Computing and Informatics, Vol. 34, 2015, 1234–1256

MULTI-AGENT BASED MODELLING OF AN ENDOGENOUS-MONEY ECONOMY

Ivan Blecic, Arnaldo Cecchini, Giuseppe A. Trunfio

Department of Architecture, Planning and Design
University of Sassari, 07041, Alghero, Italy
e-mail: {ivan, cecchini, trunfio}@uniss.it

Abstract. We present an agent-based model of a simple endogenous-money economy. The model simulates agents representing individual persons who can work, consume, invent new products and related production technologies, apply for a loan from the bank and start up a business. Through the interaction of persons with the firms, we simulate the production of goods, consumption and labour market. In order to achieve a significant level of realism of the simulations, the firms are modelled as adaptive agents using an effective reinforcement learning approach in continuous space. This setting allows us to explore how an endogenous-money economy can be built up from scratch, as an emergent property of actions and interactions among heterogeneous agents once money is injected into a non-monetary self-production (or barter) economy. In the paper, we first empirically investigate the learning capability of the firm agents. Then, we discuss the results of some computational experiments under different significant scenarios.

Keywords: Agent-based computational economics, endogenous-money economy, heterogeneous agents

1 INTRODUCTION

The endogenous-money approach to modelling of an economic system assumes that banks create money by providing loans to firms, thus simultaneously creating equivalent deposits [1, 2]. There may, of course, be restrictions on how much money the banks are allowed to create in this way, for example by way of some reserve constraints. But this approach, in the spirit of Moore [3] and Holmes [4], allows banks to look for the reserves later *after* they have extended the credit.

Such endogenously created credit money is the point of ignition of all economic activities as it allows firms to hire workers, start the production, pay wages, dividends and interests, and subsequently have workers, capitalists and bankers consume the goods produced.

The idea of such a closed monetary economy has raised two puzzles: how are profits possible if firms need to repay loans with interests? [5]; and more in general, is an ever-increasing supply of money the only way to earn profits in a steady-state economy? Keen [1, 2] has proposed a model with a set of differential equations which shows on the aggregate level that indeed no such increasing supply of money is necessary.

Based on this approach, we have developed an agent-based model of endogenous-money economy to study its evolution on disaggregated level of individual firms, banks, workers and consumers and to account for possible heterogeneities of products, production technologies, workers' skills and consumption preferences which are all common features of any real-world economy.

Agent-based computational economics (ACE) is a growing field of economic modelling [6], and several large scale models have been developed addressing macroeconomic policy issues (see [7, 8] for a review). In the ACE literature, there is an increasing use of learning techniques which make agents adaptive to their environment and capable to pursue their goals (such as, for example, to maximize long-run profits) [9]. A broad range of algorithms to represent the learning processes of computational agents has been reported in literature, including genetic algorithms [10, 11], genetic programming [12], reinforcement learning (RL) [13], classifier systems [14] and many others [15, 16]. A frequent approach is based on RL which, in the context of ACE applications, represents a form of bounded rationality in decision making [17, 13, 18, 19]. In models based on RL, agents learn through interaction with each other and with the environment. Over time, on the basis of positive reinforcement of profitable actions and negative reinforcement of unprofitable actions, they discover which actions under which conditions provide the best rewards.

In order to maximize the expected profits in the long run in our model of endogenous-money economy, firms need to make decisions on both the price of the product and the quantity to be produced. Therefore, to provide a degree of realism in firms' behaviour, we endow them with a learning capability. To this end, we use the RL approach called SARSA [20], which is a variant of the classic Q-learning method [21, 22]. To cope with the size of the state space, which is assumed as continuous, we adopt an approximation of the state-value function based on a feed forward neural network trained on-line by the agents. In the model, besides firms as adaptive agents, we also have a population of consumers which behave reactively, playing the role of defining the environment in which the former operate [23].

The purpose of our model is not so much to support policy design starting from an initial scenario populated with persons, firms and banks calibrated on a real-world economy. We rather wanted to explore if and how an endogenous-money economy may build up from scratch, as an emergent property [24] of actions and interactions among heterogeneous agents, once the money is being injected into a non-monetary self-production (or barter) economy.

This is how the remainder of the paper is organised. In the following section we specify the model. Then, in Section 3 we investigate the implemented learning mechanism and discuss the results of several computational experiments under different scenarios. Finally, in the concluding Section 4 we examine some advantages of the agent-based modelling approach and present plans for future developments.

2 THE MODEL

There are three types of agents in our model: persons, firms and a bank. At the beginning of the simulation, the world is populated only by a set of persons. *Person* represents an individual who can consume, work for firms and start a business by creating a new firm. *Firm* employs workers, runs the production and sells the produced goods. Then, at every time step, it pays the wages to workers, the dividends to the person who owns the firm, and the interests on the loan to the bank. The *Bank*'s role in the model is to evaluate the business plans for new firms proposed by persons, and to provide loans to the most promising ones.

In what follows we describe the behaviour, the decision-making procedures and the interactions among the three types of agents.

2.1 Production

For simplicity, we assume that no physical capital is used in the production, so the only production factor is labour. The workforce is differentiated, since each person is initialised with n skills whose values are randomly generated from a normal distribution. Firms produce differentiated consumption goods using technologies each represented by a Cobb-Douglas production function. In particular, given a set of workers each endowed with n skills $s_{w,i}$ $(i = 1, \ldots, n)$, the quantity produced with a specific technology is given by:

$$q = k \prod_{i \in \{\text{skills}\}} \left(\sum_{w \in \{\text{workers}\}} s_{w,i} \right)^{\alpha_i}, \tag{1}$$

where k and α's are specific parameters of the production technology.

2.2 Consumption

We assume a person's utility from the consumption of x_i quantities of n goods is given by the following constant elasticity of substitution (CES) utility function:

$$U(x_1, \ldots, x_n) = \left(\sum_{j \in \{goods\}} u_j x_j^{\rho} \right)^{\frac{1}{\rho}}, \tag{2}$$

where the share parameters u for the different products and the ρ are variable among persons. This specification of the consumption utility function is grounded on the assumption that firms may produce different products, what is an important feature of our model.

We assume the consumers are rational utility-maximising agents. Therefore, given a set of available goods, their prices p_i and the available budget B the person decides to spend on consumption during each time step, the utility-maximising consumption bundle is determined by:

$$x_i = B \frac{\left(\frac{p_i}{u_i}\right)^{\frac{1}{\rho-1}}}{\sum_{j \in \{\text{goods}\}} u_j^{\frac{1}{1-\rho}} p_j^{\frac{\rho}{\rho-1}}}, \tag{3}$$

2.3 Invention of New Products and Production Technologies

At each time step, there is a probability each person invents a new product and the related production technology. A newly invented product is defined by a randomly generated value which represents the average consumers' utility parameter for that product (i.e. the average value of the share parameter u for that product in the utility function defined by Equation (2)). In other words, if the product is put into production, this value is used as the mean of a probability density function through which we randomly assign a unique u for that product to each person. In a similar manner, the invented technology is defined by randomly generated parameters k and α's of the production function in Equation (2). Once the product and its production technology have been invented, the person applies for a loan from the bank.

