

# The Importance of *Representing* Cognitive Processes in Multi-Agent Models

Bruce Edmonds and Scott Moss

Centre for Policy Modelling,  
Manchester Metropolitan University,  
Aytoun Building, Aytoun Street, Manchester, M1 3GH, UK.  
<http://www.cpm.mmu.ac.uk/~bruce><http://www.cpm.mmu.ac.uk/~scott>

**Abstract.** We distinguish between two main types of model: predictive and explanatory. It is argued (in the absence of models that predict on unseen data) that in order for a model to increase our understanding of the target system the model must credibly *represent* the structure of that system, including the relevant aspects of agent cognition. Merely “plugging in” an existing algorithm for the agent cognition will not help in such understanding. In order to demonstrate that the cognitive model matters, we compare two multi-agent stock market models that differ only in the type of algorithm used by the agents to learn. We also present a positive example where a neural net is used to model an aspect of agent behaviour in a more descriptive manner. **Keywords:** modelling, methodology, agent, economics, neural net, genetic programming, representation, prediction, explanation, cognition, stock market, negotiation.

## 1. Introduction – types of modelling and their goals

There are two ways in which our models are constrained by reality: *verification* and *validation*. Verification is where the *design* of a model is constrained by *prior* knowledge, for example when a chemist’s model of interacting atoms is constructed so that it is consistent with atomic physics. Validation occurs *after* the model has been constructed – it is a check that the output of the model is consistent with known behaviour, for example by testing to see whether the error of a model’s predictions with respect to unseen<sup>1</sup> data can be explained by noise.

Different types of models relate to different kinds of goals and involve verification and validation in different ways. For example in a predictive model, where the goal is to predict unseen data, the validation is all important and comes first. Later the construction of the model (i.e. the verification) may be examined to try and find out *why* the model successfully predicts the phenomena.

Another important type is the explanatory model, here the goal is to explain the outcomes in terms of the design – the validation does not need to be done with unseen

---

<sup>1</sup> A hold-out sample is insufficient for a predictive model as (Mayer 1975) graphically illustrates.

data, but the design must be a credible *representation* of the structure of the process if the explanation is to be of any use. The methodology of modelling is discussed in greater detail in (Edmonds, 2000).

This approach differs significantly from the standard modelling procedures in the social sciences – especially conventional economics and its derivatives in political science and sociology. For example, in conventional economic theory, competition is an unmodelled process that is claimed to drive all economic actors to behave *as if* they were constrained optimisers. What is actually modelled in conventional economic theory is a competitive equilibrium that is said to capture the *result* of the unmodelled competitive process. A survey by Moss (forthcoming) of the 14 game theoretic papers<sup>2</sup> published in 1999 in the *Journal of Economic Theory* showed that seven of the papers proved the existence of a Nash or similar equilibrium for an n-person game, six papers reported models and results for two-person games (sometimes in round robin tournaments) and one paper reported results for a three-person game. There were no papers reporting the *process* of any game with more than three players.

A social process involves a process of interaction among individuals. By ignoring the process, or considering only very limited numbers of agents (i.e. between 1 and 3 agents), economic modellers do not even seek to capture social interaction. In consequence, it is not possible to validate their models against observed social behaviour. Instead, with extremely strong assumptions they hope to take a “short cut” to capture the outcomes only – in other words to make a predictive model. The trouble is that they do not predict, because they fail on unseen data. There seems to be a sort of “double-think” going on – the use of hold-out sets rather than unseen data would be acceptable for an explanatory model and the lack of representational content would be acceptable for a predictive model, but main-line economic models fail to be either.

The nature of a social process depends upon the behaviour of the individuals in the course of their social interactions. In multi agent systems, that behaviour is represented by *some* specification of individual cognition. That specification will not capture all of the behaviour of the individual, of course, but only what is relevant to the modelling purpose – i.e. what elements of the modelling target is going to explain the outcomes. For an explanation to be effective these elements need to be present in the target systems and they have to be necessary for the outcomes.

The use of multi-agent modelling techniques is a step towards descriptive modelling, because the separate entities (agents) in the model correspond to the separate entities in the target domain (people or institutions). In other words the *structure* of the target system is *represented* in the model at least to some extent as an essential part of the model design. In practice, because the agents in the system are identifiable as entities in the target domain, the interactions between agents in the model are programmed so that they also correspond to those between the entities modelled.

