# Modelling Bounded Rationality using Evolutionary Techniques

Bruce Edmonds and Scott Moss

Centre for Policy Modelling, Manchester Metropolitan University,
Aytoun Building, Aytoun Street, Manchester, M1 3GHG, UK.
http://www.cpm.mmu.ac.uk

**Abstract.** A technique for the credible modelling of economic agents with bounded rationality based on the evolutionary techniques is described. The genetic programming paradigm is most suited due to its meaningful and flexible genome. The fact we are aiming to model agents with real characteristics implies a different approach from those evolutionary algorithms designed to efficiently solve specific problems. Some of these are that we use very small populations, it is based on different operators and uses a breeding selection mechanism. It is precisely some of the "pathological" features of this algorithm that capture the target behaviour. Some possibilities for integration of deductive logic-based approaches and the GP paradigm are suggested. An example application of an agent seeking to maximise its utility by modelling its own utility function is briefly described.

## 1 Introduction

The purpose of this paper is to report on an approach to simulating economic agents such that their behaviour matches their known characteristics. The approach taken is to introduce characteristics of bounded rationality and learning into these simulations using evolutionary techniques.

By using an approach to modelling learning that is close to that used in genetic programming (GP) [10, 11], we open up a new range of possibilities in the credible modelling of such boundedly rational agents, where an agent has a population of candidate beliefs (or models) of its environment which evolve as it learns. This contrasts in several respects from agent modelling approaches that use "crisp" logic-like beliefs, or approaches that only involve some inductive learning. In particular multiple and frequently inconsistent beliefs are held as a resource for future model development. However, despite this contrast this approach supports integration of such a style of learning with deductive mechanisms.

## 2 Modelling Boundedly Rational Economic Agents

If you seek to model real economic agents then, unless you make some very sweeping assumptions, the entities in your software model will also need the broad characteristics of the real agents. This is in contrast to traditional economics where, by and large, the *agency* nature of the agents is ignored, in favour of trying to capture their behaviour *en masse*.

Thus the purpose of an agent in such a model is different from either agents that are designed with a particular purpose in mind (e.g. [5]) or for exploration of the most effective and flexible algorithm for a set of problems. In such modelling one seeks for as much veracity as is possible given the usual limitations of time, cost and technique and one does not necessarily look to design them to be efficient, general, or consistent in their beliefs.

In particular we are interested in agents who:

- do not have perfect information about their environment, in general it will only acquire information through interaction with its environment which will be dynamically changing;
- do not have a perfect model of their environment;
- have limited computational power, so they can't work out all the logical consequences of their knowledge [18];
- other resources, like memory are limited (so they can't hold large populations of models);

In addition to these bounds on their rationality we also add some other observed characteristics of real economic agents, namely:

- the mechanisms of learning dominate the mechanisms of deduction in deciding their action;
- they tend to learn in an incremental, path-dependent [1] (or "exploitative") way rather than attempting a global search for the best possible model [16];
- even though they can't perform inconsistent actions, they often entertain mutually inconsistent models and beliefs.

The fundamental difference between these agents and, say, logic-based agents, is that the updating of internal belief structures is done in a competitive evolutionary manner using a continuously variable fitness measure rather than in a "crisp" consistency preserving manner. This is appropriate in situations of great uncertainty caused by a rationality that is not able to completely "cope" with its environment but is more restricted in its ability.

## 3 The Agent Architecture

For the above reasons we have developed a paradigm of modelling the learning that such agents engage in, as itself a process of modelling by the agents. For more on this framework see [13].

Although economic agents primarily develop though a process of incremental learning they also use some deductive procedures. In real economic agents these processes may be arbitrarily mixed as well as developed and abstracted over different layers of an organisation. Here we will only look at a model which effectively separates out learning and deduction and comes from an essentially unitary agent structure.

The agent works within a given *a priori* body of knowledge (e.g. accounting rules). The agent may well make deductions from this in a traditional way and apply these to the current hypotheses. This body of *a priori* knowledge may also determine the syntax of the models the agent starts with, its principal goals, default actions, fitness functions and the operations to be applied to its models. Typically much of this a priori knowledge can be made implicit in the syntax of the genome (which is the approach we have tended to take).

