

The University of Oxford
MSc Project

Ethics and Artificial Life

Will Harwood

Supervisors: L.L. Floridi and J.W. Sanders

Introduction

Artificial Life is both a disparate collection of programs and models – from Conway’s Game of Life [1] to Thomas Ray’s Tierra [2] – and a particular way of approaching a system, be it the study of the universal constructor¹ or Evolution. It is an formula of simple agents acting locally to uncomplicated rules; a formula this project believes is appropriate for the study is a very different field: Ethics.

By Ethics, we mean not the study of actual human interaction (this is not a sociological dissertation), but of that body of thought which attempts to deal with and formalise good and evil, right and wrong. There have been previous attempts to mathematise Ethics: Leibniz (1646 – 1716) hoped to create a *lingua charactica* in which the settling of any problem becomes a matter of grinding away a few mechanistic calculations; George Birkhoff (1884 – 1944) developed a mathematical theory of aesthetics [Birkhoff 1933] and of ethics [Birkhoff 1968]; and so forth. However, none had the advantage of that universal laboratory: the computer. We may *run* our models, see them in action. In **Chapter Four** (*Notes on early simulations*) we will set up a situation of, say, one Hedonist operating in a group of Eudemonists, and produce graphs of the agents’ *happiness* over time, and the data of their interactions.

There is no pretence that the work here presented represents any advance in sociological understanding, or in the theory underlying Artificial Life. Rather, we present a new perspective on those well-known formalised systems of Ethics: literally, one can see them in a new way (one can study systems of *true* Eudemonists or Authoritarians, say, where arguably no such people exist).

In **Chapter One** we try to place the field of Artificial Life in the context of the development of science, and give (perfunctory) descriptions of its more important constituents. Following, we cut to Ethics, **Chapter Two**, introduce a few of the more

¹ From the work of John von Neumann; the ultimate idea that of self-copying machines able to colonise the galaxy.

famous examples, and start the process of abstraction that terminates in **Chapter Three** and the model's development. Examples ensue in the afore mentioned **Chapter Four**, and we end with the **Conclusions**, or, *Arguments against the approach*.

Chapter One

Models and abstraction

Artificial Life, in the development of science

Science may be characterised as a process of understanding through abstraction, of teasing out some thread from a chaos of information by ignoring what we decide is extraneous; that decision arbitrary, informed by history, the intellectual fashion and, of course, utility. There are no *true* models in science, as science does not deal in truths but that pending proof of falsehood. A model is a skin over reality: Newton's Universe a fine enough fit for rudimentary purposes, Einstein's distorted closer in relation to those data we have and can expect, yet rendered in its greater intricacy often inappropriate (one does not use a medical laser to light a campfire). No longer overwhelmed, we may still see the shape of things.

Consider Euler's solution to the famous Königsberg² 7-Bridge Problem: in demonstrating (in proving the impossibility of traversing all of the bridges without a repetition) the irrelevance of both bridges and the number seven, in abstracting the physical to a class of graph – in (so to speak) revealing the tree-obscured wood – topology was born, and mathematics and science advanced.

If one assumes that the universe is (how to say?) relatively uncapricious (and perhaps all it really requires is to take as one's foundation [a generalisation of Newton's observations] that a stationary object will, probably, remain stationary; and a moving object will, if it has been moving up until now, continue doing so; and that neither will cease to exist), then there need be no pretence that a model – say, Hooke's Law – is *true*, is what is *really* going on (as we might say that in the movements of chess pieces, what is *really* going on is a game of chess): it is, merely, a useful way of organising data. No material truly obeys Hooke's Law, but many approximate it, and

² Kaliningrad.

since for the most part approximations are all we need, the utility, the value of Hooke's Law is assured. There is no correct abstraction, only manifestly incorrect ones; and no true model, only those in which we find utility.

Artificial Life (AL) is to biology and ecology what Artificial Intelligence is to psychology; it is bottom-up where AI is top-down, a holistic (in a weak sense) counterpart to its older sibling, that weak sense being an implicit recognition that our goal (whatever we may decide that is) exists at a higher level than those agents whose sum behaviour (we hope) brings about that goal. Briefly, we might say that where AI starts with an Object and endeavours to fathom its parts, AL begins with parts, lots of them, and hopes to bring about the Object, and it is in this reversal we see AL's great strength: its parsimony. Simple, usually homogeneous agents can produce as a property of their society behaviour that we thinking beings consider intelligent and (clearly) the work of some designer, somewhere. Consider that favourite the ant colony: how intelligent need the Queen be? Must there be architect-ants with blueprints in their heads? AL's answers are: an absolute minimum, and no, of course not .

Important and connected ideas here are hierarchies and levels of abstraction. A particularly instructive example is Craig Reynolds' Boids program [3], which produces convincingly life-like flocking behaviour without any explicit instruction "to flock" (or *shoal*, or – removing a dimension – *herd*). At the programmed level, each Boid obeys three principles: separation, alignment and cohesion, but at the next higher level of the hierarchy, abstracting away individual Boids to leave their sum instead as subject, we have a flock that may split around obstacles and re-coalesce in a fluid and realistic manner that has proven prohibitively difficult to replicate by hand – by design, by top-down manipulation – and can do it with huge numbers of agents in real time. Order emerges from chaos.

AL encompasses models of Darwinian natural selection (Tom Ray's Tierra³ [2], [Boden, 1996]); the evolution of co-operation (The Iterated Prisoners' Dilemma,

³ Tierra's life-forms are blocks of self-copying code in an environment or soup of 30,000 bytes. After seeding the soup with a single organism of size 80 [instructions – most of which is padding], this 80aaa

[Grim 1998]) and of evolutionarily stable strategies in general ([Dawkins 1976], [Maynard Smith, 1974]). It is the modelling of societies of agents and, in that, what more appropriate subject than those most ancient of sociological theories: Ethics.

proliferates and quickly fills all available space. The process of copying is however imperfect, and that (together with death for the least successful reproducers) is sufficient for the evolution of such wonders as parasites (organisms unable to copy themselves who hijack the copy cycle of others), and even hyper-parasites. By subtly altering a few operational parameters, one can observe the emergence of monsters orders of magnitude larger than the first ancestor, or parasites a sixth the size.

Chapter 2

Ethics

There is an idea of good and evil...

...impressions, feelings of good and bad, of doing right or wrong against another person, against other people. There are social rules and conventions; there are systems in which human interaction occurs, is regulated by. One could assert that from the fear of death are built the great edifices of Theology, and upon those impressions we call “conscience” is founded Ethics: partial solutions of the dilemmas of how one ought act; systematisations of the mechanisms of reciprocation, kinship and the like. However, to hold with the above would be to contradict a goodly percentage of Ethics itself, so instead we hold to no more than: Ethics is the body of knowledge that variously commands, persuades or informs how people *should* act, and why. It is not the more modern sociology, the study of how people *do* act, but a – many – coherent ways of thinking about one’s actions, of revealing/creating the/an intellectual basis for *what is right*.

This chapter’s purpose is to introduce a few of the more prominent Ethical theories in [a hazy approximation of] a historical context; tease out the common threads and make explicit those fundamental differences; and, in a process of abstraction, render these works of prose into a form amenable to mathematical description: reduce them – indeed – to caricatures.

2.1 “Is it good because it is desired by the gods, or is it desired by the gods because it is good?”

A ubiquitous quote of Socrates, and for good reason, for it encapsulates the Authoritarian position and the disputes therein. Implicit is the assumption that ‘good’ is a quantity that exists (somehow) outside of humanity. Like the Platonic ideal of a

triangle to which all triangles in Nature are mere shadows, *good* is a part of the universe towards which our actions aspire, and on that, upon the action, does morality reside.

Belief in the omniscient, omnipresent God of monotheisms largely ‘unasks’ the question, since by definition nothing can precede the creator of everything, and the paraphrase – of whether God was compelled to create the universe in a certain way – is to contradict His omnipotence. We may therefore solve the dispute in a purely definitional, syntactic manner: no new knowledge is required, and in that sense I use the term ‘unasked’: meaning that the question is shown to be not a real question at all.

But, in the pantheistic traditions of Hellenistic Greece, with gods that are human desire given free reign and projected vast onto the clouds and sky, it is very real. The gods may desire their worshippers to do good, to act to a certain code, but they are themselves fantastically amoral, utterly hedonistic – and here we arrive at the second way to ‘unask’ Socrates’ question: rather than denying that ‘good’ and God are in any sense separable, deny the whole idea of the extra-human existence of ‘good’, deny the implicit assumption.

2.2 Hedonism

We readily say, ‘this feels good’, ‘that feels bad.’ Pleasure is, by definition, that which feels *good*, so, why invent another, greater conception of ‘good’? Why insist that ‘good’ is anything other than the sensual: good is neither more nor less than pleasure. It is, in general, satisfying – pleasurable – to help others, so one can say that *in general* it is good to help one’s fellow man, but only as a shorthand, a piece of empiricism. Solipsistic, no doubt, but are not one’s own sensations are the only things that cannot be doubted?

It was to this philosophy, this anti-ethics, that the Cyrenaics adhered, the followers of the philosopher Aristippus of Cyrene in the fourth to third centuries BCE, to whom the only virtue was the capacity for pleasure. Not that this implies a *duty* to experience pleasure, as those under Authoritarianism *ought* to do good, because the hedonistic

perspective denies duty; rather, there is no reason *not* to indulge one's vices. There should be no guilt. If this is an Ethics (in the sense of our introduction to **Chapter 2**), it supposes that people are motivated by their desires: otherwise it is incomplete, for it offers nothing to inform decisions of a purely rational nature – an assumption made explicit something short of two millennia later by Hume, and by Hobbes and his (approximation) of people as agents pursuing 'commodious living.'

Of the particulars of greater and lesser pleasures, the Cyrenaics differed. Aristippus himself held that all pleasures are equal, since there cannot exist an objective standard of pleasures as the only pleasure we can know is our own. At a different extreme was Hegesias, the Death Persuader, whose teaching coincided with the logical conclusion of another, later philosophy, owing to that philosopher most closely associated with Hedonism.

Epicurus' was a different idea of hedonism, one centred on pain. His was an ascetic hedonism: he taught that true happiness, the greatest good, is a life without pain of the body or mind, and the way to reach such a state is pleasure and moderation in all things. Good-as-pleasure is not sustainable (modulo *nothing* is sustainable) – a debauch lasts as long as one's purse or liver, and cannot end well. Moreover, drunken pleasure is overridden by the pain of the morning after. As a basis for Ethics then, pain is evil. *The hungry should not dream of gluttony but of an end to hunger.* Hurting others is evil because it causes pain, and so on. Killing someone causes anguish (mental pain) to their family (though, there is nothing wrong in painlessly killing a person without family or friends, other, I suppose, than we can never be certain that those criteria have been met). Epicurus' life is almost monastic, minus God.

To his critics (among them Cicero in his *De Finibus* – interestingly, Epicurus was *accused* of hedonism), Epicurus' philosophy is brought to absurdity by its logical conclusion: the greatest happiness is death. This is not a contradiction. We will not venture into Eastern philosophy (because our purpose is to give a fresh perspective on what is already well understood), but this total absence, this oblivion as best-state is familiar to a shallow appreciation of Buddhist thought. (And one could argue that – as eternities go – as an inevitability it is not so onerous.)

Epicureanism in the specific, Hedonism in general, eschews metaphysical interpretations of morality. It is an egoist theory, a doctrine of the individual, and a minimal one at that: minimal abstraction, a minimal idea of society – anything stronger is a sort of Pascal’s Wager. However, in defining happiness as *the* foundation of ethics, it is an easy leap for the social animal to make to shift from the knowable happiness of the individual to ‘happiness’ in the abstract. To value the happiness of one’s peers because happiness itself *ought to* be increased. This leap brings us to Eudemonism, to Utilitarianism.

2.3 The greatest happiness of the greatest number⁴

Aristotle’s state of a life active and governed by reason is eudæmonia, a perfect condition reached by steering a well-thought out path between behavioural extremes, being neither gluttonous nor fasting, etc⁵. His Virtue Ethics we will not cover in depth, for reasons of time and space rather than inappropriateness (pace Stoicism – see 2.5). From the same Greek root – eu, well; and daimōn, the spirit – comes Eudemonism, a class of theories for which happiness is the test of rectitude. For our purposes, we begin by considering Eudemonism to be synonymous with a Consequentialist theory which in its modern form owes to Jeremy Bentham⁶ (1748 – 1832): Utilitarianism, a philosophy of the greatest-good that, after Bentham’s death and championed by his successor, John Stuart Mill⁷, had and has a profound political influence (see, e.g. [Bentham 1789, 1830-41]).

It was Benthamite ideas that led in the Lunacy Act of 1845 to the legal perception of insanity shifting from a *moral* to *medical* condition – in our perspective, from the medieval to the modern conception. The Workhouse was a Utilitarian cause (from the

⁴ From Joseph Priestly's *Treatise on Government*.

⁵ As Æschylus’ (circa 525 – 456 BCE) Furies put it:

The golden mean is God's delight:
Extremes are hateful in His sight.
Hold by the mean, and glorify
Nor anarchy nor slavery.

(<http://www.nd.edu/Departments/Maritain/etext/moral105.htm>)

⁶ See, e.g. [Dimwiddy 1989] Bentham famously resides in a glass-fronted case in UCL (though not his head, allegedly following a medical-student kick about).

⁷ [Mill 1871]

New Poor Law of 1834), against which Dickens wrote *Oliver Twist*. There were no *Rights of Man*⁸.

But this is the point: Utilitarianism is an ethics of societies, a theory of legislation rather than individual action. Bentham's people are hedonists operating in a eudemonically constrained democracy of punishment and reward. (Compare with Hobbes and his hedonists 'merely' free from the fear of a violent death – free to die with their boots off.) In that it is inappropriate for the purposes of Artificial Life, at least in the simple model we will come to construct. Truly, we should design a model of sufficient richness that egoist agents organise of themselves a governance. This is beyond the scope of our dissertation.

So we draw our definition of Eudemonism down one level: while a Hedonistic agent acts for the for its own good, the Eudemonist is motivated by the good-of-all.

Here we find a clear distinction between Authoritarian philosophies (and other deontological theories – see the next section), and those based upon happiness or pleasure. For the latter, an action is always a means to an end: the act itself has no implicit moral value because the same act performed in different circumstances will produce different results⁹. To be said to knowably do good or evil, the former requires a memory; the latter, an *imagination*¹⁰, for it is the consequences that matter.

A telling (and concurrent) analogy may be drawn with the opposing Lamarckian and Darwinian models of evolution. To Lamarck, evolution was a matter of the Will: the giraffe *stretches* his neck to reach the choicest leaves; its offspring's necks may stretch that little bit further. The differences between progeny and progenitor are,

⁸ "Nonsense on stilts" to Bentham.

⁹ Although, one could widen the taxonomy of *action* to include those circumstances, and their setting, and *their* setting until, ultimately, "an action" is a snapshot of the universe, and cannot be repeated. If the universe were deterministic, *then* actions could [be said to] have moral value. It is a trick, though, legerdemain.

¹⁰ A lack of universal moral laws is troubling: what about torture, for instance. A theory that admits that *torture* could be considered *good* must surely be flawed. This is not necessarily a contradiction though. Consider: to lie we could reasonably assert is usually evil (not good), but not universally so (lying to protect another could be considered a moral act). It is trickier to imagine situations wherein *to steal* is *good* act – harder, though not impossible: do not steal (we could argue with consistency) is simply a better rule-of-thumb. Now, torture and murder lie at one extreme of this continuum: in no conceivable sense moral. One can have 'universal' laws of behaviour without introducing metaphysics. As a

then, *improvements*. To the Darwinist, such evolutionary differentials are blind mutations, their good (or, rather, their utility) determined in the process of living (see, e.g., [Kolakowski 1972]). In the dichotomy of pre- or post-determined good, Lamarckians and Authoritarians fall to the pre-, while Consequentialists and Darwinians go to the latter.

So, if “good” exists only in retrospect, can a Consequentialist *decide* to perform a good act? Here comes the imagination: the strength of *good* and *evil* in decision making is a function of the predictability of the world. Reality is predictable: one can usually imagine the immediate consequences of an action – we do, after all, act for reasons other than just to have acted. The model we will come to develop in **Chapter 3** is also predictable up to a point; however, it is not deterministic, and repeat runs from the same initial conditions may not produce the same results (although general patterns do emerge). An issue this feeds into – one which does not arise from an Authoritarian, action-oriented philosophy – is, in a Consequentialist universe, how much responsibility does an agent bear for the consequences of his or her action?

