Adapting Populations of Agents

Philippe De Wilde[1], Maria Chli[1], L. Correia[2], R. Ribeiro[2], P. Mariano[2], V. Abramov[3], and J. Goossenaerts[3]

[1] Intelligent and Interactive Systems Group, Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2BT, United Kingdom, `p.dewilde@ic.ac.uk`,
WWW home page: `http://www.ee.ic.ac.uk/philippe`
[2] Universidade Nova de Lisboa, 2825-115 Monte Caparica, Portugal
[3] Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract. We control a population of interacting software agents. The agents have a strategy, and receive a payoff for executing that strategy. Unsuccessful agents become extinct. We investigate the repercussions of maintaining a diversity of agents. There is often no economic rationale for this. If maintaining diversity is to be successful, i.e. without lowering too much the payoff for the non-endangered strategies, it has to go on forever, because the non-endangered strategies still get a good payoff, so that they continue to thrive, and continue to endanger the endangered strategies. This is not sustainable if the number of endangered ones is of the same order as the number of non-endangered ones. We also discuss niches, islands. Finally, we combine learning as adaptation of individual agents with learning via selection in a population.

1 Populations Of Software Agents

In this paper we study a population of software agents [9] that interact with each other. By drawing an analogy between the evolution of software agents and evolution in nature, we are able to use replicator dynamics [14] as a model. Replicator dynamics, first developed to understand the evolution of animal populations, has recently been used in evolutionary game theory to analyze the dynamical behaviour of agents playing a game. Agents playing a game are a good model of software agents when the latter have to make decisions.

In replicator dynamics, the mechanism of reproduction is linked to the success or utility of the agents in the interaction with other agents. We think that this process also occurs among software agents. This paper adopts this premise, and then goes on to investigate whether it pays off for a population to retain some unsuccessful strategies as an “insurance policy” against changes in the environment.

Each agent is uniquely determined by its code, just as a living organism is determined by its genetic code. For agents, there is no distinction between phenotype and genotype. Consider n different types of agents. At time t, there are $p_i(t)$ agents with code i in the population. Just as an agent is determined by i, a population is determined at time t by $p_i(t), i = 1, \ldots, n$.

The frequency of agent i in the population is

$$x_i(t) = \frac{p_i(t)}{\sum_{i=1}^n p_i(t)}. \tag{1}$$

Abbreviate $\sum_{i=1}^n p_i(t) = p$, where p is the total population. Denote the state of the population of agents by $\mathbf{x}(t) = (x_1(t), \ldots, x_n(t))$.

Now make the following basic assumptions using terminology widely adapted in evolutionary game theory [14].

Assumption 1 (Game)
If agents of type i interact with a population in state $\mathbf{x}$, all agents of type i together receive a payoff $u(e^i, \mathbf{x})$.

Assumption 2 (Replicator Dynamics)
The rate of change of the number of agents of type i is proportional to the number of agents of type i and the total payoff they receive:

$$\dot{p_i}(t) = p_i(t)u(e^i, \mathbf{x}(t)). \tag{2}$$

The proportionality constant in (2) can be absorbed in u. These assumptions are discussed in the rest of this section.

In assumption 1, the code i of an agent is identified with a pure strategy e^i in a game. The notation should not distract the reader, i could have been used instead of e^i. Identification of a code with a strategy is familiar from evolutionary genetics [12]. During replication of individuals in a population, information is transmitted via DNA. It is straightforward to identify the code of an agent with DNA.

Assumption 1, introducing a payoff to software agents, is part of the modelling of software agents as economic agents. Economic principles have been used before in distributed problem solving [8], but in [6] the author has made a start with the analysis of general software agents as economic agents. This paper is part of that project.

The replicator dynamics (2) describe asexual reproduction. Agents do sometimes result out of the combination of code from "parent" agents, but such innovative combinations do not occur very often. On a timescale of one year, the replication of agents will be far more important than the reproduction via combination of parent programs.