The agent here has many models of its environment. Once started the agent incrementally develops and propagates these models according to a fitness function which is based on its memory of past data and effects of its actions as well as the complexity and specificity of its models. It then selects the best such model according to that measure. From the best such model and its goals it attempts to determine its action using a search-based, deductive or quasi-deductive mechanism. It then takes that action and notes the effects in the environment for future use. The setup is illustrated below in figure 1.

The development of these models (i.e. the learning) is modelled by an evolutionary process on this population of internal models (similar to that described in [4]). Important restrictions on such agents include the fact that it may have only limited information gained as the result of inter-action with its environment and that any action costs it so that it can not indulge in an extensive exploratory search without this being weighed against the benefit being gained (this is especially true given the course temporal graining of typical economic simulations).

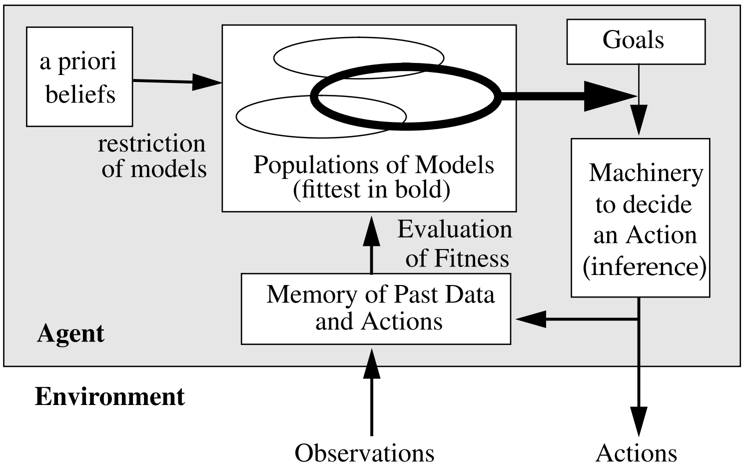


**Figure 1:** Basic Structure of a Simplified Economic Agent

## 4 The GP Paradigm

An important special case of the above approach to learning is where the range of operations includes selection and some mechanism for variation, i.e. an evolutionary algorithm. In particular the paradigm of GP is particularly appropriate, due to the structure of the genome. These techniques, however, can not be blindly applied. For example, the efficiency of the learning process is only a secondary concern when seeking to model economic agents by their software cousins, but many of the other features of this approach for modelling learning in an economic agent *are* appropriate, namely:

- the population of programs can represent a collection of multiple, competing models of the world with which it is concerned;
- there is always at least one maximally fit individual model that can be used to react to events and from which appropriate deductions can be made;
- the models are incrementally developed by the learning mechanism;
- the fitness measure can be tailored to include aspects such as cost and complexity as well as the extent of the agreement with known data;
- the language of representation of the models can be fairly general and expressive, e.g. logical expressions.

## 5 Adapting the GP Paradigm

There are several possible ways of using evolving populations to simulate a community of economic agents:

1. each member of the evolving population could represent one agent;
2. each agent could be modelled by a whole evolving population;
3. the whole population could be modelled by the whole evolving population but without an individually intended agent <-> gene correspondence.

Method (1) has been used in several models of agents which evolve (e.g. [7, 19]), here the genetic development has nothing to do with the nature of an agent's cognitive processes but helps determine its goals or strategies. Method (3) above is the most popular in economics (e.g. [1, 3]), but unless such a model predicts pertinent properties of real populations of agents it represents a bit of a fudge, and means that the observable behaviour and content of individual entities in the model do not have a clear referent in what is being modelled. This makes it far less useful if one wants to use such models to gain a detailed insight into the internal dynamics of populations. Method (2) actually addresses the cognitive process as the agent corresponds to a population of mental models. This has been done before in a limited way in [15], but here agents have a fixed menu of possible models which do not develop.