2.4 Ranking of the Applications for Loan

We assume, unrealistically but for the sake of simplicity, that there is just one bank in the system, creating credit money through loans to persons and firms. The bank evaluates the applications for loan by persons and decides which to finance. This evaluation is based on the start-up business plan which provides the quantity produced by one adequately (technology-wise) skilled worker and the product's average utility parameter. The bank finances the most promising business plan by providing the loan to the person who then creates the firm.

2.5 Labour Market

Given that each production technology is defined in terms of workers' skills (see Equation (1) above), the firms express different demand for workers with different skills. Each firm pays the same wage to all its workers, so in order to decide which persons to hire and at what wage, the firm uses a heuristics to select the most productive workers, technology-wise, with respect to their reservation wages. For already employed persons, the reservation wage is the wage they are currently working for, while for an unemployed we establish a baseline minimum wage for which the person is willing to accept a job instead of staying unemployed.

This approach was devised to model the competition for workers among firms. The competition grows as the number of unemployed drops, which is then reflected in the upward wage pressures.

2.6 Firm's Production and Pricing Decision-Making

As illustrated in Figure 1, depending on its current workforce, at each time step t a firm produces the quantity q of the product, according to Equation (1). The production replenishes the warehouse, and the current stock of product therein is denoted by w. Products are sold at a price p, and a portion of the stock is purchased by persons. The quantity purchased by each person is determined by Equation (3). Also, each firm has an amount of cash at hand, indicated by m.

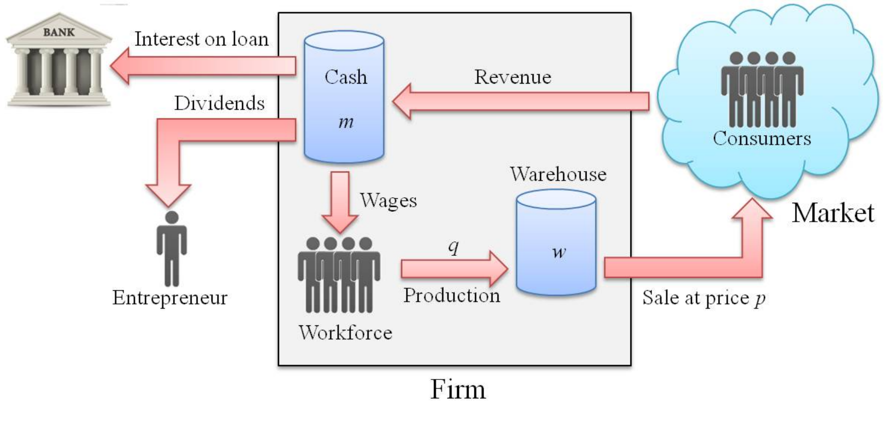


Figure 1. Scheme of the relationship between the entities involved in the model

At each step, the available cash increases with the sales revenues, and decreases with the wage payments to workers, interests on the loan to the banker, and profits to the entrepreneur. The latter is calculated as a fraction of the current value of m.

Given the general turbulence in the system (new firms get created or go bankrupt, there is a competition among firms for consumers' money and for skilled labour, wages and consumption rates may change, and so on), a price established in the past does not necessarily clears the market, nor a production plan stays optimal for a long time. Therefore, in our model the firms are adaptive agents using an on-line RL approach [13] to develop and continuously adapt a suitable policy of business decisions, which consist of setting the price p and the production quantity q, to achieve a satisfactory level of profits accumulated during the simulation.

RL is a typical approach to have software agents find optimal strategies of behaviour when there are only reinforcement (i.e. reward) signals which do not immediately point at the 'right' actions to perform, but only indicate how well the agent is performing. That is to say, the learning consists of a process in which the agent converges to the optimal policy through trials and errors. Usually, in a first phase of the learning, the agent has a higher tendency to explore the available actions, while in a subsequent phase it better exploits the learned policy while still being able to adapt it according to the received rewards. Besides ACE, RL algorithms have been successfully used in many applications, including games [25], price settings [26], robotics [27] and control tasks [28].

We use agents with adaptive abilities mainly for greater realism of the simulation. Such an improvement in the quality of model outcomes was previously reported in the literature for similar applications [26, 29]. For example, in [26] it was shown that sellers capable of learning, and thus of anticipating the longer-term effects of their actions, are less prone to engage in a *recursive price war*, which used to be observed in a population of simple *reactive* agents [30]. A further reason comes from one peculiarity of our model. In fact, we admit firms to go bankrupt. Liquidity kills them quickly, as soon as their cash reserves are not enough to make payments due: wages to workers and interests on loan to the bank. In other words, to stay in business each firm must adopt a strategy that tends to maintain a safety level of cash, in spite of the turbulence of the system.

As specified below, we use a variation of the Q-learning algorithm [21, 22], which is a technique for learning an action-value function which can estimate the long-term expected reward of taking an action for each state. Having such an action-value function, the optimal policy consists of selecting the action with the highest expected reward, given the state. However, to promote a better exploration capability of the agents, a certain level of randomness of choice of action is usually maintained.

According to a well established literature [22], for a single agent operating in a Markovian environment (i.e. in a system which evolves according to a Markovian transition probability matrix) there are theoretical results assuring, under certain conditions, the convergence of the Q-learning process to the optimal policy. However, given the description above, in the proposed model firms operate in a non-stationary and history-dependent environment for which the convergence results concerning learning procedures in Markov decision processes do not hold. Nevertheless, for the purposes of the present study it is not required that each firm constantly apply an optimal policy (e.g. the one which maximizes firm's profits). The purpose of agents' learning capability should be rather to discover and implement a good, realistic and adaptive policy.

In our model the firm perceives a set S of states in its environment which include indicators relevant for the firm's decision making. Among the candidate indicators, the agent may consider its own cash and warehouse levels, their past trends, the current price p_t and the production quantity q_t, as well as several indicators of other agents' behaviour. However, in the present version of the model we assume that firms cannot obtain direct information about the characteristics of competitors and that the set S only contains the current trends of the firm's cash amount and warehouse $S = \{\Delta m_t, \Delta w_t\}$. The value of Δm_t is significant since it can highlight a possible imbalance between revenues and costs (i.e. wages and interest payments). Also, the value of Δw_t may indicate if, given the state of the market and the current level of prices and production quantity of the firm, there is an ongoing accumulation of unsold product or if, instead, the sales are greater than the production.

At each step of the simulation, the firm can apply one of the actions a_i in the set A, as specified in Table 1. That is to say, there are five available actions, which correspond to separate increment or decrement of p and q plus a *null action*, which consists of maintaining the current level of both price and production quantity. In the present version of the model, both the price variation Δp and the quantity variation Δq are fixed. Future work may consider to include also the size of such variations in the policy. It is worth noting that to increase its production the firm needs to hire more workers, and vice versa, to reduce the production it needs to lay off workers. Thus, the $\pm\Delta q$ actually applied in practice depends on the productivity of the involved workers.

