines. We have implemented the proposed model and performed experiments in order to verify that the model works, and it works indeed: in all experiments the model behaved as expected.

As with any other model, we can point out the limitations of our approach. The primary limitation is that to create adequate results, the FSM agent-based model requires a large number of independent agents. This means that the model is applicable only to perfect competition markets and monopolistic competition markets.

This paper is, to a large extent, a proof of concept, evidence in support of the validity of the FSM model. Further work should deal with different learning algorithms and

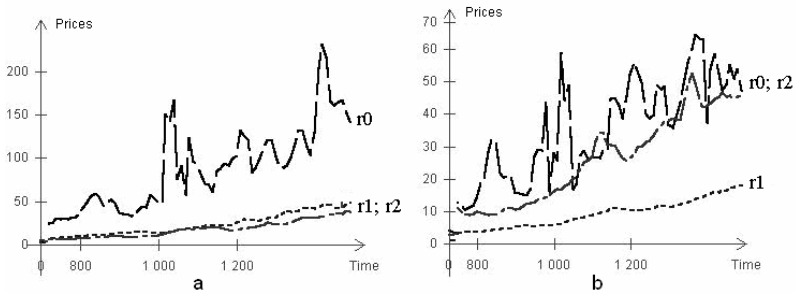

Fig. 7. Prices on complementary commodities: a – a group of three commodities; b – a group of two

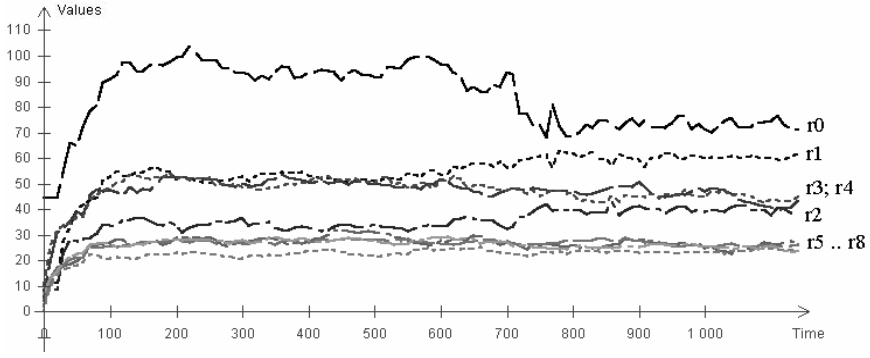

Fig. 8. Demand on substitute commodities

different pricing strategies for the agents. We plan to experiment how different strategies compete with each other and how well different FSM learning algorithms perform in this model. The model provides a natural competitive environment for testing various FSM learning algorithms against each other. And, of course, the ultimate test for our model would be to learn on real data (for example, on the stock market data) and test its predictive power.

REFERENCES

[1] F. Black. Exploring General Equilibrium. *The MIT Press, Cambridge, Massachusetts*. (1995).

[2] G. Zimmermann, R. Neuneier and R. Grothmann. Multi-agent FX-Market Modeling Based on Cognitive Systems. In: *Proceedings of the Artificial Neural Networks, ICANN 2001, Lecture Notes in Computer Science, Vol. 2130, pp. 767–774*. (2001).

[3] S. Kooths. Modelling Rule- and Experience-Based Expectations Using Neuro-Fuzzy Systems. In: *Computing in Economics and Finance – Fifth Conference of the Society for Computational Economics, Boston*. (1999).

[4] S. Kooths, T. Mitze and E. Ringhut. Forecasting the EMU Inflation Rate: Linear Econometric Versus Non-Linear Computational Models Using Genetic Neural Fuzzy Systems. *Advances in Econometrics, Vol. 19, pp. 145–173*. (2004).

[5] B. LeBaron. Agent-based Computational Finance, in Handbook of Computational Economics, L. Tesfatsion and K. Judd (eds.). *North-Holland, Amsterdam, pp. 1187–1232*. (2006).

[6] L. Tesfatsion. Agent-Based Computational Economics: A Constructive Approach to Economic Theory, in Handbook of Computational Economics, L. Tesfatsion and K. Judd (eds.). *North-Holland, Amsterdam, pp. 831–880*. (2006).

[7] I. Rudomin, E. Millan and Hernandez, B. Fragment shaders for agent animation using finite state machines. *Simulation Modelling Practice and Theory, Vol. 13, Issue 8, pp. 741–751*. (2005).

[8] A. Shalyto and N. Tukkel. SWITCH-Technology: An Automated Approach to Developing Software for Reactive Systems. *Programming and Computer Software, Vol. 27, No. 5, pp. 260–276*. (2001).

[9] L. Naumov and A. Shalyto. Automata Theory for Multi-Agent Systems Implementation. *Integration of Knowledge Intensive Multi-Agents Systems (KIMAS'03), IEEE Boston Section, pp. 65–70*. (2003).

[10] M. Lothaire. Applied Combinatorics on Words. *Cambridge University Press*. (2005).

[11] M. Dorigo and T. Stützle. Ant Colony Optimization. *MIT Press, Cambridge*. (2004).