In using the evolutionary paradigm in this sort of modelling we tend to:

- represent the agent by a whole evolving population - each gene corresponding to one of its alternative models (this is the approach taken in the example in Section 7);
- populations of agents are thus modelled as populations of evolving populations (i.e. populations of populations), with an intended agent to evolving population correspondence (e.g. [14]);
- give the agents only small populations of models, representing limited memory;
- base the fitness function on a combination of its error compared to past data, size of model and its predictivity (precision and range of applicability);
- restrict the variation operators to such as generalisation, specialisation, averaging, combining and mutating;
- and give them only a limited inferential ability to use its best model to choose its action.

This paradigm needs to be integrated with an agent-based approach and adapted to relate to credible models of economic agents. In particular the cross-over operator is somewhat arbitrary when simulating the development of models in economic agents (although undeniably efficient). This also introduces a globality to the search which is unrealistic.

In the example application presented below we use a process of combining old models together as branches from a new node and introducing randomly generated small new models. This produces more realistic results, for example it allows for better fitting by parameterisation.

## 6 Possible integration with Logic-based Agent Architectures

The structure of the agent described above and the GP style of chromosome allow for some integrations of the evolutionary learning mechanism and agents based on logic based inference mechanisms (e.g. the BDI framework of). This can occur when the chromosomes representing the internal mental models of the agent represent logical expressions. Given this there are two main possibilities:

1. The population of genes can be constrained to those that are logically consistent with a set of a priori knowledge expressed as logical expressions within some formal logical framework;
2. The inference of possible actions could be done by using a logical framework to infer the best action from its best model and its goals;

It would not be appropriate to constrain the population of internal models so that they were consistent with each other or consistent with its goals as these represent the competing partial beliefs of the agent about its world.

## 7 An Application - a model of utility learning

A simple application of the above approach is that of an economic agent that seeks to maximise its utility by dividing its spending of a fixed budget between two goods in each time period (what it does not spend on one good it spends on the other).

Unlike classical economic agents, this one does not know its utility function (even its form) but tries to induce it from past experience. It only gets information about the utility of a particular spending pattern by actually trying it. The agent wants to get the most utility from its spending. It will not speculate with alternative spending patterns merely to learn more about the utility curve.

To do this it attempts to model its utility with a function represented by a GP type chromosome using +, -, *, /, max, min, log, exp, average, “cutbetween” (a three-argument function which takes the second value if the first value is less than 1 and the third value thereafter) as branching nodes, and a selection of random constants and variables representing the amounts bought of the two products for the leaves. Thus the chromosome

```
[average
    [divide
        [amountBoughtOf 'product-2']
        [constant 2.3]]
    [constant 0.5]]
```

would predict that the utility gained would be

$$\frac{(x/2.3) + 0.5}{2}.$$

Where $x$ is the amount spent of product 2.

The fitness function is based on the RMS error of the prediction of a model compared to the actual utility gained over past spending actions. This is modified by a slight parsimony pressure in favour of shallower chromosomes and a bias in favour of chromosomes which mentioned more distinct variables based on the among bought (a rough measure of specificity - called “volume” in [13]). Only the fittest half of the population is retained each generation, so that this is a kind of selective breeding algorithm and does not use fitness proportionate random selection (thus is has some similarities to evolutionary programming [6]).

Each time period the agent:

1. carries over its previous functional models;
2. produces some new ones by either combining the previous models with a new operator or by growing a small new random one;
3. evaluates its current models using past data;
4. selects the best models in terms of fitness for survival;
5. it finds the fittest such model;
6. it then preforms a limited binary search on this model to find a reasonable spending pattern in terms of increasing its utility;
7. finally it takes that action and observers its resulting utility.

This model was realised in a language called SDML (Strictly Declarative Modelling Language) - a language that has been specifically developed in-house for this type of modelling. This is a declarative object-oriented language with features that are optimized for the modelling of such *economic, business and organisational* agents [9, 20].

Limiting the depth of the models created to 10, We preformed 10 runs over 100 time periods for each type of agent. The three types were characterised by the memory they were given and the number of new models they created each time period: respectively 10, 20 and 30. We call these 10-memory, 20-memory and 30-memory agents, they represent agents with different bounds on their rationality. The results were then averaged over these 10 runs.