Action	Specification
$a^{(0)}$	$\{p_{t+1}, q_{t+1}\} = \{p_t, q_t\}$
$a^{(1)}$	$\{p_{t+1}, q_{t+1}\} = \{p_t - \Delta p, q_t\}$
$a^{(2)}$	$\{p_{t+1}, q_{t+1}\} = \{p_t + \Delta p, q_t\}$
$a^{(3)}$	$\{p_{t+1}, q_{t+1}\} = \{p_t, q_t - \Delta q\}$
$a^{(4)}$	$\{p_{t+1}, q_{t+1}\} = \{p_t, q_t + \Delta q\}$

Table 1. The five actions in the set A for the Firm agent. The variables p_t and q_t represent the current price and quantity, respectively, while Δp and Δq are suitable fixed increments

More in detail, at each time step, given the state x_t of the environment the agent: 1. perceives the elements of S, 2. select an action a_t from A and 3. executes it.

After the implementation of an action a_t, the state of the environment becomes x_{t+1} and the firm obtains a reward $r(x_t, a_t)$. The latter is the result of the deterministic behaviour of consumers as described above, as well as of the different actions undertaken by other firms in the past. As for the dynamics of price-quantity adjustments, following [26] it is assumed that firms do not apply all their actions simultaneously. Rather, at each step only one agent undertakes its decision. This is implemented adopting a fixed interval Δt_a of time steps between every two consecutive actions. Clearly, the size of such interval depends on the maximum number of allowed firms. While the actions are applied every Δt_a steps, the evaluation of the reward $r(x_t, a_t)$ related to a_t, as well as the selection of the next action to undertake, are always performed at the subsequent time step $t + 1$. The reasons for such an alternating mechanism lie in 1. a more realistic representation of the reality and 2. its ability to allow a clearer evaluation of the effects of each action on the actual reward, so favouring a more effective learning process.

2.6.1 The Implemented Learning Strategy

In the standard *Q-learning* approach [21] the agents learn Q-functions, which estimate the return associated with each action $a \in A$. At each visited state x_t, the agent undertakes the action a_t and updates the Q-functions as follows:

$$Q(x_t, a_t) = r(x_t, a_t) + \gamma \max_{a \in A} Q(x_{t+1}, a) \tag{4}$$

where $\gamma \in [0, 1]$ is a *discount factor* and $\max_{a \in A} Q(x_{t+1,a})$ is the estimation of the sum of payoffs received from time t onwards, assuming that a *greedy policy* is followed. The latter consists of always taking the action with the highest predicted return.

Typically, at the beginning of the learning the Q-values are unknown (e.g. they are set to some default value). Subsequently the stored Q-values are updated on the basis of the experienced state-action pairs using the received rewards. This is done gradually, using a learning rate $\eta \in [0, 1]$ as follows:

$$Q^{(new)}(x_t, a_t) = (1 - \eta) Q^{(old)}(x_t, a_t) + \eta \left[r(x_t, a_t) + \gamma \max_{a \in A} Q^{(old)}(x_{t+1,a}) \right] \tag{5}$$

The idea is not to completely discard the previous estimate of a Q-value when a new reward is obtained. Instead, the update is carried out as a weighted combination of the old Q-value for that state-action pair and the new information obtained from the environment.

According to Equation (5), the learning process requires some data structure to store the current Q values for each state and each possible action. In general, this can be done using a lookup table when the state space is discrete and composed of few elements. However, as illustrated above, the *Firm* agent of our model has a multidimensional continuous state space and its discretization [21, 31, 13, 32] would imply the use of a large data structure and a significant component of arbitrariness in choosing the size of the discrete sub-spaces. To avoid the discretization and to cope with continuous state spaces, in RL applications it is common the use of function approximation [32, 13]. The approach consists of expressing the Q-values as a function of some relevant state variables in order to obtain, through some generalization capabilities, reliable predictions even for states which the agent has not yet experienced during the simulation. In this study we use a typical approach for approximating Q-values in RL [13, 33, 34, 20], namely Artificial Neural Networks, and in particular multi-layer perceptrons.

In general, in order to operate with an approximate Q-function, a learning algorithm may store a state-action value function $Q : S \times A \to \mathbb{R}$, or a state value function $V : S \to \mathbb{R}$ [20, 35]. In terms of ANNs, the first approach would imply the use of a single network with $|A| + |S|$ inputs and one output, while the second approach could be implemented using $|A|$ networks with $|S|$ input and one output.

As in [34, 20], we choose to approximate the Q-function using one ANN for each action (i.e. five ANNs in total). Each ANN has two inputs (the current values of Δm_t and Δw_t) plus a bias input and one output (the Q-value estimate). We use a single hidden layer with 3 nodes and a tanh activation function. The input and output layers use the identity function. Therefore, since we use fully connected networks, for each action we have 12 weights to learn. As for the ANN architecture, we have tested more and less hidden nodes and different activation functions and found no improvements in the results.

Instead of using Equation (5), which refers to the use of a lookup table, in our implementation learning is done with a gradient descent update using standard backpropagation [36] with a fixed learning rate η. In addition, following the State-Action-Reward-State-Action(SARSA) approach proposed in [20], the target of the ANN learning is defined as:

$$Q(x_t, a_t) = r(x_t, a_t) + \gamma Q(x_{t+1}, a_{t+1}) \tag{6}$$

where, in practice, $Q(x_{t+1}, a_{t+1})$ is used instead of $\max_{a \in A} Q(x_{t+1}, a)$. The method is justified by the fact that in the early stage of learning the ANNs are not sufficiently trained to provide reliable predictions of the total return associated with the state x_{t+1}. Also, there are evidences [37] that the $\max_{a \in A} Q(x_{t+1}, a)$ obtained by the Q-function is likely to be overestimated. Another consideration is that Equation (4) refers to the case in which greedy actions are taken at each step. However, this is not the case of the present application since we maintain a certain level of random exploration of the state space.

To apply the SARSA algorithm, first the new action a_{t+1} is selected through the approach explained in the next subsection, then the weights of the ANN associated to the action a_t are updated using the target computed by Equation (6).

It is worth noting that in the SARSA approach adopted in this study, only the Q-value $Q(x_t, a_t)$ of the previous state with respect of the new one x_{t+1} is updated. However, there are more sophisticated learning strategies, (such as SARSA(λ)) where the reward obtained at the time $t + 1$ can be propagated backwards to many states. Although it has been reported that such methods may learn more efficiently [13], the simple one-step SARSA proved to be quite effective for this application, and more sophisticated methods will be subject to further studies.

2.6.2 Action Selection

Given the Q-function, at each time step the agent can choose between two options: it can select from the set A the action with the highest Q-value, i.e. the so-called *greedy strategy*, or it can take a random action. In the first case the agent exploits what it has already learned in order to achieve the maximum estimated reward according to its current Q-function. In the second case, the agent explores new directions and has the chance to learn more.

