A Laboratory for Agent Based Computational Economics: The Self-development of Consistency in Agents' Behaviour*

Pietro Terna

Dipartimento di Scienze economiche e finanziarie, corso Unione Sovietica 218bis, 10134 Torino, Italy; pietro.terna@torino.alpcom.it

Abstract. The use of agents based models in the field of computational economics is widely spreading. Here we introduce a tool (Cross Target method), useful in building artificial laboratories, for experimenting with learning, self-developed consistency and interaction in artificial worlds of agents, to observe the emergence of rationality and complexity. Two examples of environments created with our technique will be introduced.

1 Introduction

In December 1996, http://www.econ.iastate.edu/tesfatsi/abe.htm, a site maintained by Leigh Tesfatsion, Iowa State University, contains the following definition of computational economics:

> Agent-based computational economics (ACE) is roughly characterized as the computational study of economies modelled as evolving decentralized systems of autonomous interacting agents. A central concern of ACE researchers is to understand the apparently spontaneous appearance of global regularities in economic processes, such as the unplanned coordination of trade in decentralized market economies that economists associate with Adam Smith's invisible hand. The challenge is to explain these global regularities from the bottom up, in the sense that the regularities arise from the local interactions of autonomous agents channeled through actual or potential economic institutions rather than through fictitious top-down coordinating mechanisms such as a single representative consumer.

At http://www.econ.iastate.edu/tesfatsi/, a site always maintained by Leigh

* This research has been supported by grants from the Italian Ministero dell'Università e della Ricerca scientifica e tecnologica, inside the project "Intermediazione finanziaria, funzionamento dei mercati ed economia reale".

Tesfatsion, the definition is:

> Agent-based computational economics (ACE) is the computational study of economies modelled as evolving decentralized systems of autonomous interacting agents. ACE is thus a specialization to economics of the basic Alife paradigm.

Agent-Based Computational Economics or ACE was the current term in December 1996; few months ago, as reported in Tesfation (1996), at the February 1996 UCLA Economic Simulation conference organised by Axel Leijonhufvud, participants suggested the name Agent Based Economics (ABE).

The introduction of ACE approach is also related to the necessity of incorporating bounded rationality in our economic models, with real life complexity emerging from agent interaction and not from agent self-complexity. See Conlisk (1996) and Berk, Hughson and Vandezande (1996) about bounded rationality models and behaviour.

2 Cross-Target Method, Neural Networks and Other Tools

To develop ACE experiments, we introduce the following general hypothesis (GH): an agent, acting in an economic environment, must develop and adapt her capability of evaluating, in a coherent way, (1) what she has to do in order to obtain a specific result and (2) how to foresee the consequences of her actions. The same is true if the agent is interacting with other agents. Beyond this kind of internal consistency (IC), agents can develop other characteristics, for example the capability of adopting actions (following external proposals, EPs) or evaluations of effects (following external objectives, EOs) suggested from the environment (for examples, following rules) or from other agents (for examples, imitating them). Those additional characteristics are useful for a better tuning of the agents in making experiments.

To apply the GH, first we introduce a tool useful to build an artificial laboratory for ACE, employing artificial neural networks (ANNs); the program is developed in C language and can be obtained from the author as public domain code. We observe, anyway, that the GH can be applied using other algorithms and tools, reproducing the experience-learning-consistency-behaviour cycle with or without neural networks.

An introductory general remark: in all the cases to which we have applied our GH, the preliminary choice of classifying agents' output in actions and effects has been useful (1) to clarify the role of the agents, (2) to develop model plausibility and results, (3) to avoid the necessity of prior statements about economic rational optimising behaviour (Terna 1991, 1992a, 1992b, 1993a, 1993b; Beltratti et al. 1996).

Economic behaviour, simple or complex, can appear directly as a by-product of IC, EPs and EOs. To an external observer, our AAAs are apparently operating with goals and plans. Obviously, they have no such symbolic entities, which are inventions of the observer. The similarity that we recall here is that the observations and analyses about real world agents' behaviour can suffer from the same bias. Moreover, always to an external observer, AAAs can appear to apply the rationality paradigm, with maximising behaviour.

Complexity can be more frequently found out of the agents - in the framework emerging from their interaction, adaptation and learning - than within them; exactly in the same way, also rationality (and Olympic rationality) can be found out of agents, simply as a by-product of environment constraint and agents' bounded capabilities. The same, for optimisation, as a by-product of interaction and constraints, emerging out of agents' mind.

The main problem is: obviously agents, with their action, have the goal of increasing or decreasing something, but it is not correct to deduce from this statement any formal apparatus encouraging the search for complexity within agents, not even in the "as if" perspective. On optimisation, see Schoemaker (1991).