The first graph shows the (RMS) error of the agent's best model of the utility function compared with the actual function (figure 2). It shows a great improvement between the 10-memory agent's and 20-memory agents, but only a marginal improvement between 20 and 30-memory agent's, suggesting the existence of a sort of minimum capacity for this task.

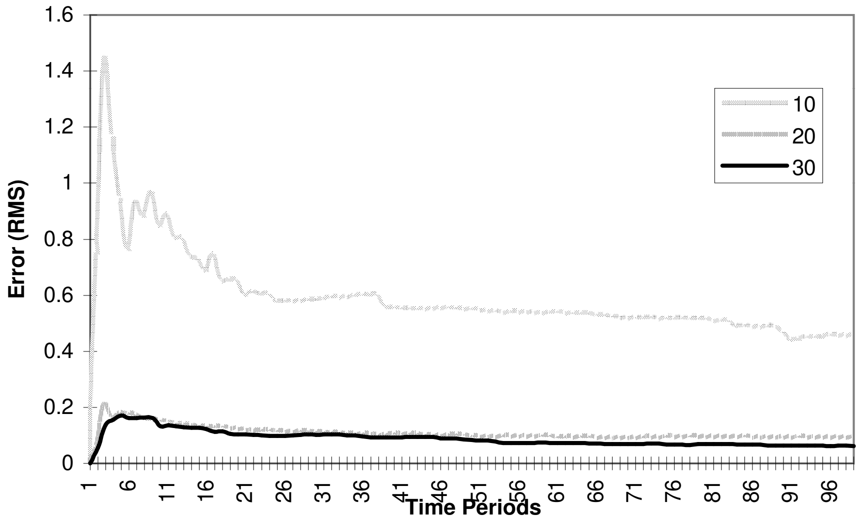


**Figure 2:** Error in Agent's Best Model for Different Memories, Averaged Over 10 Runs

When you look at the utilities achieved by the agents with different memories (figure 3), you see that a memory capacity (above 10) does not significantly increase the average utility over time, but it *does* dramatically effect the reliability of the utility it gains. If this were a firm with the utility being its profits, this reliability would almost as important as its average profit level.

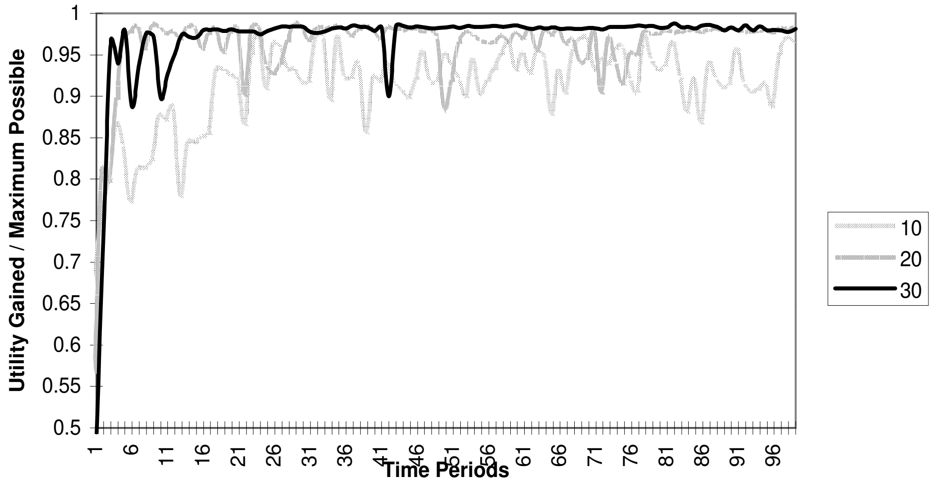


**Figure 3:** Utility Ratio Achieved for Agents with Different Memories, Averaged over 10 Runs

To give a flavour of the sort of models these agents develop, in run 1 of the 30-memory agent batch the agent achieved the following model by date 75:

```
[average
    [[divide
        [[add [[constant 1.117] [amountBoughtOf 'product-2']]]
        [average [[amountBoughtOf 'product-2'] [constant 4.773]]]]]
    [min
        [[amountBoughtOf 'product-2']
        [cutBetween
            [[average [[amountBoughtOf 'product-2'] [constant 4.773]]]
            [constant 1.044]
            [add [[constant 1.117] [amountBoughtOf 'product-2']]]]]]]]].
```

The extent of the fit learnt by the agent is shown in figure 4.