In addition to DNA exchange, our species also passes information between individuals via cultural inheritance. This tends to result in behaviour that is a close copy to the behaviour of the "cultural" parent. If agents are to represent humans in an evolving society, they will also exhibit cultural inheritance or social learning, which follows assumption 2 [7].

In biological systems, one can distinguish long term macroevolution [13], and shorter term microevolution [12]. Assumption 2 can be situated in the field of microevolution. On an even shorter timescale, psychologists observe reinforcement learning. Although the results of reinforcement learning are not passed on to offspring (central dogma of neo-Darwinism), it is possible to cast this learning as replicator dynamics [2]. This adds to the credibility of assumption 2, because software agents will often use reinforcement learning together with replication of code [5].

Biological organisms as well as software agents live in an environment. This is actually the same environment, because software agents act for humans, who live in the biological environment. In the model (2), the change of the environment will be reflected in the change of the payoff function $u(e^i, \mathbf{x})$, which has to be written $u(e^i, \mathbf{x}(t), t)$ to make the time dependence explicit. It is very important to be able to model this change in environment, because a successful strategy or agent type should be as robust as possible against changes in environment.

Another type of evolution that software agents have in common with biological agents is mutation. Strategies should be evolutionary stable if they are to survive mutations [12]. However, mutations can positively contribute to the evolution of a system. Current models tend to concentrate on stochastically perturbing the choice of strategies used [7, 3], rather than the random creation of new strategies. Much work still needs to be done in this area.

2 The Burden Of Maintaining Diversity

Agents are pieces of software that act on behalf of humans. Software has a lifetime, so have agents, and humans in a population. The human lifetime is not necessarily the biological lifetime, it may be the time that a human operates in a certain environment. Unsuccessful strategies will die out. This is an essential part of the model (2). Recently there has been much interest in biodiversity [11] and sustainable development [10]. As in all population dynamics, one can ask the question whether it is worth artificially maintaining strategies (or agent types) with a low payoff. The research on biodiversity suggests that this is worthwhile indeed.

An agent type is a number $i \in K = \{1, \ldots, n\}$. The set K_e of endangered agent types is defined by $K_e = \{i \in K | u(e^i, \mathbf{x}) < a\}$, they have a payoff lower than a. The set K_e will change in time, but the threshold a is fixed.

To indicate that the payoffs have been changed, we will use q instead of p for the population, and y instead of x for the frequencies. Assume now that a is the minimum payoff required to sustain a viable population. It is now possible to redistribute the payoff between the non-endangered strategies outside K_e, and the endangered ones in K_e in the following way

$$\begin{aligned} &u(e^i, \mathbf{x}) = a, \quad i \in K_e, \\ &u(e^i, \mathbf{x}) = u(e^i, \mathbf{x}) - \frac{\sum_{j \in K_e} [a - u(e^j, \mathbf{x})]}{q - |K_e|}, \quad i \notin K_e. \end{aligned} \tag{3}$$

This transformation conserves the total payoff

$$\sum_{i \in K} u(e^i, \mathbf{x}).$$

Abbreviate

$$b = \frac{\sum_{j \in K_e} [a - u(e^j, \mathbf{x})]}{q - |K_e|} \tag{4}$$

To derive the differential equations in the state variables y_i, start from (1),

$$q(t)y_i(t) = q_i(t), \tag{5}$$

take the time derivative of left and right hand side to obtain

$$q\dot{y_i} = \dot{q_i} - \dot{q}y_i. \tag{6}$$

Using (3), we obtain, for $i \in K_e$,

$$\begin{aligned} q\dot{y_i} &= q_i a - \sum_{j \in K} \dot{q_j} y_i, \\ &= q_i a - \sum_{j \in K_e} q_j a y_i - \sum_{j \notin K_e} q_j [u(e^j, \mathbf{y}) - b] y_i, \end{aligned} \tag{7}$$

or

$$\dot{y_i} = y_i \left\{ a - \sum_{j \in K_e} y_j a - \sum_{j \notin K_e} y_j [u(e^j, \mathbf{y}) - b] \right\}. \tag{8}$$

