

Multi-agent simulation of different human societies

ISO project report

Roger Kingdon
MAC 2008-09

Abstract

In *The Republic* Plato hypothesises that there are five different types of city-state and that these correspond to five different individual personality types. It is noted that this 'Five State Hypothesis' (FSH) has been overlooked in the modern study of politics and society, but that it might be a suitable application for multi-agent system (MAS) modelling. Accordingly the FSH is reviewed in relation to modern political and personality typologies, and also in relation to existing MAS simulations of social phenomena. A very simple MAS is constructed to simulate agents independently foraging for food in a common environment. It is shown that by varying the values of global control parameters it is possible to promote the interests of particular agents while demoting others. This is interpreted as the ability of the MAS to simulate a number of different societies. Thus the MAS supports the central notion of the FSH that there is a one-to-one relationship between individual behaviour and collective behaviour. This constitutes a firm basis for further research into the behaviour of multiple macro-states (societies) whose characteristics are determined by interactions on the micro-state (agent) level.

Contents

1. Motivation: Plato’s ‘Five State Hypothesis’	1
2. Review of the Five State Hypothesis in relation to current multi-agent systems	2
2.1 Five personality types and five systems of governance	2
2.2 Grand descriptions and underlying mechanisms	3
2.3 Multi-agent system modelling of social phenomena	5
3. The design and operation of a simple multi-agent system for the simulation of different human societies.....	7
3.1 Program overview	7
3.2 Algorithm details.....	11
4. Simulation results	12
4.1 Control macro-state.....	12
4.2 Time-convergence.....	13
4.3 Extreme macro-states	13
4.4 Robustness	17
5. Discussion.....	18
Acknowledgements	18
References	19
Appendix: Multi-agent system code listing	21

1. Motivation: Plato's 'Five State Hypothesis'

In Book VIII of *The Republic* [1] Plato attributes to Socrates the extraordinary hypothesis that since there are five different types of person then there are five different ways of organising a city-state:

- Socrates:* The States are as the men are; they grow out of human characters.
Glaucon: Then if the constitutions of States are five, the dispositions of individual minds will also be five?
Socrates: Certainly.

In the subsequent discourse the five 'constitutions' of 'aristocracy', 'timocracy', 'oligarchy', 'democracy' and 'tyranny' are described in relation to particular human characteristics. What makes this hypothesis extraordinary is Plato's sheer audacity in assuming (without evidence) that (i) there are five different types of person, and (ii) there is a direct relationship between individual behaviour and collective behaviour. Perhaps it is this extreme audacity that has led scholars to reject out of hand what shall be called here the 'Five State Hypothesis' (FSH). Or perhaps the FSH has been rejected because, until recently, the only way that it could have been rigorously tested is to have conducted a wholly unethical series of 'social engineering' experiments on artificially-isolated human populations selected according to their perceived individual characteristics. Whatever the reason, although Plato's constitutions survive to this day – in name alone, and not even that in the case of timocracy – the FSH has been neglected to the point that it simply does not appear in modern political or sociological discussions. But this does not mean that it does not merit a second look; for, as it turns out, in multi-agent system (MAS) modelling we now have precisely the means of testing the FSH without resorting to unethical social engineering. As described above, the FSH presents the different ways of organising human society as the emergent consequence of the characteristic behaviour of many individuals acting independently within a common environment. That is, the FSH predicts a close relationship between individual and collective behaviour in the operation of a suitably-constructed MAS. Thus through an appropriate series of MAS simulations it should be possible to test this predicted relationship and thereby come to a more reasoned assessment of the FSH than has been possible heretofore. It is this exciting prospect that motivates the current study.

In section 2 the FSH is described and assessed in relation to modern personality typologies and political categories. This is followed by a brief review of existing MAS simulations of social phenomena and a discussion of the place of the FSH in this field of study. It is observed that integral to the FSH is the notion that several different human societies can be described and, by implication, modelled; and it is observed that this is an area of research that has yet to be explored in any depth using MAS. Accordingly, section 3 sets out the design and operation of a simple MAS for the simulation of different human societies. Simulation results are described in section 4, and the implications of these results are discussed in section 5. The full MAS code listing is given in an Appendix.

2. Review of the Five State Hypothesis in relation to current multi-agent systems

The FSH expresses the idea that there are five different types of person that give rise to five different systems of governance. What are these types of person, and what are the political systems? How do they relate to what is known today? As defined here, the FSH is a 'grand description', a political meta-theory on a par with Imperialism and Communism, but these historicist accounts have been discredited and abandoned in favour of analyses focussing on the underlying socio-economic mechanisms; so where does this leave FSH? Finally, in investigating the behaviour of underlying socio-economic mechanisms, it turns out that MAS modelling is a useful tool; so can this be used in application to the FSH? These questions are addressed in turn.

2.1 Five personality types and five systems of governance

Plato's five 'constitutions' and their determining 'dispositions of individual minds' are as follows [1]:

1. Aristocracy, the rule of the wise, is Plato's preferred system. He envisages for aristocrats the common ownership of property, wives and children. Aristocrats are distinguished by their love of wisdom.
2. Next best is Timocracy, the system of the Spartans. The Timocrat "is often the young son of a brave father", dominated by his mother, who complains that "his father is only half a man", and this leads the son to less noble pursuits. Timocrats are distinguished by their love of honour.
3. Oligarchy follows from Timocracy, through the accumulation of wealth. "Oligarchies have both the extremes of great wealth and utter poverty." Oligarchs are distinguished by their love for wealth.
4. Democracy follows from Oligarchy, through the revolution of the poor. Democrats are distinguished by their love of liberty.
5. Tyranny springs from Democracy. "The excess of liberty, whether in States or individuals, seems only to pass into excess of slavery." Tyrants are distinguished by their (paradoxical) enslavement to their own lusts and passions.

These constitutions are inherently stable: "Clearly, all political changes originate in divisions of the actual governing power; a government which is united, however small, cannot be moved." Thus, for example, tyranny can persist in a stable fashion and if a tyrannical regime remains united then (by implication) it can be overturned only by outside intervention.

Is there anything in the recent research literature to support the FSH? As it happens – yes there is. Looking first at the proposition that there are five distinct personality types, we find the following:

- The Five-Factor Model (also known as FFM, Big Five or OCEAN) is a psychometric test based on five distinct personality traits (openness, conscientiousness, extroversion, agreeableness and neuroticism) identified following a lexical analysis by Allport and Odbert [2], [3]. In using this model for estimating the effect of personality on earnings, Mueller and Plug [4] comment that "The five-factor categorization of personality is pervasive in the current personality and social psychology literature", and "The factorial structure has since been replicated in a large number of studies, cross-validated using a variety of questionnaire scales, and found to generalize across languages and cultures".
- The popular Myers-Briggs psychometric test is based on Jung's model of four functional types (sensation, thinking, feeling and intuition) combined with two attitude types (introvert and extrovert) to give a total of eight personality types (the 'extroverted sensation' type, and so on). Arguably the four functional types are all aspects of the introverted mind, so with the inclusion of extroversion we recover a set of five types equivalent to the Five-Factor Model.
- Building on his theory of 'multiple intelligences', psychologist Howard Gardner has identified 'five minds for the future' [5].
- Maslow's fivefold hierarchy of human needs [6] has been extremely influential throughout the social sciences, for example, in development studies and in the framing of human rights legislation. In what

may be taken as a complementary study, Kübler-Ross has identified five stages in the process by which individuals experience grief and tragedy [7].

- Classically there are five types of love: agape, eros, philia, storge and xenia [8]. By coincidence, perhaps, there are five types of stalker [9].
- There are five types of speech act [10] and five negotiation styles [11].

The evidence for five systems of governance is similarly abundant:

- Freedman [12] identifies five dominant political ideologies of the nineteenth and twentieth centuries: Communism, Conservatism, Fascism, Liberalism and Socialism.
- Toynbee [13] identifies five ancient civilisations that he considers to be distinct and unaffiliated to others: Aegean, Egyptian, Indus, Sinic and Sumerian-Akkadian.
- Scottish Enlightenment thinker John Millar has identified five distinct 'ranks' in society [14].
- There are five Hindu castes, that is, four brahminical castes plus outcast pariahs.
- Business guru Peter Senge has identified five essential 'disciplines' of a learning organisation [15].

These and similar 'quincidences' are discussed in detail in my forthcoming book [16], which also sets out a novel analysis providing a strong rationale for the persistent appearance of the number five in these studies.

What about the second part of the FSH, that is, the assumption that there is a direct relationship between individual behaviour and collective behaviour? Whilst the observation of 'sets of five' on both individual and collective scales indicate that there is some kind of relationship, it is clear that we need to understand the underlying mechanism before ascribing any degree of causality or determinism. And on this underlying mechanism Plato is silent.

2.2 Grand descriptions and underlying mechanisms

The FSH is an attempt to describe and explain the origin and operation of entire city-states. As such it may be grouped with other grand descriptions (beloved of futurologists and quasi-religious prophets) of the mysterious and transcendent forces that drive society forward as one. To better understand the function and validity of such theories, it is instructive to take a closer look at two famous examples: Smith's 'invisible hand' and Marx's 'spectre of communism'.