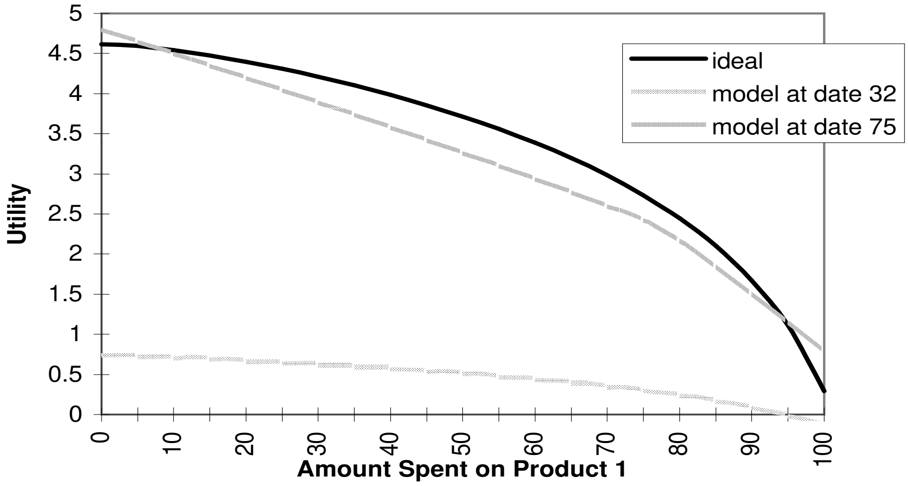


**Figure 4:** Learnt vs. Actual Utility Functions, Run 1 of 30-memory Agents

The purpose of this simulation is *not* to be an efficient maximiser of utility, but to model economic agents in a more credible way. It will only be vindicated (or otherwise) when compared to real economic data. However, the model does show traits found in the real world. For example, one phenomenon that is observed is that agents sometimes get "locked" into inferior models for a considerable length of time (as in [2]) - the model implies an inferior course of action, but this course of action is such that the agent never receives disconformation of its model. Thus this remains its best model in terms of the limited data it has, so it repeats that action. If, for example, some consumers find a satisfactory brand at an early stage in the development of their tastes and then they may never try any others - their (limited) experience will never disconfirm their model of what would give them most satisfaction, even when they would like other brands better.

Other related applications have included a model of intelligent price fixing in Cournot Duopoly tournaments [12], and a model of emerging markets where the agents are simultaneously building models of the economy they inhabit (and mutually create) [14].

## 8 Discussion

Such modelling using evolutionary techniques, where there is an explicit one-one correspondence between items modelled and the genes in the population typically deal with very small populations (in evolutionary terms).

In the example above we had populations of mental models as small as 10. Most of the models of abstract evolutionary algorithms deal only with large populations (many assume an infinite population for formal purposes). The behaviour of small populations may be pathological from the point of view of an efficient search mechanism, but here we have different goals in using evolutionary algorithms. It is precisely the pathological aspects of the process that capture the qualitative behaviour observed: sharp path-dependence, lock-in, exploitative search, a large spread of behaviours between different populations and limited overall optimization.

Also it is not always the case that the usual genetic operators are very efficient in such small populations. It is known that selective breeding can work well with small populations [8]. In addition (in the example above) we found that a traditional GP mixture of tree-crossover and propagation did substantially worse than that of combining together old models, generating small new random ones and propagation. This is important as the mechanism chosen has to be credible for realistically small populations of mental models. Much work needs to be done to understand the evolutionary dynamics of small populations.