With our GH, and hereafter with the Cross Target (CT) method, we work at the edge of Alife techniques to develop Artificial Worlds of simple bounded rationality AAAs: from their interaction, complexity, optimising behaviour and Olympic rationality can emerge, but "out of agents".

Finally, we want to consider learning from the point of view of the bounded rationality research program; as Arthur (1990) points out (see also Arthur, 1991):

> In designing a learning system to represent human behaviour in a particular context, we would be interested not only in reproducing human rates of learning, but also in reproducing the "style" in which humans learn, possibly even the ways in which they might depart from perfect rationality. The ideal, then, would not simply be learning curves that reproduce human learning curves to high goodness-of-fit, but more ambitiously, learning behaviour that could pass the Turing test of being indistinguishable from human behaviour with its foibles, departures and errors, to an observer who was not informed whether the behaviour was algorithm-generated or human-generated.

In order to implement this ideal target without falling in the trap of creating models that are too complicated to be managed, we consider artificially intelligent agents founded upon algorithms which can be modified by a trial and error process. In one sense our agents are even simpler than those considered in neoclassical models, as their targets and instruments are not as powerful as those assumed in those models. From another point of view, however, our agents are much more complex, due to their continuous effort to learn the main features of the environment with the available instruments.

Agents calibrated with this kind of limitations, but also with adaptation capabilities, can be the basis upon which to build large interacting models, which arise directly from the economic or financial framework, or like the more abstract models of the Artificial Worlds (AW) literature (Lane, 1993a and 1993b). AW are a class of models designed to give insights about emergent hierarchical organisations. Many systems, in chemistry and biology as well as in human society, appear to have the capability of achieving, over time, more complex organisation. Mainly, emerging organisations are hierarchical. That is, the systems are composed of a number of different levels, each of them consisting of entities that interact with the others. For example, economic activities involve interaction between individual decision-makers, firms and households, industries, and national economies.

2.1 The Cross-Target Method

The name cross-targets (CTs) comes from the technique used to figure out the targets necessary to train the ANNs representing the artificial adaptive agents (AAAs) populating our ACE experiments.

Following the GH, the main characteristic of these AAAs is that of developing internal consistency between what to do and the related consequences. Always according to the GH, in many (economic) situations, the behaviour of agents produces evaluations that can be split in two parts: data quantifying actions (what to do); forecasts of the outcomes of the actions. So we specify two types of outputs of the ANN and, identically, of the AAA: (1) actions to be performed and (2) guesses about the effects of those actions.

Both the targets necessary to train the network from the point of view of the actions and those connected with the effects are built in a crossed way, originating the name Cross Targets. The former are built in a consistent way with the outputs of the network concerning the guesses of the effects, in order to develop the capability to decide actions close to the expected results. The latter are similarly built with the outputs of the network concerning the guesses of the actions, in order to improve the agent's capability of estimating the effects emerging from the actions that the agent herself is deciding.

CTs, as a fulfillment of the GH, can reproduce economic subjects' behaviour, often in internal "ingenuous" ways, but externally with apparently complex results.

The method of CTs, introduced to develop economic subjects' autonomous behaviour, can also be interpreted as a general algorithm useful for building behavioural models without using constrained or unconstrained optimisation techniques. The kernel of the method, conveniently based upon ANNs (but it could also be conceivable with the aid of other mathematical tools), is learning by guessing and doing: control capabilities of the subject can be developed without defining either goals or maximising objectives.

The CT method can appear to be related to Temporal Difference (TD) Learning of Barto and Sutton (Sutton 1988, Tesauro 1992), which learns from the differences between temporally successive predictions - or action outcomes - of the system, having a final target perfectly known at the end of the run. In the TD method we have a special and powerful case of true supervised learning, where an external teacher can suggest correct target values. Also TD, as CT, addresses the issue of consistent learning, but with delayed feedback founded upon a true target value; CT uses immediate tentative targets, self-generated and never corrected by an external teacher. The aim of CT is in effect that of generating time paths for relevant variables, without any final or intermediate externally known objective, operating only with simple rules to adapt both behaviour and predictions.

2.2 The CT Algorithm

Surely, as we will see above, the CT algorithm introduced here is not the only way for dealing with AAAs in ACE context. However, it represents a useful simulation structure because it does not require injections of rules, optimising behaviour, planning capabilities, but only a limited computational ability: that necessary to take simple decisions and to compare guesses with results, developing self-consistency.