As reported in the literature, the greedy method can get trapped in local minima during the convergence process, performing suboptimal actions in the long run of the simulation [13]. However, different approaches are reported in literature for balancing the two objectives of exploration and exploitation. For example, an alternative called ε-greedy consists of applying the greedy behaviour most of the time, but with a small uniform probability ε selecting an action at random.

Another typical approach, called *softmax*, consists of a random selection of the action in which the probabilities depend on a graded function of the corresponding estimated Q-value. In this case, the best action has still the highest selection probability, but all the actions are ranked according to their Q-value estimates. Typically, the softmax method is based on a Boltzmann distribution of probability [13].

In the model illustrated in this paper, after some preliminary comparison between the performance of the *softmax* and the ε-greedy methods, we have chosen the latter which showed a better performance. Furthermore, the ε starts with an high value ε_i, which is gradually and linearly lowered up to a steady state value ε_f after a specified number T_L of time steps. In this way, the Firm agents try to explore as much as possible of the state space at the beginning of the simulation, taking more greedy actions later to exploit what they have learned.

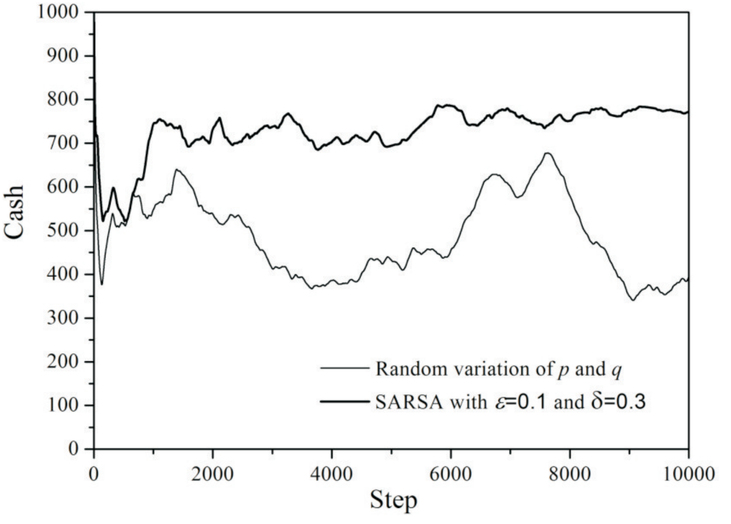
3 COMPUTATIONAL EXPERIMENTS

In this section, we first empirically investigate the learning capability provided by the SARSA RL approach described in Section 2.6. Then we present and discuss the results of some significant run of the model, under different scenarios.

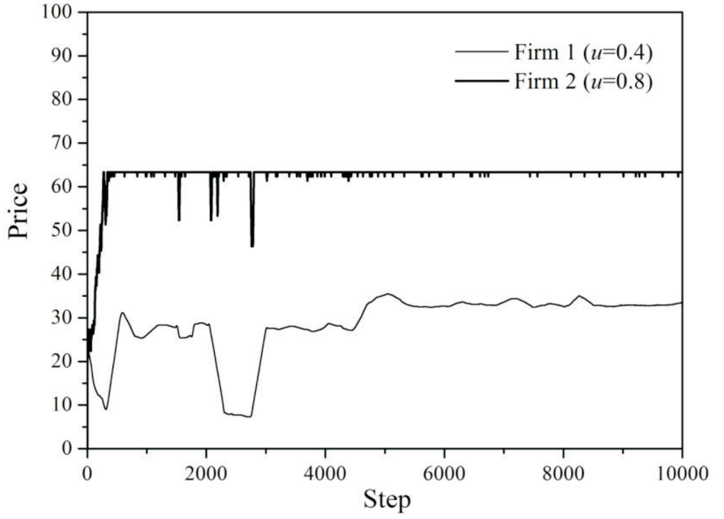
3.1 Investigation of the Learning Process

In order to quantify the effectiveness of the learning process, we first performed a number of runs in which two firms with the same product and the same technology competed for customers in the market. One of the firm was endowed with the learning procedure described above, while the other maintained a constant price and production quantity. Both firms started with the same level of production and the same price.

In the numerical investigation, we carried out five experiments composed of 50 runs with the length of 10 000 steps. Each run was initialized with a different random seed. The value of T_L was set to 1 000 steps, while ε_i was initialized with 0.5 and ε_f with 0.1. The value of the discount factor γ was set to 0.3. The first experiment was performed using a random choice of the action from the set A at each step. In the remaining four experiments, we used different values of the learning rate η for the backpropagation update of ANNs, namely 0.05, 0.1, 0.2 and 0.3. At the end of each run, the average value $\bar{m}$ of the variable m (i.e. the cash amount) in the last 5 000 steps was recorded. Then, for each experiment we computed the means and standard deviations of $\bar{m}$ over the 50 runs. The numerical results are shown in Table 2, while in Figure 2 a) we show an example of the obtained convergence plots for $\varepsilon = 0.1$. Each experiment with $\varepsilon \neq 0$ was compared with the random one using a two-tailed t-test, rejecting the null hypothesis (i.e. equivalence between



a)



b)

Figure 2. a) Comparison between the cash of a firm that uses the SARSA learning algorithm and a firm with the same characteristics which randomly varies the price and quantity. b) Comparison between the prices set by two firms with different products.

the algorithms under comparison) if the p-value was smaller than the significance level 0.05. As shown in Table 2, the implemented SARSA algorithm had the capability of achieving a quite stable level of cash. According to the t-test, the firm endowed with the SARSA mechanism was always better than that with the simple random variation of the price and quantity. This indicates that our adaptive firm can learn a good direction in the decision space.

We carried out many other tests to investigate the quality of the learning process in order to obtain realistic simulations. For example, in Figure 2 b) we show the convergence towards a quite stationary price for two firms with products having different average share parameters u (see Equation (2)), namely $\bar{u} = 0.8$ and $\bar{u} = 0.4$. According to Equation (3), it is reasonable to expect that a firm with the largest value of $\bar{u}$ can increase more its price while maintaining approximately the same market share. Obviously, the exact dynamics is influenced by the fact that the value of $\bar{u}$ is only an average of those of the consumers and that various random processes affect the system. However, it is satisfactory to show in Figure 2 b) that the two agents are able to reasonably set their prices without knowing Equation (3), but only by observing the trend of their sales and cash.

ε	Mean	Std. Dev	Mean	Std. Dev	p
0.05	645.2	79.3	497.2	84.3	0.000
0.10	766.5	56.2	497.2	84.3	0.000
0.20	732.2	71.2	497.2	84.3	0.000
0.30	683.7	86.9	497.2	84.3	0.000

Table 2. Results of the comparison between a firm agent endowed with the SARSA learning mechanism (second and third columns) and a firm with the same characteristics which operate randomly (fourth and fifth columns).

3.2 Model Results

We carried out three computational experiments each ran under a different scenario in relation to the structure of competition among firms and to the relative 'abundance' of workforce. All scenarios were initialised with 200 persons and ran for 500 time steps. The learning rate η used in backpropagation was set to 0.1, and the value of the discount factor γ was set to 0.3. As before, the greedy parameter ε was initialized with 0.5 and decreased up to 0.1 in 1 000 steps.