Every individual... generally, indeed, neither intends to promote the public interest, nor knows how much he is promoting it. By preferring the support of domestic to that of foreign industry he intends only his own security; and by directing that industry in such a manner as its produce may be of the greatest value, he intends only his own gain, and he is in this, as in many other cases, led by an invisible hand to promote an end which was no part of his intention. [17]

A spectre is haunting Europe – the spectre of communism. All the powers of old Europe have entered into a holy alliance to exorcise this spectre: Pope and Tsar, Metternich and Guizot, French Radicals and German police-spies. [18]

The mechanisms underlying these 'mysterious and transcendent forces' are the 'division of labour' and 'class struggle', respectively:

The greatest improvement in the productive powers of labour, and the greater part of the skill, dexterity, and judgment with which it is anywhere directed, or applied, seem to have been the effects of the division of labour. The effects of the division of labour, in the general business of society, will be more easily understood by considering in what manner it operates in some particular manufactures... To take an example, therefore, from a very trifling manufacture; but one in which the division of labour has been very often taken notice of, the trade of the pin-maker... I have seen a small manufactory of this kind where ten men only were employed, and where some of them consequently performed two or three distinct operations. But though they were very poor, and therefore but indifferently accommodated with the necessary machinery, they could,

when they exerted themselves, make among them about twelve pounds of pins in a day. There are in a pound upwards of four thousand pins of a middling size. Those ten persons, therefore, could make among them upwards of forty-eight thousand pins in a day. Each person, therefore, making a tenth part of forty-eight thousand pins, might be considered as making four thousand eight hundred pins in a day. But if they had all wrought separately and independently, and without any of them having been educated to this peculiar business, they certainly could not each of them have made twenty, perhaps not one pin in a day; that is, certainly, not the two hundred and fortieth, perhaps not the four thousand eight hundredth part of what they are at present capable of performing, in consequence of a proper division and combination of their different operations. In every other art and manufacture, the effects of the division of labour are similar to what they are in this very trifling one; though, in many of them, the labour can neither be so much subdivided, nor reduced to so great a simplicity of operation. The division of labour, however, so far as it can be introduced, occasions, in every art, a proportionable increase of the productive powers of labour. [17]

The history of all hitherto existing society is the history of class struggles. Freeman and slave, patrician and plebian, lord and serf, guild-master and journeyman, in a word, oppressor and oppressed, stood in constant opposition to one another, carried on an uninterrupted, now hidden, now open fight, a fight that each time ended, either in a revolutionary reconstitution of society at large, or in the common ruin of the contending classes. In the earlier epochs of history, we find almost everywhere a complicated arrangement of society into various orders, a manifold gradation of social rank. In ancient Rome we have patricians, knights, plebians, slaves; in the Middle Ages, feudal lords, vassals, guild-masters, journeymen, apprentices, serfs; in almost all of these classes, again, subordinate gradations. The modern bourgeois society that has sprouted from the ruins of feudal society has not done away with class antagonisms. It has but established new classes, new conditions of oppression, new forms of struggle in place of the old ones. Our epoch, the epoch of the bourgeoisie, possesses, however, this distinct feature: it has simplified class antagonisms. Society as a whole is more and more splitting up into two great hostile camps, into two great classes directly facing each other – bourgeoisie and proletariat. [18]

Smith’s theory is based on a model of society in which individuals are distinguished by their trade, while Marx’s theory is based on a model of society in which individuals are distinguished by their class. Smith’s theory was the catalyst for an extraordinary research programme that has generated a mass of perceptive observations and tractable analytical results that today constitute the field of economics and underpin the global economic order. These include the theories of supply and demand, comparative advantage, equilibrium (Ricardo, Pareto), resources, labour, market failure (Malthus, Keynes), games and rational expectations (von Neumann, Nash, Muth). Marx’s theory, on the other hand, has been conclusively demonstrated to be false by the vast social experiment of Europe 1945-1990: Draw an ‘Iron Curtain’ across a continent, operate the two sides by two different theories, then after 45 years observe the difference. Why did Smith succeed and Marx ultimately fail? Partly because “Communism is like prohibition, it’s a good idea but it won’t work” [19]. But mainly because, whilst Marx seems addicted to his grand notion of an idealised communist world, Smith makes scant reference to the ‘invisible hand’, rather he focusses almost exclusively on describing the features of its underlying mechanism, that is, the division of labour. This difference in focus becomes strikingly clear if we count the occurrences of the relevant key phrases in *The Wealth of Nations* (380k words) and *The Communist Manifesto* (12k words), see Table 1.

Key phrases	<i>Wealth of Nations</i>	<i>Communist Manifesto</i>
‘invisible hand’	1	0
‘division of labour’	35	0
‘communist’, ‘communism’	0	57
‘class struggle’	0	8

Table 1. Occurrences of key phrases relating to grand descriptions and underlying mechanisms in *The Wealth of Nations* [17] and *The Communist Manifesto* [18].

In choosing to overlook the grand description and focus instead on the underlying mechanism, Smith set an example to his followers of the correct application of what has come to be known as ‘scientific method’, whereby theorising is restricted to that which can be tested empirically by individuals operating within a common culture that values (in particular) open scrutiny and iterative learning [20], [21], [16]. In the case of economics this has led to the development of a robust and reliable set of concepts and procedures that may be applied to a wide variety of practical situations. But can this ever be said to constitute a complete and convincing critique of the associated grand description of the ‘invisible hand’? Arguably not. Indeed, until the advent of MAS the only way to test a grand description directly would be to perform a series of ‘Iron Curtain’-style experiments, which is perfectly possible but wholly unethical, given the degree of suffering that may be experienced by some of the populations in the resulting societies. MAS modelling gives a way of directly testing grand descriptions such as the invisible hand without causing suffering. The approach is valid as long as we follow what has been described as scientific method and focus on aspects of the grand description and its underlying mechanism that are tractable and testable using the available tools.

The implication of the preceding analysis for the FSH is clear: In order to understand the dynamics of its ‘grand description’ (of the assumed direct relationship between individual and collective behaviour), it is necessary to first identify a plausible underlying mechanism. Although *The Republic* describes a number of underlying mechanisms (including, incredibly, division of labour and class struggle), no such explanation is given for the five ‘dispositions of individual minds’. And since subsequent scholars have chosen not to pursue this line of research, there can be few grounds to oppose the use of the model proposed in my forthcoming book [16]. This model, termed ‘IDEAL’, describes human personality traits in terms of combinations of five distinct ‘learning styles’:

1. ‘Empiricists’ learn about the world through observation.
2. ‘Idealists’ learn about the world through comparing and correlating information within a pre-existing framework.
3. ‘Theorists’ learn about the world through constructing and operating representative models.
4. ‘Activists’ learn about the world through action and reaction.
5. ‘Conformists’ learn about the world through reference to an established authority.

As with the Five-Factor Model, Myers-Briggs and Kolb’s four-stage learning cycle model [22], IDEAL learning styles are a normalised basis set. That is, the learning style of any person can be represented by five numerical factors, one for each of the five IDEAL learning styles, and these factors sum to unity. Clearly, within this model it is possible to define five ‘extremist’ personality types who each score highly on one of the five factors, as well as a sixth type having a ‘balanced’ profile. Using these five learning styles, [16] analyses a variety of published studies across the social sciences, including Plato’s FSH, for which: aristocrats are mapped to idealists; timocrats to theorists; oligarchs to conformists; democrats to empiricists; and tyrants to activists. Accordingly the IDEAL model provides a framework by which well-defined character traits can be ascribed to Plato’s five different types of person in a consistent fashion. This is the required ‘underlying mechanism’ that enables us to devise an appropriate MAS by which we might test the relationship between individual and collective behaviour.

2.3 Multi-agent system modelling of social phenomena

There is a well-established and ongoing research programme for the MAS modelling of social phenomena. For reasons discussed already, this programme has focussed on reproducing and building on tractable results within the economics literature. To take a few examples:

- The typical starting-point is the MAS simulation of the division of labour, trade specialisation, and the related asymptotic convergence to an optimal price-based equilibrium [23].
- Variations on this theme include looking at different global optimisation criteria (maximum utility, minimum variance *etc.*). Such models are ‘goal-setting’ in nature [24], [25], [26].
- Social welfare questions have been addressed in the literature using a variety of means, but most clearly formulated as game theory [27], [28]. Some of these idealised games are intractable using conventional analyses but turn out to be suitable applications for MAS. These include the Ultimatum Game used to study fair dealing [29], [30], and an application of the Prisoner’s Dilemma used to study ethnocentric favouritism [31].

At the other end of the scale there is a thriving ‘God sim’ software industry where computational agents interact to build virtual relationships, families, communities (*The Sims*) and even nations and empires (*Civilization, Age of Empires*). Again, for reasons already discussed, the relationship between this popular gaming activity and serious socio-economic research remains tenuous. Time and again, when playing these commercial games or their more ‘academic’ variants, the question arises: Why this choice of parameters, and none other? Many of the models are underpinned by a favourite theory of human behaviour; but, ultimately, the reason for the choice of this or that parameter is pragmatic, *i.e.* ‘it works’. This ‘heuristic’ approach to modelling is justifiable just as long as a preferred model does in fact perform as advertised and there is no other model that works better. In the present case of Plato’s FSH we observe the following:

- Integral to the FSH is the notion that several different human societies can be described and modelled. This is an area of research that has yet to be explored in any depth using MAS.
- MAS design requires an ‘underlying mechanism’ (or theoretical structure) by which the parameters that define agent behaviour may be derived. In the case of the FSH this underlying structure is given by the five IDEAL learning styles.
- In order to justify the claim that a MAS performs ‘as advertised’, it is necessary to first advertise its requirements! In the case of the FSH we require, as a minimum, (i) the ability to generate several (*i.e.* more than two, and preferably five) different ‘societies’, each distinguishable by the characteristics of its dominant class of agents, and (ii) the ability to switch between any two of these societies through the simple expedient of altering the values of ‘global’ parameters (*i.e.* parameters affecting the environment experienced by all agents). Any MAS displaying these features may be said to be supporting the FSH notion that there exist several different societies that may be defined in terms of a one-to-one relationship between individual and collective behaviour.
- The heuristic approach allows any choice of MAS, as long as it performs as advertised. However, MAS models are notoriously complex, having solution spaces that expand exponentially with the addition of each new free parameter, and in order to obtain the required limiting cases of several identifiable, stable ‘societies’ within a reasonable time, it is necessary to make the MAS as simple as possible. This is ‘Ockham’s Razor’ applied with a purpose: If the MAS is more sophisticated than is strictly necessary, then it is unlikely that it will easily manifest the several different societies as required.

These points effectively define the requirements for the MAS developed for this application.

3. The design and operation of a simple multi-agent system for the simulation of different human societies

A MAS has been developed that has the ability to generate several different stable macro-states, each distinguishable by the characteristics of its dominant class of agents, selected indirectly by altering the values of global parameters. In order to clearly identify the characteristic responses of each type of agent, their environment has been simplified to the bare minimum, specifically, it is a one-dimensional space through which the agents may roam in order to forage for 'food'. These food supplies are available in 'parcels' placed at particular locations in the environment at particular times. The parcels contain food and 'poison' in equal proportion. Food increases agent 'energy'; poison decreases it. Agents with more energy may travel faster through the environment, thereby enhancing their access to parcels. Different environments may be realised through the adjustment of global control parameters. Agent 'faculties' are application-specific distillations of the five IDEAL learning styles. These features have been coded as a Visual Basic for Applications (VBA) program that is accessed and operated by means of a MS Excel workbook called 'forage5.xls'. The following detailed design notes refer to screenshots of the workbook and to the listing of the VBA code that is given in the Appendix.

3.1 Program overview

On opening forage5.xls a warning box appears, asking the user whether the workbook's macros should be enabled or not. Selecting 'Enable Macros' activates the VBA program and opens the workbook at its Overview worksheet (Figure 1).