How can we say that, for instance, James assisting Dave on Tuesday is responsible for the happiness of their town being, on Wednesday, point eight two. Perhaps sense can be made of the idea if we could calculate that, say, had James hindered Dave, the town’s happiness *would have been* point seven nine – the good of an action is a matter of its alternatives. But even granting the existence of a “happiness function” (integral to the *felicific calculus* of Bentham), is it possible to determine what the results of an alternative action would have been? (In removing the metaphysical basis of morality, it seems a different metaphysics is needed to save it as an objective concept.)

This another issue we will not venture much further into, but, one which modelling in the manner of our dissertation could serve to clarify (in a non-deterministic model, I would imagine talking of an action being *good* or *evil* in “the greatest possible number of worlds”), and not venture further because it is a distortion of Utilitarianism: it is to remodel Consequentialism into an action-oriented system like Authoritarianism, or, indeed, Kantian Deontology.

defence of Eudemonism this is disingenuous; that, however, is a different dissertation.

2.4 The Categorical imperative

Deontology, from the Greek *that which is binding*, means ethics as moral-duty, as rule-following creeds, and in that it encompasses the Authoritarian systems with which we began the chapter. For us it specifically refers to a distillate of Kant's (1724 – 1804) ethical teachings:

- (1) The Categorical Imperative: *people are ends*; and
- (2) Universality: “Act only on that maxim through which you can at the same time will that it should become a universal law.” (See [Kant 1948].)

The second dictate is Biblical, “Do unto others...”, but Kant's metaphysic is for Reason to fathom; his people act morally towards others not for the glory of God, not for their personal ends, but for the very fact of being moral. Others are not stepping-stone's to be trodden to some ultimate good: they *are* the ultimate good. In this, Deontologism differs from Hedonism and Eudemonism in being action-oriented, and diverges from Authoritarianism in the method by which those moral laws are discovered. (We might say, though have no right to, that it is Consequentialist in inception (*what* if everyone were to behave thus?), and Authoritarian in execution (that is the nature of the Law).)

In our model, we conflate all rule-following theories into the class Deontologism, since, in action, it makes no difference how our agents' Laws were arrived at, so long as they are immutable. One could construct a model of “true” Kantians, agents able to “reason” their own universalities. The possibilities and intricacies of defining and implementing such a model are endless, and beyond anything that approaches the mandate of this project: here is Ethics in operation.

2.5 In conclusion

To “be philosophical” means to be stoical, if not a Stoic, deriving from that school of the Hellenistic era reputedly founded by Zeno of Citium (in the fourth to third centuries BCE – see [Rist 1969]). Stoicism, unlike its contemporary Epicureanism, is not appropriate for inclusion to a model as simple as that presented here. While in its rival we find a caricature, hedonism, by which our agents are motivated, the same treatment to the Stoic school¹¹ gleans an ethics of inaction. Rather than tell us what we must do, as in Deontology, or ought not do, as in Authoritarianism, or should work towards, á la the eudemonistic theories, it teaches a passivity, and there is no motivation in *acceptance* of one’s fate. We will, in the following chapter, ask our theories to give value to our agents’ courses of action: courses that are, by definition, within their power. There is no place for Stoicism in the proceeding abstraction.

¹¹ A complete way of life, a metaphysical opponent to Hedonism’s materialism and gods too busy carousing to run the universe. For a concise treatment, see <http://plato.stanford.edu/entries/stoicism>.

Chapter 3

Thud

To an Artificial-Life informed model of Ethics

Our platform is AL; from it we inherit a certain schema in modelling (see **Chapter 1**), namely, that the model should consist of,

A society of homogenous agents
Operating under simple rules
Acting as individuals

Any apparent group-activity that comes about is to happen as a result of individual decisions (see especially the Boids for a very visual example of group activity built from individual decision making).

Without then much thought we arrive at a system of three parts. First, the agents, creatures of a notional existence that extends no farther than the indices of a few arrays; second, the actions, the behaviour we allow of our actors; and third, Ethics, the systems by which an agent decides to pick one act from the multitude.

There is no environment because we do not require it. Our agents need neither to range over space nor eat, drink, use: consume. The sharing of resources is perhaps a key ethical fact – one could conjecture that in a world of unlimited abundance, concepts of “good” and “evil” would not have arisen, that Ethics itself is no more than the problem of equitable division of insufficient resources. But that is to stray into the territory of economics; this project intends to make steps towards a new way of visualising and working with Ethical theories, which concern (in the main) the interaction of people. For our purposes, we no more need to place our agents in a world than give them hats and shoes: abstraction is all. Their world is subsumed by those properties by which we know them. The environment if any is the rest.

3.1 Agents

First, is a set Agents. By convention,

$$\text{card Agents} = N$$

and we refer to agents by index in $[0, N)$; for example, agent i acts on j , where $0 \leq i, j < N$ (and $i \neq j$). In practice, Agents (an abyss holding our ethically ideal agents in all their parts) is no more than $\{0, 1, \dots, N - 1\}$ – a distinction we could make more ontologically pleasing by use of an enumeration of Agents, but no clearer, and certainly no easier to work with.

The finitude of Agents is an executional necessity, but even in mathematical definition there are compelling reasons for not generalising to infinite sets (*pace* the floating-point approximation of real numbers), for disallowing the infinite. Agents that decide their actions by enumerating and evaluating the possibilities (see **3.3**) cannot be reconciled with a world in which such an enumeration could take forever, or if $\text{card Agents} > \aleph_0$, (by definition) does not exist. Granting our agents such extraordinary powers is to compound the exercise's meaninglessness. (Things work in the infinite that simply do not in the finite world: pyramid schemes, for instance.)

Admitting an infinity (or, by the same token, very large number) of agents militates a different approach in modelling: it imposes the incorporation of *space*, so that agents tend to act in their own locality and need not consider acting against agents unrealistic distances away. The lack of specialisation assumes then small N ; in simulations (see **Chapter 4**), typically tens of agents are used.

Agents have one property: happiness, represented by a real number in the unit interval $[0, 1]$, chosen for its mathematical simplicity, and bounded (at all) because of the physiological assumption that, unlike wealth, there is only a certain range of happiness that one can possess, analogous to there being (say) only a certain extent of *red* discernible to the eye and associated sensory machinery, or of *salt* gleaned by the tongue. This is not a straightforward assumption and is, in *Arguments against the*

approach (the **Conclusion**), given a thorough treatment.

The comparison with wealth is important since “happiness” is not intended to be some composite measurement of money, health, stature and so on, but just *happiness* (care should be taken to avoid too strongly presupposing a Eudemonist standpoint). Much of our reasoning hinges upon this supposition. We are, in this simulation, looking at a slice of life – to say an agent is “doing extremely well” tells us nothing about his or her health, wealth...

The state of our system, on which our simulation operates and transforms turn-to-turn, is no more than the happiness of all agents concerned; that being, a function from Agents to the unit interval:

$$\text{State} = \text{Agents} \rightarrow [0, 1]$$

The current state we will by convention call h ,

$$h: \text{State}$$

and the next state we shall habitually refer to as h' , the result of our agents' actions on h .

3.2 Actions

An action is what an agent *does*, and does to another agent: the reagent. One could conjure interest from a single action and the problem of against whom to act, but I think modern computers can admit of more than one possibility, so we have two: *assist* and *hinder*:

$$\text{Actions} = \{\text{assist}, \text{hinder}\}$$

We could overtly call these “evil” and “good”, but this would not work because for

most ethical theories (that I cover) an action in itself has no ethical value; rather, it is by the (speculated) consequences of an action that one determines its good or ill. To a deontologist it could (and is, in our implementation) be *evil* to hinder, to act selfishly, but a hedonist would consider such a judgement to be meaningless, since only if by hindering another he harms himself does *that* hindering acquire a value of *evil*. We cannot label our actions “good” and “evil” because an *action* has no such property – verbs do not have moral value.

If we can legitimately say that overall it is better to help than hinder, assist than attack, in no good conscience can these rules-of-thumb be generalised into Laws (with the proviso that that is exactly what some theories have done, rendering this passage somewhat less than Authoritative). But rather, we create actions with a real and intuitive meaning, chosen for their ethical flavour. An agent may either *assist* or *hinder* another agent, where assist and hinder correspond to co-operation and selfishness, operating mutually or at the expense of another. Do you: look after number one, or work together?

(The factors of health and wealth are irrelevant, excepting the assumption that our agents do not *die*.)

The grammar of action requires a subject and an object; we will refer to agent, action, reagent triples, or (actor, act, actee), for which the subject is not the object: an agent cannot act on itself. (We have no interest in systems for $N = 1$. Again, on a tangential point as regards our pov on Ethics, *could* The Only Person in the World do evil?)

An action is a binary operation on the State; specifically, on the happiness of its protagonists. Thus, having applied some action (i, act, j) to h to produce h', we insist that,

(i)

$$(k: \text{Agents} \mid k \neq i, j \Rightarrow h.k = h'.k)$$

We regard this approach as more fundamental; if desired, actions with more far-

reaching consequences can be constructed in terms of these.

3.2.1 Visualisation

We visualise an action (i, act, j) by means of two graphs plotting the happiness of i against j , and showing (in colour) the advantage or disadvantage to i and, in the second, the balance likewise for j , where red denotes disadvantage, blue advantage, and white: no change. **Fig 3.a** gives an example plot for an unsophisticated variant of *assistance*, showing graphically how it is to i 's advantage to help the needy (and i gains the most if he, too, is worse off), while j benefits at the prosperity of its helper:

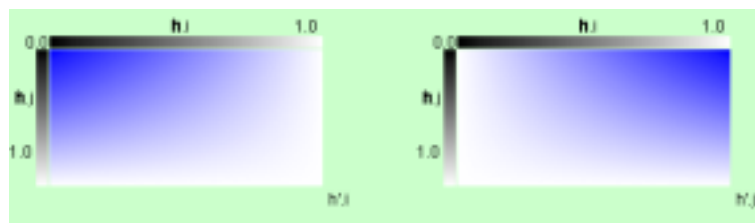


Fig 3.a

(For the rudimentary graph plotting program's Java source code, see **Appendix B**.)

3.2.2 Assist

Restating (i) in terms of a function from state, agent and reagent to (updated) state,

$$h' = \text{assist}(h, i, j),$$

the above graphs (**Fig 3.a**) describe an implementation of assist motivated by the ideas that i) more satisfaction is to be gained in helping the less rather than better off; and ii) a man gains more if assisted by the better rather than less well off. (Perhaps an inconsistency here in our meaning of "happiness" – see the **Conclusion**.), defined:

$$\text{assist}: \text{State} \times \text{Agents}^2 \rightarrow \text{State}$$

$$\text{assist}(h, i, j) = \begin{cases} h[i \mapsto h.i + (1 - h.i)(1 - h.j)], (j \mapsto h.j + h.i(1 - h.j)) & \leftarrow i \neq j \\ \perp & \leftarrow i = j \end{cases}$$

(So, we are decreasing the distance between $h.k$ and 1 – an elegant way of ensuring that the bounds $[0, 1]$ cannot be overstepped.) Or, in the Guarded Command Language, the assignment (explicitly for $i \neq j$)¹²,

$$h.i, h.j := h.i + (1 - h.i) \times (1 - h.j), h.j + h.i \times (1 - h.j)$$

An implementation of a subtler idea of assistance, in which the actor is penalised for “assisting” an agent who is better off than himself, is plotted in **Fig 3.b** (the i plot is blue above and red below the diagonal),

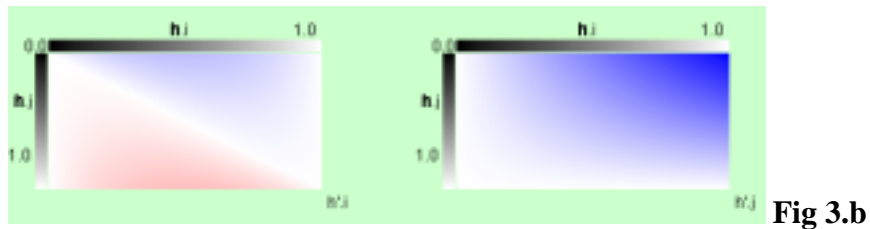


Fig 3.b

– a visualisation of the assignments,

```

{ i ≠ j }
if h.i < h.j    →    h.i, h.j := h.i × (1 + h.i - h.j), h.j + h.i × (1 - h.j)
[] h.i ≥ h.j    →    h.i, h.j := h.i + (1 - h.i) × (h.i - h.j), h.j + h.i × (1 - h.j)
fi

```

The above renditions of *assistance* stress the idea of *help*: i helps j in some fashion, and it is not unreasonable to suppose that there is greater satisfaction to be had in the act of helping the less fortunate than the more.

¹² Since using arrays can lead to a contradiction of the form $x, y := 0, 0; h.x, h.y := 1, 0$ (see [Kaldewaij 1990]), the parallel assignment should be read as an abbreviation of the type:

```

h.i, h.j := c(h.i, h.j), d(h.i, h.j)
≡
[[
    var hi : [0, 1];
    hi := h.i;
    h.i := c(h.i, h.j);
    h.j := d(hi, h.j);
]]

```

The general principle of assistance is that its effect is not directly detrimental to the reagent,

(ii)

$$h' = \text{assist}(h, i, j) \wedge i \neq j \Rightarrow h'.j \geq h.j$$

(It can be indirectly detrimental, over time, and in many cases is, e.g. **Fig 4.d**) There is an ambiguity here: we refer to the “actions” *assist* and *hinder* but rely on the fact that by “i assists j” we mean *some implementation* of assistance, one of many that we look at. Assistance is then a set of actions, of functions $\text{assist}_x: \text{State} \times \text{Agents}^2 \rightarrow \text{State}$ that meet the criterion of action (i) and of assistance (ii):

$$\begin{aligned} \text{All-Actions} = & (\text{act}: \text{State} \times \text{Agents}^2 \rightarrow \text{State} \mid h: \text{State} \wedge i, j: \text{Agents} \wedge i \neq j \bullet \\ & k: \text{Agents} \wedge k \neq i, j \Rightarrow h.k = \text{act}(h, i, j).k) \end{aligned}$$

$$\begin{aligned} \text{assist} = & (\text{assist}: \text{All-Actions} \mid h: \text{State} \wedge i, j: \text{Agents} \wedge i \neq j \bullet \\ & \text{assist}(h, i, j).j \geq h.j) \end{aligned}$$

Where it is ambiguous *which* assistance we mean, a subscript can be used (in the program it is `assist_one`, `assist_two` etc, see **Appendix A**). There is no technical reason to restrict the agents to one version of assistance; an easy extension of the work presented here would be establish richer worlds with any number of subtly distinct ways that an agent can do good or ill. In the simulations of **Chapter 4** we keep to a binary simplicity, because this is not an extension to anything.

3.2.3 Hinder

The second class of action, *hinder*, is specifically not to the object’s advantage. That is, our realisations of *hinder*,

$$\text{hinder}: \text{State} \times \text{Agents}^2 \rightarrow \text{State}$$

are such that,

$$h' = \text{hinder}(h, i, j) \wedge i \neq j \Rightarrow h'.j \leq h.j$$

That is, are members of the set:

$$\begin{aligned} hinder = & (\text{hinder: All-Actions} \mid h: \text{State} \wedge i, j: \text{Agents} \wedge i \neq j \bullet \\ & hinder(h, i, j).j \leq h.j) \end{aligned}$$

A few implementations are pictured below: the first (**Fig 3.c**) has the hindered's disadvantage (the right graph is in red, the left, blue) proportional only to the amount it has to lose, while the hinderer's advantage increases with the prosperity of its object – the circumstances of the attacker are irrelevant, all that matters is that we have been attacked.

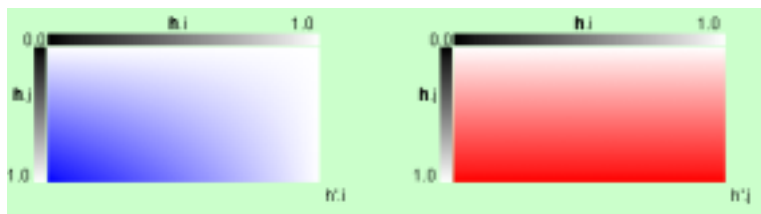


Fig 3.c

In **Fig 3.d**, our image of the agent is unchanged, but the reagent's loss is now proportional to the happiness of the agent. It is a subtler picture of *hinder* we try to intimate in this, not an outright attack but advantage taken nonetheless. We might try the example of, say, if another takes a seat on a bus that I consider mine, I can hardly feel aggrieved if that somebody is old and frail, but this does not work. Happiness does not equal vitality. Instead, we imagine a more general idea, less *i attacking j* than taking some opportunity, some pleasure otherwise open to *j*.