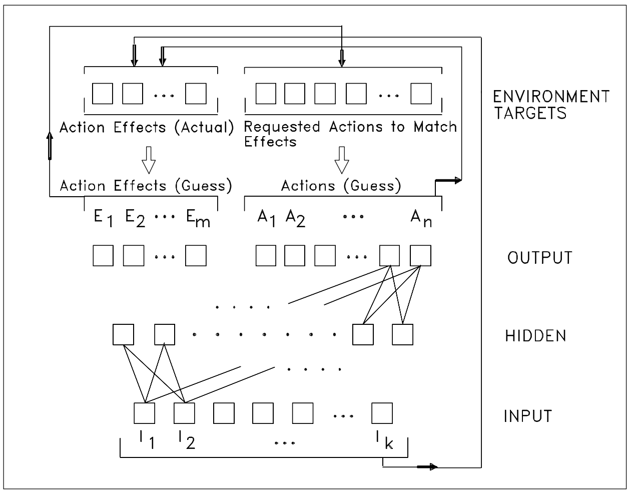

Fig. 1 - The cross-target construction.

Following also other authors' works (Parisi et al. 1990), we choose the neural networks approach to develop CTs, mostly as a consequence of the intrinsic adaptive capabilities of neural functions. Here we will use feed forward multilayer networks.

Figure 1 describes an AAA learning and behaving in a CT scheme. The AAA has to produce guesses about its own actions and related effects, on the basis of an information set (the input elements are I_1, ..., I_k). Remembering the requirement of IC, targets in learning process are: (1) on one side, the actual effects - measured through accounting rules - of the actions made by the simulated subject; (2) on the other side, the actions needed to match guessed effects. In the last case we have to use inverse rules, even though some problems arise when the inverse is indeterminate. Technical explanations of CT method are reported in the Appendix, following Beltratti et al. (1996).

A first remark, about learning and CT: analysing the changes of the weight during the process we can show that the matrix of weights linking input elements to hidden ones has little or no changes, while the matrix of weights from hidden to output layer changes in a relevant way. Only hidden-output weight changes determine the continuous adaptation of ANN responses to the environment modifications, as the output values of hidden layer elements stay almost constant. This situation is the consequence both of very small changes in targets (generated by CT method) and of a reduced number of learning cycles.

The resulting network is certainly under-trained: consequently, the simulated economic agent develops a local ability to make decisions, but only by adaptations of outputs to the last targets, regardless to input values. This is short term learning as opposed to long term learning.

Some definitions: we have (1) short term learning, in the acting phase, when agents continuously modify their weights (mainly from the hidden layer to the output one), to adapt to the targets self-generated via CT; (2) long term learning, ex post, when we effectively map inputs to targets (the same generated in the acting phase) with a large number of learning cycles, producing ANNs able to definitively apply the rules implicitly developed in the acting and learning phase.

We think that it would be useful to develop the GH also in other ways, eventually employing neither ANN nor CT, to verify the reproducibility of our results in other contexts. We have to introduce algorithms capable of the same performances in order to obtain the same results of short and long term learning, as before, without the aid of ANN. Our algorithm must be capable of modifying its outputs in a smooth way, following cross-target suggestion about actions and guesses of action effects, to produce the self-development of behavioural skills of the acting and adapting phase. The algorithm - may be not the same - has also to develop a strong mapping capability between input and output (target) vectors, to definitively develop the ability of producing the same behavioural results on the basis of input data.

A second remark, about both external objectives (EOs) and external proposals (EPs). If used, these values substitute the cross targets in the acting and adapting phase and coherently are included in the data set for ex post learning. Despite the target coming from actions, the guess of an effect can be trained to approximate a value suggested by a simple rule, for example increasing wealth. This is an EO in CT terminology. Its indirect effect, via CT, will modify actions, making them more consistent with the (modified) guesses of effects. Vice versa, the guess about an action to be accomplished can be modified via an EP, affecting indirectly also the corresponding guesses of effects. If EO, EP and IC conflict in determining behaviour, complexity may emerge also within agents, but in a bounded rationality perspective, always without the optimisation and full rationality apparatus.

2.3 Other Tools

In the future, we are planning to use Swarm, as a more standard platform. As one can read at the Swarm site (http://www.santafe.edu/projects/swarm/):

> Swarm is a software package for multi-agent simulation of complex systems being developed at The Santa Fe Institute. Swarm is intended to be a useful tool for researchers in a variety of disciplines, especially artificial life. The basic architecture of Swarm is the simulation of collections of concurrently interacting agents: with this architecture, we can implement a large variety of agent based models.

Other tools that one can plan to introduce in this kind of applications are, for example, the "National Micropopulation Simulation Resource" which is located within the Medical School's Department of Laboratory Medicine and Pathology from the National Center for Research Resources of the National Institutes of Health (at http://www.nmsr.labmed.umn.edu/nmsr/NMSR.html).