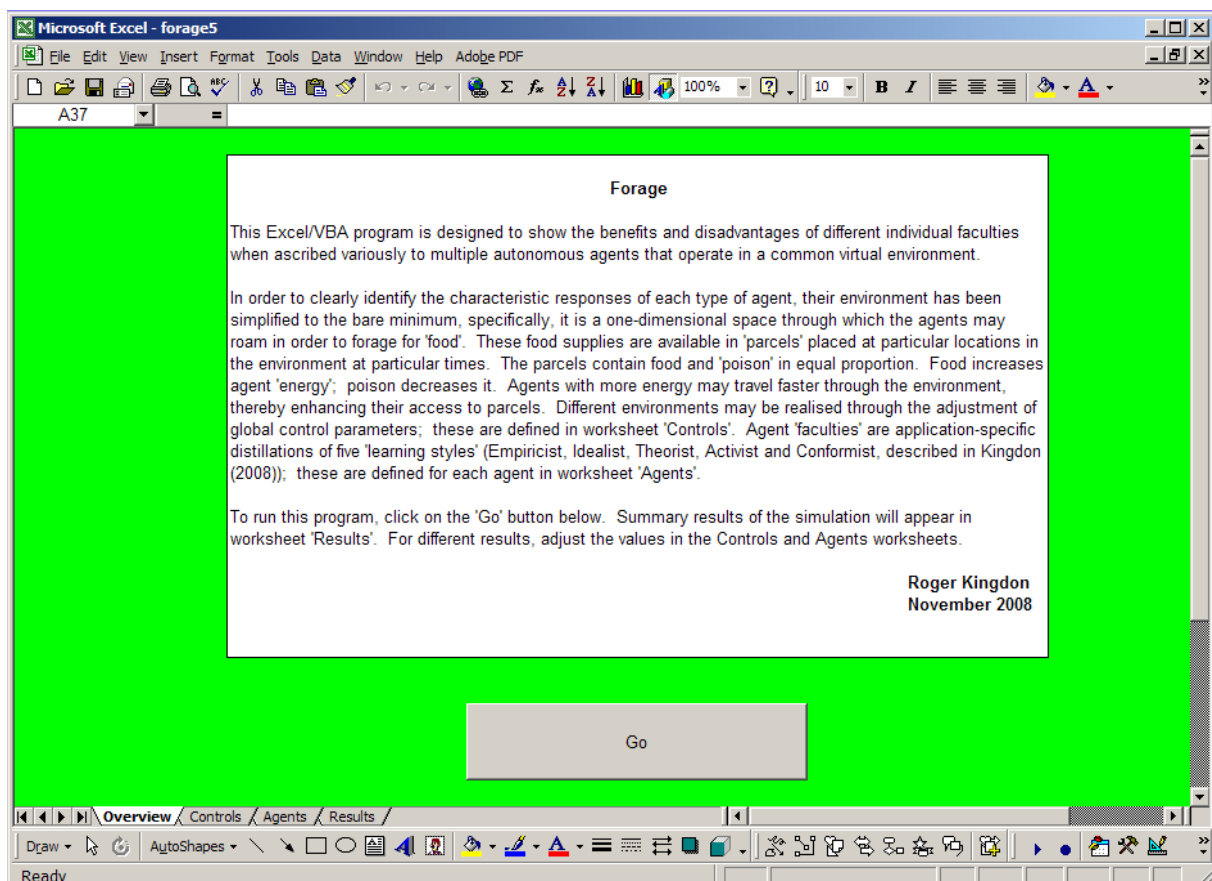


Figure 1. Overview worksheet of forage5.xls.

By clicking on the Go button the user runs the Forage subprogram (see Appendix). This program inputs parameter values from worksheets Controls and Agents, executes the MAS algorithm for a specified number of iterations, and writes the results of the computation to worksheet Results.

Values of global control parameters are specified in cells B2-B16 of worksheet Controls (Figure 2). Definitions of these parameters have been given in Table 2.

	A	B	C	D	E	F	G	H	I	J	K
1	Global control parameters	Value		Equal	Vision	Recognition	Efficiency	Precedence	Memory		
2	NumSpace	1000		1000	1000	1000	1000	20	1000		
3	NumTime	1000		1000	1000	1000	1000	1000	1000		
4	EnergyUseMax	0.2		0.2	0.2	0.2	0.2	0.2	0.2		
5	SpeedMax	5		5	250	5	5	5	5		
6	VisionMax	20		20	20	20	20	20	20		
7	RecognitionMax	10		10	10	2	10	10	10		
8	EfficiencyMax	10		10	10	10	0.2	10	10		
9	PrecedenceFraction1	0.5		0.5	0.5	0.5	0.5	0.5	0.5		
10	PrecedenceFraction2	0.3		0.3	0.3	0.3	0.3	0.3	0.3		
11	PrecedenceFraction3	0.2		0.2	0.2	0.2	0.2	0.2	0.2		
12	MemoryMax	20		20	20	20	20	20	2		
13	ParcelSpaceFreq	20		20	20	20	20	2	5		
14	ParcelTimeFreq	2		2	2	2	2	2	20		
15	ParcelRange	2		2	2	2	2	2	2		
16	ParcelSize	1000		1000	1000	1000	1000	1000	1000		
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											

Figure 2. Controls worksheet of forage5.xls.

Global control parameter	Definition
NumSpace	Number of discrete spatial cells in the one-dimensional environment. Integer.
NumTime	Number of discrete iteration timesteps. Integer.
EnergyUseMax	Maximum proportion of energy available for agent motion. Double.
SpeedMax	Maximum speed attainable by an agent. Integer.
VisionMax	Scale factor on the distance an agent can detect parcels. Integer.
RecognitionMax	Scale factor on an agent's ability to distinguish food from poison. Double.
EfficiencyMax	Scale factor on an agent's ability to convert food into energy. Double.
PrecedenceFraction1	Proportion of parcel available to agent first in pecking order. Double.
PrecedenceFraction2	Proportion of parcel available to agent second in pecking order. Double.
PrecedenceFraction3	Proportion of parcel available to agent third in pecking order. Double.
MemoryMax	Scale factor on an agent's ability to record parcel locations. Integer.
ParcelSpaceFreq	Modulus determining the spatial frequency of parcels. Integer.
ParcelTimeFreq	Modulus determining the temporal frequency of parcels. Integer.
ParcelRange	Spatial range within which parcels are distributed. Integer.
ParcelSize	Quantity of food or poison in each parcel. Double.

Table 2. Definitions of global control parameters specified in worksheet Controls.

Columns D-I of this worksheet contain alternative values of the global control parameters that have been used to generate different stable macro-states. These settings are implemented by copying and pasting from

the appropriate column to column B. Coloured cells indicate where parameter values differ from those for the default 'Equal' macro-state that favours agents having a balanced set of faculties.

Values defining agent faculties, and their initial locations and energies, are specified in columns A-H of worksheet Agents (Figure 3). Table 3 gives definitions of these parameters.

AgentID	Empiricist	Idealist	Theorist	Activist	Conformist	Location0	Energy0
1	0.6	0.1	0.1	0.1	0.1	50	100.0
2	0.1	0.6	0.1	0.1	0.1	50	100.0
3	0.1	0.1	0.6	0.1	0.1	50	100.0
4	0.1	0.1	0.1	0.6	0.1	50	100.0
5	0.1	0.1	0.1	0.1	0.6	50	100.0
6	0.2	0.2	0.2	0.2	0.2	50	100.0
7	0.6	0.1	0.1	0.1	0.1	150	100.0
8	0.1	0.6	0.1	0.1	0.1	150	100.0
9	0.1	0.1	0.6	0.1	0.1	150	100.0
10	0.1	0.1	0.1	0.6	0.1	150	100.0
11	0.2	0.2	0.2	0.2	0.2	150	100.0
12	0.6	0.1	0.1	0.1	0.1	250	100.0
13	0.1	0.6	0.1	0.1	0.1	250	100.0
14	0.1	0.1	0.6	0.1	0.1	250	100.0
15	0.1	0.1	0.1	0.6	0.1	250	100.0
16	0.1	0.1	0.1	0.1	0.6	250	100.0
17	0.2	0.2	0.2	0.2	0.2	250	100.0
18	0.6	0.1	0.1	0.1	0.1	350	100.0
19	0.1	0.6	0.1	0.1	0.1	350	100.0
20	0.1	0.1	0.6	0.1	0.1	350	100.0
21	0.1	0.1	0.1	0.6	0.1	350	100.0
22	0.1	0.1	0.1	0.1	0.6	350	100.0
23	0.2	0.2	0.2	0.2	0.2	350	100.0
24	0.6	0.1	0.1	0.1	0.1	450	100.0
25	0.1	0.6	0.1	0.1	0.1	450	100.0
26	0.1	0.1	0.6	0.1	0.1	450	100.0
27	0.1	0.1	0.1	0.6	0.1	450	100.0
28	0.1	0.1	0.1	0.1	0.6	450	100.0
29	0.2	0.2	0.2	0.2	0.2	450	100.0
30	0.1	0.1	0.6	0.1	0.1	450	100.0

Figure 3. Agents worksheet of forage5.xls.

Agent parameter	Definition
AgentID	Unique numerical identifier for each agent. Integer.
fV	Factor on the distance an agent can detect parcels. Double in range 0-1.
fR	Factor on an agent's ability to distinguish food from poison. Double, 0-1.
fE	Factor on an agent's ability to convert food into energy. Double, 0-1.
fP	Factor determining an agent's position in the pecking order. Double, 0-1.
fM	Factor on an agent's ability to record parcel locations. Double, 0-1.
Location0	Initial agent location. Integer in range 1-NumSpace.
Energy0	Initial agent energy. Double.

Table 3. Definitions of parameters specified in worksheet Agents.

These values are input to a dynamic data structure by subprogram ReadAgents, for as many agents as have been specified (one per row). All parameters are specified by the user, who is responsible for ensuring the uniqueness of AgentID (although this does not affect the computation), and that the five factors defining the agent faculties {fV, fR, fE, fP, fM} sum to unity. The default values specify five 'extremist' agents who each score highly on one of the five faculties (value 0.6, with value 0.1 for each of the rest), and a sixth agent having a balanced profile (value 0.2 each), at each of ten equally-spaced locations in the range 1-NumSpace, total 60 agents. This population can of course be modified or augmented by the user.

Once the computation has finished the user is alerted with an 'End of program' message and the Results worksheet is activated. This contains new parameter values that have been generated by the MAS algorithm. Figure 4 is an example screenshot of this worksheet.