The first scenario is a monopolistic setting with a single firm (i.e. we set the limit of only one firm financed by the bank during the simulation). The second scenario is a more competitive setting with five firms, each with its product and production technology. As we shall see, this setting does not hit the upper bound of absorbing all the available workforce, due to the interplay of the demand and the structure of production costs related to production technologies. Finally, the third scenario allows unlimited number of firms. In this case, we will see that there is a high demand of workforce, which toughens the competition among firms for the skilled labour.

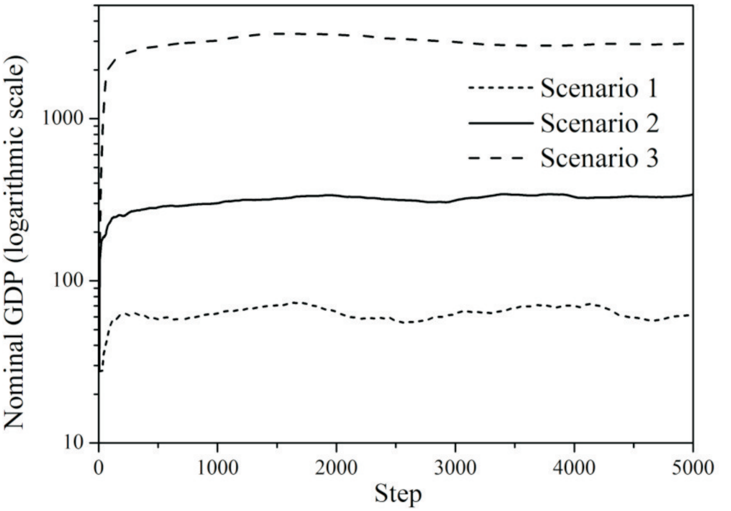
Figure 3 shows the evolution of several macroeconomic aggregates. As it was to be expected, the economy in the third scenario, with no limits on the number of firms, produced the greatest overall output (Figure 3 a)). It is important to remember that firms in principle may be created and die at any time step of the simulation. Multiple Monte Carlo runs of the simulation under the third scenario yielded on average a maximum of around 55 firms operating simultaneously at some point in time, but on average only about 40 ended up active at the end of the simulation, while the others were outcompeted and went bankrupt. These numbers are of course sensitive to the model parameters as well as to the learning parameters used by firms, but they in principle show that given the nature of the production technology and a limited population there is an upper bound of firms that may operate in a steady-state.

The differences among the three scenarios in the overall output are reflected by the unemployment rates (Figure 3 b)). In the case of the monopoly, only a small portion – around 10% – of the available workforce was absorbed by the firm, while in the second scenario 20% was employed at the end of the simulation. In the third scenario the economy was able to employ around 80% of its production potential.

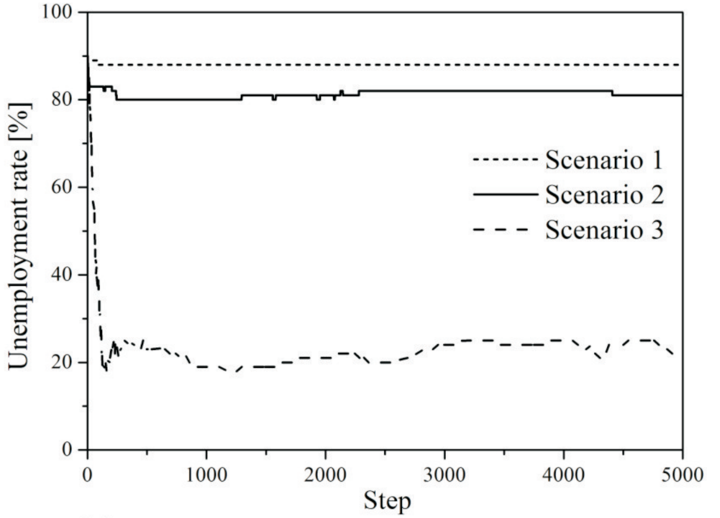
What is relevant here to see in a combination with the unemployment rates is the wages dynamics (Figure 3 c)). In the first and the second scenario the wages paid to workers remained at the level of the unemployment reservation wage. In the case of the second scenario, this of course is partly due to the fact that, given the internal randomness, in this specific simulation run the five operating firms happened to use sufficiently different production technologies (putting different 'weights' on worker skills) in relation to the available pool of skills so as not to bring about a competition among firms for 'rare' skills. Had it been otherwise, there would have been some upward wage pressure. In any case, nothing of the magnitude observed in the third scenario where the final average wage was more than twice the unemployment reservation wage.

We do not model rigidities and transaction costs for hiring, lay-offs and job switching. So, during the simulation under the third scenario we frequently observed highly turbulent intervals of time where workers change jobs from step to step, often back-and-forth among firms trying to outbid the competitors by offering higher wages.

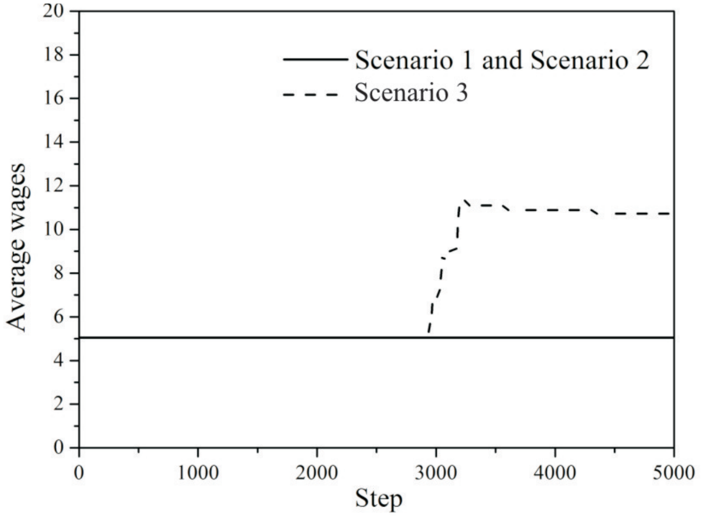
Possibly the most interesting macroeconomic result is told by the Figure 3 d). It shows how the income is distributed among the firm owners and workers (and the banker). Here we represent the dividends paid to firm owners as a share of all the incomes of the economy (which is the sum of dividends, wages paid to workers, and interest payments to the banker). We see that in the case of monopoly the firm owner manages to capture the greatest share of the overall income, followed by the second and then the third scenario. Again, the numbers themselves are not as important as the general story they tell.