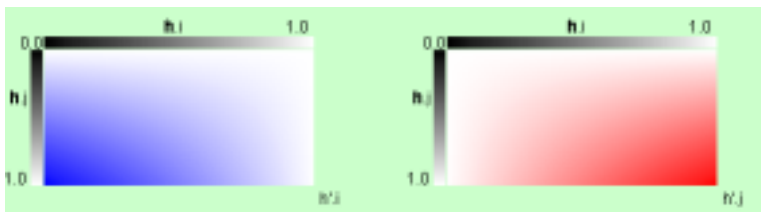


Fig 3.d

3.2.4 Further actions

Adding actions to the simulation is a matter of appending a name to the Actions set and providing one or more implementations – no other parts need modification since, but for the Deontological theories' edicts, individual actions are not referred to by name.

We could (but have not executed), do,

$$\text{Actions}' = \text{Actions} \cup \{\text{nothing}\}$$

a *skip* action defined (by its action on the state),

$$\text{nothing}: \text{State} \times \text{Agents}^2 \rightarrow \text{State}$$

$$\text{nothing}(h, i, j) = h$$

Or introduce a class of explicitly altruistic actions *altruism*, a subset of the *assists*, such that,

$$h' = \text{altruism}(h, i, j) \wedge i \neq j \Rightarrow h'.i \leq h.i \wedge h'.j \geq h.j$$

The possibilities are restricted by what we can find meaningful, by the need to find a counterpart in reality for our agents' behaviour (otherwise the abstraction is barren); that conduct given – in the manner of *h* – by a function: *d*.

3.2.5 *d*

The agents' intended actions are given by a function from Agents (actors) to Action–Agent pairs (act and actee), named *d* for decisions,

$$d: \text{Agents} \rightarrow \text{Actions} \times \text{Agents}$$

where $d.i = (\text{act}, j)$, under the condition that,

$$(i: \text{Agents} \mid \pi_2(d.i) = j \Rightarrow i \neq j)$$

The set of all action-combinations is then,

$$\text{All-Actions} = \{ d: \text{Agents} \rightarrow \text{Actions} \times \text{Agents} \mid (i: \text{Agents} \mid \pi_2(d.i) = j \Rightarrow i \neq j) \}$$

(π_n a projection function that returns the n th element of a tuple.) The problem of how to carry these actions out is addressed in **3.6**, while that of computing d , of assigning agents to actions, is treated in **3.5**, and uses the modes of Ethics.

3.3 Ethics

For each “turn” (to be defined in **3.4**), an agent has,

$$(N - 1) \times \text{card Actions}$$

combinations of action and reagent to choose from; a choice informed solely by the particular theory of Ethics it follows, and no other factor. As a model of *ethics* (intuitive ideas of right and wrong), ignoring pragmatic considerations represents a crippling dilution of the strength of our model; however, as stressed in **Chapter 1**, we deal with Ethics, and therein lies the reason for ignoring money, land, consumables... for operating with agents that are no more than “happiness.” Money is an obstruction, health obfuscation.

Agents, in fact, are slightly more than happiness: they have beliefs. They have Ethics, a theory apiece, chosen from the set Theories and given by a function named (again, by convention) t ,

$$t: \text{Agents} \rightarrow \text{Theories}$$

We make a distinction between act- and result-oriented theories (see **Chapter Two**) thus,

$$\text{Theories} = \text{Theories-Act} \cup \text{Theories-Result}$$

$$\text{Theories-Act} = \{\text{Hedonism, Eudemonism, Consequentialism, ...}\}$$

$$\text{Theories-Result} = \{\text{Deontology, Authoritarianism}\}$$

For example, a Hedonist theory would return to i that action and reagent which, on application to h , maximises i 's happiness: maximises $h'.i$. That is, given h and assuming that $act(h, actor, actee)$ is (in a (currently) weakly defined sense) the result of applying $(actor, act, actee)$ to h – $actor$ acting on $actee$ – the act and reagent determined for i will be a member of the set,

$$\{(i, act, j) : \text{Agents} \times \text{Actions} \times \text{Agents} \mid i \neq j \wedge ((i, act', j') : \text{Agents} \times \text{Actions} \times \text{Agents} \mid i \neq j' \bullet act(h, i, j).i \geq act'(h, i, j').i)\}$$

that being, an action triple such that no other legal triple applied to h results in a greater happiness for index i . It is irrelevant *which* of these (possibly many) hedonistically equal actions is chosen since they are, from the only perspective we have, equivalent; hence, the choice is non-deterministic.

Of course, returning some (i, act, j) on this basis is not to say that (i, act, j) *will be* the best action (indeed, it will often not be: see **Fig 3.c** for tall poppy effects and so on), neither does it mean that $act(h, i, j)$ is a state that will *ever* exist; the reason being that there are $N - 1$ other agents operating under similar bases, acting for “their” own good. What the Hedonist theory returns, in this instance, is a supposition, an imagined approximation from i 's pov. This use of “imaginary states” is common to our result-oriented theories, whereas deontological action-oriented theories need no imagination.

Now, consider a strain of Deontology that holds that one must always assist those worse off than oneself. Given a state h , the action of a Deontologist i ($t.i =$ Deontology) is a member of the set,

$$\{(i, assist, j) : \text{Agents} \times \text{Actions} \times \text{Agents} \mid i \neq j \wedge h.i > h.j \}$$

(of action-triples meeting the Authoritarian criterion) chosen arbitrarily and passed back to agent i . The bind is, what if *we* are the least happy; what if the set is empty? One could of course reformulate the law to eliminate *this instance* of the problem, but a problem it is: it is unreasonable to assume that a theory will find *any* association of action and reagent which meets with its approval.

One solution is to let the agent (explicitly devoid of other mechanisms of decision) act at random. But, by reconsidering what a theory does, we can solve the problem without recourse to special cases – and in any case, our current conception of a sort of *external* black-box or Oracle choosing *for* the agents is highly artificial. Rather than telling an agent what it must do, a *theory* will become a means of valuation, a method for making sense of the myriad possibilities open to even our very simple agents. An ethical theory is a metric for moral worth.

There is, as mentioned, a difference between those theories of Theories-Act and Theories-Result; the former consisting of deontological systems for which the point of ethical judgement is the nature of the action we are to perform, the latter of theories for which the result counts, and the precise nature of the action is irrelevant. In the former, Act-oriented mould, our metric operates on the agent, current state, and (proposed) action and reagent,

$$m_{t \in \text{Theories-Act}} : \text{Agents} \times \text{Actions} \times \text{Agents} \times \text{State} \rightarrow [0, 1]$$

The latter requires us to evaluate the outcome, the imagined state,

$$m_{t \in \text{Theories-Result}} : \text{Agents} \times \text{State} \rightarrow [0, 1]$$

In **Chapter One** we went some way towards reformulating Ethics in simulation-friendly terms, with emphasis on systematics over appeals to “common sense.” Following, we complete the abstraction, and render the theories into functions.

3.3.1 Hedonism

(See 2.2) A Hedonist agent operates for its own happiness so, for i evaluating an imagined state h' , we return no more than,

$$\begin{aligned} m_{\text{Hedonism}}: \text{Agents} \times \text{State} &\rightarrow [0, 1] \\ m_{\text{Hedonism}}(i, h') &= h'.i \end{aligned}$$

We might think to use instead the difference $h'.i - h.i$, reasoning that an action which takes us to (a happiness of) 0.6 should count higher if our initial happiness had been 0.3 than 0.5. Indeed, an act that results in our happiness going down from 0.9 to 0.7 is given a greater value than one that *increases* happiness from 0.1 to 0.6. However, there are several telling reasons to retain the first rendition, not least that,

$$m_{\text{Hedonism}'}(i, h, h') = h'.i - h.i$$

is a function to $[-1, 1]$ (although normalisation could fix that). First, in practice $m_{\text{Hedonism}'}$ is no different from m_{Hedonism} – it is equivalent to maximise $h'.i$ or $h'.i - h.i$, the latter just does slightly more work. Second, and more seriously, is its inconsistency. In a Hedonist system the only point of meaning is happiness: our previous state is as ethically irrelevant as the action we have performed; the present state is of no more direct ethical relevance than the action we are to perform.

3.3.2 Eudemonism

(See 2.3) Expressing “the greatest good to the greatest number” within $[0, 1]$ is to return no more than the imagined state’s mean happiness,

$$\begin{aligned} m_{\text{Eudemonism}}: \text{Agents} \times \text{State} &\rightarrow [0, 1] \\ m_{\text{Eudemonism}}(i, h') &= \frac{1}{N} \sum_{n=0}^{N-1} h'.n \end{aligned}$$

(Clearly, if $i \in \text{Agents}$ and h' is a legal state, $m_{\text{R-Eudemonism}}(i, h') \in [0, 1]$.)

3.3.3 Consequentialism

In 2.3 we introduced Consequentialism with Utilitarianism, and proceeded to develop the idea of Eudemonism as expressed in 3.3.2. Consequentialism is a synonym for “result-oriented”, and in that, both Hedonism and Eudemonism are Consequentialist theories, albeit weak ones, operating on states that will not exist and making no allowance for the influence our actions will have on those of others. Here, we work towards a stronger version, one that does account for the influence our actions will have on others.

At turn t , agent i acts on j , which, together with the other agents’ actions, brings us to turn $t + 1$, and another process of individual decision, until all have decided by what activity they will step to $t + 2$, those decisions dependent entirely on the state at $t + 1$, which is dependent in part on the action of i at turn t . Therefore, to judge an action by its consequences we should look at least one full turn ahead (where Hedonism and Eudemonism foresee $1/N^{\text{th}}$ of a turn).

There are distinct paths: i) we could construct a function that takes a state and proposed action to a value (always within $[0, 1]$) by going through and valuing as a whole the (possible) next-states. This does not rely on troublesome imagined states, but takes no real account of an action’s influence on others. We should, then, value an action in terms of all possible states that arise from all possible states in which the action is performed – imposing an additional form of metric (one from agent, action and state), and a ridiculous burden of computation. Instead ii), Consequentialism is addressed with, again, a function from agent and imagined state to $[0, 1]$,

$$m_{\text{Consequentialism}}: \text{Agents} \times \text{State} \rightarrow [0, 1]$$

but one for whom “valuation” is applied to the (many) successor states, and those values incorporated by some method into *the* result. So, if we have a procedure α for judging the quality of a state,

$$\alpha: \text{State} \rightarrow [0, 1]$$

and a method β for conflating a sequence of these “qualities” into a real of $[0, 1]$,

$$\beta: [0, 1]^* \rightarrow [0, 1]$$

then the Consequentialist metric is defined, for i : Agents, h : State,

$$m_{\text{Consequentialism}}(i, h) = \beta \langle \alpha h^d \mid d : \text{All-Actions} \rangle$$

(Where h^d is the next-state, the result of all agents’ actions d – see **3.4.1**, and **3.2.5** for All-Actions.) “Deeper” Consequentialism is defined with a “depth” superscript, $p > 1$,

$$m_{\text{Consequentialism}}^p(i, h) = \beta \{ m_{\text{Consequentialism}}^{p-1}(i, h^d) \mid d : \text{All-Actions} \}$$

and,

$$m_{\text{Consequentialism}}^1 = m_{\text{Consequentialism}}$$

One could in β construct a complex system of possibility-weighting and the like, but, as we are ultimately going to dispose of this class of metrics, a first approximation will do – the average:

$$\beta C = \text{mean } C$$

Apropos of α , we distinguish three flavours of Consequentialism: egoist, Utilitarian, and altruistic. For the first two, we have already defined our methods: Hedonism and Eudemonism. Altruism is a variant of Eudemonism that excludes the actor. These are defined, respectively, for i : Agents and h' : State (an imagined state),

$$m_{\text{Consequentialism-Egoist}}(i, h') = \text{mean} \langle m_{\text{Hedonism}}(i, h'^d) \mid d : \text{All-Actions} \rangle$$

$$m_{\text{Consequentialism-Utilitarian}}(i, h') = \text{mean} \langle m_{\text{Eudemonism}}(i, h'^d) \mid d : \text{All-Actions} \rangle$$

$$m_{\text{Consequentialism-Altruist}}(i, h') = \text{mean} \langle m_{\text{Altruism}}(i, h'^d) \mid d: \text{All-Actions} \rangle$$

where,

$$m_{\text{Altruism}}(i, h') = \frac{\left(\sum_{n=0}^{N-1} h'.n \right) - h'.i}{N-1}$$

A problem lurks: All-Actions is huge, in the region of N^N . Practical computability does not intrude on the project as-such (computing, for example, a hundred Eudemonists for a thousand turns takes five minutes on a mid-range PC), but the combinatorial explosion involved in enumerating the possibilities just one turn ahead renders this, in a project, in a *field* the nature of which is to be run, of moot interest. The sheer impracticality of it is slightly absurd though, when contrasted with the real-life situation our Consequentialism caricatures: in a world of two actions and ten people, for example, considering all the (immediate!) consequences of ones actions is unfeasible; for a hundred people, it cannot be done. In practice, it is impractical to use more than six agents with a Consequentialist (if the computation is not to be left overnight). One could drastically cut the computational burden: evaluate, say, only 1% of the possibilities, or use more information as to the other agents' likely behaviour. But the former is a dilution and the latter I imagine imposes an even greater computational burden, so little more will be said about Consequentialism.

Of course, in our rarefied world, it makes no sense to construct Consequentialist versions of the action-oriented theories (Consequentialist-Authoritarianism is an oxymoron) – those theories here subsumed by the class: Deontologism.

3.3.4 Deontologism

This is also not *a* theory in the sense of **3.3.1** and **2**, but a set of theories, a conflation of mutually contradictory rules and laws. A deontologistic theory is a rule-of-action: it delineates exactly that which is good and evil. The Deontologistic metrics are functions from proposed action (that is, agent, action, reagent) and current state to $\{0, 1\}$ – an act is either permitted (good), or not (evil). One obeys or disobeys. An

instance of Deontology is to *help those worse off than yourself*, expressed:

$$m_{\text{Deontology-Samaritan}}(i, act, j, h) = \begin{cases} \perp \leftarrow i = j \\ 1 \leftarrow i \neq j \wedge h_i > h_j \wedge act = \text{assist} \\ 0 \leftarrow \text{otherwise} \end{cases}$$

Whether we are using the Authoritarianism of **2.1** or (Kantian) Deontology (see **2.4**) is relevant only in formulating what those laws are; having deduced or been told how to behave, nothing is left but to obey. A Kantian, reasoning that he does not wish to be hindered by others, makes it a rule never to hinder. He might wish to be assisted by everyone else, but cannot similarly make that a law since it cannot be made universal (though, “always assist the agent to your (numerical) right” has potential), so our Kantian is satisfied with:

$$m_{\text{Deontology-Peaceful}}(i, act, j, h) = (act = \text{assist})$$

A note on evil: our agents are nothing but strict followers of their ethics; if they “do evil” it is as a by-product or side-effect of them attempting to do good – or, for the Deontologist, if the theory (in a sense) fails, and offers nothing but evil for them to perform. We will not in this dissertation work on purposefully evil agents – it brings unneeded complication to our already overburdened word, “evil” (an evil hedonist, here, looks more like a depressive), see **Conclusion** (*Arguments against the approach*) – but, in our metrics, we can easily define agents that take a contrary path to their ethical teaching, who (in a sense) intend to do evil: rather than using the result m of the metric, go by $1 - m$. (But this is by the bye.)

3.3.5 Virtue Ethics

Virtue Ethics, the steering of a well-thought path between behavioural extremes, slides awkwardly into the theory as developed thus far. Could it, for instance, be expressed by giving our agents a ratio of assist to hinder, so they will neither be forever helpers nor serial attackers, this ratio to be determined by experience, one supposes. In the rudimentary simulations of **Chapter 4**, it is evident that always-

assisting is the “best” strategy, so the agents should reason that a ratio of 1:0 is their “path between extremes”; but, if there is a cheater present, a hedonist amid the eudemonists say, the individual might conclude that 0:1 is actually the best. (See **Fig 4.g.**)

One could envisage starting with a large field of Virtue Ethics following agents, each with a random ratio of assist to hinder, where periodically the worse off adopt the ratio of the better off, and there being a gradual convergence to some “optimal” ratio. This result would, I expect, be a function of the implementation of assist and hinder. However, one can imagine situations when the best action to do is exactly what the majority are *not* doing, in which case this stable ratio should not exist, and from our agents we would expect more interesting behaviour.