It is still common in multi-agent modelling to take an existing algorithm (designed with another purpose in mind) and use it for the cognition of the agent, regardless of

---

<sup>2</sup> A paper was judged to be concerned with game theory if “game theor\*” appeared in its title, abstract or among its key words.

whether this can be justified in terms of what is known about the behaviour of the modelled entities. For, example there have been many economic models which have utilised standard Genetic Algorithm techniques<sup>3</sup> without regard for whether this can be justified by reference to the target domain (Chattoe 1998).

In this paper we will argue that neural network techniques can be a valuable part of the multi-agent modellers palette, but that simply “plugging in” existing techniques will not help in *understanding* what we model. In the absence of models that predict unseen data this seems to be simply an anachronism. Rather, if we construct models whose structure *including that of the cognition* corresponds to a description of the target domain, then the behaviour of the whole system can provide an explanation of this behaviour in terms of the elements of that description.

Thus the algorithm that is to drive an agent’s behaviour must be, at least, *chosen* and *adapted* to correspond to what is known concerning the modelled entity’s behaviour and cognition. This is as true for neural network techniques as for other types of algorithms – the fact that at a micro level there might be a correspondence between brain structure and NN techniques does not give any guarantee that this is true at the macro level of agent behaviour or in social phenomena arising from agent interaction.

Section 2 of this paper compares two artificial models of a stock market, which are identical except for the cognitive model of the traders – these are not designed to be realistic but merely to demonstrate that the type of cognitive model *matters*. Section 3 is a more positive example of multi-agent modelling which seeks to *represent* negotiators cognition using a mixture of techniques. We conclude in section 4.

## 2. A Negative Example – A Model of Stock Market Traders

To emphasis that the *type* of the cognitive algorithm is crucial to the behaviour of a whole system (and not merely its adjustment or parameterisation), I will describe two versions of a model of stock market traders (roughly following Palmer et al. 1994) where the only difference is that different types of algorithm were used for their decision making mechanism were used: a GP based algorithm and a NN algorithm. The different cognitive models lead to qualitatively different outcomes – differences that I suggest will not be eliminated by any simple “tuning” of the algorithms. The corollary of this is that it matters that we use *appropriate representations* of cognition, and do not simply “plug in” algorithms “off the shelf”.

### 2.1 The inter-agent structure

First I will describe what is common to both versions of this model, namely what is exterior to the trading agents. Their environment and the structure and type of their interactions.

---

<sup>3</sup> Sometimes these are altered so that they more efficiently optimise on simple problems but are even *less* realistic!

There are seven traders, three stocks and one market maker. Each time cycle the market maker posts prices for the stocks and traders can buy (if they have cash or are also selling stocks) or sell (if they have that stock) at the price posted in each of the three stocks. They pay a fixed cost for each trade they make. They do not have to trade. There is a dividend for each stock which varies upwards and downwards in a slow random walk, so that different stocks have the best dividend rate at different times. This dividend is paid each cycle and is the only fundamental underlying the value of each stock. Cash and levels of the stocks are adjusted at the end of each cycle.

The market maker always accepts trades, unless it has run out of stock to sell, in which case it sells what it has left. The prices it sets are initially random but from then on are moderated according to demand – that is, if there traders are buying a stock it raises its price and if there are net sales it lowers the price.

The traders are initialised with 100 units of cash at the start and a small but random level of each stock. During the first five time cycles the traders can learn about the market but make small random purchases or sales. There after the assets of each trader depends on its trading decisions (to buy, sell or keep stocks). The traders' learning algorithms attempt to find actions that will increase its assets or, equivalently, to correctly predict stock price movements. The internal algorithm decides on its intentions, which is then modified by what is possible given its resources – this moderation is the same for both versions. The maximum purchase or sale of each stock is 5 units.

The inputs that the internal learning algorithm has in order to guide its actions are as follows:

- The prices of the three stocks during the previous two cycles and 5 cycles ago;
- The value of the stock index (average of the prices) during the previous two cycles and 5 cycles ago;
- The last actions of its contacts in terms of buying and selling the three stocks;
- And one constant: the Boolean value of “true”.

At the beginning of the simulation a randomised “contact structure” is imposed upon the traders, with each trader having three contacts, that is a specified and fixed set of three of the other traders. Each trader can observe its contacts, that is each cycle the actions of its contacts in terms of buying or selling each of the three stocks are employed as additional inputs for its decision making algorithm. Thus each trader could learn to imitate another trader's actions.