Similarly, for $i \notin K_e$, we find

$$\begin{aligned} \dot{y_i} = y_i \{ u(e^i, \mathbf{y}) - b - \sum_{j \in K_e} y_j a \\ - \sum_{j \notin K_e} y_j [u(e^j, \mathbf{y}) - b] \}. \end{aligned} \tag{9}$$

We can now simplify this using

$$\begin{aligned} &- \sum_{j \in K_e} y_j a + \sum_{j \notin K_e} y_j b \\ &= -\frac{|K_e|}{q} a + \frac{q - |K_e|}{q} \frac{\sum_{l \in K_e} [a - u(e^l, \mathbf{x})]}{q - |K_e|}, \\ &= -\frac{1}{q} \sum_{l \in K_e} u(e^l, \mathbf{y}). \end{aligned} \tag{10}$$

This quantity will be much smaller than both a and $u(e^i, \mathbf{y}), i \notin K_e$, the payoff of the non-endangered strategies, if $u(e^i, \mathbf{y}) \ll a, i \in K_e$, or if $|K_e| \ll q$. These plausible assumptions mean, in words, that the conservation value, a, of the payoff is much larger than the payoff of the endangered strategies, or that there are only a small number of endangered strategies. We will give practical examples in the next section.

Hence we can write the population dynamics with conservation of endangered strategies as

$$\dot{y_i} = y_i[a - \sum_{j \notin K_e} y_j u(e^j, \mathbf{y})], \quad i \in K_e,$$

$$\dot{y_i} = y_i[u(e^i, \mathbf{y}) - b - \sum_{j \notin K_e} y_j u(e^j, \mathbf{y})], \quad i \notin K_e. \tag{11}$$

Compare this to the population dynamics without conservation [14],

$$\dot{x_i} = x_i[u(e^i, \mathbf{x}) - \sum_{j \in K} x_j u(e^j, \mathbf{x})], \quad i \in K. \tag{12}$$

The term subtracted from the payoff for strategy i is the population average payoff

$$u(\mathbf{x}, \mathbf{x}) = \sum_{j \in K} x_j u(e^j, \mathbf{x}). \tag{13}$$

These effects of artificially increasing the utility for endangered species are illustrated in figures 1 and 2.

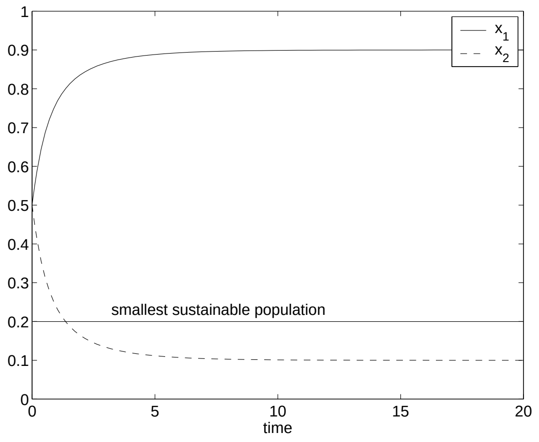


Fig. 1. The evolution of the proportion of two population types, x_1 and x_2, for payoffs $u(e^1, \mathbf{x}) = 5x_2$ and $u(e^2, \mathbf{x}) = 0.5$. This is a situation where the first type depends on the second type to survive. Both types survive, but x_2 is low and can become accidentally extinct.