It would look something like,

$$m_{\text{Virtue Ethics}}: \text{Agents} \times \text{Actions} \rightarrow \{0, 1\}$$

$$m_{\text{Virtue Ethics}}(i, act) = \begin{cases} 1 \leftarrow e \in [0,1] \mid e < r(i, act) \\ 0 \leftarrow o/w \end{cases}$$

$(e \in [0, 1] \mid e < r(i, act))$ is an ad hoc way of expressing a random number from $[0, 1]$ being less than $r(-)$ where r is a function from agent-action pairs to $[0, 1]$ such that,

$$r(i, assist) + r(i, hinder) = 1$$

(In general, $\sum_{a \in \text{Actions}} r(i, a) = 1$) i.e. i 's ratio of assist to hinder is $r(i, assist) : r(i, hinder)$; for $r(i, assist).100\%$ of the time, i will assist, the rest, hinder.

We would have another function to be called from time-to-time that lets the poorest-off agent alter his ratio of actions to that of the best-off,

$$\text{update-r}: (\text{Agents} \times \text{Actions} \rightarrow [0, 1]) \times \text{State} \rightarrow \text{Agents} \times \text{Actions} \rightarrow [0, 1]$$

$$\text{update-r}(r, h) = r[(i, assist) \mapsto r(j, assist), (i, hinder) \mapsto r(j, hinder)]$$

(where $i, j \in \text{Agents}$ • $h.i = \min(h)$ and $h.j = \max(h)$)

But alas this is by the bye. We are extemporising on Virtue Ethics until the plans (setting agents in competition) have but a parenthetic connection to Aristotle's doctrine. It might be of interest if one could connect it to, say, some problem of distributed computing, but otherwise it is pretty void of purpose. There is a level of complication here absent from the other theories, an additional component to an Architecture we wish, in fact, to make the epitome of simplicity.

3.4 Architecture

Having established the constituents of our model: namely, the sets,

$$\text{Agents} = \{0, \dots, N - 1\}$$

$$\text{State} = \text{Agents} \rightarrow [0, 1]$$

$$\text{Actions} = \{\text{assist}, \text{hinder}\}$$

$$\text{Theories} =$$

$$\text{Theories-Act} \cup \text{Theories-Result} =$$

$$\{\text{Hedonism}, \text{Eudemonism}, \text{Consequentialism-}__, \text{Deontologism-}__\}$$

and functions,

$$t: \text{Agents} \rightarrow \text{Theories}$$

$$h: \text{State}$$

$$d: \text{Agents} \rightarrow \text{Actions} \times \text{Agents}$$

$$m_{t \in \text{Theories-Result}} : \text{Agents} \times \text{State} \rightarrow [0, 1]$$

$$m_{t \in \text{Theories-Act}} : \text{Agents} \times \text{Actions} \times \text{Agents} \times \text{State} \rightarrow [0, 1]$$

the question is, how to bring them into a working whole. Where do the dependencies lie; how are we to bring **3.1**, **2**, **3** into connexion. Our decisions here are informed by parsimony, and by taking – if this makes sense – the least arbitrary path.

The question of *clocks*, whether we imagine a soup of agents acting independently to their own time, or a regiment of agents, acting to a global clock, has been decided by the work above, and to the latter. *First* we compute the function d (**3.2.5**), *then* we apply it to h (**3.1**).

Thus, the happiness at turn $t + 1$ depends entirely upon the happiness at turn t . A strong assumption, one that precludes, most obviously, the agents' histories. This is a barrier to a real simulation of human behaviour, one would think, but not necessarily in the terms we have set ourselves: the idealised codes of conduct offered by Ethics. Still, a judgement flagged for future revision.

Do we include “thinking time” into our calculations. Should, say, a Consequentialist have one action to the Deontologist's three? No. Why should we think that a Consequentialist *does less* than a Deontologist? Indeed, if we divide up the turns according to processor time, a Consequentialist would hardly ever act (if there are nine other agents, a Consequentialist would operate in the region of twelve orders of magnitude slower than an agent advised by a different theory – twenty agents, and it is *thirty one* orders of magnitude). This is not to say we could not meaningfully weight this time (thinking time, we might suppose, is a small component of the total time it takes to act), but such is the subject of refinements, not prototypes, as unfolded:

3.4.1 Next state

To take one state h to its subsequent state h' is a two-step process: first, compute the agents' actions; then, apply those actions d to h . If the theories t (Agents \rightarrow Theories) is a constant, we have,

$$d = \text{decisions}(h)$$

$$h = \text{happiness}(d, h)$$

and,

$$\text{next-state: State} \rightarrow \text{State}$$

$$\text{next-state}(h) = \text{happiness}(\text{decisions}(h), h);$$

where d is defined (as the not necessarily unique),

$$\text{decisions: State} \rightarrow \text{Agents} \rightarrow \text{Actions} \rightarrow \text{Agents}$$

$$\begin{aligned} \text{decisions}(h) = \{ & (i, \text{act}, j) : \text{Agents} \times \text{Actions} \times \text{Agents} \mid i \neq j \bullet \\ & ((i, \text{act}', j') : \text{Agents} \times \text{Actions} \times \text{Agents} \mid i \neq j' \bullet \\ & \quad t.i \in \text{Theories-Act} \Rightarrow m_{t.i}(i, \text{act}, j, h) \geq m_{t.i}(i, \text{act}', j', h) \wedge \\ & \quad t.i \in \text{Theories-Result} \Rightarrow m_{t.i}(i, \text{act}(h, i, j)) \geq m_{t.i}(i, \text{act}'(h, i, j')) \\ & \left. \right\} \end{aligned}$$

i.e. for each agent i , $d.i$ is an action-reagent pair such that no pair has a greater value (under the metric $t.i$).

$\text{decisions}(h)$ is not strictly a function, and its coding presents an interesting subtlety in interpretation: assuming a serial, imperative program that for all agents i traverses the space of potential actions, applying the metric $t.i$ to each and retaining the best, in which order should the space,

$$\text{Actions} \times \text{Agents}$$

be traversed? Do we make use of the ordering implicit in our definition of Agents: try $(\text{assist}, 0)$, $(\text{assist}, 1)$, \dots , $(\text{hinder}, N - 1)$. No, because this would suppose that our agents are *aware* of their relative order; if (in the initial state, say) all agents are equal, they will choose as a body to act on 0 (except for 0, who will act on 1) for a wholly arbitrary reason. Unreasonable, and hence, what is non-deterministic above translates not merely to the arbitrary but the *random* in code (random impressing upon the programmer the additional obligation that while we do not care at any single instance

in which order the agents' decisions are evaluated, we trust that no one order will be favoured over another).

Thus, we simultaneously insist that d is Ethically determined and indeterminable.

In defining happiness(d, h), we introduce some (ad hoc) notation,

$$h^{(i, \text{act}, j)} =_{\text{df}} \text{act}(h, i, j)$$

Then, given $a = (i, \text{act}, j) \in \text{Agents} \times \text{Actions} \times \text{Agents}$, $i \neq j$,

$$h^a$$

is defined and uniquely so (under the implementation of act); the problem of how to apply d to h becomes that of finding a meaningful definition for,

$$h^d$$

(As used in **3.3.3**.) If we have two actions $a = (i, \text{act}, j)$, $b = (i', \text{act}', j') \in \text{Agents} \times \text{Actions} \times \text{Agents}$, $i \neq j$, $i' \neq j'$, the result of applying first a then b is known, and defined,

$$(h^a)^b = \text{act}'(\text{act}(h, i, j), i', j')$$

Clearly,

$$(h^b)^a = (h^a)^b$$

holds for all $a = (i, \text{act}, j)$, $b = (i', \text{act}', j')$ if $i \neq i'$ and $j \neq j'$, since $\text{act}(h, i, j)$ is only permitted to alter the i and j indices of h.

Let,

$$h^d =_{\text{df}} (\dots(h^a)^b)\dots^z,$$

where $\{a\} \cup \{b\} \cup \dots \cup \{z\} = d$ and $\text{card}(a, b, \dots, z) = \text{card } d$.

In the vein of `decisions()`, the question arises: in which order should the Agents of h be plucked? There is no good reason to imagine a queue of agents, a prescribed sequence. Indeed, the formulations above assume that agents are not aware of their order, otherwise `decisions()` should be changed to take this into account (so that agent 1 knows that 0 will have the first crack at hindering 2, say, reducing the benefit to 1).

We could so constrain our actions to ensure that their order of application to h is irrelevant. If, for instance, the agents' happiness were reset to 0.5 (having computed d and before applying d to h), then defining `assist` and `hinder` along the lines of adding or subtracting some constant n , where $n \leq \frac{1}{2}N$, is enough to guarantee that the bounds of h will not be overstepped, and that $(h^b)^a = (h^a)^b$ holds in general. But then it makes no difference whom an agent acts against, or whosoever acts upon them. We could refine this, make use of the previous state, but now it makes no odds to the actors whether they are one or many (we imagine three agents hindering a fourth simultaneously and, curiously, not). In lieu of a more sophisticated solution, we conclude that h^d is not unique, that `happiness(d, h)` is not in fact a function: the order of application is random.

3.4.2 *Guarded command program code*

```
h := [0.5, 0.5, ...]    { some initialisation }
                       { card h = card d }
```

```
do true →
    d := decisions(h);
    h := happiness(d, h);
od
```

The architecture for fickle agents, in which the theories t becomes variable, is the pleasingly symmetric,

$h := [\alpha, \beta, \gamma, \dots]; \{ \text{some initialisation, } \alpha, \beta, \gamma, \dots \in [0, 1] \}$

$t := [A, B, \Gamma, \dots]; \{ A, B, \Gamma, \dots \in \text{Theories} \}$

$\{ \text{card } t = \text{card } h = \text{card } d \}$

do **true** \rightarrow

$d := \text{decisions}(h, t);$

$h := \text{happiness}(d, h);$

$t := \text{theories}(h, t);$

od

Incidentally, from this, backwards and with the imposition of a vowel, we get the model's name: Thud.

3.5 $d = \text{decisions}(h)$

value := -1;

actees := Agents

actions := Actions

$i := 0;$

do $i \neq N \rightarrow$

do actees $\neq \emptyset \rightarrow$

$\{ \text{clumsy way of saying: remove a random element from the set} \}$

act : actions;

do actees $\neq \emptyset \rightarrow$

$j : \text{actees};$

if $i \neq j \wedge (t.i \in \text{Theories-Result} \wedge m_{t.i}(i, \text{act}(h, i, \text{act}, j))) \geq \text{value})$

value := $m_{t.i}(i, \text{act}(h, i, \text{act}, j));$

$d.i := (\text{act}, j);$

```

    [] i ≠ j ∧ (t.i ∈ Theories-Action ∧ mt,i(i, act, j, h) ≥ value) →
        value := mt,i(i, act, j, h);
        d.i := (act, j);
    fi

    actees := actees - {j};
od
    actions := actions - {act}
od
    i := i + 1;
od

```

Three loops or duration N , N and card.Actions , the latter a constant 2, means a time complexity of $O(N^2)$. [However, the Consequentialist metrics do not compute in constant time but (reckoning similarly), $O(N^2)$.]

3.6 $h' = \text{happiness}(d, h)$

```

actors := Agents;
j: Agents;
act: Actions;

do actors ≠ ∅ →
    i : actors;
    (act, j) := d.i;
    h := act(h, i, j);
    actors := actors - {i};
od

```

3.7 $t' = \text{theories}(h, t)$

Apostasy: of potential interest (allowing the unhappiest to switch ethics, setting Ethical theories in competition). Here for completeness.

```
actors := Agents;
unhappiest, happiest : Agents;
min, max := 1, 0;

do actors  $\neq \emptyset$   $\rightarrow$ 
    i: actors;
    if  $h.i \geq \text{max}$   $\rightarrow$ 
        happiest, max := i, h.i;
    []  $h.i \leq \text{min}$   $\rightarrow$ 
        unhappiest, min := i, h.i;
    fi
    actors := actors - {i};
od

t.unhappiest := t.happiest;
```

And now, to look at some simulations.

Chapter 4

Notes on early simulations

The Java command-line implementation of the program developed in **Chapter 3** (see **Appendix A** for the source code) takes the following parameters:

<i>turns</i>	the number of turns we are to compute
<i>initial level</i>	the agents' happiness at turn 0, where less than 0 signifies random (within the interval [0, 1])
<i>assist number</i>	the specific assist function we wish the agents to use (we do not look into the use of more than one implementation of “assistance”)
<i>hinder number</i>	
<i>hedonists</i>	the number of hedonists
<i>eudemonists</i>	...eudemonists
<i>deontologists</i>	etc...
<i>consequentialist-hedonists</i>	
<i>consequentialist-eudemonists</i>	

and produces three text files, all suffixed with the particular instantiations of the parameters above, with prefixes,

<i>graph-</i>	happiness data turn-by-turn
<i>averages-</i>	average happiness (to better see developing trends and the “overall good”)
<i>actions-</i>	what each agent did to whom

The first two are formatted for simple inclusion into a spreadsheet (below I use

Microsoft Excel¹³). For example, running two hedonists, three eudemonists and five deontologists (their rule: always help the needier) for ten turns:

```
>> Computing 10 turns of 10 agents
N=10 T=10 lev=0.5 H=2 E=3 V=0 CH=0 CE=0 CV=0 DW=5 DS=0
ass=1 hin=1
 0 1 2 3 4 5 6 7 8 9
Final state:
 0.5891853274552397 0.6056009883256158
 0.604613929854067 0.5868659406544714
 0.6243784825267418 0.5703121542599574
 0.6138344891254892 0.6315138917655185
 0.6128348620742157 0.5946387343386068
>>
```

creates the files:

```
actions-N=10 T=10 lev=0.5 H=2 E=3 V=0 CH=0 CE=0 CV=0 DW=5 DS=0
ass=1 hin=1.txt
```

```
0.5      0.5      0.5      0.5      0.5      0.5      0.5      0.5      0.5      0.5
0 ?? 0   1 ?? 0   2 ?? 0   3 ?? 0   4 ?? 0   5 ?? 0   6 ?? 0   7 ?? 0   8 ?? 0   9 ?? 0

0.499    0.451    0.564    0.529    0.531    0.441    0.576    0.562    0.480    0.469
0 hi 8   1 hi 9   2 as 0   3 as 7   4 as 6   5 as 2   6 hi 5   7 as 8   8 hi 0   9 hi 1

0.550    0.557    0.587    0.586    0.561    0.601    0.420    0.587    0.512    0.543
0 hi 6   1 hi 6   2 as 5   3 as 5   4 as 5   5 as 3   6 as 1   7 as 1   8 as 9   9 as 5

0.613    0.603    0.611    0.614    0.589    0.437    0.602    0.611    0.567    0.598
0 hi 5   1 hi 5   2 as 6   3 as 6   4 as 6   5 as 6   6 as 9   7 as 0   8 as 6   9 as 8

0.646    0.644    0.630    0.457    0.615    0.634    0.630    0.629    0.640    0.517
0 hi 3   1 hi 3   2 as 5   3 as 5   4 as 5   5 hi 9   6 as 5   7 as 8   8 as 5   9 as 8

0.570    0.589    0.686    0.569    0.638    0.667    0.647    0.650    0.655    0.605
```

¹³ A not difficult extension to the program of **Appendix A** would be a graphical interface by which one could see the agents' turn-to-turn progress and, as it were, make adjustments "on the fly". This would not be helpful for the dissertation, though, for which other people's graphing software gives clearer and more flexible results than my taking screenshots of some Java Applet window.

```

0 hi 1 1 hi 0 2 as 3 3 as 9 4 as 3 5 as 2 6 as 2 7 as 9 8 as 5 9 as 3
0.634 0.674 0.500 0.659 0.673 0.683 0.666 0.666 0.670 0.643
0 hi 2 1 hi 2 2 as 3 3 as 0 4 as 3 5 as 9 6 as 3 7 as 4 8 as 1 9 as 1
0.710 0.711 0.606 0.709 0.693 0.490 0.684 0.680 0.683 0.661
0 hi 5 1 hi 5 2 as 0 3 as 2 4 as 2 5 as 0 6 as 2 7 as 3 8 as 3 9 as 2
0.517 0.643 0.664 0.726 0.711 0.604 0.699 0.707 0.696 0.690
0 hi 1 1 hi 0 2 as 5 3 as 5 4 as 5 5 hi 0 6 as 2 7 as 9 8 as 7 9 as 2
0.649 0.680 0.693 0.514 0.729 0.643 0.711 0.719 0.720 0.717
0 hi 3 1 hi 3 2 as 0 3 as 0 4 as 0 5 as 0 6 as 8 7 as 2 8 as 9 9 as 5