## 2.2 The GP cognitive algorithm version

The GP learning algorithm is a strongly typed GP algorithm following (Montana 1995). Each trader has 20 “models”, where each model is made up of a GP tree for each stock. The nodes are as follows:

```
IDidLastTime indexLastTime indexNow maxHistoricalPrice
AND divide doneByLast greaterThan lastBoolean
lastNumeric lessThan minus NOT OR plus priceDownOf
priceLastWeek priceNow priceUpOf times
```

The terminals consist of the names of its contacts, the names of the stocks, *True* and *False*. The evaluation of the models is done by assessing what the trader's total assets would be if it had employed the model over the last 5 time periods. The operations are 85% tree crossover, 5% new randomly generated models and 10% propagation.

Figures 1 and 2 give a flavour of the resulting interactions of the agents. Figure 1 below shows the prices of the three stocks and figure 2 the total assets (at current stock prices) of the agents.

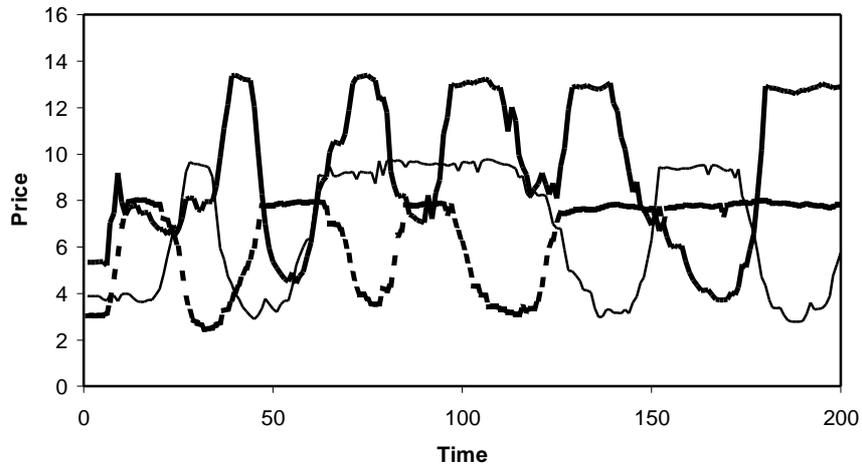


Fig. 1. Price of stocks in the GP version

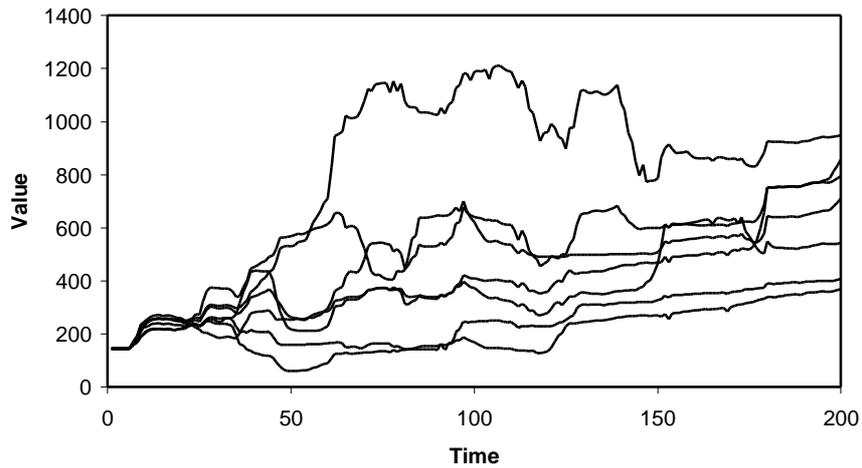


Fig. 2. Total value of trader's assets in the GP Version

There is quite a lot of market activity – traders continue to drop in out of stocks resulting in speculative “cycles”. As a result the fortunes of the different traders vary over time in comparison with each other.

### **2.3 The NN cognitive algorithm version**

The neural net version of the model broadly follows (Chialvo and Bak 1999). This has a standard fixed and layered topology with weights on the arcs which are adjusted. This is a winner-take-all network where only the strongest node in each layer (after the input) fires. Feedback is entirely on the basis of error, the arcs that fired leading to an erroneous prediction are slight depressed. The NN used is a variation of this to allow more than one set of arcs to fire in parallel – a “critical level” is set and all those nodes whose inputs sum to over this level fire, the feedback is as before but with the addition that the critical level is adjusted if the network is generally under firing.