Global control parameters		Value									
1	Global control parameters	Value									
2	NumSpace	1000									
3	NumTime	1000									
4	EnergyUseMax	0.2									
5	SpeedMax	5									
6	VisionMax	20									
7	RecognitionMax	10									
8	EfficiencyMax	10									
9	PrecedenceFraction1	0.5									
10	PrecedenceFraction2	0.3									
11	PrecedenceFraction3	0.2									
12	MemoryMax	20									
13	ParcelSpaceFreq	20									
14	ParcelTimeFreq	2									
15	ParcelRange	2									
16	ParcelSize	1000									
17											
AgentID	Vision	Recognitio	Efficiency	Precedenc	Memory	Location	Energy				
19	1	0.6	0.1	0.1	0.1	0.1	61	157189			
20	2	0.1	0.6	0.1	0.1	0.1	171	29500			
21	3	0.1	0.1	0.6	0.1	0.1	20	96834			
22	4	0.1	0.1	0.1	0.6	0.1	14	42708			
23	5	0.1	0.1	0.1	0.1	0.6	14	17298			
24	6	0.2	0.2	0.2	0.2	0.2	46	214958			
25	7	0.6	0.1	0.1	0.1	0.1	166	100288			
26	8	0.1	0.6	0.1	0.1	0.1	24	22148			
27	9	0.1	0.1	0.6	0.1	0.1	150	46200			
28	10	0.1	0.1	0.1	0.6	0.1	14	43598			
29	11	0.1	0.1	0.1	0.1	0.6	490	41100			
30	12	0.2	0.2	0.2	0.2	0.2	159	205371			

Figure 4. Results worksheet of forage5.xls.

The results worksheet contains the following data:

1. The list of global control parameter values used in the computation.
2. The list of agents with their faculty values and final locations and energies.
3. Final lists of remembered parcel locations for each agent.
4. The final spatial distribution of parcels.

Of these, the main parameter of interest is the final energy of each agent, for this indicates which agents have prospered and which have suffered in the prevailing environment.

3.2 Algorithm details

When initiated by clicking the Go button on the Overview worksheet, subprogram Forage inputs the data from the worksheets (subprograms ReadControls and ReadAgents), executes the MAS algorithm in NumTime iterations (subprograms Provide, Survey, Travel and Consume), writes the results (subprogram WriteResults), and finally returns control to the user. The following discussion focusses on the central MAS algorithm that is executed on each iteration:

1. Subprogram Provide, which distributes the parcels into the spatial environment.
2. Subprogram Survey, where agents look around for parcels.
3. Subprogram Travel, which moves agents towards where there are (or were) parcels.
4. Subprogram Consume, where agents consume the contents of parcels, adjusting their energies accordingly.

In the following these parts of the algorithm are described in turn.

Subprogram Provide. First, the spatial domain is emptied of parcels. Then, for all spatial cells: If the current timestep is modulo ParcelTimeFreq and the current spatial cell number is modulo ParcelSpaceFreq, a cell is selected at random within domain ParcelRange centred on the current spatial cell, and this selected cell is populated with a parcel.

Subprogram Survey. First, a direction is chosen at random for all agents to survey their environment (*i.e.* from right to left or from left to right). Then, for each agent: The agent examines the contents of the cells within its visual range, specified by the product $fV * VisionMax$. If it finds a parcel, the agent compares the location of the parcel with the previous locations of parcels stored in its memory. If the parcel location is new then it is recorded in memory. The memory of each agent is organised in time order with the most recent observation first, so that as observations are added their predecessors are pushed along, until they exceed the agent's memory capacity (given by the product $fM * MemoryMax$) and are deleted. If the agent does not find a parcel at a new location then it continues to look until it has scanned every cell within its visual range.

Subprogram Travel. For each agent: The agent measures the distance (in terms of number of cells) from its current location to the last-recorded parcel location. This distance is also the required speed of the agent if it is to reach the parcel location in the current (unit) timestep. The agent checks that this speed is less than SpeedMax and that the energy required to maintain this speed is less than a fraction EnergyUseMax of its current energy, and adjusts accordingly. Energy is calculated as energy = work = (constant) force * distance, *i.e.* energy = distance. Travelling at its adjusted speed, the agent moves towards the parcel location, with the agent location and energy being updated accordingly. If on arrival at the target cell the agent finds that it is empty, and if it has sufficient energy available, then it travels to the next-previous-recorded parcel location, and so on. If the agent does not find any parcels then it wanders in a random direction until it has burned fraction EnergyUseMax of its energy.

Subprogram Consume. For each agent, if it finds that there is a parcel in its cell:

1. If $fR * RecognitionMax$ is greater than a random number in range 0-1 then the agent consumes the food in the parcel, gaining energy; otherwise, it mistakenly consumes the poison and loses energy. To account for these two cases variable Recognise is given value +1.0 or -1.0 respectively.
2. Variable Efficiency is given value $fE * EfficiencyMax$ (or 1.0, whichever is the smaller).
3. The agent's position in the pecking order, n , is determined according to the fP values of the agents in the cell. Variable Peck is given value $PrecedenceFractionn$ where n is 1, 2 or 3. Agents lower than third in the pecking order ($n > 3$) have Peck = 0.0.
4. The agent's energy is incremented by quantity $ParcelSize * Recognise * Efficiency * Peck$.

4. Simulation results

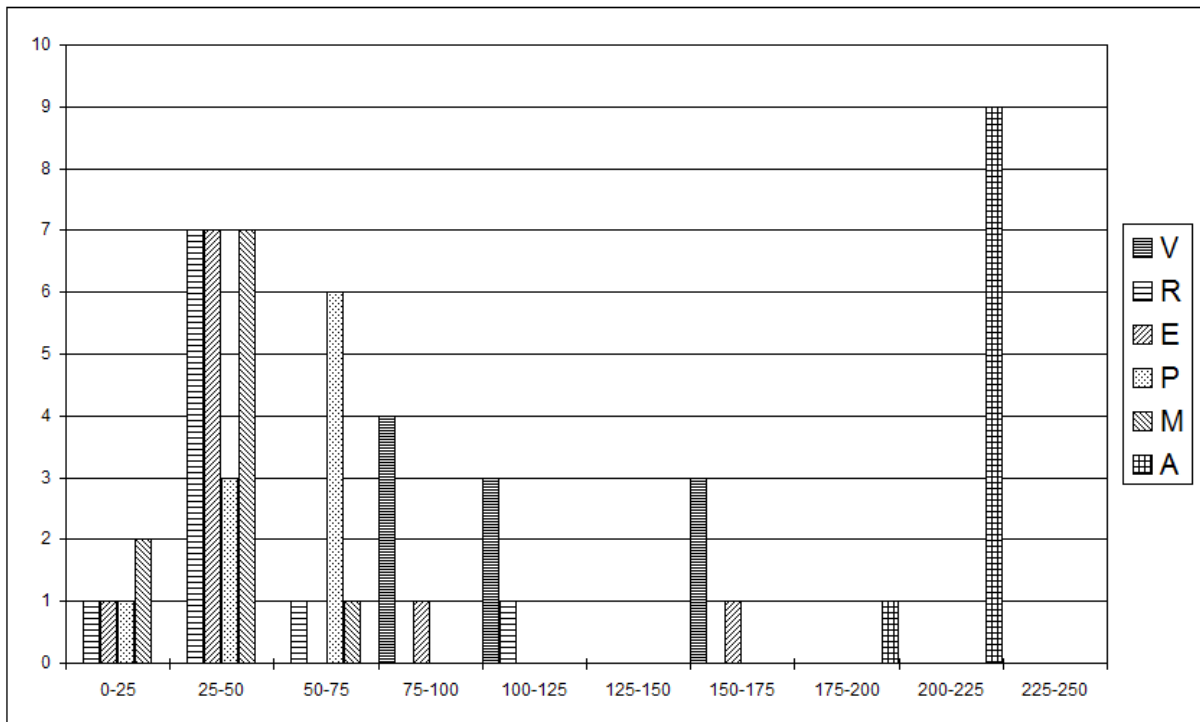
The MAS described in the previous section was applied to the FSH problem as follows:

1. Through systematic and careful adjustment of the global control parameters, a stable macro-state was obtained that favours agents having a balanced set of faculties ($fV = fR = fE = fP = fM = 0.2$). This was designated the ‘control’ or default state from which other states could be obtained.
2. This default macro-state was tested for time-convergence by doubling the number of iterations.
3. Further adjustments to global control parameters generated five other macro-states that each favour agents having ‘extremist’ profiles.
4. The robustness of these results was tested by scoping the global control parameters and finding the stable domains of the macro-states.

These results are described in turn.

4.1 Control macro-state

The values of the global control parameters selected to give the required control macro-state are given in Figure 2 (column B or D). Graph 1 is a histogram showing the energy distribution of six different types of agent {V, R, E, P, M, A}, these are defined in Table 4.



Graph 1. Histogram of the energy distribution of agents for the control macro-state. The abscissa scale is energy ÷ 1000 and the ordinate scale is number of agents.

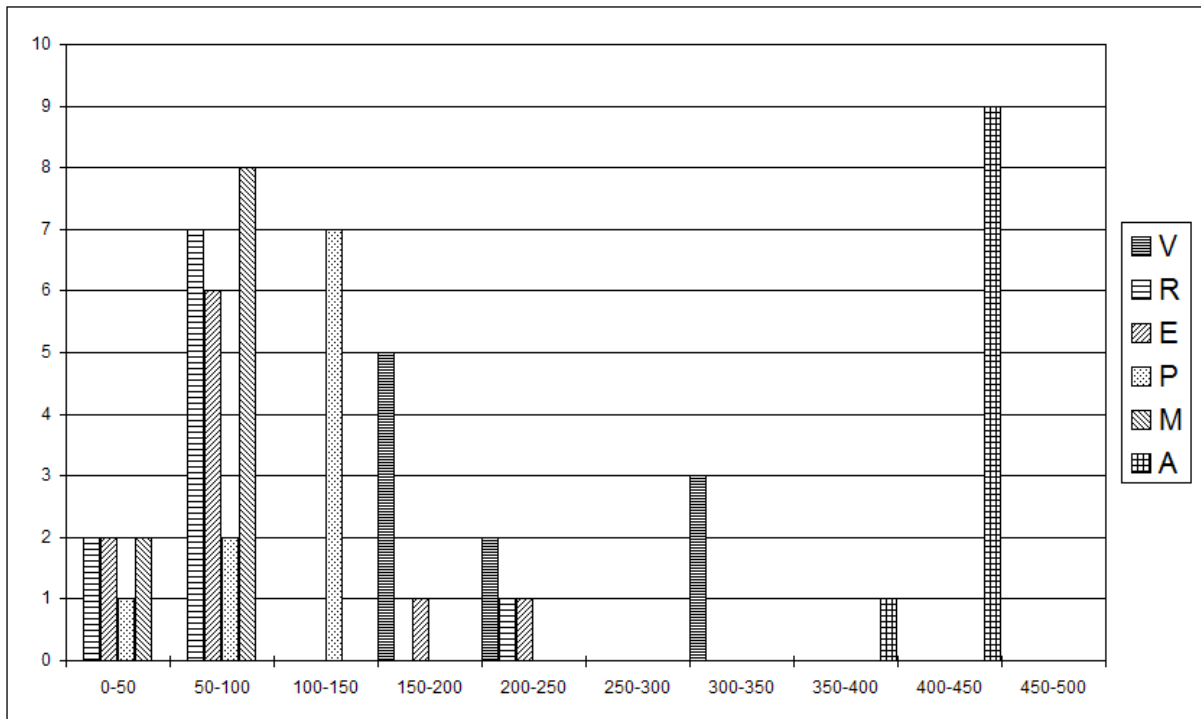
Type	Character	fV	fR	fE	fP	fM
V	High Vision	0.6	0.1	0.1	0.1	0.1
R	High Recognition	0.1	0.6	0.1	0.1	0.1
E	High Efficiency	0.1	0.1	0.6	0.1	0.1
P	High Precedence	0.1	0.1	0.1	0.6	0.1
M	High Memory	0.1	0.1	0.1	0.1	0.6
A	High All (<i>i.e.</i> balanced)	0.2	0.2	0.2	0.2	0.2

Table 4. Definitions of types of agent.