Nominal GDP (logarithmic scale)
10
100
1000
0
1000
2000
3000
4000
5000
Step
Scenario 1
Scenario 2
Scenario 3

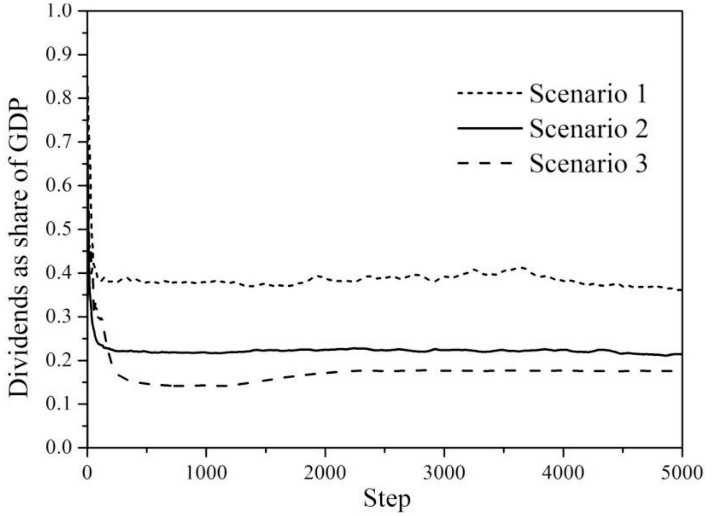
a)


Unemployment rate [%]
0
20
40
60
80
100
0
1000
2000
3000
4000
5000
Step
Scenario 1
Scenario 2
Scenario 3

b)



c)



d)

Figure 3. The evolution of macroeconomic variables of the system under three scenarios: a) Nominal GDP; b) Unemployment rate; c) Average wage; d) Dividends as a share of GDP

Let us turn to some micro analysis. In Table 1. we summarise few descriptive statistics of the firm population at the end of the simulation ran under the three scenarios.

	Scenario 1	**Scenario 2**	**Scenario 3**
Number of firms	1	5	42
Average number of workers	24	8	4
Min. number of workers	24	5	1
Max. number of workers	24	39	18
Median wage	5	5	10

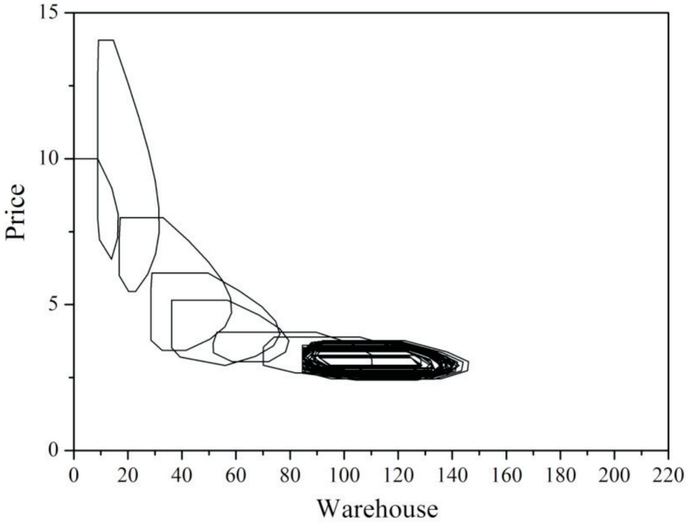
Table 3. Firms at the end of the simulation

An interesting demonstration of a firm's behaviour is the phase diagram of the prices it sets and the corresponding quantities of the product it manages to sell at those prices. In Figure 4 we present such phase diagrams for the firm in the Scenario 1 and for one of the five firms in the Scenario 2. They both start from low production levels and settle for high prices. Then, as they expand the production the trajectory gravitates towards lower price levels.

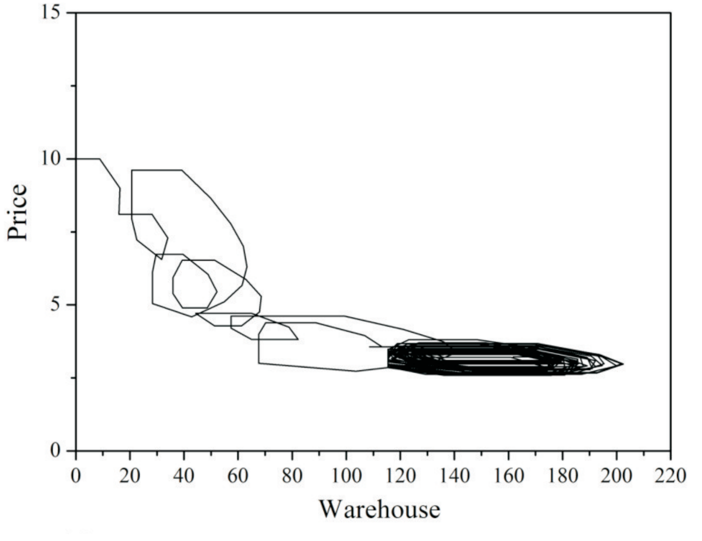
Richer interplays among firms occur when they enter into a stronger mutual competition, as in the Scenario 3. In Figure 5 we show the evolution of the wages and the number of workers hired by two firms under Scenario 3. The firm represented in the Figure 5 a) is the first created in the simulation. During an initial phase, it hires the workers at the unemployment reservation wage of 5, and then increases its workforce up to a maximum of 10 workers. Then, when an attempt to further increase the workforce occurs, the wage level rises. Thus, the firm finds more convenient to go back to the number of 10 workers. This phenomenon may be due to two distinct effects. One is the rise of wages (due to the competition for workers from other firms) which increases the production costs. The other effect is the product competition from other firms which may negatively influence the demand for the firm's products.

A similar phenomenon can be recognized in the change in workforce occurred in the firms represented in Figure 5 b). As shown, the firm increased its workforce up to the number of 18. However, in the meantime, the cost of labour increased due to the competition between firms. Also because of this, the firm chose to reduce production in order to balance revenues and costs.

Finally, we want to make a few more general remarks in relation to the results. On the macro level, our agent-based model produces the same central result of the Keen's aggregate model: in principle it is possible to have a steady-state with a constant flow of monetary profits and wages, while still paying interests on loan to the bank. This of course shouldn't come as a surprise if one understands the difference between a stock and a flow. Our model neatly illustrates the difference: loans are a stock (the total amount of money present in the economy), while profits, wages and interest payments are in a flow, the firms, the banker and the workers juggle back and forth among themselves month by month. Things would break

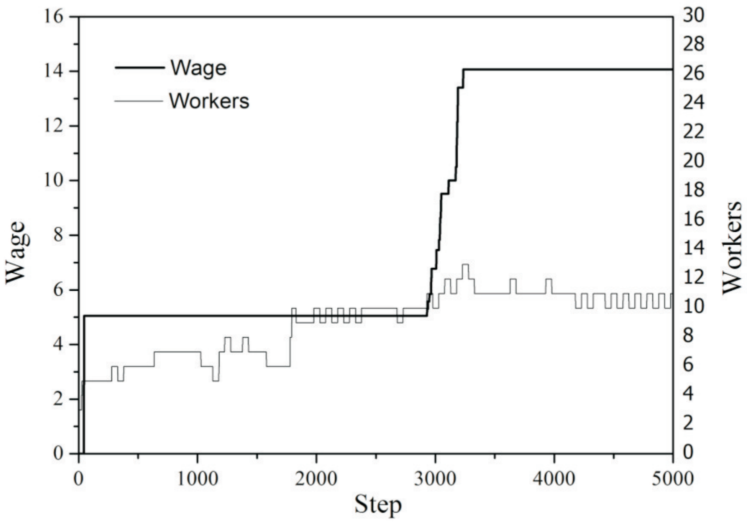


a)