## 9 Future Work

Research into this style of modelling agents is at a very early stage. Future work is almost unbounded. There some more immediate shortcomings of this approach that we intent to focus on soon. In no particular order these include:

- the introduction of intentions and planning;
- the development of techniques to evolving logical expressions, representing beliefs;
- the integration of the evolutionary module, with existing logic based approaches as described above;
- the investigation of the effects of different syntaxes (as exhorted by [17]);
- the increasing formalisation of the structure;
- the investigation of the effects of different genetic operators in very small populations

## References


1. Arifovic, J. (1994). Genetic Algorithm Learning and the Cobweb Model, Journal of Economic Dynamics and Control 18, 3-28.
2. Arthur, W. B. (1995). Increasing Returns and Path Dependence in the Economy. University of Michigan Press, Ann Arbor, MI.
3. Axelrod, R. (1984). The Evolution of Cooperation, Basic Books, New York.

4. Dennett, D.C. (1995). Consciousness Explained. Philosophy and Phenomenological Research, 53, 889-892.
5. Bonasso, R. P.; Kortenkamp, D.; Miller, D. P.; Slack, M. (1995). Experience with an Architecture for Intelligent Reactive Agents. ATAL'95 at IJCAI'95, Montreal. Published in (1996), Lecture Notes in Artificial Intelligence, 1037, 187-202.
6. Fogel, D.B. (1994) Evolutionary Programming an introduction and some current directions. Statistics and Computing, 4, 113-129.
7. Holland, J. H. (1992). Adaptation in Natural and Artificial Systems, 2nd Ed., MIT Press, Cambridge, MA.
8. Mühlenbein, H.; Schlierkamp-Voosen, D. (1993). The science of breeding and its application to the breeder genetic algorithm (BGA). Evolutionary Computation, 1, 335-360.
9. Edmonds, B.; Moss, S.; Wallis, S. (1996). Logic, Reasoning and A Programming Language for Simulating Economic and Business Processes with Artificially Intelligent Agents, AIEM96, Tel Aviv, 1996.
10. Koza, J. R. (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA.
11. Koza, J. R. (1994). Genetic Programming II: Automatic Discovery of Reusable Programs. MIT Press, Cambridge, MA.
12. Moss, S.; Dixon, H.; Wallis, S. (1995). Evaluating Competitive Strategies. Intelligent Systems in Accounting, Finance and Management. 4, 245-258. Also available at URL: http://www.cpm.mmu.ac.uk/cpmrep02.html
13. Moss, S.; Edmonds, B.; (1994) Modelling Learning as Modelling, CPM Report 3, Centre for Policy Modelling, Manchester Metropolitan University. Available at URL: http://www.cpm.mmu.ac.uk/cpmrep03.html
14. Moss, S.; Kuznetsova, O. (1995). Modelling the Process of Market Emergence. MODEST (Modelling of Economies and Societies in Transition), Warsaw, 1995.
15. Palmer, R.G. et. al (1994). Artificial Economic Life - A simple model of a stockmarket. Physica D, 75, 264-274.
16. Penrose, E.T. (1959) The theory of the Growth of the Firm, Blackwell, Oxford.
17. Russell, S.J.; Grosof, B.N. (1990): A Sketch of Autonomous Learning using Declarative Bias. In: Machine Learning, Meta-Reasoning and Logics. (Eds: Brazdil, P.B.; Konolige, K.) Kluwer Academic, Boston, 19-53.
18. Simon, H.A. 1972. Theories of Bounded Rationality. In McGuire, C.B.and Radner, R. (eds.) Decision and Organization. North-Holland.: Amsterdam.
19. Vriend, N.J. (1995). Self-organization of markets: an example of a computational approach. Computational Economics, 8, 205-232.
20. Wallis, S.; Edmonds, B.; Moss, S. (1995). The Implementation and Logic of a Strictly Declarative Modelling Language (SDML). ES'95, Cambridge, UK.
21. Wooldridge, M. (1995). Time, Knowledge, and Choice. ATAL'95 at IJCAI'95, Montreal. Published in (1996), Lecture Notes in Artificial Intelligence, 1037, 79-96. Available at URL: http://www.doc.mmu.ac.uk/STAFF/mike/atal95.ps