```

averages-N=8 T=10 lev=0.5 H=3 E=5 V=0 CH=0 CE=0 CV=0 DW=0 DS=0
 ass=1 hin=1.txt

2 Hedonists (S1,2)
 3 Eudemonists (S3, 4, 5) and
 5 Deontologists (S6, 7, 8, 9, 10) -
 - average

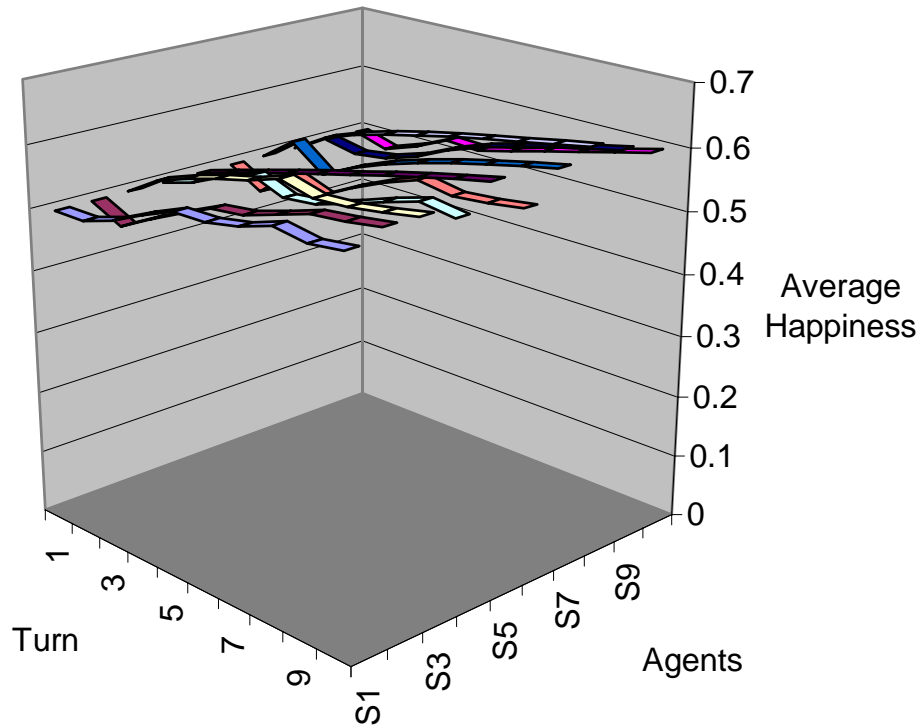


Fig 4.a

and,

graph-N=10 T=10 lev=0.5 H=2 E=3 V=0 CH=0 CE=0 CV=0 DW=5 DS=0
ass=1 hin=1.txt

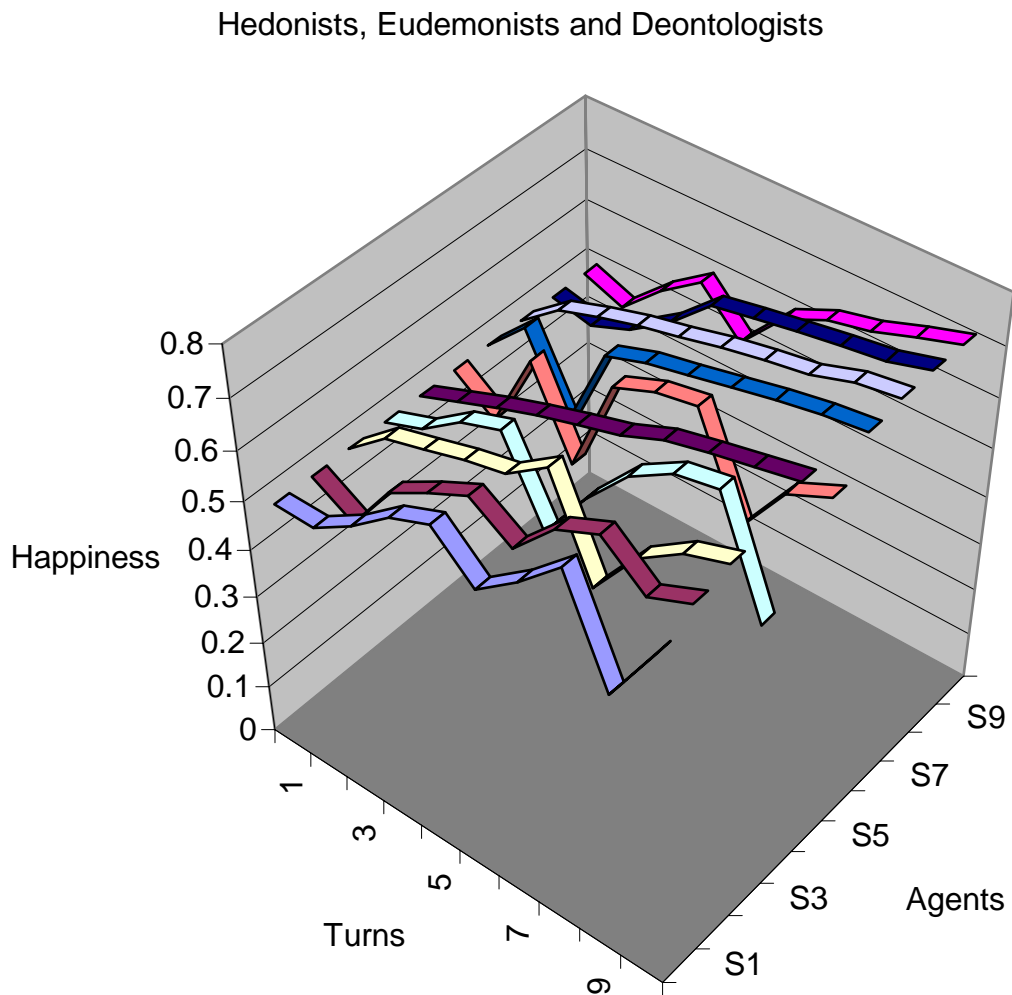


Fig 4.b

But this is to *jump the gun*; so we start anew with an aesthetically pleasing run of Eudemonists. As a consequence of our first implementations of assistance and hindering (**Fig 3.c**), the agents will always choose to assist the least well off of their number; their happiness rapidly converges to 1 (although may only reach it by a rounding error, since we are removing every time a fraction of the difference between its happiness and 1).

Interest it to be found in (Fig 4.c in) the first few turns of the following graph, where we see that that agent which happens to be the least happy at turn 1 (because none other assists him and she he assists, he assists last) is catapulted to the highest at 2, having been helped by all others.

Thirteen Eudemonists

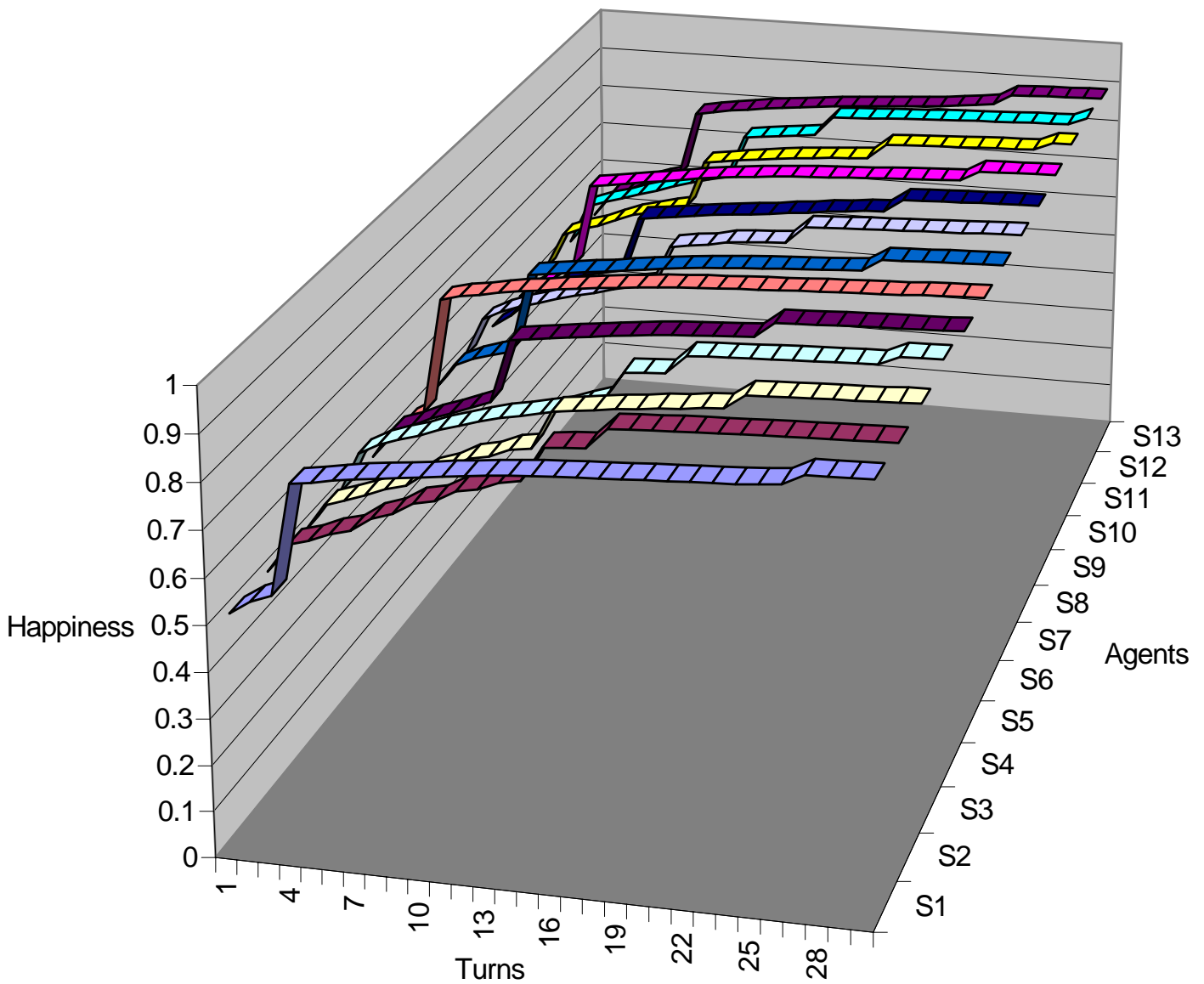


Fig 4.c

This communal aid may be compared with the mob hindrance of Hedonists, as the regular plummets of **Fig 4.d** testify,

Thirteen Hedonists

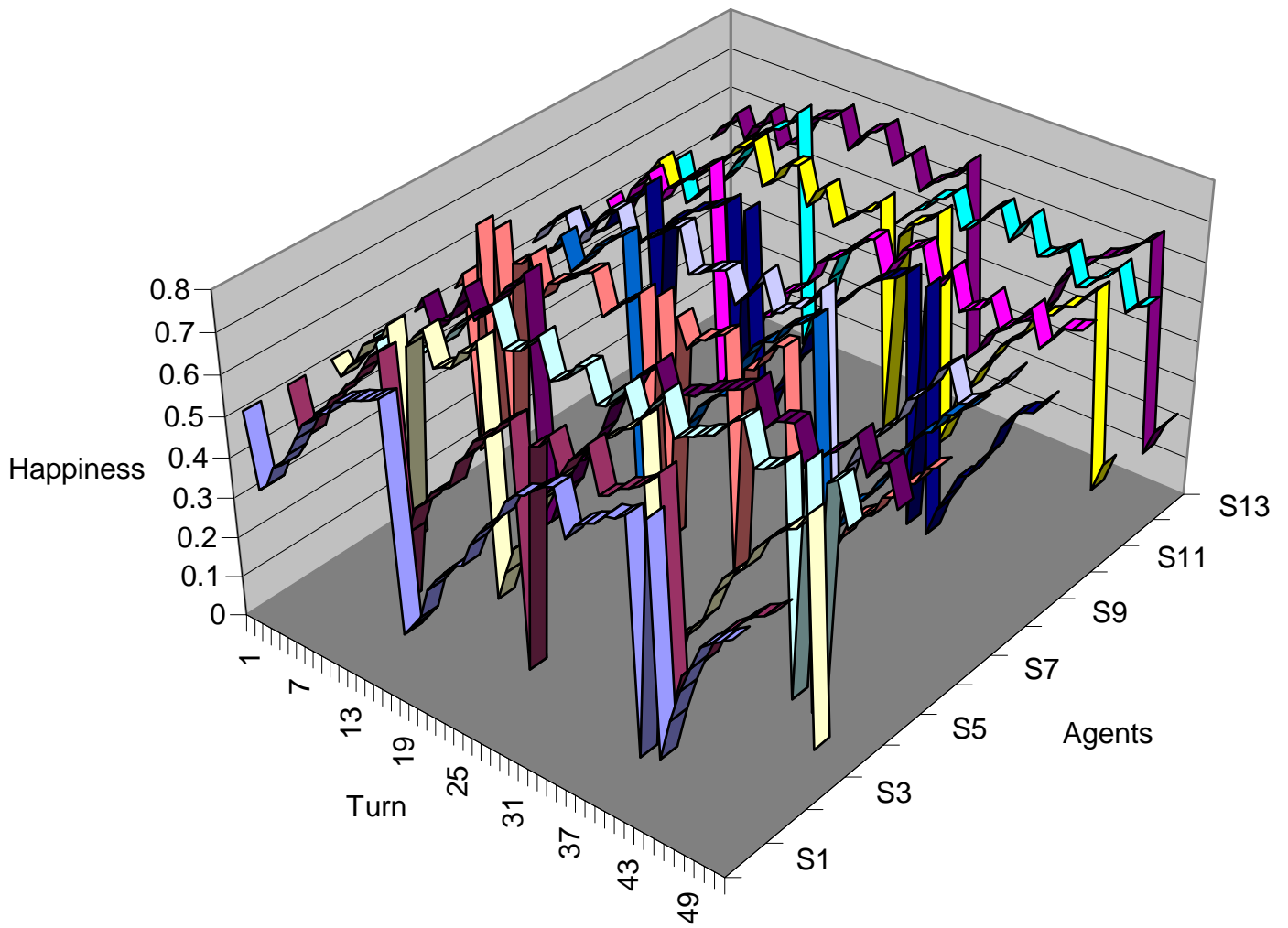


Fig 4.d

– those catastrophic falls in happiness occurring always to the happiest.

That we find all agents performing the same action is a function of the simplicity of our actions. We could mitigate this herding by, say, defining benefit to be relative to the difference between actor and actee's happiness, the question is, how to justify that? Without some reasoning grounded in the systems our simulations approximate, those simulations are utterly meaningless.

Now, compare a sample of Deontologists with the Eudemonists of **Fig 4.c**; agents motivated to maximise group happiness against agents commanded to, in this case, always help those less happy than yourself.

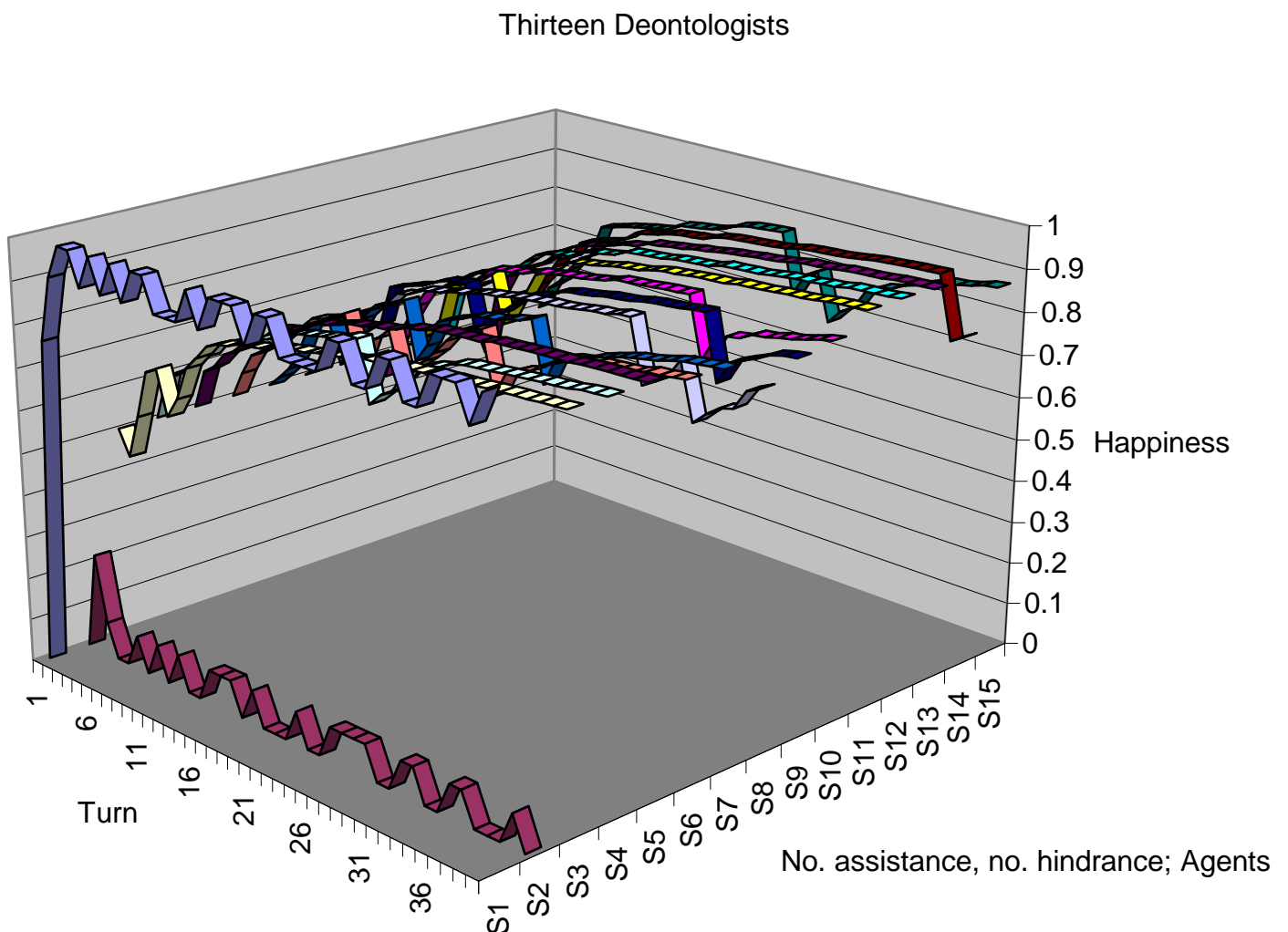


Fig 4.e

In **Fig 4.e**, S1 represents the (normalised) incidence of assistance, and S2 that of hindrance. (S3 – S15 are the agents, as per usual.) We see that every other turn, or so, one agent decides to *hinder* another; consequently, the overall performance is rather “worse” than we find in **Fig 4.c**. The reason is, that agent who is (or, especially for turn 0, those that are) the least happy finds no instruction: all courses of action are equally bad, so it chooses randomly, a random action and random reagent.