3 Niches and Islands

Comparing equations (11) and (12) can teach us a lot about the cost and implications of conservation. In the natural world, endangered strategies tend to

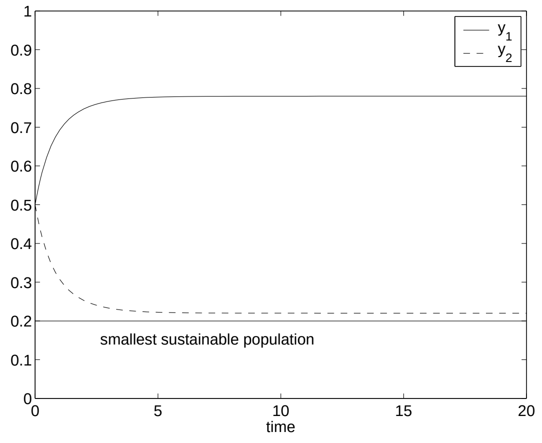


Fig. 2. The evolution of the proportion of two population types, y_1 and y_2, as in figure 1, but the payoffs have now been adjusted according to (3), with $K_e = \{2\}$, and $a = 0.8$. This implies $u(e^1, \mathbf{y}) = 5y_2 - 0.3$ and $u(e^2, \mathbf{y}) = 0.8$. The proportion of type 2 is now higher, and not in danger of extinction. The price to pay is that the proportion of type 1 is now lower.

become extinct, unless a niche can be found for them. A niche is an area in state space where there is little or no competition. We will say that here is a niche in the system if the equations (13) are uncoupled. In that case, the payoff does not depend on the whole state $\mathbf{x}$ anymore, but on a projection of $\mathbf{x}$ on a subspace of the state space.

The existence of a niche prevents strategies from going extinct because it imposes a particular structure on the payoff function u. For a fixed u, there is no particular advantage or disadvantage in the existence of a niche, the replicator dynamics go their way, and that is all. However, the environment can change, and this will induce a change in u, the function modeling the payoff or success of strategies. Certain feedback loops in the dynamics can now become active.

Assume a system with two strategies that each operate in their niche

$$\begin{aligned}\dot{x_1} &= x_1[u(e^1, x_1, \mathbf{z}) - x_1 u(e^1, x_1, \mathbf{z}) \\ &\quad - x_2 u(e^2, x_2, \mathbf{z})], \\ \dot{x_2} &= x_2[u(e^2, x_2, \mathbf{z}) - x_1 u(e^1, x_1, \mathbf{z}) \\ &\quad - x_2 u(e^2, x_2, \mathbf{z})]. \end{aligned} \tag{14}$$

Strategy 1 does not have to compete with strategy 2 because $u(e^1, x_1, \mathbf{z})$ is independent of x_2. Similarly, Strategy 2 does not have to compete with strategy 1 because $u(e^2, x_2, \mathbf{z})$ is independent of x_1. The frequencies x_1, x_2 of the strategies remain non-zero. The frequencies of the other strategies $x_3, \ldots, x_n$ are grouped in the partial state vector $\mathbf{z}$. If the function u changes, the dynamics of the frequencies of strategies 1 and 2 will now in general be

$$\begin{aligned}\dot{x_1} &= x_1[u(e^1, x_1, x_2, \mathbf{z}) - x_1 u(e^1, x_1, x_2, \mathbf{z}) \\ &\quad - x_2 u(e^2, x_1, x_2, \mathbf{z})], \\ \dot{x_2} &= x_2[u(e^2, x_1, x_2, \mathbf{z}) - x_1 u(e^1, x_1, x_2, \mathbf{z}) \\ &\quad - x_2 u(e^2, x_1, x_2, \mathbf{z})]. \end{aligned} \tag{15}$$

It is now possible that there is a positive feedback that causes x_1 and x_2 to increase over time. This positive feedback is not guaranteed, but if one of the strategies had become extinct, then the positive feedback could never have occurred at all when u changed.

Remark that such a positive feedback was already possible in (14), because both equations are coupled via the partial state vector $\mathbf{z}$. We are not concerned with this process here. More complex mechanisms of the benefits of altruism have been studied in [1].