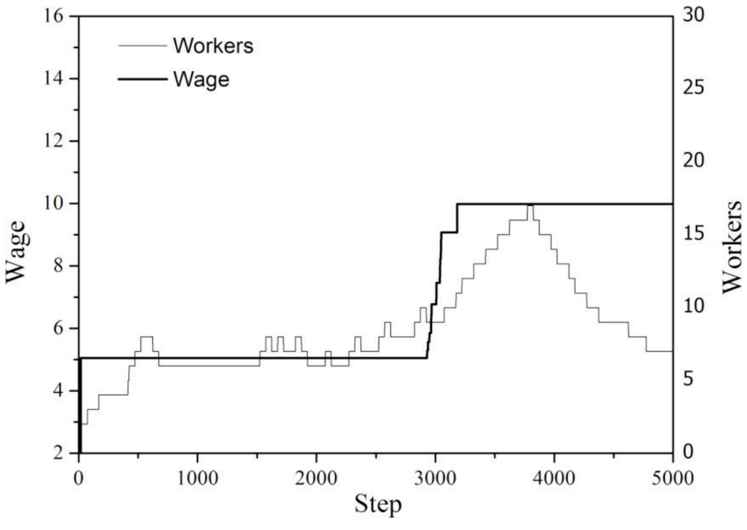


b)

Figure 4. Firm's behaviour in the price-warehouse phase space: prices set by the firm and the quantity of product sold during the simulation. The diagram to the left shows the case of the only firm present in Scenario 1, to the right the case of one of the five firms in Scenario 2.



a)



b)

Figure 5. The number of workers and wages through time steps for two of the firms in Scenario 3

apart if the firms on aggregate start to repay the principal. All is good as long as the firms *on aggregate* pay the interests, but as so soon as the principal repayment is greater than the loans issued, this basically amounts to withdrawing money from circulation which may bring the system to a halt.

Having said that, the sustainability and the possibility of a sustainable steady state of positive profits and wages depend on how the consumption rates, wages and profits are set relative to each other. Of course, total profits, wages and interests are just monetary aggregates, and what is important here are their relative values which tell us the thing that really matters: how the produced goods are distributed among the firms owners, the workers and the banker.

Higher consumption rates (and hence less saving), especially from the workers who make the bulk of it, bring in more cash to the firms, and thus higher profits to their owners. Actually, that may be slightly misleading, for the cash brought into the firms can be used *both* for higher profits or for higher wages (or for a higher interest rate for the banker).

Now, all these conclusions can already be drawn from the Keen's model, so what is our model adding to the story? Instead of just modelling the interaction on the aggregate among three sectors (firms, workers, bank), we provide a microfoundation with having many firms (each producing a different goods) and many workers (with different skills and consumption utility functions, that is to say, different tastes). Here the firms compete for workers' skills, for their money, and for loans. Workers compete to get better jobs. From such a microfoundation the crucial *deus ex machina* element of Keen's model *emerges* – the question of how the surplus is shared between the firm owners and the workers (and the banker). While in his model Keen cannot but postulate that some fraction of the surplus goes to the workers (here Keen references Marx), in our model this fraction emerges *endogenously*, from the interplay of the competition in the consumption and labour market resulting from the numerosity of firms and workers, the diversification of goods, production technologies, and workers' skills, and the role of the bank as a gatekeeper. In other words, what emerges from these interplays in our model is the relative *market power* of agents, which is then reflected in how the economy's surplus is shared among them.

Such endogenous emergence of the market power of agents, we hold, is the primary advantage and advancement of our agent-based modelling approach over the aggregate modelling (like Keen's).

4 CONCLUSIONS

The results we obtained on the macroeconomic level confirm those of Keen's model [1, 2], namely that a constant flow of profits is, in principle, possible in a steady-state economy without an ever-increasing supply of money. But our agent-based approach to modelling endogenous-money economy has, so it seems to us, a few advantages over the aggregate modelling with differential equations and systems dynamics, as it allows several features to arise as emergent properties of the interaction among agents. One notable example is the distribution of the income (and thus of the production surplus) between workers and the firm owners. While this is something that needs to be postulated in the modelling on the aggregate level, it appears instead as an emergent property in our model.

An important distinctive feature of our model is the possibility of products differentiation among firms. This, for instance, is not contemplated in one of the most complete ACE models [38] where no difference in the quality of goods is assumed. Besides the fact this is a notable characteristics of any real economy; some properties exhibited by our model relevant for the economic analysis emerge precisely because of the assumption of product differentiation among firms.

The model we presented is still quite rudimentary and there are plenty of things we plan to develop in the future. One limiting assumption we make in our model is that the production takes place without a physical capital. This greatly simplified our task, for we did not need to model the production of capital and intermediate goods, or the procurement of natural resources. This though comes at a cost, because then we are not able to simulate some features which, we hold, are probably relevant in this context, such as the impact of fixed costs and the related economies of scale.

The financial sector is another area which needs to be entirely developed. Instead of a single bank, we plan to implement multiple banks competing among each other. Finally, in order to make it potentially useful for policy analysis, we would need to model the government sector and to allow for a more realistic representation of different institutional settings.

REFERENCES

[1] Keen, S.: The Dynamics of the Monetary Circuit. In: Rossi, S., Ponsot, J. F. (Eds.): The Political Economy of Monetary Circuits: Tradition and Change. Palgrave Macmillan, 2009, pp. 161–187.

[2] Keen, S.: Solving the Paradox of Monetary Profits. Economics: The Open-Access, Open Assessment E-Journal, Vol. 4, 2010, No. 2, Art. No. 2010-31.

[3] Moore, B. J.: The Endogenous Money Supply. Journal of Post Keynesian Economics, Vol. 10, 1988, No. 3, pp. 372–385.

[4] Holmes, A. R.: Operational Contraints on the Stabilization of Money Supply Growth. In: Morris, F. (Ed.): Controlling Monetary Aggregates. The Federal Reserve Bank of Boston, 1969, pp. 65–77.

[5] Rochon, L. P.: The Existence of Monetary Profits within the Monetary Circuit. In Fontana, G., Realfonzo, R. (Eds.): Monetary Theory of Production: Tradition and Perspectives. Palgrave Macmillan, 2005.

[6] Farmer, J. D.—Foley, D.: The Economy Needs Agent-Based Modelling. Nature, Vol. 460, 2009, pp. 685–686.

[7] Tesfatsion, L.—Judd, K. L.: Agent-Based Computational Economics. Handbook of Computational Economics, Vol. 2, North Holland, 2006.

[8] Fagiolo, G.—Roventini, A.: Macroeconomic Policy in DSGE and Agent-Based Models. Working Paper Series, Vol. 7, Department of Economics, University of Verona, 2012.