IBM presents the Intelligent Agent Resource Manager, which one can see at http://www.software.ibm.com./openblue/id1n2/cover.htm. The presentation states:

> Intelligence reflects the degree of reasoning and learned behavior in an agent. Intelligence describes the agent's ability to accept the user's statement of goals and carry out the task the user delegated to it. The agent's goals and behaviors could be encoded in a simple script that is executed by an interpreter in response to an event. Or, the reasoning could be provided by a set of rules that encodes strategy and goals. Sophisticated agents could learn and adapt to their observed environment, both in terms of the user's objectives and in terms of the resources available to the agent to carry out its task.

Another interesting tool is Starlogo, applied in Resnick (1994) and described at http://lcs.www.media.mit.edu/groups/el/Projects/starlogo/.

3 Agent Based Computational Experiments

We introduce here two examples of experiments in which agent's behaviour arise from the GH developed with CT. The first experiment is fully reported in Beltratti et al. (1996); the second one, about money, is under development and has been partially reported in Terna (1995, 1996). Technical contents of the experiments are summarised; readers interested in a thorough explanation can obtain both the complete description as reported in the referenced papers and the CT files necessary to repeat the experiments from the author.

3.1 Agents Foraging for Food: External Objectives vs. Imitation

In this first experiment on motion of agents foraging for food, we apply the following scheme. On a plain with (x,y) coordinates, the subject is initially in (10,10) while the food is fixed in (0,0). The ANN simulating the subject has the following inputs: X(t-1), position in the x direction at the time t-1; Y(t-1), position in the y direction at the time t-1; dX(t-1), step in the directions x, at time t-1 (bounded in the range ±1); dY(t-1), step in the directions y, at time t-1 (bounded in the range ±1).

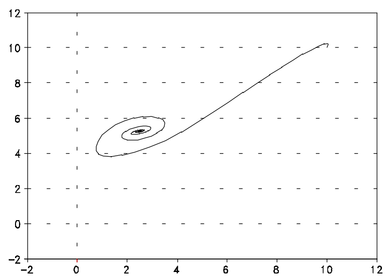

Fig. 2 - Moving toward food, without EO.

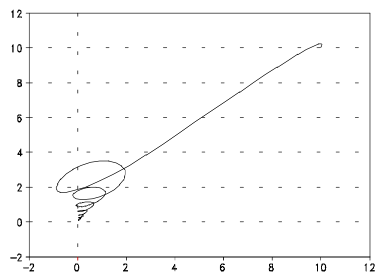

Fig. 3 - Moving toward food, with EO.

Using CT terminology, the ANN produces as outputs two guesses about effects and two guesses about actions. Guesses about effects are X(t), Y(t). Guesses about actions are dX(t) and dY(t), all with the same meaning of the input values. Positions X(t) and Y(t) have also the meaning of distance of the artificial subject from the food (distance evaluated employing rectangular coordinates). Summarising, the ANN representing the AAA has the following structure: 4 input, 6 hidden and 4 output nodes.

In Figure 2 we report the movement of the agent in 200 cycles of acting and learning. The agent goes toward the food on the basis of a simple implicit mechanism, which explains also the situation of locking in the middle of the path. The mechanism works in the following way: at the beginning of the experiment, the ANN produces random outputs, in a small interval around the central value between minimum and maximum. This effect is always present and is easily explained by considering the consequence of the initial random choice of the weights, that gives on average a null sum of the inputs of the sigmoidal transformation. In the case of the logistic functions, that input gives an output of about 0.5, corresponding to the mean between minimum and maximum values. As a consequence, the initial guesses about the effects of the movement give estimated positions around the central point where food is placed, with some variability.

In other term, the initial guess is that of being near the food. CTs immediately correct this wrong estimate, but they also correct the guesses about actions (the movements), to develop their consistency with the (wrong, but positively optimistic) guesses of effects. So, the artificial agent moves in the correct direction, but the process rapidly goes in a locking situation, with mutual consistency between effects and actions.

Now, imposing an EO on the side of the effects, that is the target of reducing in each cycle the distance from food to the 75% of the distance of the previous cycle, the food is easily gained, as reported in Figure 3. We underline that no suggestion is introduced about the direction of the movement.

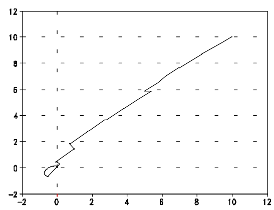

Fig. 4 - After relearning.