It is clear from Graph 1 that the selected values of global control parameters have created an environment that favours exclusively agents of type A. This defines the required control macro-state.

4.2 Time-convergence

The previous simulation was repeated with double the number of iterations, *i.e.* NumTime = 2000. The results are given in Graph 2.



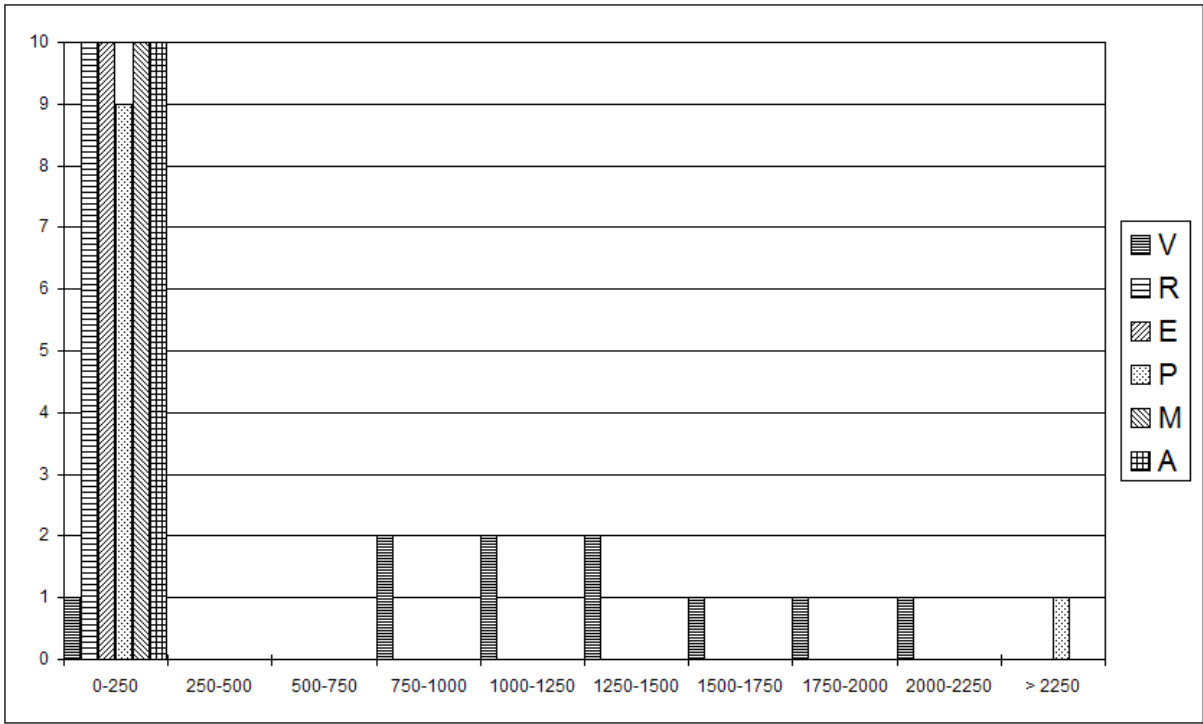
Graph 2. Histogram of the energy distribution of agents for the control macro-state after 2000 iterations. The abscissa scale is energy ÷ 1000 and the ordinate scale is number of agents.

The shapes of the energy distributions in Graph 2 are very similar to those in Graph 1, indicating that the control macro-state is a time-converged solution.

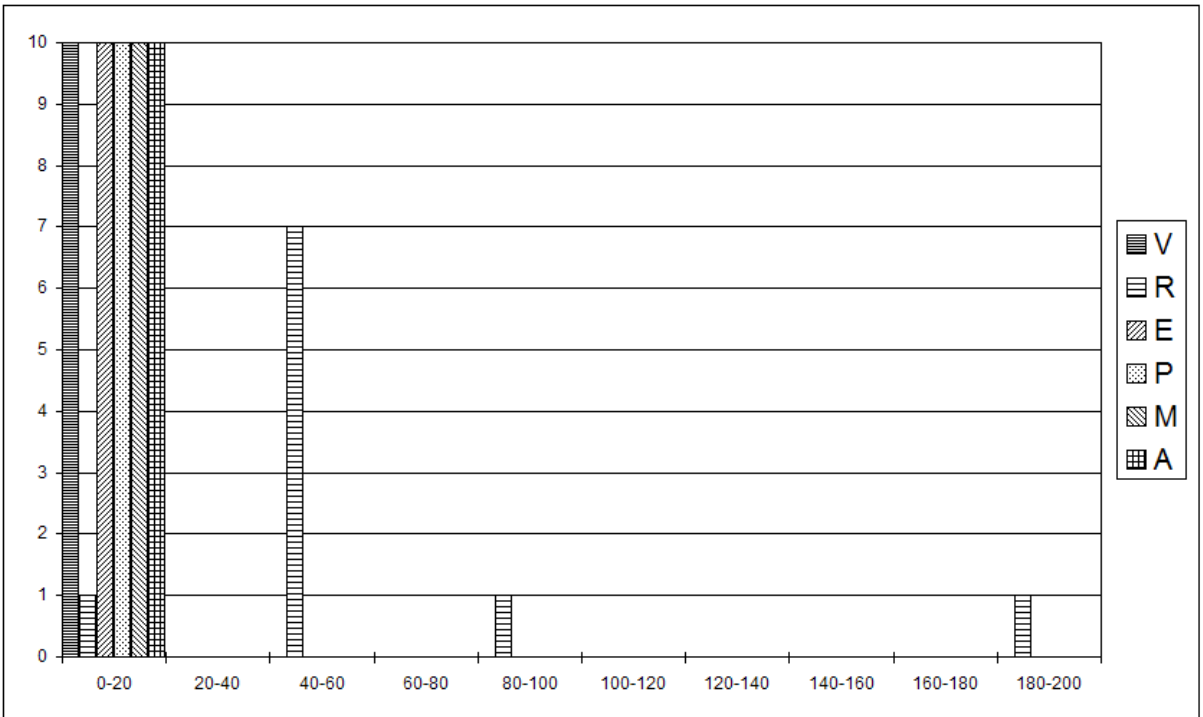
4.3 Extreme macro-states

By adjusting the global control parameters it was possible to generate five other macro-states, each favouring agents having 'extremist' profiles. In each case a premium was placed on adjusting as few parameters as possible in order to switch states.

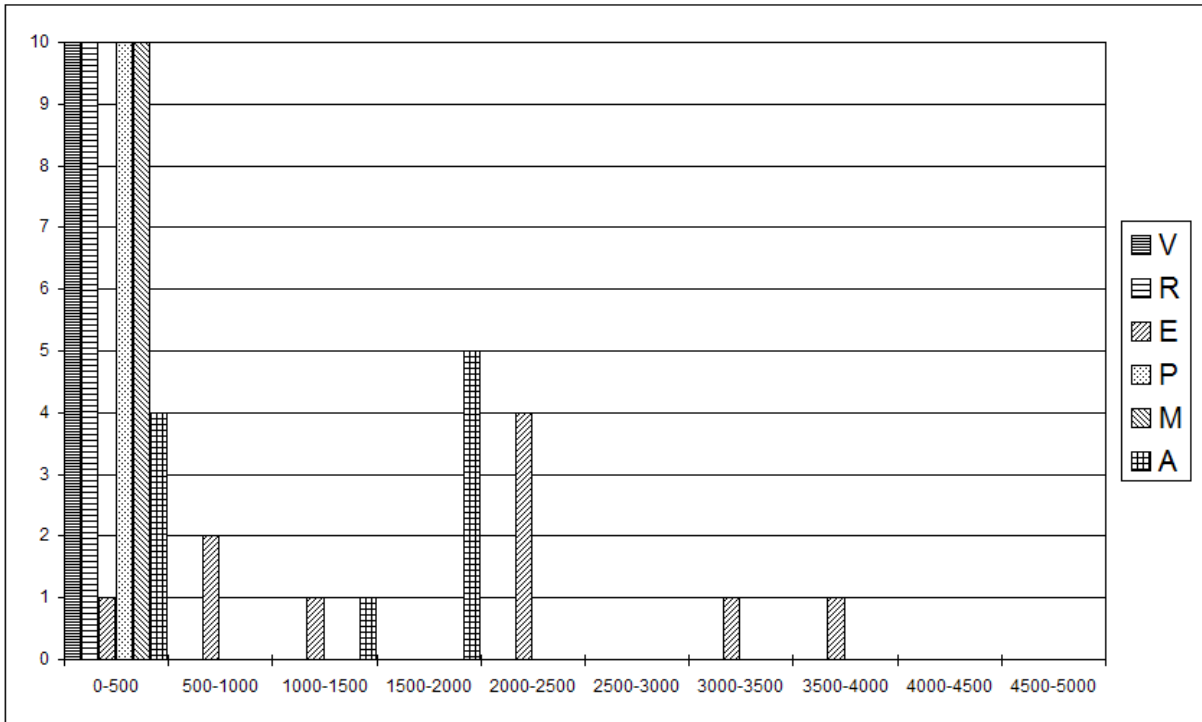
1. By increasing SpeedMax from 5 to 250 (Figure 2, column E), a macro-state was obtained that favours High Vision agents (and one High Precedence 'outlier' having energy 46,427). See Graph 3.
2. By decreasing RecognitionMax from 10 to 2 (Figure 2, column F), a macro-state was obtained that favours High Recognition agents. See Graph 4.
3. By decreasing EfficiencyMax from 10 to 0.2 (Figure 2, column G), a macro-state was obtained that favours High Efficiency agents. See Graph 5.
4. By decreasing NumSpace from 1000 to 20 and ParcelSpaceFreq from 20 to 2 (Figure 2, column H), a macro-state was obtained that favours High Precedence agents. See Graph 6.
5. By decreasing MemoryMax from 20 to 2 and ParcelSpaceFreq from 20 to 5, and increasing ParcelTimeFreq from 2 to 20, (Figure 2, column I), a macro-state was obtained that favours High Memory agents. See Graph 7.