[9] Tesfatsion, L.: Agent-Based Computational Economics: Growing Economies from the Bottom Up. Artificial Life, Vol. 8, 2002, No. 1, pp. 55–82.

[10] Miller, J. H.: A Genetic Model of Adaptive Economic Behavior. Technical Report, Working Paper. University of Michigan, Ann Arbor, MI, 1986.

[11] Dawid, H.: Adaptive Learning by Genetic Algorithms: Analytical Results and Applications to Economic Models. Springer-Verlag, Lecture Notes in Economics and Mathematical Systems, Vol. 441, 1996.

[12] Chattoe, E.: Just How (Un)Realistic Are Evolutionary Algorithms as Representations of Social Processes? Journal of Artificial Societies and Social Simulation, Vol. 1, 1998, No. 3.

[13] Sutton, R. S.—Barto, A. G.: Reinforcement Learning: An Introduction. MIT Press, 1998.

[14] Holland, J. H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, Cambridge, MA, USA, 1992.

[15] Gjerstad, S.—Dickhaut, J.: Price Formation in Double Auctions. Games and Economic Behavior, Vol. 22, 1998, pp. 1–29.

[16] Tesauro, G. J.—Kephart, J. O.: Foresight-Based Pricing Algorithms in Agent Economies. Decision Support Systems, Vol. 28, 2000, pp. 49–60.

[17] Roth, A. E.—Erev, I.: Learning in Extensive Form Games: Experimental Data and Simple Dynamic Models in the Intermediate Term. Games and Economic Behavior, Vol. 8, 1995, pp. 164–212.

[18] Weiss, G.: Adaptation and Learning in Multi-Agent Systems: Some Remarks and a Bibliography. In: Wei, G. and Sen, S. (Eds.): Adaption and Learning in Multi-Agent Systems. LNCS, Springer, Vol. 1042, 1996, pp. 1–21.

[19] Noble, J.—Franks, D. W.: Social Learning in a Multi-Agent System. Computing and Informatics, Vol. 22, 2003, No. 6, pp. 561–574.

[20] Rummery, G. A.—Niranjan, M.: On-Line Q-Learning Using Connectionist Systems. Technical Report, Cambridge University Engineering Department, 1994.

[21] Watkins, C. J. C. H.: Learning from Delayed Rewards. Ph.D. thesis, King's College, Cambridge, UK, 1989.

[22] Watkins, C. J. C. H.—Dayan, P.: Q-Learning. Machine Learning, Vol. 8, 1992, pp. 279–292.

[23] Bandini, S.—Manzoni, S.—Vizzari, G.: Agent Based Modeling and Simulation: An Informatics Perspective. Journal of Artificial Societies and Social Simulation, Vol. 12, 2009, No. 4.

[24] Paunovski, O.—Eleftherakis, G.—Cowling, A. J.: Disciplined Exploration of Emergence Using Multi-Agent Simulation Framework. Computing and Informatics, Vol. 28, 2009, No. 3, pp. 369–391.

[25] Tesauro, G.: Td-Gammon, a Self-Teaching Backgammon Program, Achieves Master-Level Play. Neural Computation, Vol. 6, 1994, No. 2, pp. 215–219.

[26] TESAURO, G.—KEPHART, J. O.: Pricing in Agent Economies Using Multi-Agent Q-Learning. Autonomous Agents and Multi-Agent Systems, Vol. 5, 2002, No. 3, pp. 289–304.

[27] PETERS, J.—VIJAYAKUMAR, S.—SCHAAL, S.: Reinforcement Learning for Humanoid Robotics. Proceedings of the 3rd IEEE-RAS International Conference on Humanoid Robots (Humanoids 2003), Karlsruhe, Germany, 2003.

[28] WERBOS, P. J.: Backpropagation and Neurocontrol: A Review and Prospectus. Proceedings of the International Joint Conference on Neural Networks (IJCNN), 1989, Vol. 1, pp. 209–216.

[29] WALTMAN, L.—KAYMAK, U.: Q-Learning Agents in a Cournot Oligopoly Model. Journal of Economic Dynamics and Control, Vol. 32, 2008, No. 10, pp. 3275–3293.

[30] HU, J.—WELLMAN, M. P.: Self-Fulfilling Bias in Multiagent Learning. Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96), 1996, pp. 118–125.

[31] SUTTON, R. S.: Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding. Advances in Neural Information Processing Systems 8, NIPS, MIT Press, 1996, pp. 1038–1044.

[32] SANTAMARÍA, J. C.—SUTTON, R. S.—RAM, A.: Experiments with Reinforcement Learning in Problems with Continuous State and Action Spaces. Adaptive Behaviour, Vol. 6, 1997, No. 2, pp. 163–217.

[33] TESAURO, G.: Pricing in Agent Economies Using Neural Networks and Multi-Agent Q-Learning. Sequence Learning, Springer, LNCS, Vol. 1828, 2001, pp. 288–307.

[34] LIN, L. J.: Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching. Machine Learning, Vol. 8, 1992, pp. 293–321.

[35] WIERING, M. A.—VAN HASSELT, H.: The QV Family Compared to Other Reinforcement Learning Algorithms. Proceedings of IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL '09), 2009, pp. 101–108.

[36] RUMELHART, D. E.—HINTON, G. E.—WILLIAMS, R. J.: Learning Representations by Back-Propagating Errors. Neurocomputing: Foundations of Research, MIT Press, Cambridge, MA, USA, 1988, pp. 696–699.

[37] THRUN, S.—SCHWARTZ, A.: Issues in Using Function Approximation for Reinforcement Learning. Proceedings of the Fourth Connectionist Models Summer School, Erlbaum, 1993.

[38] CINCOTTI, S.—RABERTO, M.—TEGLIO, A.: The EURACE Macroeconomic Model and Simulator. In: Aoki, M., Binmore, K., Deakin, S., Gintis, H. (Eds.): Complexity and Institutions: Markets, Norms and Corporations. Palgrave Macmillan, 2012.

Ivan Blecic is Researcher in the Department of Architecture, Design and Urban Planning of the University of Sassari, Italy. He holds his Ph.D. in territorial planning and public policy from the IUAV University of Venice. His research interests are planning and territorial analysis, evaluation models, construction of future scenarios and urban simulation. The research activities regard techniques, models and tools for decision support and evaluation of complex projects, plans and territorial policies.

Arnaldo Cecchini is Full Professor of Urban Planning and the Head of the Department of Architecture Design and Urban Planning of the University of Sassari. He is Director of the Laboratory of Analysis and Models for Planning – LAMP. He is an expert in techniques of urban analysis and modelling such as catastrophe theory, simulation, gaming simulation, cellular automata, scenario techniques and information systems for public participation.

Giuseppe A. Trunfio is Researcher in computer science in Department of Architecture, Design and Urban Planning of the University of Sassari, Italy. His main research interests focus on decision support, modelling and simulation, high performance computing and optimization metaheuristics. He is a member of IEEE Computer Society and a professional member of the ACM.