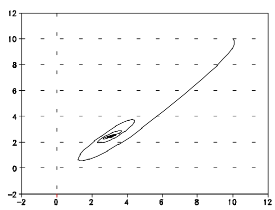

Fig. 5 -With EP.

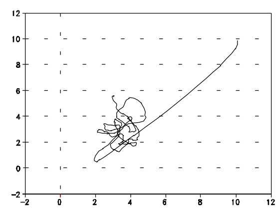

Fig. 6 -With random EP.

In Figure 4 we present again the case of Figure 3, but with an ANN whose weights come from an ex post repeated learning process of 200,000 cycles. The simulated agent reaches the food in few steps, going directly toward it, despite some uncertainty. We have here the effects of long term learning vs. short term one (Figure 3).

The last step in this experiment is to develop a population of similar agents, all acting without EO, as the one in Figure 2, but with an EP. In Figure 5 we have the itinerary toward the food of an agent having as EP the suggestion of imitating the current movement of another agent. The imitative mechanism works as follows: the agent, whose action has to be imitated, is chosen randomly; the imitation occurs only if the action of the other agent is greater in absolute value than that of the imitating subject.

The artificial subjects, also without EO, come close to the food: the reason is that the choice of imitating another agent reproduces a situation of inconsistency among CTs, from which - as seen before - the implicit simple mechanism driving the agent restarts, avoiding situations of locking. Also note that if the imitated action is directed towards the food, the consequence is immediately convenient; if the imitated action goes in the opposite direction, the process corrects the error rapidly, always strengthening the global convergence process.

If the EP is randomly generated, as in the case of Figure 6, the effect of avoiding locking situations is replicated, but the random behaviour prevails: it is more difficult to recognise a plan in this kind of artificial crazy agent, which is anyhow capable of going close to the food.

3.2 The Emergence of Money

The model is founded upon a multiple barter market, with couples of AAAs randomly meeting. Each AAA owns four goods in different quantities, bearing a conservation cost per unit of each good in each day. Each good can have a different conservation cost, with multiple meanings: uselessness of the commodity, difficulties of conservation, etc.

Random meetings among agents allow the fulfillment of multiple barters, covering the set of all possible exchanges. In this context, we look for the emergence of commodity money in exchanges, without theoretic a priori hints implied into agents. Recent key works in the direction of money emergence are Kiyotaki and Wright (1989) and Marimon et al. (1990), both based upon theoretic assumptions. We recall also Ritter (1995), focusing on fiat money (inconvertible paper money made legal tender by a Government decree).

Only some (antithetic) suggestions are here presented to agents (as EOs): (1) to balance the quantities of the various goods owned or, on the contrary, to increase the quantity of a preferred one, (2) to reduce the cost arising from the conservation of goods, (3) to reduce the total amount of transactions.

Among the four goods considered in the experiment, one has a systematic disadvantage or advantage in its conservation cost for all the agents. In both cases the "different" good emerges as medium of exchanges.

We have four AAAs founded upon ANNs, with 6 input, 24 hidden and 30 output nodes. The experiments run for 200 days (or cycles of four steps, as introduced in the previous section).

The inputs are the quantities of the four goods held by the agent at the beginning of a day, the total conservation cost arising from the quantities of the four goods held at the end of the previous day, the total amount of transactions accomplished by the subject in the previous day.

Output nodes represent both guesses about the effects of the actions of each AAA (from 1 to 18) and the measure of the actions that the subject is planning to accomplish (from 19 to 30).

Outputs from 1 to 12 measure the exchanges between the considered AAA (defined 'first') and another randomly chosen (defined 'second'); the first sells the good x and purchases an equal quantity of good y; the second does the opposite operations. With four goods and a double exchange for each couple of goods (x vs. y and y vs. x), we have twelve possible transactions.

For each exchange we state a random match between the two proposals. In this case we both avoid any form of systematic prevalence and force agents to exchange. In a subtler way, agents could be forced to exchange, splitting the goods that they produce or posses and that one that they can consume, but from the point of view of exchange simulation the effect would be the same.

Outputs from 13 to 16 represent the quantities of goods after the exchanges of the day. The successive two outputs measure respectively the total cost coming from conserving goods (output 17) after the exchanges of the day, and the total amount of the daily transaction (output 18).

Finally, outputs from 19 to 30 are guesses of proposals of exchange between the considered AAA (defined 'first') and another to be chosen randomly (defined 'second'); the actual exchange will arise when the random meeting will be established, following the rules of targets related to outputs from 1 to 12.