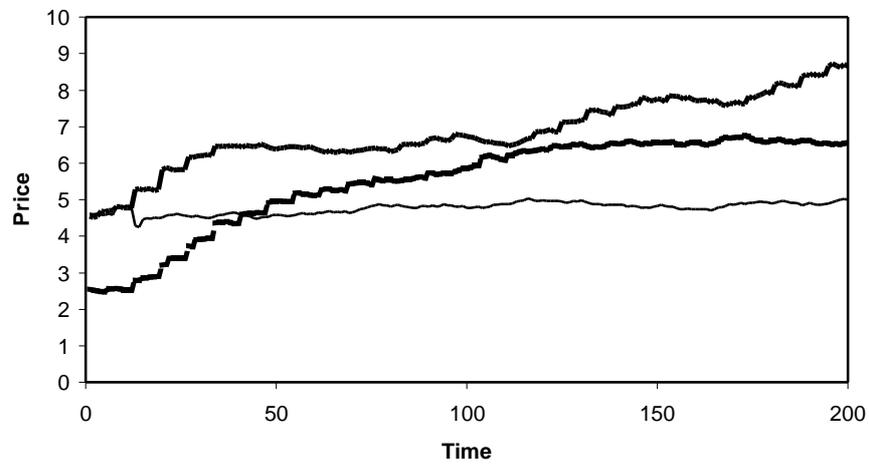
In this network there were 47 inputs two intermediate layers of 35 nodes each and 11 output nodes. The inputs were as follows:

- For each of the three stocks, for the last and previous time whether it has gone significantly up, down or not changed;
- For the index, for the last and previous time whether it has gone significantly up, down or not changed;
- Whether the index is historically high or low;
- And for each of its contacts and each stock whether they bought, sold or did nothing.

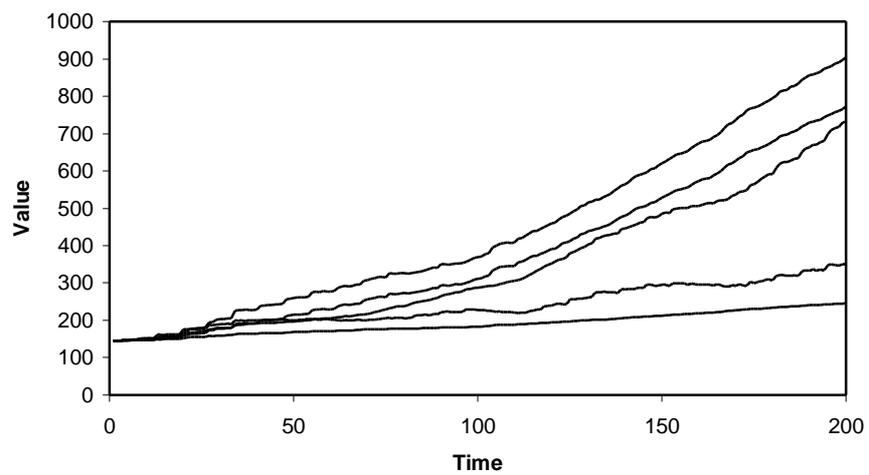
The predictive outputs were:

- For each of the three stocks whether it has gone significantly up, down or not changed;

The arcs were initialised with weights around 100 (the initial critical level) and decremented by a factor of 0.99 when leading up to an erroneous prediction.



**Fig. 3.** Price of stocks in the NN version



**Fig. 4.** Total Value of trader's assets in the NN Version

Here we have a very different picture – traders invest early, predicting that prices in a couple of stocks will go up, which turns out to be a self-fulfilling prophecy. Those traders that bought into those stocks first did best. The assets of traders increase largely as a result of the dividends they gain on those stocks.

## 2.4 Some speculations about the causes of the different outcomes

It would be almost impossible to prove that the two algorithms could not be “tuned” so they produced qualitatively similar results, but I would argue that this unlikely. It is unlikely because the qualitative difference in outcomes can be explained in terms of the different characteristics of the two algorithms.

The GP algorithm is far more “brittle” and noisy – there is a continual stream of new models being created as part of its operation, and small changes in a model’s content can mean a very large change in its impact. The NN algorithm is also a critical system – paths that are just below the critical level are not further depressed and have the potential for “resurfacing” when the current paths fail, but whilst there is a small probability that a successful model in the GP algorithm can be forgotten, this is not the case in the NN, where a successful path will remain as long as it is successful. Finally, while the GP algorithm employs a retrospective evaluation of models, the NN is incremental – it can build up the feedback a little bit each time period. The incrementally means that the NN will tend to be more consistent in its behaviour whilst the GP can be more context-sensitive (for example if the market entered a new phase exactly 5 periods ago).