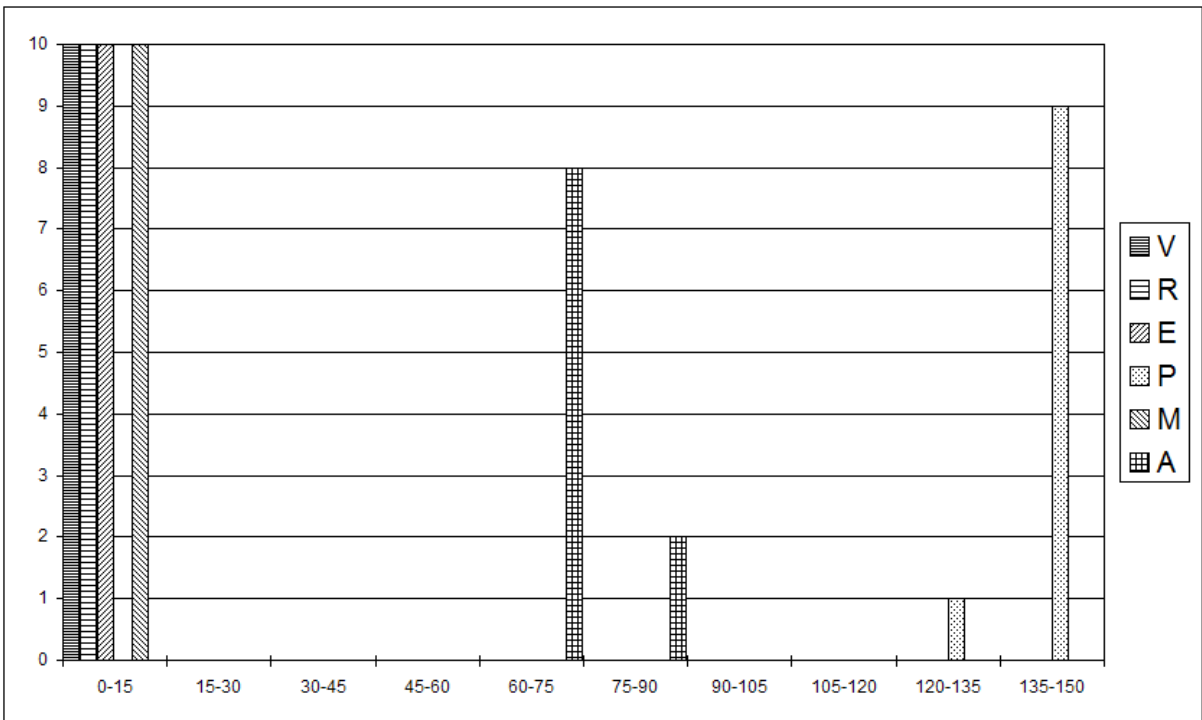
Graph 3. Histogram of the energy distribution of agents for the macro-state favouring High Vision agents. The abscissa scale is energy and the ordinate scale is number of agents.



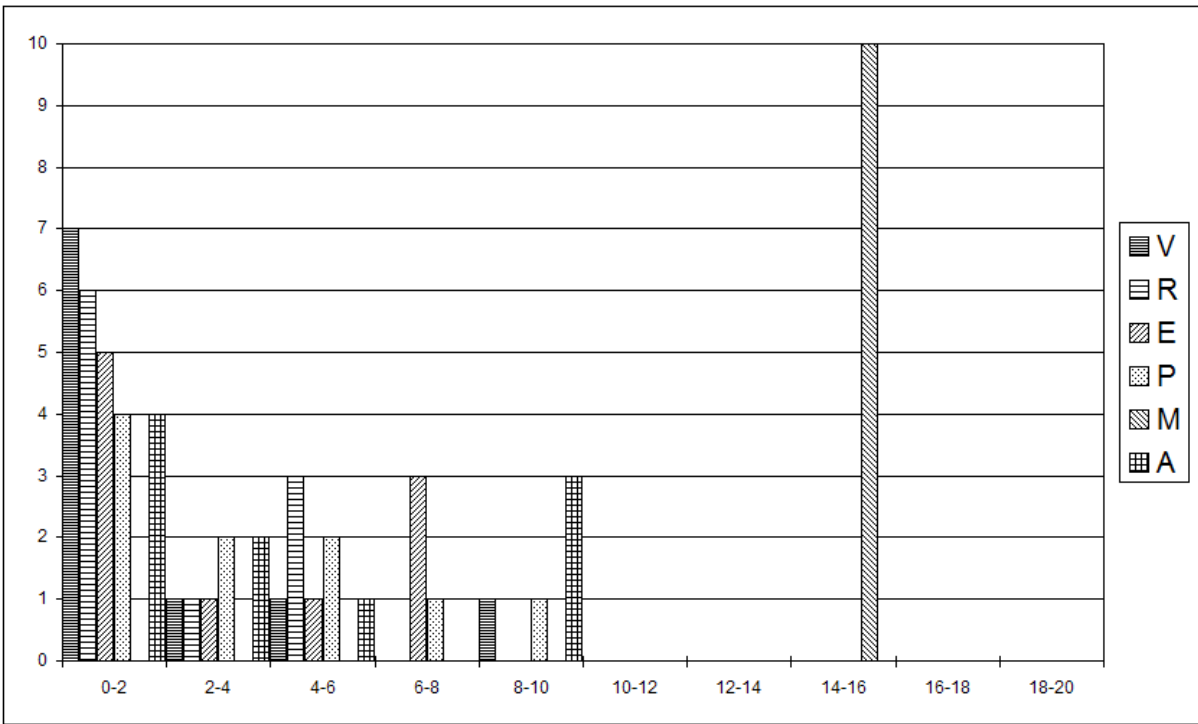
Graph 4. Histogram of the energy distribution of agents for the macro-state favouring High Recognition agents. The abscissa scale is energy ÷ 1000 and the ordinate scale is number of agents.



Graph 5. Histogram of the energy distribution of agents for the macro-state favouring High Efficiency agents. The abscissa scale is energy and the ordinate scale is number of agents.



Graph 6. Histogram of the energy distribution of agents for the macro-state favouring High Precedence agents. The abscissa scale is energy ÷ 1000 and the ordinate scale is number of agents.



Graph 7. Histogram of the energy distribution of agents for the macro-state favouring High Memory agents. The abscissa scale is energy ÷ 1000 and the ordinate scale is number of agents.

4.4 Robustness

By scoping the relevant global control parameters it was possible to find the stable domains of each of the macro-states, defined in terms of their dominant agent types (using the notation of Table 4). Table 5 shows the results of this scoping exercise. In each case only the specified parameters were altered, with all other parameters retaining their default (control macro-state) values. Generally it was found that, whilst stable macro-states exist for each of the extremist agent types, these macro-states have very restricted domains, indicating that they correspond to outlying and/or rare environmental conditions.

Scoping parameters	Values	Dominant agent type
SpeedMax	5	A
	10 - 200	V with significant others
	250 - 1000	V
RecognitionMax	≥ 10	A
	5	A
	4	A followed by R
	3	A and R equal
	2	R
	1	R
	0.5	R
	0.25	Mainly dead
	≥ 10	A
EfficiencyMax	5	A
	1	A followed by E
	0.5	A followed by E
	0.2	E followed by A
	0.1	E
	0.05	Mainly dead
	≥ 10	A
NumSpace, ParcelSpaceFreq	1000, 20	A
	500, 10	A
	200, 6	A and P equal
	100, 4	P followed by A
	50, 3	P followed by A
	20, 2	P followed by A
	10, 1	P followed by A
MemoryMax, ParcelSpaceFreq, ParcelTimeFreq	20, 20, 2	A
	2, 20, 2	No overall dominance
	20, 10, 10	A with significant others
	10, 10, 10	A
	5, 10, 10	A
	2, 10, 10	No overall dominance
	2, 10, 20	No overall dominance
	2, 5, 20	M
	2, 5, 30	M
2, 5, 50	No overall dominance	

Table 5. The results of scoping over the key global control parameters.

5. Discussion

The results of the preceding modelling exercise indicate that it is quite straightforward to construct a MAS capable of simulating a number of fundamentally different societies. By keeping to a simple MAS design it was possible to draw on ‘common sense’ knowledge when choosing the values of control parameters. For example, the High Precedence agents will only be dominant in a society where many agents are competing for food, so in this case NumSpace was reduced from 1000 to 20, thereby increasing the population density by factor 50. A weakness of the model is the somewhat artificial nature of the control parameters RecognitionMax and EfficiencyMax, whose values are direct factors for the recognition and efficiency faculties of each agent, whereas it was intended that the global control parameters should influence the agents in an indirect manner only, by changing their environment. Nevertheless, this indirect mechanism is clearly operating in the case of agent vision, precedence and memory, and this is sufficient to prove the point that the MAS can establish a nontrivial one-to-one relationship between individual and collective behaviour as suggested by the FSH. Therefore the modelling results can be taken to support this remarkable hypothesis first set down by Plato some 2350 years ago.

One benefit of the simplicity of the present MAS model is the ease with which it can be developed to investigate more complex questions. A number of possible developments spring to mind:

- *Parasitical behaviour.* Just as carnivores learn the significance of circling vultures, so too can agents be equipped to notice where other agents are congregating, and head in that direction. This ability could replace the unsatisfactory recognition faculty in the current model.
- *Cooperative team behaviour.* Just as humans (and many animals) learn to work in teams, so too can agents be equipped with an ‘agreeableness’ faculty that constitutes the ‘glue’ enabling them to coordinate their activities and share the proceeds. This ability could replace the unsatisfactory efficiency faculty in the current model, and could be used to investigate whether particular combinations of agent types are more compatible than others, as suggested in [16].
- *Feedback.* Once a particular agent type has become dominant then it is in a position to alter its environment in order to consolidate its control. For example, the MAS could be adapted to allow the spoiling and/or hoarding of food by dominant agents. It would be of interest to see how much this feedback mechanism contributes to the natural stability of states first noted by Plato. In turn this should inform the study of mass trauma and other collective behaviour.

It is hoped that these pointers prove useful in a future research programme.

Acknowledgements

This research was conducted under the supervision of Professor Abbas Edalat and it has benefited considerably from his ideas and guidance.