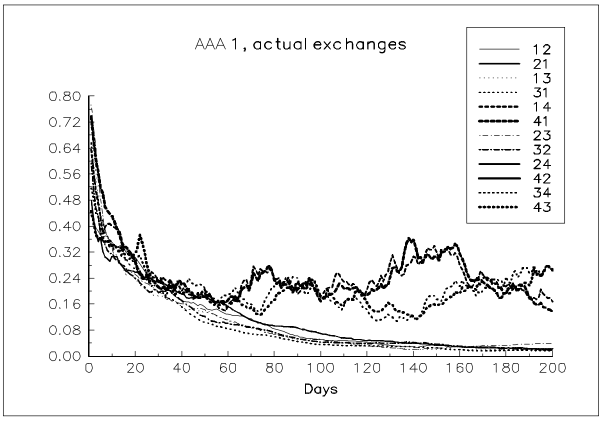

Fig. 7 - Actual exchanges made by AAA #1, considering jointly the three EOs.

Environmental data included in the model structure are the costs of conservation of the four goods and the initial assignment of goods to agents, set to 1 for all goods and AAAs.

Money emerges when we consider jointly the three EOs. The AAAs behave to balance the quantities owned of each good, but simultaneously they are attempting to reduce good 4, which is the most expensive as conservation cost (exchanging it, being this the unique way for them to achieve the goal); alternatively, if good 4 becomes the less expensive, all agents attempt to increase the quantity owned; the effect about the emergence of money is the same. Besides this, they have to reduce the total amount of exchanges. As a natural consequence, only transactions of good 1, 2, and 3 with good 4 are taken into account and good 4 emerges as commodity money.

In Figure 7 it is easy to distinguish transactions made with the commodity money (those indicated with 14, 41, 24, 42, 34 and 43).

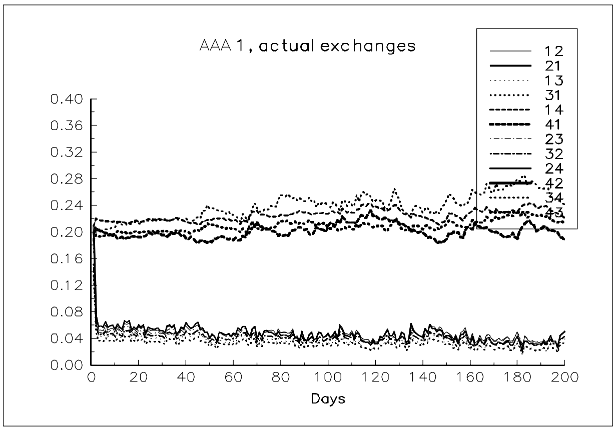

Fig. 8 - Actual exchanges made by AAA #1, considering jointly the three EOs, after relearning.

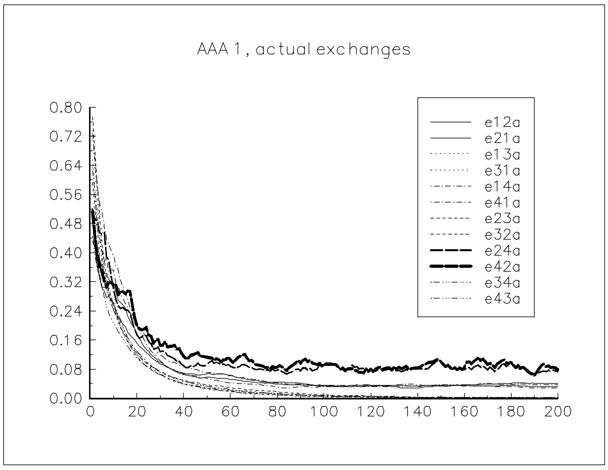

Fig. 9 - Actual exchanges made by AAA #1, considering jointly the three EOs.

Is the emergence of money strictly related to the learning and acting situation produced by CTs or can it be definitely learned by our AAAs? In Figures 8 we have the results of a long term relearning experiment (see Section 2.1), obtained repeating for 100,000 cycles the ANN training at the end of the 200 days of experience. Afterwards, we use the weight coming from relearning to build the four agents whose behaviour is now no more influenced by targets: the role of commodity money (see exchanges 14, 41, 24, 42, 34 and 43) is clearly confirmed.

We point out that the actions producing these results, as outcomes of AAAs interacting in our environment, are all developed in an autonomous way. Surely, the mechanisms behind actions are very simple, but what is surprising is that those mechanisms, although ingenuous, are endogenously developed by the model.

In the previous model, the random match between the proposals of the two AAAs meeting makes exchanges mandatory and opposite EOs determine the emergence of money. We can relax the exchanging obligation, choosing the minimum one of the two proposals: we obtain a structure in which each agent has a predetermined preference and the objective of reducing the conservation cost, as before. Newly exchanges emerge, as in Figure 9, where good 2 is the preferred one and good 4 emerges as commodity money.