The point is *not* that one algorithm is a *better* than the other, but which more accurately reflects the happenstance of real traders’ behaviour. In this case the one algorithm might be a better representation of one kind of trader and the NN of another.

## 3. A Positive Example – A Model of Multilateral Negotiation

The model of multilateral negotiation reported in this section is under development as part of a European project on using multi agent systems to inform and support the management of water resources across Europe under anticipated conditions of climate change. A characteristic of water issues is that they involve conflicting interests. Extraction of water for irrigation conflicts with leisure and environmental interests. Flood protection schemes can rely on dykes, which shift the flooding problem downstream, or on the creation of flood plains that enthruse environmentalists but impact heavily on farmers. Industrial water use conflicts with domestic water quality. Even the list of conflicts is far from exhaustive. Moreover, the number of issues relevant to the various stakeholders is large.

The purpose of the model is to identify, both in general terms and in relation to particular cases within European river basins, how to structure negotiations over a large number of conflict-laden issues among interested stakeholders in order to achieve a commonly acceptable set of outcomes. The implementation was designed specifically to investigate the importance of different attitudes among negotiations towards the behaviour of other negotiators.

It is possible – and certainly should not be excluded by assumption – that over the course of any complicated negotiation process, there will be changing criteria for determining with which other agents one should seek agreement. Consequently, any representation of cognition that supports the selection of negotiating partners must

allow for flexibility and the evolution of selection criteria. The Chialvo-Bak (1999) learning algorithm is a clear candidate in this regard. Because synaptic strengths are only altered by small amounts when an expected or required result is not realised, agents incorporating this representation of cognition remain flexible and adaptive. Moreover, the winner-takes-all strategy, whereby the strongest of alternative synapses is always chosen, ensures fast execution.

Negotiators' cognition is represented by a problem space architecture within which the Chialvo-Bak specification of neural networks is used to implement learning behaviour (Chialvo and Bak 1999). There is a middle layer network for each agent composed of a random network of 200-300 neurons.

An abstract, canonical model has been implemented that can represent specific local sets of issues. This was achieved by ascribing to each agent a digit string to represent its negotiating position. Each digit in the string could take any integer value within a user-specified range. A second digit string of the same length indicated the importance to the agent of the values of the digits at corresponding positions on the string representing the negotiating position. So if the negotiating position string were

1 5 3 7 6 9 2

and the string indicating the importance of the elements of the negotiating position were

3 1 2 5 8 3 4

then the value of the first string at position 5 (with a value of 8 in the second string) would be the most important element in the agent's negotiating position. The least important would be the negotiating issue represented by the digit at position 2 corresponding to the value 1 in the importance string. Evidently the positions represented by the first and sixth digits in the position string were of equal and moderate importance.

The simulation experiments were run with either 30 or 40 negotiating positions which could take any of five values. There were nine negotiating agents.

In line with the conflict resolution literature, each agent would seek agreement with other agents on the positions which it found least important and then on the next least important, and so on until full agreement were reached. Agents will agree to less important positions held by another agent when, by so doing, the other agent agrees to a position that the first agent finds more important.

It is not too surprising that we have found the *sine qua non* of successful negotiation to be a common commitment to achieve some agreement. We have also found that the standard prescription for conflict resolution – agree on the least important elements first – does not result in overall agreement in a setting with many negotiators. This is because, once differences have been resolved with one agent, resolving differences with another agent might involve reopening or opening up new differences with the previous negotiating partner. It seems likely that a necessary characteristic of the negotiation process will be the emergence and evolution of coalitions among negotiators. Further simulations and development of the model will be required to investigate appropriate means of creating such.

The inputs to the neural network are collections of endorsements of other agents. Another agent will be endorsed as "trustworthy" if it reaches an agreement and is subsequently found to have observed the agreement. An agent will be endorsed as "reliable" if it responds positively to an offer made by the endorsing agent. An agent

is “helpful” if it brings together two other agents that reach some sort of agreement. An agent is similar if it shares more of the same negotiating position than any other agent known to the endorsing agent. There is also an endorsement “known” to cover those agents that are known to the endorsing agent. Finally, there are three negative endorsements: untrustworthy, unreliably and unhelpful which obvious meanings.