So far the pedestrian justification of conservation: once a strategy is extinct, you cannot benefit from it anymore in the future, if the environment changes. In mathematical terms, once the state space is reduced, many trajectories are excluded, and some could benefit your strategy if the environment changes.

Since Darwin's time, it is known that local optima in a population distribution $\mathbf{x}$ can exist on islands. And recently, we have seen how ending the isolation of islands can destroy the local optimum by the invasion of more successful species. Should we isolate information systems and software agents, so that new types can develop? In that case the replicator dynamics (12) will be altered again. For the evolution on an island, it appears that all species not present on the island are extinct. Call K_r the set of strategies represented on the island. Then the population dynamics is

$$\dot{x_i} = x_i[u(e^i, \mathbf{r}) - u(\mathbf{r}, \mathbf{r})], \quad i \in K_r, \tag{16}$$

where $\mathbf{r}$ is the state vector with zeroes for the strategies not in K_r. When the isolation is ended, these zeroes become non-zero, and we obtain the dynamics

$$\dot{x_i} = x_i[u(e^i, \mathbf{x}) - u(\mathbf{x}, \mathbf{x})], \quad i \in K_r \tag{17}$$

for the strategies on the island. This is illustrated in figure 3. The dynamics (16) and (17) are just the general formulations for the two-strategy niche dynamics described by equations (14) and (15).

4 Learning By Individual Agents

The dynamics of software agents differs in another aspect from that of biological systems. Learned behaviour can be passed on to offspring. Agents can be duplicated, retaining all that was learned, e.g. via a neural network [5]. The replicator dynamic has to take learning into account. If learning is successful, the payoff for states encountered earlier will increase. If the learning also has a generalization

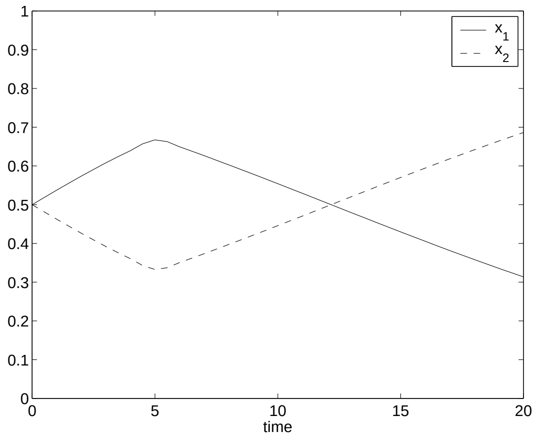


Fig. 3. The evolution of the proportion of two population types, x_1 and x_2, with payoffs $u(e^1, \mathbf{x}) = x_1 x_2$ and $u(e^2, \mathbf{x}) = 0.1$. At $t = 5$, the populations become separated, and the positive feedback maintaining type 1 disappears.

capacity, as happens for neural networks [4], then the payoff for states similar to those encountered earlier will also increase. The payoff now changes explicitly with time, and (12) becomes

$$\dot{x_i} = x_i[u(e^i, \mathbf{x}, t) - u(\mathbf{x}, \mathbf{x}, t)], \quad i \in K. \tag{18}$$

If all the payoffs u were simply multiplied by the same number during learning, the dynamics (18) would be equivalent to (12) in that the orbits remain the same, but they are traversed at a different speed (faster for a constant larger than one). When the payoffs are multiplied by a time-varying factor,

$$\dot{x_i} = x_i \alpha(t)[u(e^i, \mathbf{x}, t) - u(\mathbf{x}, \mathbf{x}, t)], \quad i \in K, \tag{19}$$

the factor $\alpha(t)$ can be absorbed in the time derivative, and the orbits are traversed with a speed varying in time. When the learning factor now described as $\alpha_i(t)$ becomes dependent on the strategy i however, the orbits are changed, and we cannot compare the population evolution with and without learning any more. A non-trivial learning algorithm for a population of 2 different types is illustrated in figure 4.

5 A Population Of Traders

We present here a simulation of a population of traders, under realistic assumptions. We investigate what happens when the payoff (price) is perturbed.