4 Further Developments

We are now working to an experiment about fiat money, in which each AAA owns and wants different goods, without symmetric potential partners. The following step will the porting of the laboratory in Swarm, preliminary with ANNs as AAAs.

Appendix

Now we may introduce some technical explanations about CTs, with the aid of the general scheme of Figure 1, observing that (1) the inputs of the model are mainly data coming from the environment or from other agents' behaviour, (2) they can be dependent or independent from the previous actions of the simulated artificial subject, (3) targets are known only when actions take place.

The CT algorithm is a learning and acting one: action is necessary to produce the information by which we can construct targets to train the ANN that simulates the subject. A training set cannot be constructed here in the usual way because rules linking inputs and outputs of the ANN have 'to be discovered' by the experiments led by AAAs.

Learning and acting take place in four steps each 'day'; a day is the sum of the four steps required to perform a full cycle of estimation of outputs and of backpropagation of errors, correcting the neural network weights. Initial weights are randomised in a given range.

Looking at Figure 4, the four steps can be introduced in the following sequence.

(1) Outputs of the ANN: the actions to be accomplished, reported in the right side of Figure 1, and the effects of these actions, reported in the left side of the same figure, are guessed following inputs and network weights.

(2) Targets for the left side of the network: the targets for the effects supposed to arise from actions, as guessed in the left side of the output layer in Figure 1, are figured out by the independently guessed actions. In this way, guesses about effects become more close to the true consequences of actual actions.

(3) Targets for the right side of the network: the differences measured in step (2) among targets and ANN outputs on the effect side can be inversely interpreted as starting points for action modifications, to match the guessed effects. So they are used to build the targets for the mechanism that guesses the actions. Being the inverses of the formulas shown below often undefined, corrections are shared randomly among all the targets to be constructed; besides, when several corrections concern a target, only the one with the largest module is chosen. In this way, we would like to imitate the actual behaviour of a subject requested of obeying to several independent and inconsistent commands: probably the most imperative, here the largest value, will be followed.

(4) Backpropagation: learning takes place, correcting weights in order to obtain guessed effects closer to the consequences of guessed actions, and guessed actions more consistent with guessed effects. Thus, we have two learning processes, both based upon the guesses of the elements of the opposite side of the network.

This double side process of adaptation, with interaction among agents and long term learning introduced in Section 2.1, ensures the emergence of non trivial self-developed behaviour, from the point of view of time paths of the values generated by the outcomes of the agents.

We can now explain in a formal way the acting and learning algorithm of CTs, introducing a generic effect E_1 arising from two actions, named A_1 and A_2. The target for the effect is:

$$E_1' = f(A_1,A_2) \tag{1}$$

where $f(\cdot)$ is a definition, linking actions to effects on an accounting basis.

Our aim here is to obtain an output E_1 (the guess made by the network) closer to E_1', which is the correct measure of the effect of actions A_1 and A_2. The error related to E_1 is:

$$e = E_1' - E_1$$

or, by convention in ANN development, one half of the square of $E_1' - E_1$. To minimise the error, we backpropagate it through network weights.

Our aim is now finding the actions, as outputs of our network, more consistent with the outputs produced by the effect side. So we have to correct A_1 and A_2 to made them closer to A_1' and A_2', which are actions consistent with the output E_1. We cannot figure out the targets for A_1 and A_2 separately. From (1) we have:

$$A_1 = g_1(E_1',A_2) \tag{2}$$

$$A_2 = g_2(E_1',A_1) \tag{3}$$

Choosing a random value τ_1 from a random uniform distribution whose support is the closed interval [0,1] and setting $\tau_2 = 1 - \tau_1$, from (2) and (3) we obtain:

$$A_1' = g_1(E_1' - e \cdot \tau_1, A_2) \quad (4)$$

$$A_2' = g_2(E_1' - e \cdot \tau_2, A_1) \quad (5)$$

Functions g_1 and g_2, being obtained from definitions that link actions to effects mainly on an accounting basis, usually have linear specifications; so equations (4) and (5) generally give solutions that are globally consistent. The errors to be minimised are:

$$a_1 = A_1' - A_1$$

$$a_2 = A_2' - A_2$$

Eqs. 4 and 5 would be unacceptable as inversions of true dynamic functions, but they are used here as a simplifying tool (mainly for the presence of random separation obtained by τ_1 and τ_2 values), always to generate time paths for variables, without a priori or external suggestions.

When the actions determine multiple effects, they are included in multiple definitions of effects. So, those actions will be affected by several corrections; as reported in point 3 above, only the largest absolute value is chosen.