We can clarify the overall difference by adjoining the *averages* graphs of seven Eudemonists and seven Deontologists:

Seven Deontologists' average, then, Seven Eudemonists' average

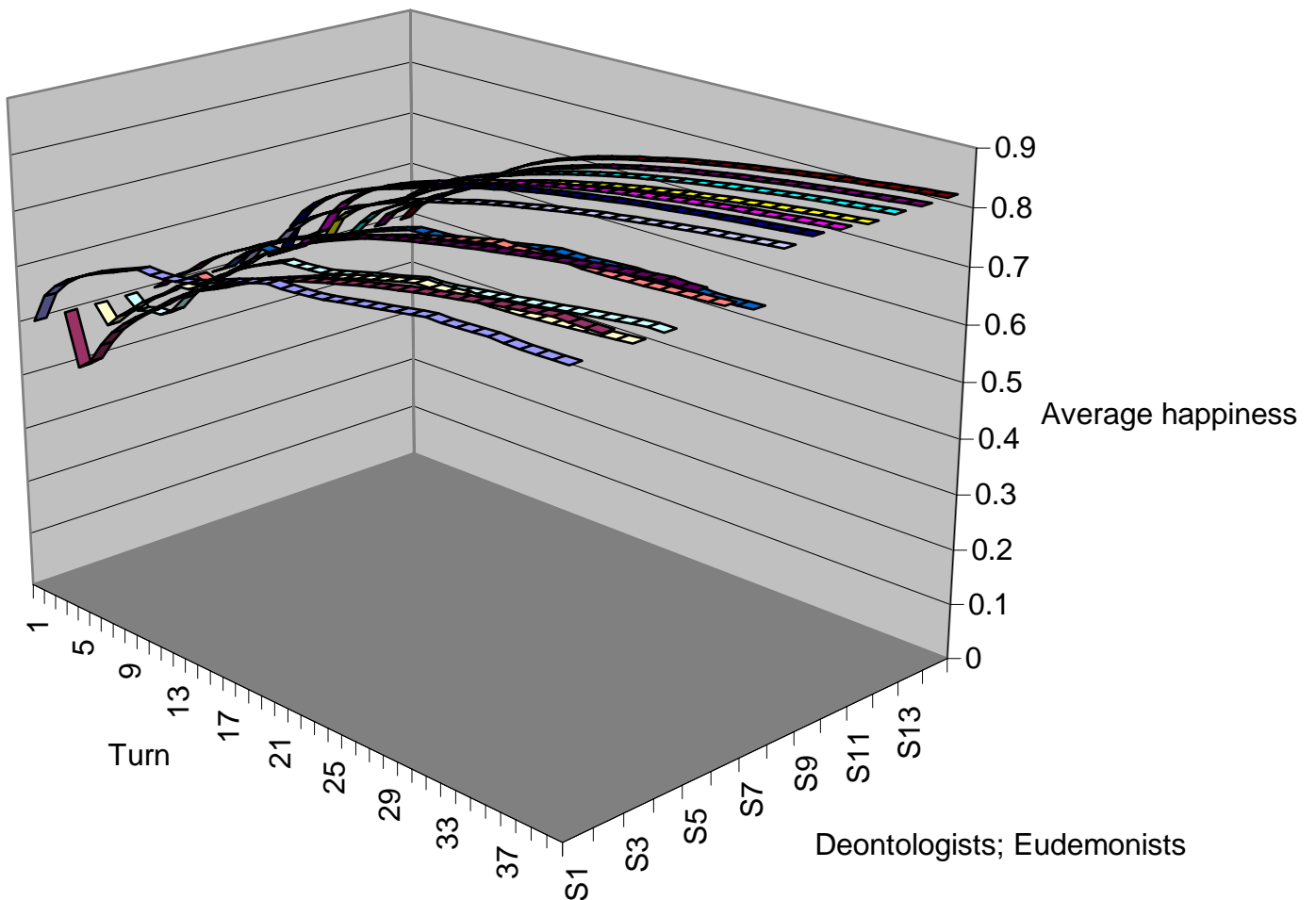


Fig 4.f

Having composed two separate runs, we will return to considering mixed fields. Namely, one Hedonist in a group of Eudemonists, which goes some way to demonstrate that while “always assist” works as a universal rule, it is vulnerable to cheaters.

One Hedonist amid the Eudemonists

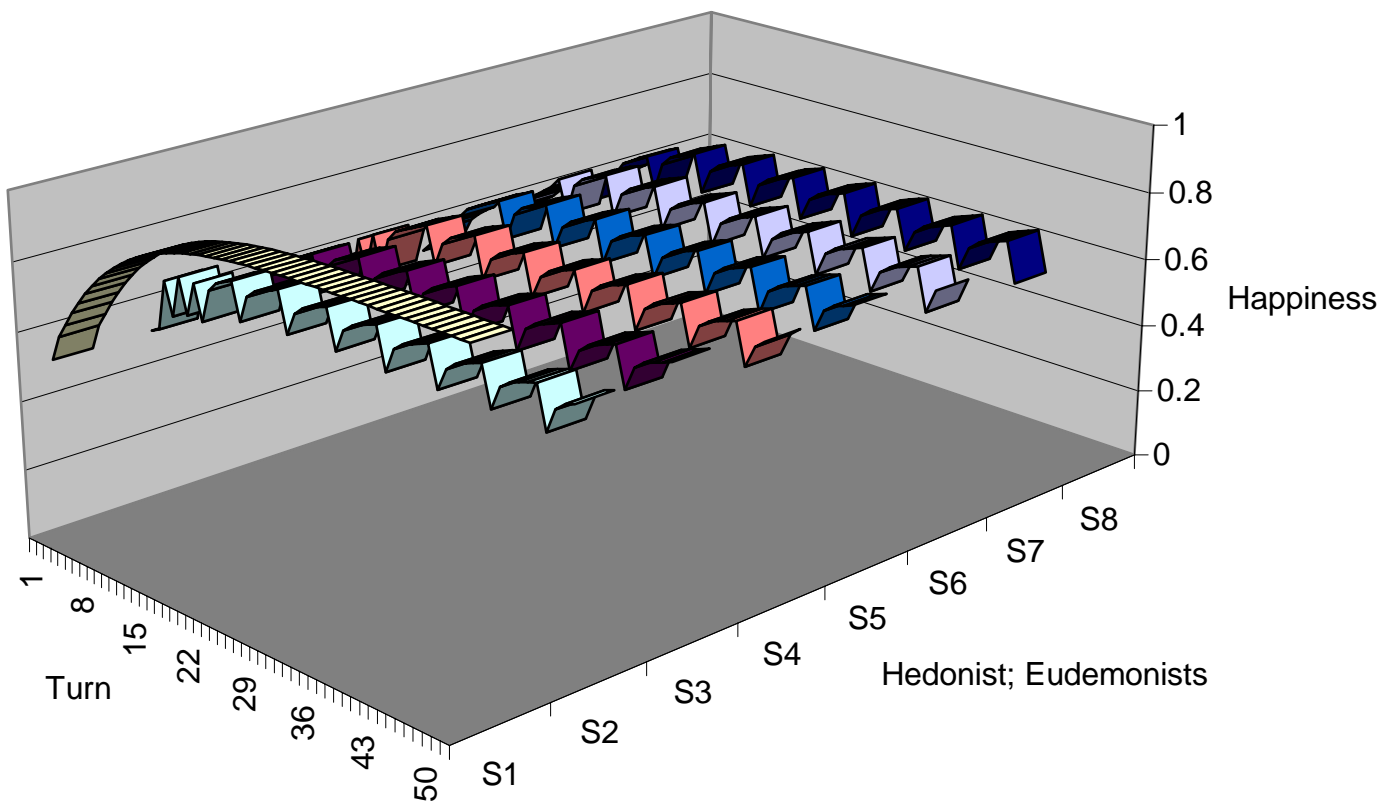


Fig 4.g

The Hedonist, S1, quickly attains maximum happiness; the Eudemonists (S2 – 8), however, are alternately beaten back into a pleasing but depressive waves.

Looking at larger simulations, we find this pattern breaking down:

One Hedonist, many Eudemonists

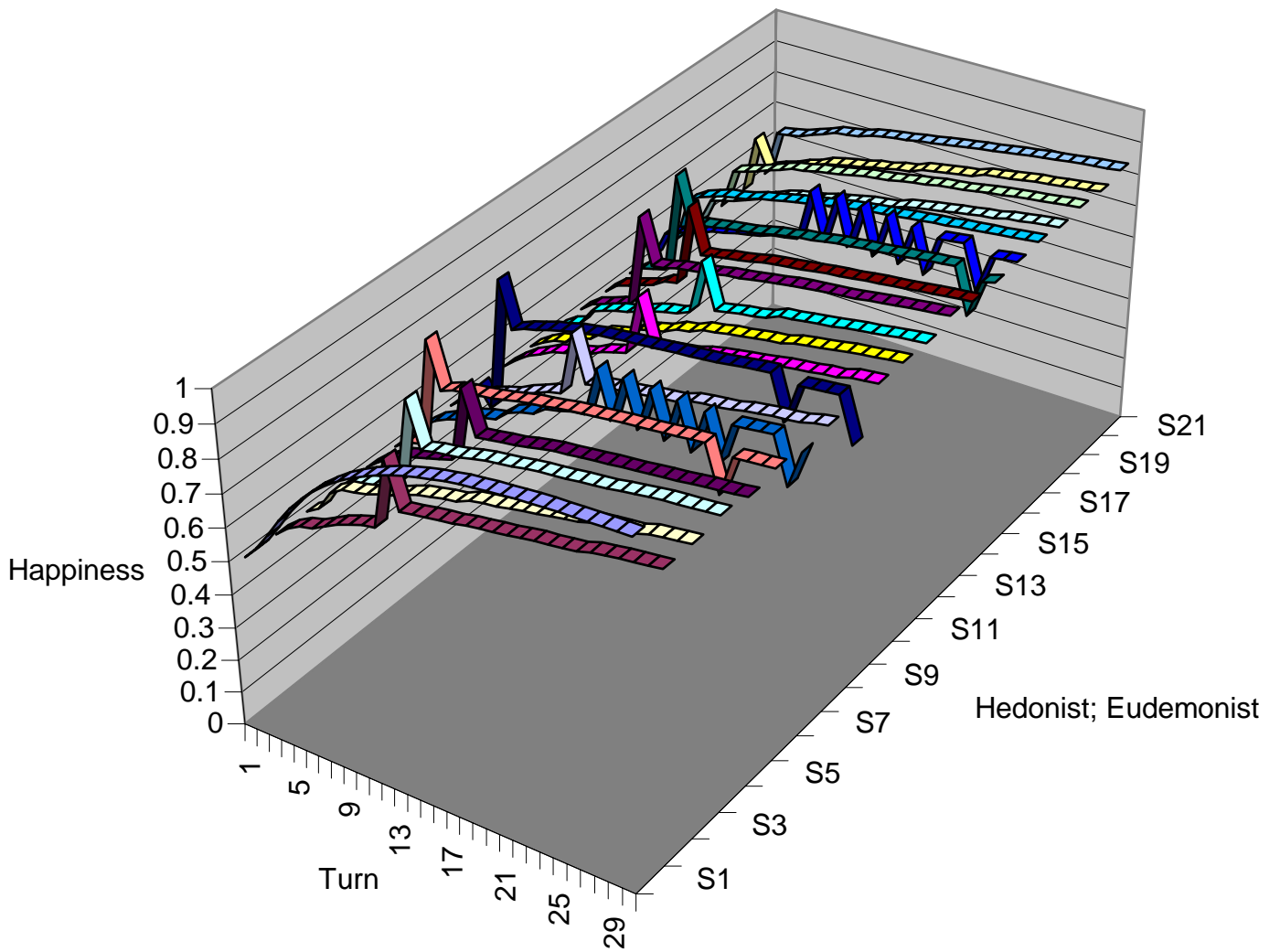


Fig 4.h

However, with a longer plot, it seems what we see above is an initial period of irregularity before a periodic (canon) system is reached (uncertainty is at the beginning, and is quickly wiped out, leaving but minor fluctuations.):

One Hedonist, many Eudemonists

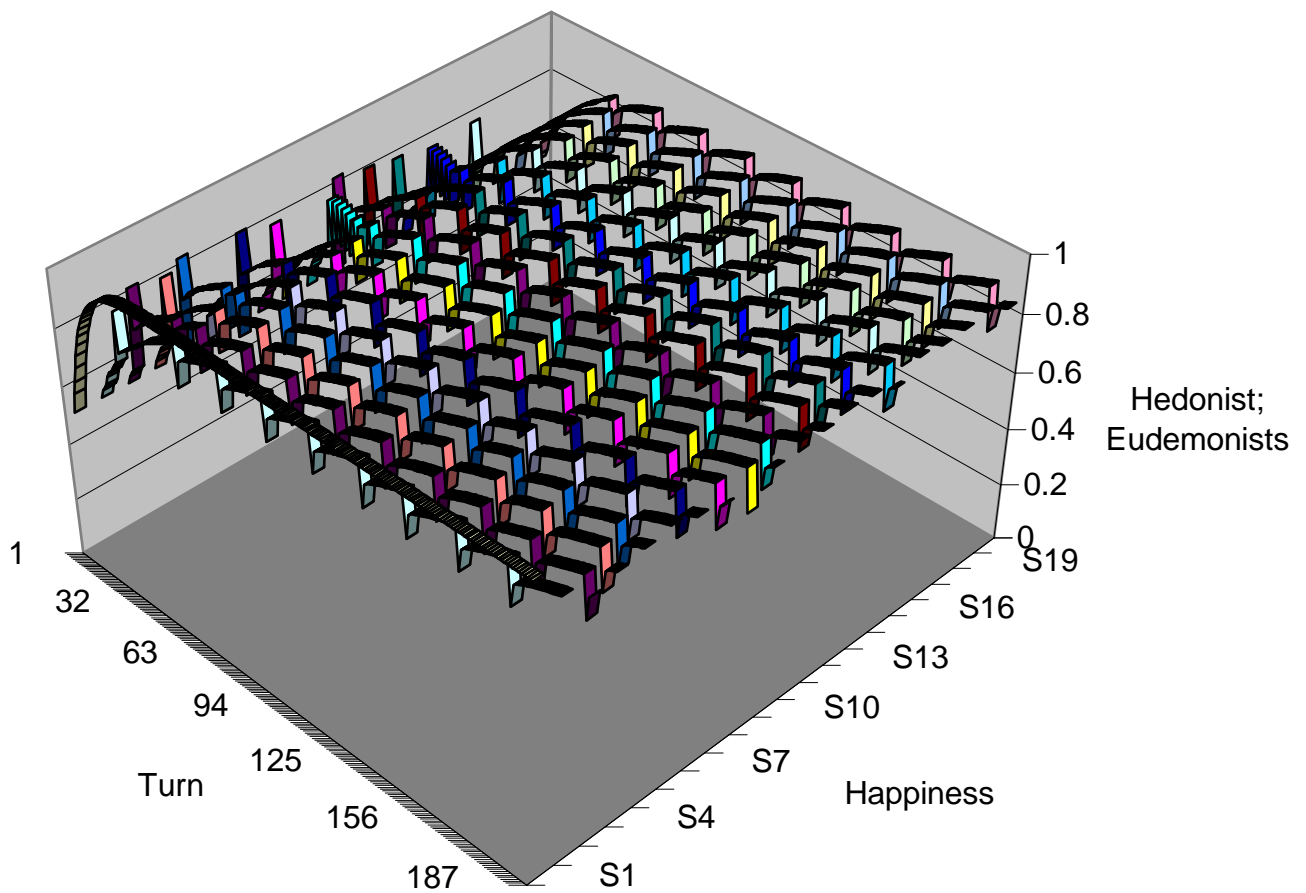


Fig 4.i

Increasing the number of Hedonists – of cheaters – moderates the success of those cheaters.