5.1 Scenario

There is an array of N different *resources* in this model. There are M *traders* each one with its capital, its discounting percentage and its available resources.

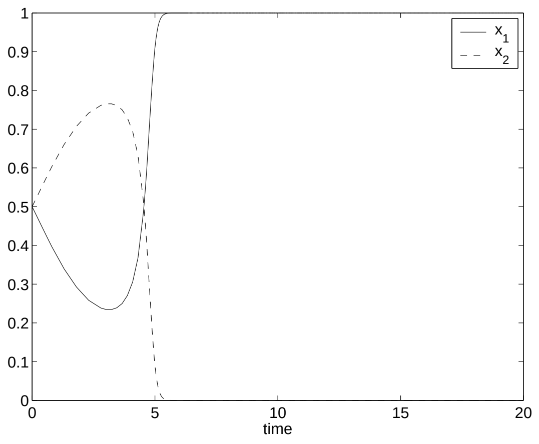


Fig. 4. The evolution of the proportion of two population types, x_1 and x_2. Initially type 1 looses out, x_1 goes down. However, type 1 adapts its utility in time as $u(e^1, \mathbf{x}) = t^2 x_1/3$, and this allows it to win over type 2, that uses no learning, $u(e^2, \mathbf{x}) = x_2$.

All traders are endowed with the same amount of wealth, the amount of each resource given to each trader is randomly calculated. A discounting percentage is used when the trader recalculates its prices. Its function is explained in more detail in the paragraph "Pricing of the Resources" below. Finally, the scenario involves *tasks* that are generated by agents. A task requires some resources and produces some others, loosely modelling generic economic activity.

At each clock tick every trader with its turn issues a task and advertises it to the other traders. Each task carries a random combination of required and produced resources. Every trader gives an offer for the task (provided that they possess the required resources). If the issuer cannot pay for any offer then the task is not executed. Otherwise, it selects the cheapest offer, and the task is executed. The required resources are subtracted from the task executor's set of resources, the produced resources are added to the issuer's set of resources and the issuer pays to the executor an amount of money equal to the price for executing the task.

The prices each trader sets for each resource are different. After the execution of a task all the traders that gave offers for the task recalculate their prices. Only the prices of the resources required for the task are altered on recalculation. There are three ways in which recalculation occurs:

1. The trader whose offer was accepted increases the prices of the required resources of the task as follows:
   ```
   resourcePrices[i] +=
   this.resourcePrices[i] * discountingFactor;
   ```
2. The traders whose offers were not accepted decrease the prices of the required resources of the task as follows:
   ```
   resourcePrices[i] -=
   ```

```
(1 - selectedPrice/myPrice) *
this.resourcePrices[i] * discountingFactor;
```

3. In case no offer was accepted all traders that gave offers for the task decrease the prices of the required resources of the task as follows:
```
resourcePrices[i] -=
(1 - maxPricePayable/myPrice) *
discountingFactor * this.resourcePrices[i];
```

When a trader is sufficiently rich, i.e. its wealth exceeds a certain threshold, it generates a new trader to which it gives half its wealth and half of its resources. The new trader inherits its generator's discounting factor. When a trader's wealth goes below zero it is destroyed.

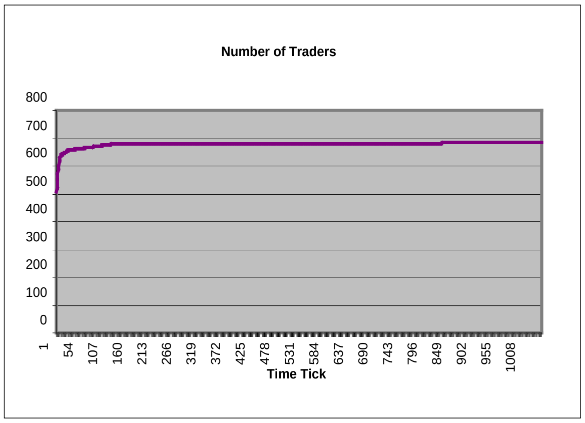
One could say that the system described above is stable when the prices do not rise or fall unexpectedly, or when they do not fluctuate outside some set limits. Also, we would perceive the system as being stable when the traders' number does not increase or decrease too much. Similarly having zero traders, or all prices set to zero are situations where the system is stable. However, we are not interested in these trivial cases, and we would prefer to avoid them.