References

- [1] Plato, *The Republic* (circa 350 BC), tr. Benjamin Jowett (1856-93).
- [2] Gordon Allport and Henry Odbert, 'Trait-Names – A Psycho-Lexical Study', *Psychological Monographs*, vol. 47, no. 211 (1936).
- [3] Daniel Nettle, *Personality – What makes you the way you are* (Oxford 2007).
- [4] Gerrit Mueller and Erik Plug, 'Estimating the Effect of Personality on Male and Female Earnings', *Industrial and Labor Relations Review*, vol. 60, no. 1, pp. 3-22 (2006).
- [5] Howard Gardner, *Five Minds for the Future* (Harvard 2007).
- [6] Abraham Maslow, 'A Theory of Human Motivation', *Psychological Review*, vol. 50, pp. 370-96 (1943).
- [7] Elizabeth Kübler-Ross, *On Death and Dying* (Routledge 1973).
- [8] URL <http://en.wikipedia.org/wiki/Love> (23 December 2008).
- [9] Paul Mullen, Michele Pathé and Rosemary Purcell, *Stalkers and their Victims* (Cambridge 2000).
- [10] John Searle, *Speech Acts* (Cambridge 1969).
- [11] Douglas Walton and Erik Krabbe, *Commitment in Dialogue* (SUNY Press 1995).
- [12] Michael Freeden, *Ideology – A Very Short Introduction* (Oxford 2003).
- [13] Arnold Toynbee, *A Study of History* (Oxford 1972).
- [14] John Millar, *The Origin of the Distinction of Ranks* (third edition, 1779).
- [15] Peter Senge, *The Fifth Discipline – The Art and Practice of the Learning Organization* (Doubleday 1990).
- [16] Roger Kingdon, *Principia Intellegentia – The principles governing human and machine intelligence*, in preparation for publication (December 2008). The current draft may be downloaded for review purposes from http://www.idealectic.com/PIv4_1.pdf.
- [17] Adam Smith, *An Enquiry into the Nature and Causes of the Wealth of Nations* (1776).
- [18] Karl Marx and Friedrich Engels, *Manifesto of the Communist Party* (1848).
- [19] Will Rogers, *Weekly Articles* (1927).
- [20] Karl Popper, *The Logic of Scientific Discovery* (Hutchinson 1959).
- [21] Thomas Kuhn, *The Structure of Scientific Revolutions* (second edition, Chicago 1970).
- [22] David Kolb, *Experiential Learning – Experience as the Source of Learning and Development* (Prentice-Hall 1984).
- [23] Allen Wilhite, 'Self-Organizing Production and Exchange', *Computational Economics*, vol. 21, pp. 107-123 (2003).

- [24] Michael Wooldridge, *An Introduction to Multiagent Systems* (Wiley 2002).
- [25] Marco Janssen and Elinor Ostrom, 'Empirically based, Agent-based models', *Ecology and Society*, vol. 11, no. 2, art. 37 (2006).
- [26] Ulle Endriss, Nicolas Maudet, Fariba Sadri and Francesca Toni, 'Negotiating Socially Optimal Allocations of Resources', *Journal of Artificial Intelligence Research*, vol. 25, pp. 315-348 (2006).
- [27] URL http://en.wikipedia.org/wiki/Social_simulation (23 December 2008).
- [28] Colin Camerer, *Behavioural Game Theory: Experiments in Strategic Interaction* (Princeton 2003).
- [29] Fang Zhong, D.J. Wu and Steven Kimbrough, 'Cooperative Agent Systems: Artificial Agents Play the Ultimatum Game', *Group Decision and Negotiation*, vol. 11, pp. 433-447 (2002).
- [30] Steven de Jong, Karl Tuyls, Katja Verbeeck and Nico Roos, 'Priority Awareness: Towards a Computational Model of Human Fairness for Multi-agent Systems', in Karl Tuyls *et al* (eds.), *Adaptive Agents and MAS III*, LNAI 4865, pp. 117-128 (2008).
- [31] Ross Hammond and Robert Axelrod, 'The Evolution of Ethnocentrism', *Journal of Conflict Resolution*, vol. 50, no. 6, pp. 926-936 (2006).

Appendix: Multi-agent system code listing

```
'
' VBA program to simulate multiple autonomous agents operating in a common environment
' Roger Kingdon 2008
'
' Global declarations
'
Option Explicit
' Subtype maxima (required by annoying VB limitation)
'
Public Const intMAXMEM As Integer = 20
'
' Type definitions - subtypes first
'
Public Type MEMORY_TYPE
    intLocation As Integer
End Type
'
Public Type AGENT_TYPE
    intfV As Integer
    dblfR As Double
    dblfE As Double
    dblfP As Double
    intfM As Integer
    intLocation As Integer
    dblEnergy As Double
    udtMemory(intMAXMEM) As MEMORY_TYPE
End Type
'
Public Type SPACE_TYPE
    blnParcel As Boolean
End Type
'
' Data structure definitions
'
Public udtAgent() As AGENT_TYPE
Public udtSpace() As SPACE_TYPE
'
' Data structure sizes
'
Public intNumAgent As Integer
'
' Control parameters
'
Public intNumSpace As Integer
Public intNumTime As Integer
Public dblEnergyUseMax As Double
Public intSpeedMax As Integer
Public intVisionMax As Integer
Public dblRecognitionMax As Double
Public dblEfficiencyMax As Double
Public dblPrecedenceFraction1 As Double
Public dblPrecedenceFraction2 As Double
Public dblPrecedenceFraction3 As Double
Public intMemoryMax As Integer
Public intParcelSpaceFreq As Integer
Public intParcelTimeFreq As Integer
Public intParcelRange As Integer
Public dblParcelSize As Double
'
' End of global declarations
'
```



```

Sub Forage()
'
' Main program, initiated by clicking Go button on Overview worksheet
'
Dim intTime As Integer
'
' Input data from worksheets
'
Call ReadControls
Call ReadAgents
'
' Run program for the specified number of timesteps
'
For intTime = 1 To intNumTime
    Call Provide(intTime)
    Call Survey
    Call Travel
    Call Consume
Next intTime
'
' Output results
'
Call WriteResults
MsgBox ("End of program")
'
' End of Forage
'
End Sub

Sub ReadControls()
'
' Read data from worksheet Controls
'
With Excel.Sheets("Controls")
    intNumSpace = .Cells(2, 2)
    intNumTime = .Cells(3, 2)
    dblEnergyUseMax = .Cells(4, 2)
    intSpeedMax = .Cells(5, 2)
    intVisionMax = .Cells(6, 2)
    dblRecognitionMax = .Cells(7, 2)
    dblEfficiencyMax = .Cells(8, 2)
    dblPrecedenceFraction1 = .Cells(9, 2)
    dblPrecedenceFraction2 = .Cells(10, 2)
    dblPrecedenceFraction3 = .Cells(11, 2)
    intMemoryMax = .Cells(12, 2)
    intParcelSpaceFreq = .Cells(13, 2)
    intParcelTimeFreq = .Cells(14, 2)
    intParcelRange = .Cells(15, 2)
    dblParcelSize = .Cells(16, 2)
End With
'
' Set up memory space corresponding to 1D discretised physical space
'
ReDim udtSpace(intNumSpace)
'
' End of ReadControls
'
End Sub

Sub ReadAgents()
'
' Read data from worksheet Agents
'
Dim intRow As Integer, intRec As Integer, intRow1 As Integer, intM As Integer
'
intRow1 = 4
With Excel.Sheets("Agents")
    intRow = intRow1
    Do Until .Cells(intRow, 1) Like ""
        intRow = intRow + 1
    Loop

```

```

intNumAgent = intRow - intRow1
'
If intNumAgent > 0 Then
  MsgBox (CStr(intNumAgent) + " records in spreadsheet Agents")
Else
  MsgBox ("Error: Empty spreadsheet: Agents")
  End
End If
'
If intNumAgent > 0 Then
  ReDim udtAgent(intNumAgent)
  For intRec = 1 To intNumAgent
    intRow = intRec + intRow1 - 1
    udtAgent(intRec).intfV = CInt(intVisionMax * .Cells(intRow, 2))
    udtAgent(intRec).dblfr = .Cells(intRow, 3)
    udtAgent(intRec).dblfe = .Cells(intRow, 4)
    udtAgent(intRec).dblfp = .Cells(intRow, 5)
    udtAgent(intRec).intfM = CInt(intMemoryMax * .Cells(intRow, 6))
    udtAgent(intRec).intLocation = .Cells(intRow, 7)
    udtAgent(intRec).dblEnergy = .Cells(intRow, 8)
  Next intRec
End If
End With
'
Ensure that the memory of each agent is empty
'
For intRec = 1 To intNumAgent
  With udtAgent(intRec)
    For intM = 1 To .intfM
      .udtMemory(intM).intLocation = 0
    Next intM
  End With
Next intRec
'
End of ReadAgents
'
End Sub

Sub Provide(intTime As Integer)
'
' Distribute parcels into the space
'
Dim intSpace As Integer, intLoc As Integer
'
Empty the space of old parcels
'
For intSpace = 1 To intNumSpace
  udtSpace(intSpace).blnParcel = False
Next intSpace
'
Depending on these characteristics, populate the space with new parcels
'
If intTime Mod intParcelTimeFreq = 0 Then
  For intSpace = 1 To intNumSpace
    If intSpace Mod intParcelSpaceFreq = 0 Then
      intLoc = intSpace + intParcelRange * (Rnd - 0.5)
      intLoc = iMin(intNumSpace, iMax(1, intLoc))
      udtSpace(intLoc).blnParcel = True
    End If
  Next intSpace
End If
'
End of Provide
'
End Sub