Two Hedonists in a field of Eudemonists

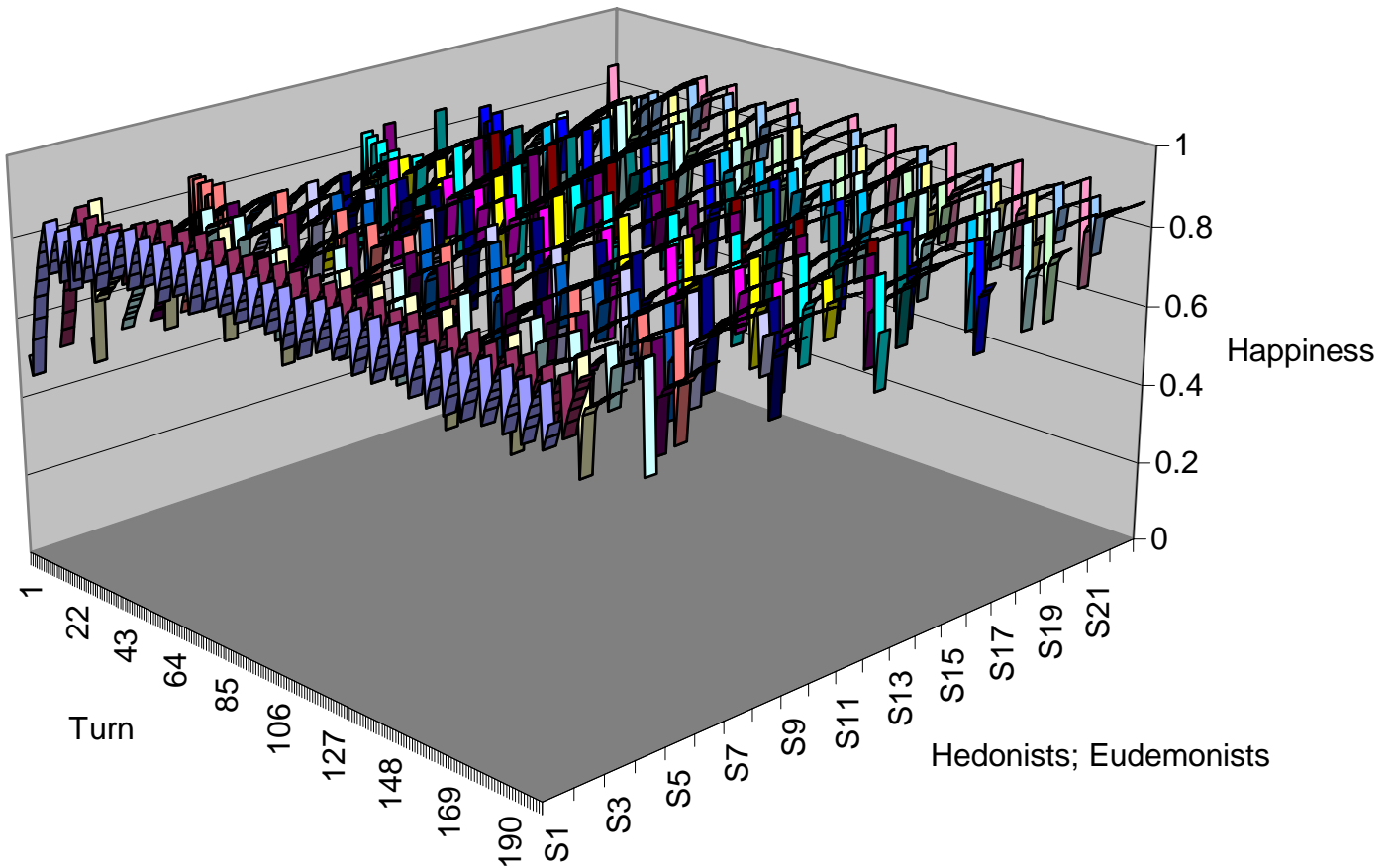


Fig 4.j

Having presented the above examples, the question of their interpretation – or, indeed, to what extent our interpretations have meaning – is left for the Conclusion, and *The arguments against the approach.*

Conclusion

Arguments against the approach

Whether the model is useful in its stated aim – in providing a new perspective on Ethics – is a question that cannot be answered here. However, there are a number of factors that militate against its utility (its value, as defined in **Chapter One**).

That it is a computer model at all is a valid argument against it – an argument valid and beyond the bounds of a computer science dissertation. In taking as exemplars such programs as Tierra and Conway's Game of Life, arguments against computer modelling per se need not be answered. But, while the strength of Tierra, for instance, is that one can draw definite parallels from the assembler mnemonic life-forms to organisms in general, it is less clear how exactly we are to interpret the workings of our program. By *i hinders j*, do we (as **Fig 3.c**) mean that *j* is attacked by *i*, or (as suggested by **Fig 3.d**) just that some advantage has been taken; that *i* has some benefit that would otherwise have gone to *j*. There is a lack of a coherent story with which to motivate and give meaning to the simulations – a lack of insufficient development, though, that can be answered (*vide* the arguments made against the inclusion of Virtue Ethics of **3.3.5**). A more fundamental problem is to do with bias.

There is, in the opening discussion of **Chapter One**, a presupposition of Utilitarianism – there is no truth but utility. This bias is central to the model: the idea that one *could* reduce everything important to people to the single quantity named *happiness* is the very basis of Utilitarianism; and is, indeed, anathema to an Authoritarian position on ethics. It seems that one is compelled to either ignore the Deontological theories (**2.1, 2.4**), or append a caveat to the effect that the *good* of our agents is not available to us. (Perhaps alternative models could be constructed to look at how “good” a Eudemonist is from a Kantian bias.)

There is no clearly defined end to the project: no results there to be sought or obtained, no necessity to answer, no problem to solve. But Artificial Life is a young subject – the Game of Life but 30 years old – and in perceiving Ethics to be a subject amenable to study as part of that field, we hope, in howsoever modest a sense, to have broadened the discipline.

References

- [1] John Conway, the Game of Life
www.reed.edu/~jwalton/gameoflife.html
www.radicaleye.com/lifepage/

For a Turing-machine implemented in the Game of Life:
www.rendell.uk.co/gol/tm.htm
- [2] Thomas Ray, Tierra
www.hip.atr.co.jp/~ray/tierra
- [3] Craig Reynolds, Boids
www.red3d.com/cwr/boids/
www.harry-space.ndirect.co.uk/boids_pro/boidsproj.htm
- [Bentham 1789] Jeremy Bentham; *An Introduction to the Principles of Morals and Legislation*, 1789
- [Bentham 1830-41] Jeremy Bentham; *Constitutional Code*, 1830-41
- [Birkhoff 1933] George Birkhoff; *Aesthetic Measure*; Harvard University Press, 1933
- [Birkhoff 1968] George Birkhoff; *Collected Mathematical Papers. Vol. 3*; New York: Dover Publications, 1968
- [Boden 1996] Margaret Boden; *The philosophy of artificial life*; OUP, 1996.
- [Dawkins 1976] Richard Dawkins; *The Selfish Gene*; Oxford University Press, 1976
- [Dimwiddy 1989] John Dimwiddy; *Bentham*; OUP 1989
- [Gardener 1970] Martin Gardener; *Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life"*; Scientific American 223 (pp. 120 – 123), October 1970
Online at:
acf5.nyu.edu/~mm64/x52.9264/october1970.html
- [Grim 1998] Patrick Grim; *The philosophical computer: essays in philosophical computer modelling*; MIT Press, 1998
- [Kant 1948] Immanuel Kant; *The Moral Law, or, Kant's Groundwork of the Metaphysic of Morals*, trans. H. J. Paton; Hutchinson University Press, 1948
- [Kolakowski 1972] Leszek Kolakowski; *Positivist Philosophy: From Hume to the Vienna Circle*; Pelican, 1972
- [Maynard Smith 1974] Maynard Smith, John; *The theory of games and the evolution of animal conflicts*; Journal of Theoretical Biology 47, 277-349, 1974
- [Mill 1871] John Stuart Mill; *Utilitarianism*; Longmans, Green, Reader, Dyer, 1971
- [Mullender 1993] Sape Mullender; *Distributed Systems*; Addison-Wesley 1993
- [Rist 1969] J. M. Rist; *Stoic Philosophy*; Cambridge, 1969

Appendices

The following rudimentary programs will compile on Java 1.2 or later.

Appendix A:

h.java

(Work-in-progress reflections and to-do notes deleted. The minimal commenting echoes the simplicity of the code here presented: no real O-O – it is all-but written in C, the work of a few days, and chocked with artefacts of aborted ideas.)

```
import java.util.Vector;
import java.io.*;
import java.math.*;

public class h {

    // constants
    final static int HEDONISM = 0, EUDEMONISM = 1,
                    VIRTUEETHICS = 3, TRIVIAL = 4,

                    CONSEQUENTIALISM_HEDONISM = 5,
                    CONSEQUENTIALISM_EUDEMONISM = 6,
                    CONSEQUENTIALISM_VIRTUEETHICS = 7,
                    DEONTOLOGISM_WEAK = 8,
                    DEONTOLOGISM_STRONG = 9,
                    MAXTHEORIES = 10,

                    ASSIST = 0, HINDER = 1,
                    MAXACTIONS = 2,

                    RANDOM = 0, LINEAR = 1;

    final static int HINDER_ONE = 1, HINDER_TWO = 2, HINDER_THREE = 3,
                    HINDER_FOUR = 4,
                    ASSIST_ONE = 1, ASSIST_TWO = 2, ASSIST_THREE = 3,
                    ASSIST_FOUR = 4;

    static int assistnumber = ASSIST_ONE, hindernumber = HINDER_ONE;

    /*
        call java h T[URNS] k
                    LEVEL (set agents' h to:) s (in [0, 1],
                        or,
                        <=-1 is random)
                    H[EDONISM] m
                    E[UDEMONISM] n
                    V[IRTUEETHICS] o
                    C[ONSEQUENTIALISM]H[EDONISM] p
                    C[ONSEQUENTIALISM]E[UDEMONISM] q
                    C[ONSEQUENTIALISM]V[IRTUEETHICS] r
                    D[EONTOLOGISTM]W[EAK] z
                    D[EONTOLOGISTM]S[TRONG] x

                    a[ssist number] v
                    h[inder number] w

        (Agents are displayed on the above order by graph)

    */

    public static void main(String [] args) {

        int turns = 52,
            N = 0;
```



```

double level = 0.5;

int [] numbers = Ancillary.blankintarray(MAXTHEORIES, 0);

for (int n = 0; n < args.length - 1; n++) {
    if (args[n].equals("T"))
        turns = Integer.decode(args[n+1]).intValue();

    else if (args[n].equals("a"))
        assistnumber = Integer.decode(args[n+1]).intValue();

    else if (args[n].equals("h"))
        hindernumber = Integer.decode(args[n+1]).intValue();

    else if (args[n].equals("H"))
        numbers[HEDONISM] =
            Integer.decode(args[n+1]).intValue();

    else if (args[n].equals("E"))
        numbers[EUDEMONISM] =
            Integer.decode(args[n+1]).intValue();

    else if (args[n].equals("V"))
        numbers[VIRTUEETHICS] =
            Integer.decode(args[n+1]).intValue();

    else if (args[n].equals("CH"))
        numbers[CONSEQUENTIALISM_HEDONISM] =
            Integer.decode(args[n+1]).intValue();

    else if (args[n].equals("CE"))
        numbers[CONSEQUENTIALISM_EUDEMONISM] =
            Integer.decode(args[n+1]).intValue();

    else if (args[n].equals("CV"))
        numbers[CONSEQUENTIALISM_VIRTUEETHICS] =
            Integer.decode(args[n+1]).intValue();

    else if (args[n].equals("DW"))
        numbers[DEONTOLOGISM_WEAK] =
            Integer.decode(args[n+1]).intValue();

    else if (args[n].equals("DS"))
        numbers[DEONTOLOGISM_STRONG] =
            Integer.decode(args[n+1]).intValue();

    else if (args[n].equals("L"))
        level = Double.valueOf(args[n+1]).doubleValue();

}

for (int n = 0; n < MAXTHEORIES; n++)
    N += numbers[n];

if (N == 0) { // Use 9 hedonists by default
    N = 9;
    numbers[HEDONISM] = N;
}

double []      h = new double[N],
averageh = new double[N],
totalh = new double[N];

int [] actions = new int[N],
reagents = new int[N],
theories = new int[N];

double [][] agentvirtues = new double[N][MAXACTIONS];

for (int i = 0; i < N; i++)
    for (int a = 0; a < MAXACTIONS; a++)
        agentvirtues[i][a] = 0.5;

String suffix = ""+"N="+N+" T="+turns+" lev="+level+
                " H="+numbers[HEDONISM]+
                " E="+numbers[EUDEMONISM]+
                " V="+numbers[VIRTUEETHICS]+

```

```

        CH="+numbers[CONSEQUENTIALISM_HEDONISM]+
        CE="+numbers[CONSEQUENTIALISM_EUEDEMONISM]+
        CV="+numbers[CONSEQUENTIALISM_VIRTUEETHICS] +
        DW="+numbers[DEONTOLOGISM_WEAK] +
        DS="+numbers[DEONTOLOGISM_STRONG] +
        ass="+assistnumber+" hin="+hindernumber;

System.out.println("Computing "+turns+" turns of "+N+" agents");
System.out.println(suffix);

try {

// note: use .file_separator() rather than '\\'
PrintWriter graphout = new PrintWriter(new
    FileWriter("graphs\\graph-"+suffix+".txt"), true);
PrintWriter actionsout = new PrintWriter(new
    FileWriter("graphs\\actions-"+suffix+".txt"), true);
PrintWriter averagegraphout = new PrintWriter(new
    FileWriter("graphs\\average-"+suffix+".txt"), true);

for (int n = 0; n < N; n++) {
    if (level >= -0.1)
        h[n] = level;
    else    h[n] = Math.random();

    totalh[n] = 0.0;

    actions[n] = -1;
}

int n = 0;
for (int i = 0; i < MAXTHEORIES; i++) {
    for (int j = 0; j < numbers[i]; j++) {
        theories[n++] = i;
    }
}

int hmode = RANDOM;

for (int t = 0; t < turns; t++) {
    // compute averageh
    Ancillary.adddoublearrays(totalh, h);
    Ancillary.divideddoublearray(averageh, totalh, (double) (t + 1));

    // Output data for turn t
    print_state(graphout, h, actions);

    print_averagestate(averagegraphout, averageh, actions);

    //print_state(actionsout, h, actions);
    print_actions(actionsout, h, actions, reagents);

    // decide on our actions
    ethics(h, theories, actions, reagents, agentvirtues);

    // compute
    happiness(hmode, h, actions, reagents);
    System.out.print(" "+t);
}

} catch (IOException e) {}

// and display the f-f-finishing values
System.out.println();
System.out.println("Final state:");
for (int i = 0; i < N; i++)
    System.out.print(""+averageh[i]+'\\t');
System.out.println();
}