Input and target variability, generated both in deterministic and random ways, is required to ensure the economic plausibility of the experiments, but is also necessary to ensure that the outputs and the targets of the ANN change. Lacking such variability, on the basis of initial random weights of the network and following CTs, in most cases all outputs would be frozen at about 0.5, with perfect but merely apparent learning results.

With the proper variability, we repeat for a given number of cycles (days) the four steps introduced describing Figure 1. The learning following the fourth step of each day gives a sort of local adaptation to the changes of the environment.

5 References

Arthur, W.B. (1990) A Learning Algorithm that Mimics Human Learning, *Santa Fe Institute Working Paper* 90-026.

Arthur, W.B. (1991) Designing Economic Agents that Act Like Human Agents: A Behavioral Approach to Bounded Rationality. *American Economic Review*, 2, 353-9.

Beltratti, A., Margarita, S., Terna, P. (1996) *Neural Networks for Economic and Financial Modelling*. International Thomson Computer Press, London.

Berk, J.B., Hugson, E., Vandezande, K. (1996) The Price Is Right, But Are the Bids? An Investigation of Rational Decision Theory. *American Economic Review*, 3, pp. 954-970.

Conlisk, J. (1996), Why Bounded Rationality? *Journal of Economic Literature*, 2, pp. 669-700.

Kiyotaki, N., Wright, R. (1989) On Money as a Medium of Exchange. *Journal of Political Economy*, 4, pp. 927-955.

Lane, D.A. (1993a) Artificial Worlds and Economics, part I. *Journal of Evolutionary Economics*, 3, 89-107.

Lane, D.A. (1993b) Artificial Worlds and Economics, part II. *Journal of Evolutionary Economics*, 3, 177-97.

Marimon, R., McGrattan, E., Sargent, T.J. (1990) Money as a Medium of Exchange in an Economy with Artificially Intelligent Agents. *Journal of Economic Dynamics and Control*, 2, pp. 329-373.

Parisi, D., Cecconi, F., Nolfi, S. (1990) Econets: Neural networks that learn in an environment. *Network*, 2, pp. 149-168.

Resnick, M. (1994) T*urtles, termites, and traffic jams: explorations in massively parallel microworlds*. MIT Press, Cambridge, MA.

Ritter, J.A. (1995) The Transition from Barter to Fiat Money. *American Economic Review*, 1, pp. 134-49.

Schoemaker, P.J.H. (1991) The quest for optimality: A positive heuristic for science? *Behavioral and brain sciences*, 2, pp. 205-245.

Sutton, R.S. (1988) Learning to Predict by the Methods of Temporal Differences. *Machine Learning*, 3, 9-44.

Terna, P. (1991) Labour, Consumption and Family Assets: A Neural Network Learning from Its Own Cross-Targets, in *Artificial Neural Networks* (eds T. Kohonen et al.). Elsevier Science Publishers B.V. (North-Holland), Amsterdam, pp. 1759-62.

Terna, P. (1992a) Artificial Economic Agents. SPIE Vol. 1709*, Proceedings of Applications of Artificial Neural Networks III*, pp. 1003-14.

Terna, P. (1992b) Microeconomic Experiments by Neural Networks, *in Artificial Neural Networks 2* (eds I. Aleksander and J. Taylor). Elsevier Science Publishers B.V. (North-Holland), Amsterdam, pp. 1339-42.

Terna, P. (1993a) Artificial Interacting Agents for Stock Market Experiments: the Cross-Target Method, in *Proceedings of ICANN 93* (eds S. Gielen and B. Kappen). Springer-Verlag, Berlin.

Terna, P. (1993b) Randomness, Imitation or Reason to Explain Agents' Behaviour into an Artificial Stock Market, in *Advances in Artificial Intelligence* (ed P. Torasso). Springer-Verlag, Berlin.

Terna, P. (1995) An artificial laboratory to experiment the emergence of money. CECOIA IV - *Fourth international conference on the cognitive foundations of economics and management*, September 6-8, 1995, Cachan (Paris).

Terna, P. (1996) Neural Networks for Economic and Financial Modelling: Summing Up Ideas Emerging from Agent Based Simulations and Introducing an Artificial Laboratory. LaSCoMES, Laboratory of Cognitive, Methodological and Socio-Economic Sciences, *Perspectives and Limits of Cognitive Economics*, March 22, 1996, Milano (in print).

Tesauro, G. (1992) Practical Issues in Temporal Difference Learning. *Machine Learning*, 8, 257-77.

Tesfation, L. (1996) How Economists Can Get Alife, to appear in *The Economy as an Evolving Complex System*, II, Santa Fe Institute in the Sciences of Complexity. Addison-Wesley, Reading, MA.