```

```

Sub Survey()
'
' Agents look around for a parcel
' If one is found in a new location then this is recorded in available memory space
'
Dim intAgentID As Integer, intSpace As Integer, intM As Integer
Dim int1 As Integer, int2 As Integer, intStep As Integer
'
intStep = 1
If Rnd > 0.5 Then
    intStep = -1
End If
'
For intAgentID = 1 To intNumAgent
    With udtAgent(intAgentID)
        int1 = .intLocation - .intfV * intStep
        int1 = iMin(intNumSpace, iMax(1, int1))
        int2 = .intLocation + .intfV * intStep
        int2 = iMin(intNumSpace, iMax(1, int2))
        If intStep = 1 And int1 > int2 Then
            Stop
        ElseIf intStep = -1 And int1 < int2 Then
            Stop
        End If
        For intSpace = int1 To int2 Step intStep
            If udtSpace(intSpace).blnParcel Then
                For intM = 1 To .intfM
                    If intSpace = .udtMemory(intM).intLocation Then
                        GoTo NextParcel
                    End If
                Next intM
                For intM = .intfM To 2 Step -1
                    .udtMemory(intM).intLocation = .udtMemory(intM - 1).intLocation
                Next intM
                .udtMemory(1).intLocation = intSpace
                GoTo NextAgent
            End If
        NextParcel:
        Next intSpace
    NextAgent:
    End With
Next intAgentID
'
' End of Survey
'
End Sub

Sub Travel()
'
' Agents use energy to move towards where there are (or were) parcels
' If on arrival there is no parcel, try the next remembered location, and so on
' Algorithm assumes unit timestep, i.e. speed = distance
' and energy = work = force (assumed constant) * distance, i.e. energy = distance
'
Dim intAgentID As Integer, intM As Integer
Dim intParcelLoc As Integer, intDistance As Integer, intSpeed As Integer
Dim dblEnergy0 As Double, dblEnergyUsed As Double, dblEnergy As Double
'
For intAgentID = 1 To intNumAgent
    With udtAgent(intAgentID)
        dblEnergy0 = .dblEnergy
        If dblEnergy0 < 1# Then
            GoTo NextAgent
        End If
        dblEnergyUsed = 0#
        For intM = 1 To .intfM
            intParcelLoc = .udtMemory(intM).intLocation
            If intParcelLoc > 0 Then
                intDistance = Abs(.intLocation - intParcelLoc)
                intSpeed = iMin(intDistance, intSpeedMax)
                dblEnergy = dMin(CDbl(intSpeed), (.dblEnergy * dblEnergyUseMax))
            End If
        Next intM
    End With
Next intAgentID
'
' End of Travel
'
End Sub

```

```

        intSpeed = CInt(dblEnergy)
        intDistance = intSpeed
        If intParcelLoc < .intLocation Then
            intDistance = intDistance * -1
        End If
        dblEnergyUsed = dblEnergyUsed + dblEnergy
        .dblEnergy = dMax(0#, (.dblEnergy - dblEnergy))
        .intLocation = .intLocation + intDistance
        .intLocation = iMin(intNumSpace, iMax(1, .intLocation))
        If dblEnergyUsed > (dblEnergy0 * dblEnergyUseMax) Then
            GoTo NextAgent
        End If
        If udtSpace(.intLocation).blnParcel Then
            GoTo NextAgent
        End If
    End If
Next intM
'
' Still no luck: just wander around for a bit
'
        intSpeed = intSpeedMax
        dblEnergy = dMin(CDbl(intSpeed), (.dblEnergy * dblEnergyUseMax))
        intSpeed = CInt(dblEnergy)
        intDistance = intSpeed
        If Rnd < 0.5 Then
            intDistance = intDistance * -1
        End If
        .dblEnergy = dMax(0#, (.dblEnergy - dblEnergy))
        .intLocation = .intLocation + intDistance
        .intLocation = iMin(intNumSpace, iMax(1, .intLocation))
NextAgent:
    End With
Next intAgentID
'
' End of Travel
'
End Sub

Sub Consume()
'
' Agents consume contents of parcels, adjusting energy accordingly
'
    Dim intAgentID As Integer, intLoc As Integer
    Dim dblRecognise As Double, dblEfficiency As Double
    Dim intPeck As Integer, intCompetitor As Integer, dblPeck As Double
'
    For intAgentID = 1 To intNumAgent
        With udtAgent(intAgentID)
            intLoc = .intLocation
            If udtSpace(intLoc).blnParcel Then
'
' There is a parcel, so try to distinguish food from poison
'
                dblRecognise = -1#
                If .dblFR * dblRecognitionMax > Rnd Then
                    dblRecognise = 1#
                End If
'
' Efficiency factor
'
                dblEfficiency = dMin(1#, (.dblFE * dblEfficiencyMax))
'
' Determine position in pecking order and consequent fraction of parcel available
'
                intPeck = 1
                For intCompetitor = 1 To intNumAgent
                    If udtAgent(intCompetitor).intLocation = intLoc Then
                        If intCompetitor <> intAgentID Then
                            If udtAgent(intCompetitor).dblFP >= .dblFP Then
                                intPeck = intPeck + 1
                            End If
                        End If
                    End If
                Next intCompetitor
            End With
        End For
    End Sub

```

```

        End If
    End If
Next intCompetitor
,
    If intPeck = 1 Then
        dblPeck = dblPrecedenceFraction1
    ElseIf intPeck = 2 Then
        dblPeck = dblPrecedenceFraction2
    ElseIf intPeck = 3 Then
        dblPeck = dblPrecedenceFraction3
    Else
        dblPeck = 0#
    End If
,
' Apply food/poison recognition, efficiency and pecking order
,
        .dblEnergy = .dblEnergy + (dblParcelSize*dblRecognise*dblEfficiency*dblPeck)
        .dblEnergy = dMax(.dblEnergy, 0#)
    End If
    End With
Next intAgentID
,
' End of Consume
,
End Sub

Sub WriteResults()
,
' Write results to worksheet Results
,
    Dim intRow As Integer, intCol As Integer
    Dim intRec As Integer, intRow1 As Integer
,
    Excel.Sheets("Results").Activate
    Cells.Select
    Selection.ClearContents
    Cells(1, 1).Select
,
' Echo Controls definition for this run
,
    intRow1 = 0
    Excel.Cells(intRow1 + 1, 1) = "Global control parameters"
    Excel.Cells(intRow1 + 2, 1) = "NumSpace"
    Excel.Cells(intRow1 + 3, 1) = "NumTime"
    Excel.Cells(intRow1 + 4, 1) = "EnergyUseMax"
    Excel.Cells(intRow1 + 5, 1) = "SpeedMax"
    Excel.Cells(intRow1 + 6, 1) = "VisionMax"
    Excel.Cells(intRow1 + 7, 1) = "RecognitionMax"
    Excel.Cells(intRow1 + 8, 1) = "EfficiencyMax"
    Excel.Cells(intRow1 + 9, 1) = "PrecedenceFraction1"
    Excel.Cells(intRow1 + 10, 1) = "PrecedenceFraction2"
    Excel.Cells(intRow1 + 11, 1) = "PrecedenceFraction3"
    Excel.Cells(intRow1 + 12, 1) = "MemoryMax"
    Excel.Cells(intRow1 + 13, 1) = "ParcelSpaceFreq"
    Excel.Cells(intRow1 + 14, 1) = "ParcelTimeFreq"
    Excel.Cells(intRow1 + 15, 1) = "ParcelRange"
    Excel.Cells(intRow1 + 16, 1) = "ParcelSize"
,
    Excel.Cells(intRow1 + 1, 2) = "Value"
    Excel.Cells(intRow1 + 2, 2) = intNumSpace
    Excel.Cells(intRow1 + 3, 2) = intNumTime
    Excel.Cells(intRow1 + 4, 2) = dblEnergyUseMax
    Excel.Cells(intRow1 + 5, 2) = intSpeedMax
    Excel.Cells(intRow1 + 6, 2) = intVisionMax
    Excel.Cells(intRow1 + 7, 2) = dblRecognitionMax
    Excel.Cells(intRow1 + 8, 2) = dblEfficiencyMax
    Excel.Cells(intRow1 + 9, 2) = dblPrecedenceFraction1
    Excel.Cells(intRow1 + 10, 2) = dblPrecedenceFraction2
    Excel.Cells(intRow1 + 11, 2) = dblPrecedenceFraction3
    Excel.Cells(intRow1 + 12, 2) = intMemoryMax
    Excel.Cells(intRow1 + 13, 2) = intParcelSpaceFreq

```

```

Excel.Cells(intRow1 + 14, 2) = intParcelTimeFreq
Excel.Cells(intRow1 + 15, 2) = intParcelRange
Excel.Cells(intRow1 + 16, 2) = dblParcelSize
intRow = intRow1 + 16
,
' Echo Agents definition for this run
,
intRow1 = intRow + 2
Excel.Cells(intRow1, 1) = "AgentID"
Excel.Cells(intRow1, 2) = "Vision"
Excel.Cells(intRow1, 3) = "Recognition"
Excel.Cells(intRow1, 4) = "Efficiency"
Excel.Cells(intRow1, 5) = "Precedence"
Excel.Cells(intRow1, 6) = "Memory"
Excel.Cells(intRow1, 7) = "Location"
Excel.Cells(intRow1, 8) = "Energy"
,
For intRec = 1 To intNumAgent
    intRow = intRow1 + intRec
    With udtAgent(intRec)
        Excel.Cells(intRow, 1) = intRec
        Excel.Cells(intRow, 2) = CDb1(.intfV) / CDb1(intVisionMax)
        Excel.Cells(intRow, 3) = .dblFR
        Excel.Cells(intRow, 4) = .dblFE
        Excel.Cells(intRow, 5) = .dblFP
        Excel.Cells(intRow, 6) = CDb1(.intfM) / CDb1(intMemoryMax)
        Excel.Cells(intRow, 7) = .intLocation
        Excel.Cells(intRow, 8) = .dblEnergy
    End With
Next intRec
,
' Output final memory for each agent
,
intRow1 = intRow + 2
Excel.Cells(intRow1, 1) = "Final memory of each agent"
For intRec = 1 To intNumAgent
    intRow = intRow1 + intRec
    Excel.Cells(intRow, 1) = intRec
    With udtAgent(intRec)
        For intCol = 1 To .intfM
            Excel.Cells(intRow, intCol + 1) = .udtMemory(intCol).intLocation
        Next intCol
    End With
Next intRec
,
' Output final distribution of parcels
,
intRow1 = intRow + 2
Excel.Cells(intRow1, 1) = "Parcel distribution"
For intRec = 1 To intNumSpace
    intRow = intRow1 + intRec
    Excel.Cells(intRow, 1) = intRec
    Excel.Cells(intRow, 2) = udtSpace(intRec).blnParcel
Next intRec
,
' End of WriteResults
,
End Sub

Function iMax(int1 As Integer, int2 As Integer)
,
' Returns the larger of two integers
,
    If int1 > int2 Then
        iMax = int1
    Else
        iMax = int2
    End If
,
' End of iMax
,

```

```

End Function

Function iMin(int1 As Integer, int2 As Integer)
'
' Returns the smaller of two integers
'
  If int1 < int2 Then
    iMin = int1
  Else
    iMin = int2
  End If
'
' End of iMin
'
End Function

Function dMax(dbl1 As Double, dbl2 As Double)
'
' Returns the larger of two doubles
'
  If dbl1 > dbl2 Then
    dMax = dbl1
  Else
    dMax = dbl2
  End If
'
' End of dMax
'
End Function

Function dMin(dbl1 As Double, dbl2 As Double)
'
' Returns the smaller of two doubles
'
  If dbl1 < dbl2 Then
    dMin = dbl1
  Else
    dMin = dbl2
  End If
'
' End of dMin
'
End Function

```