Each combination of endorsements is associated with an input neuron with synaptic connections to seven neurons of the middle layer network. The initial order of goodness is determined by randomly ordering the individual endorsements from best (with value 5) to least good (with value 1). The pejorative endorsements are given the negative of the value of their positive counterparts. Then all combinations of endorsements are valued at the sum of the values of their components and, for each value, an output neuron is created.

When all of the neurons are created, each neuron that is not an output neuron is linked by synapses to seven other neurons within that network. Each synapse is given a strength of  $1 \pm$  a small random perturbation in the interval  $[0, 0.02)$ . On the first occasion when an agent encountered a set of endorsements, the network was trained to associate that set with its predetermined initial value. The training took the form of finding a synaptic path from the neuron associated with the endorsement set to an output neuron. The synapse with the largest weight from each neuron was selected to form the path. Following the Chialvo-Bak algorithm, if the output neuron did not correspond to the correct value of the endorsement set, weight of every synapse in that path was reduced by 0.02 and the sum of the weight reductions was distributed among all of the other synapses in the network. This procedure was repeated until the input neuron associated with the endorsement set was linked by the strongest synapses to the correct output neuron. The choice of which agent’s negotiation offer to accept is determined by which agent is associated with the highest ranked set of endorsements. Having chosen an agent’s offer in this way, if an agreement on any of the negotiating issues results, the synapse values are left unchanged. If, however, no agreement is reached, then the weights of the synapses connecting the input neuron to the rank-determining output neuron are reduced and the weights redistributed as described above.

In the simulation experiments, we capture the ranking of endorsement sets triggered by each agent at each time step in order to determine whether the specific ranking makes any difference to the pace and outcome of the negotiations. If the specific ranking does make a difference, then it is of obvious interest to note which endorsements are most effective in choosing agents with which an agreement will in fact be reached.

#### **4. Conclusion**

Neural network algorithms can be a valuable addition to the palette of techniques available to the multi-agent modeller. However their use needs to be justified by reference to the behaviour of individuals in the target domain if the model is to have any value in terms of understanding, just as with any other approach. It is

theoretically possible that a model might be predictively successful on unseen data but, at present this seems unlikely without greater understanding being gained first.

There are some problem with using algorithms that are produced by fields such as neural networks or evolutionary computing, and that is that there is very little understanding about their domain of applicability. This makes it very difficult to know when a particular technique is appropriate. The tendency in many papers is to imply that a certain technique is *generally* better, rather than it has advantages for certain types of problem. This specificity of advantage seems inevitable given the “No Free Lunch” theorems of (Wolpert and Macready 1995, and 1997), which show (in a very abstract way) that no technique is better than the others over all possible problem domains.

## 5. References

- Edmonds, B. (2000) The Use of Models - making MABS actually work. In. Moss, S. and Davidsson, P. (eds.), Multi Agent Based Simulation, *Lecture Notes in Artificial Intelligence*, **1979**:15-32.
- Chattoe, E. (1998) Just How (Un)realistic are Evolutionary Algorithms as Representations of Social Processes?, *Journal of Artificial Societies and Social Simulation*, **1**(3), <http://www.soc.surrey.ac.uk/JASSS/1/3/2.html>
- Chialvo, D. R., Bak, P. (1999) Learning from mistakes, *Neuroscience* 90:1137-1148.
- Mayer, T. (1975). Selecting Economic Hypotheses by Goodness of Fit. *The Economic Journal*, **85**:877-883.
- Montana, D. J. (1995). Strongly Typed Genetic Programming, *Evolutionary Computation*, **3**:199-230.
- Moss, S. (forthcoming). Game Theory: Limitations and an Alternative, *Journal of Artificial Societies and Social Simulation*. Also available at <http://www.cpm.mmu.ac.uk/cpmrep80.html>
- Palmer, R.G. et. al. (1994). Artificial Economic Life - A Simple Model of a Stockmarket. *Physica D*, **75**:264-274.
- Wolpert, D. H. and Macready, W. G. (1997) No Free Lunch Theorems for Optimization, *IEEE Transactions on Evolutionary Computation*, **1**:67-82.
- Wolpert, D. H. and Macready, W. G. (1995) No Free Lunch Theorems for Search, Technical Report, Santa Fe Institute, Number SFI-TR-95-02-010, 1995. <http://www.santafe.edu/sfi/publications/Working-Papers/95-02-010.ps>