```

```

// Our program is in two parts: First, ethics(), which decides the actions of
// our
// agents, then, happiness(), which performs them. Different versions we
// reference
// with switches (not OO cleverness, alas, as the EthicalDiners sported)

static void ethics(double [] h, int [] theories, int [] actions,
                  int [] reagents,
                  double [][] agentvirtues) {
    // act agent-by-agent
    for (int n = 0; n < h.length; n++) {
        doethics(n, h, theories[n], actions, reagents, agentvirtues);
    }
}

static void doethics(int agent, double [] h, int theory, int [] actions,
                   int [] reagents,
                   double [][] agentvirtues) {

    int act = -1, re = -1;
    double best = -1.0, test = -1.0;

    Vector as = new Vector();
    for (int n = 0; n < MAXACTIONS; n++)
        as.addElement(new Integer(n));

    // goes through every action (at random)
    while (as.size() > 0) {
        Integer A = (Integer) as.elementAt(
            Ancillary.random_int(as.size()));
        as.removeElement(A);
        int a = A.intValue();

        Vector js = new Vector();
        for (int n = 0; n < h.length; n++)
            if (n != agent)
                js.addElement(new Integer(n));

        // enumerates every reagent (at random)
        while (js.size() > 0) {
            Integer J = (Integer) js.elementAt(
                Ancillary.random_int(js.size()));
            js.removeElement(J);
            int j = J.intValue();

            double [] imagination = Ancillary.copypdoublearray(h);

            doaction(agent, a, j, imagination);

            if (theory == HEDONISM) {
                test = hedonism(agent, imagination);
            } else if (theory == EUDEMONISM) {
                test = eudemonism(agent, imagination);
            } else if (theory == CONSEQUENTIALISM_HEDONISM) {
                test = consequentialism(agent, imagination,
                    HEDONISM);
            } else if (theory == CONSEQUENTIALISM_EUDEMONISM) {
                test = consequentialism(agent, imagination,
                    EUDEMONISM);
            } else if (theory == VIRTUEETHICS) {
                test = virtueethics(actions[agent],
                    agentvirtues[agent]);
            } else if (theory == DEONTOLOGISM_WEAK) {
                // One example, easy expansion natch
                // give it agent, act, reagent and state
                test = deontologism_weak(agent, a, j, h);
            } else if (theory == DEONTOLOGISM_STRONG) {
                test = deontologism_strong(agent, a, j, h);
            } else

```

```

        System.out.println("doethics: unknown theory error");

        if (test > best) {
            best = test;
            act = a;
            re = j;
        }
    }
    actions[agent] = act;
    reagents[agent] = re;
}

static double hedonism(int agent, double [] h) {
    return h[agent];
}

static double eudemonism(int agent, double [] h) {
    return Ancillary.sumdoublearray(h) / ((double) h.length);
}

static double consequentialism(int agent, double [] h, int metric) {
    double average = 0.0;

    int ticker = 0;

    int [] reagents = Ancillary.blankintarray(h.length, 0);

    int N = h.length;
    int maxa = N;
    for (int i = 0; i < (N - 1); i++)
        maxa *= N;

    int maxb = 1;
    for (int i = 0; i < N; i++)
        maxb *= MAXACTIONS;

    /*
       on max.
       first, all poss is N ^ N. However, most of these are illegal,
       since reagent[i] != i (we cannot act on ourselves). So, take max and
       remove
       1/N, and do it again and again, N times. That is (in calculator speak)
       ans = N^N
       for i = 1 to N
           ans = ans - (ans / N)
       all integer arithmetic, of course (?)
       pictorial proof works (look how the list of numbers is built up)
       show that ans always ends up an integer.
       Looks like gcd, doesn't it?
    */

    for (int i = 0; i < N; i++)
        maxa -= maxa / N;

    for (int i = 0; i < maxa; i++) {
        // add a bean.
        Ancillary.addbeanintarray(reagents, reagents.length, true);

        int [] actions = Ancillary.blankintarray(h.length, 0);
        for (int j = 0; j < maxb - 1; j++) {
            Ancillary.addbeanintarray(actions, MAXACTIONS, false);

            // And here we have a possible world
            // and a state
            double [] possworld = Ancillary.copydoublearray(h);
            happiness(RANDOM, possworld, actions, reagents);

            // Do NOT use consequentialism.
            if (metric == HEDONISM)
                average += hedonism(agent, possworld);
            else if (metric == EUDEMONISM)
                average += eudemonism(agent, possworld);
            else System.out.println("consequentialism: unknown
                metric (hope you're not trying
                to do CONSEQUENTIALISM)");
        }
    }
}

```

```

        ticker++;
    }
    }
    return average / ((double) ticker);
}

static double virtueethics(int action, double [] virtues) {
    if (action < 0)
        return 0.0;
    if (Math.random() <= virtues[action])
        return 1.0;
    else    return 0.0;
}

static double deontologism_weak(int agent, int act, int reagent, double [] h) {
    if (act == ASSIST && (h[agent] > h[reagent]))
        return 1.0;
    else    return 0.0;
}

static double deontologism_strong(int agent, int act, int reagent, double [] h)
{
    if (act == ASSIST && h[agent] > h[reagent]) {
        for (int j = 0; j < h.length; j++) {
            if (j != reagent && j != agent && h[j] < h[reagent])
                return 0.0;
        }
        return 1.0;
    }

    return 0.0;
}

static void happiness(int mode, double [] h, int [] actions, int [] reagents) {
    if (mode == LINEAR) {
        for (int i = 0; i < h.length; i++) {
            doaction(i, actions[i], reagents[i], h);
        }
    } else if (mode == RANDOM) {

        Vector agents = new Vector();
        for (int n = 0; n < h.length; n++)
            agents.addElement(new Integer(n));

        while (agents.size() > 0) {
            Integer I = (Integer) agents.elementAt(
                Ancillary.random_int(agents.size()));
            agents.removeElement(I);
            int i = I.intValue();
            doaction(i, actions[i], reagents[i], h);
        }

    } else System.out.println("happiness: unrecognised mode error");
}

// i = agent, j = reagent by natty convention
static void doaction(int i, int action, int j, double [] h) {
    if (action == ASSIST) {
        assist(i, j, h);

    } else if (action == HINDER) {
        hinder(i, j, h);

    } else System.out.println("doaction: unknown action error "+ action);
}

// assist chooser!
// unpleasant global variables
static void assist(int i, int j, double [] h) {
    if (assistnumber == ASSIST_ONE)
        assist_one(i, j, h);
    else if (assistnumber == ASSIST_TWO)
        assist_two(i, j, h);
    else if (assistnumber == ASSIST_FOUR)
        assist_four(i, j, h);
}

```

```

        else System.out.println("unknown ASSIST_NUMBER with "+assistnumber);
    }

    static void hinder(int i, int j, double [] h) {
        if (hindernumber == HINDER_ONE)
            hinder_one(i, j, h);
        else if (hindernumber == HINDER_TWO)
            hinder_two(i, j, h);
        else if (hindernumber == HINDER_THREE)
            hinder_three(i, j, h);
        else if (hindernumber == HINDER_FOUR)
            hinder_four(i, j, h);

        else System.out.println("unknown HINDER_NUMBER with "+hindernumber);
    }

    static void assist_one(int i, int j, double [] h) {
        double hi = h[i];
        h[i] += (1 - h[i])*(1 - h[j])/8;
        h[j] += (1 - hi)*(1 - h[j])/7;
    }

    static void hinder_one(int i, int j, double [] h) {
        double take = h[j] / 6;
        h[j] -= take;
        h[i] += (1 - h[i]) * take;
    }

    static void assist_two(int i, int j, double [] h) {
        double hi = h[i];
        h[i] += (1.0 - h[i])*(1.0 - h[j]) / 5;
        h[j] += (hi)*(1.0 - h[j]);
    }

    // add modifiers (weakeners) on 2.08.01
    // turns out, without these modifiers you get horrible "ceiling" trouble,
    // wherein the E's go from near-max plateaux to the depths to the plateau
    // again with very little in between. /5 and /6 reduce the effect, and /8 and /9
    // put us in a situation that looks rather like assist_one. With the vanilla and
    // /5 and /6 cases the H's WILL assist. But /8 and /9 and: no. Another reason why
    // it looks familiar.
    // we look for a point at which *both* H's do assist and the ugly plateau behaviour
    // is lost.
    // 5 6 H's (sometimes) assist. 6 6 they don't.
    // there is of course a dynamic going on with hinder.

    static void assist_four(int i, int j, double [] h) {
        double hi = h[i];
        if (h[i] < h[j])
            h[i] *= (1.0 + h[i] - h[j]);
        else h[i] += (1.0 - h[i])*(h[i] - h[j]) / 5.0;
        h[j] += (hi)*(1.0 - h[j]) / 10.0;
    }

    static void hinder_two(int i, int j, double [] h) {
        double hi = h[i];
        h[i] += (1.0 - h[i])*h[j] / 5;
        h[j] *= hi;
    }

    static void hinder_three(int i, int j, double [] h) {
        double hi = h[i];
        h[i] += (1.0 - h[i])*h[j] / 5;
        h[j] *= (1.0 - hi);
    }

    static void hinder_four(int i, int j, double [] h) {
        double hi = h[i];
        h[i] += (1.0 - h[i])*h[j];
        h[j] /= hi + 1.0;
    }

    // ancillary

```

```

static void print_state(PrintWriter out, double [] h, int [] actions) {
    double [] totals = Ancillary.blankdoublearray(MAXACTIONS, 0.0);

    for (int a = 0; a < actions.length; a++)
        if (actions[a] > -1)
            totals[actions[a]] += 1.0;

    for (int a = 0; a < totals.length; a++)
        out.print((" "+ (totals[a] / ((double) h.length) ) +'\t');

    for (int n = 0; n < h.length; n++) {
        out.print((" "+h[n]) + '\t');
    }

    out.println();
}

static void print_averagestate(PrintWriter out, double [] averageh,
    int [] actions) {

    for (int n = 0; n < averageh.length; n++) {
        out.print((" "+averageh[n]) + '\t');
    }

    out.println();
}

static void print_actions(PrintWriter out, double [] h, int [] actions,
    int [] reagents) {

    for (int n = 0; n < h.length; n++)
        out.print(Ancillary.getstring(h[n], 5) + '\t');
    out.println();

    for (int a = 0; a < actions.length; a++) {
        out.print(a + " ");
        if (actions[a] == ASSIST) {
            out.print("as ");
        } else if (actions[a] == HINDER) {
            out.print("hi ");
        } else out.print("?? ");
        out.print(reagents[a]+" \t");
    }
    out.println();
    out.println();
}

}

// useful but dull
import java.util.Vector;
import java.io.*;
import java.math.*;

public class Ancillary {

    static int random_int(int max) {
        int i = (int) Math.floor(Math.random() * ((double) (max)));
        if (i < max)
            return i;
        else { System.out.println("overflow "+i);
            return random_int(max);
        }
    }

    static double [] copydoublearray(double [] a) {
        double [] b = new double[a.length];
        for (int i = 0; i < a.length; i++)
            b[i] = a[i];
        return b;
    }

    static double sumdoublearray(double [] a) {
        double sum = 0.0;
        for (int i = 0; i < a.length; i++)
            sum += a[i];
    }
}

```

```

        return sum;
    }

    static int [] blankintarray(int length, int value) {
        int [] inta = new int[length];
        for (int i = 0; i < inta.length; i++)
            inta[i] = value;
        return inta;
    }

    static double [] blankdoublearray(int length, double value) {
        double [] da = new double[length];
        for (int i = 0; i < da.length; i++)
            da[i] = value;
        return da;
    }

    static void adddoublearrays(double [] a, double [] b) {
        for (int i = 0; i < a.length; i++)
            a[i] += b[i];
    }

    static void divideddoublearray(double [] a, double [] b, double d) {
        for (int i = 0; i < a.length; i++)
            a[i] = b[i] / d;
    }

    static String getstring(double r, int l) {
        if (r < 0.001 && r > 0.0)
            return "<10E3";
        else {
            String str = "+r";
            while (str.length() < (l+1))
                str = str + " ";
            return str.substring(0, l);
        }
    }

    static void addbeanintarray(int [] a, int modulo, boolean yep) {
        int index = 0;
        boolean stop = false;
        while ((stop == false) && (index < (a.length))) {
            stop = addintindex(index, a, modulo);

            index ++;

            // better way to do this
            //boolean go = true;
            //for (int i = start; i <= end && go; i++)
            //    if (a[i] == i

            boolean go = true;
            if (yep) {
                for (int i = 0; i < a.length && go; i++)
                    if (a[i] == i) {
                        go = false;
                        addbeanintarray(a, modulo, yep);
                    }
            }
        }

    static boolean addintindex(int index, int [] a, int modulo) {
        if (a[index] < (modulo - 1)) {
            a[index] += 1;
            return true;
        }
        a[index] = 0;
        return false;
    }
}

```


Appendix B

actions.java

```
import java.util.Vector;
import java.io.*;
import java.math.*;

import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class actions extends Applet {

    Image bufferImage;
    Graphics buffer;

    public static void main(String [] args) {
        actions acts = new actions();
    }

    public void init() {
        Dimension d = getSize();

        bufferImage = this.createImage(d.width, d.height);
        buffer = bufferImage.getGraphics();

        double [] a = new double[2];

        buffer.setColor(new Color(0.8f, 1.0f, 0.8f));
        buffer.fillRect(0, 0, d.width, d.height);

        (Almost no interface – should use, say, tags, or 10 lines of GUI code.)

        graph(buffer, a, 20, 20, 200, 100, HINDER_ONE, 0);
        graph(buffer, a, 320, 20, 200, 100, HINDER_ONE, 1);

        graph(buffer, a, 20, 190, 200, 100, HINDER_TWO, 0);
        graph(buffer, a, 320, 190, 200, 100, HINDER_TWO, 1);

        graph(buffer, a, 20, 360, 200, 100, HINDER_THREE, 0);
        graph(buffer, a, 320, 360, 200, 100, HINDER_THREE, 1);

    }

    // modes
    final int ASSIST_ONE = 0, ASSIST_TWO = 1, ASSIST_THREE = 6,
             ASSIST_FOUR = 7,
             HINDER_ONE = 2,
             HINDER_TWO=3, HINDER_THREE=4, HINDER_FOUR = 5, MAXACTIONS = 2;

    void graph(Graphics g, double [] a, int x, int y, int width, int height,
              int mode, int n) {

        double xfactor = 1.0 / ((double) width),
              yfactor = 1.0 / ((double) height);

        double xcol, ycol;

        for (int j = 0; j <= height; j++) {
            ycol = ((double) j) * yfactor;
            g.setColor(new Color(((float) ycol),
                                ((float) ycol), ((float) ycol)));
            g.fillRect(x, y+j*10, 9, 1);
        }

        for (int i = 0; i <= width; i++) {
            xcol = ((double) i) * xfactor;

            g.setColor(new Color(((float) xcol), ((float) xcol),
```

```

        ((float) xcol));
g.fillRect(x+i+10, y, 1, 9);

for (int j = 0; j <= height; j++) {
    ycol = ((double) j) * yfactor;

    a[0] = xcol;
    a[1] = ycol;

    float d = (float) a[n];

    if (mode == ASSIST_ONE)
        assist_one(0, 1, a);
    else if (mode == ASSIST_TWO)
        assist_two(0, 1, a);
    else if (mode == ASSIST_THREE)
        assist_three(0, 1, a);
    else if (mode == ASSIST_FOUR)
        assist_four(0, 1, a);

    else if (mode == HINDER_ONE)
        hinder_one(0, 1, a);

    else if (mode == HINDER_TWO)
        hinder_two(0, 1, a);
    else if (mode == HINDER_THREE)
        hinder_three(0, 1, a);
    else if (mode == HINDER_FOUR)
        hinder_four(0, 1, a);

    d = d - ((float) a[n]);

    if (d >= 0)
        g.setColor(new Color(1.0f, 1.0f - d, 1.0f - d));
    else
        g.setColor(new Color(1.0f + d, 1.0f + d, 1.0f));

    // draw the graph, b&w
    g.fillRect(x+i+10, y+j+10, 1, 1);
}

}

g.setColor(Color.black);
g.drawString("h.i", x + (width / 2), y - 3);
g.drawString("h", x + (width/2) - 1, y - 3);
g.drawString("h", x + (width/2), y - 4);

g.drawString("h.j", x - 15, y + (height / 2));
g.drawString("h", x - 15, y + (height / 2) - 1);
g.drawString("h", x - 16, y + (height / 2));

String legend = "h'.";
if (n == 0)
    legend += "i";
else
    legend += "j";

g.drawString(legend, x + width + 15, y + height + 25);

g.drawString("1.0", x + width - 10, y - 3);
g.drawString("1.0", x - 20, y + height + 10);

g.drawString("0.0", x - 8, y + 9 );
}

public void update(Graphics g) { paint(g); }

public void paint(Graphics g) {
    g.drawImage(bufferImage, 0, 0, this);
}

}

(Action duplicated in h.java have been deleted here.)
}

```