5.2 Experiments

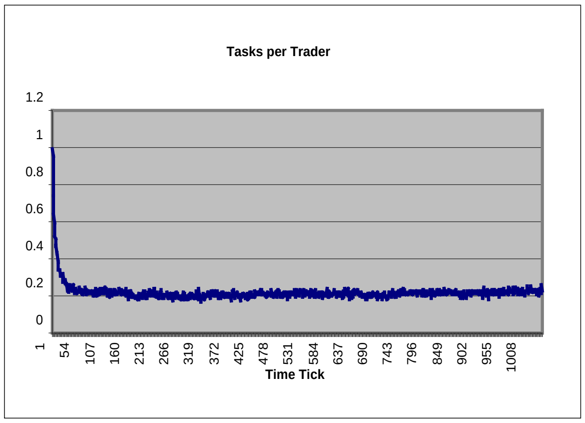
Having in mind the criteria of stability listed above, we now devise metrics of stability for the model. It is of interest to measure the proportion of traders that execute tasks during a time tick. It is also useful to know the prices of the resources in the course of time.

The graphs shown below are for an experiment with 500 traders, 10 different types of resources. The simulation was left running for 10 000 time ticks. Each trader is endowed with EURO 100 000 and a random amount of each of the 10 resources. The amount from each resource it gets is of the order of 1000 (calculated randomly). The prices of resources are initially of the order of EURO 100 (calculated randomly). A trader can generate a new trader if its wealth exceeds EURO 150 000 and it is destroyed if its wealth goes below EURO 0. We only show the first 1000 ticks of the simulation in figure 5 as the rest are more or less similar. We can observe that the number of traders stabilizes after some time. Also the tasks each trader has executed is more or less stable, fluctuating around 0.20 of a task per trader during the course of one time tick. The prices of the resources seem to fluctuate evenly close to EURO 100. The few spikes we observe in this graph are due to traders who have a relatively big discounting factor and increase their prices.

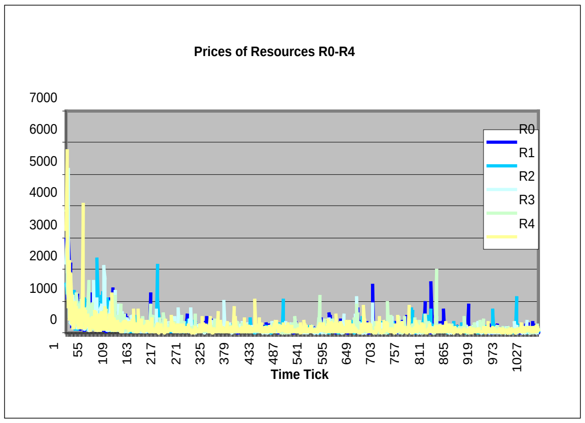
We now try to inject a shock into the system. At time tick 350, the prices of all the resources of each trader are multiplied arbitrarily by 1000. We then allow the simulation to run until time tick 10 000 and see what happens. Again the graphs shown are up to tick 1000. The following graph, figure 6 is a 'zoom in' on the region of time when the shock is injected. We see the prices of five resources rising momentarily at t=350. Then they go into a transient recovery phase slowly converging to a stable state, with prices slightly higher than before.



a.



b.



c.

Fig. 5. Simulation Statistics: a.Number of Traders,b.Tasks per Trader,c.Prices of Resources R0-R4

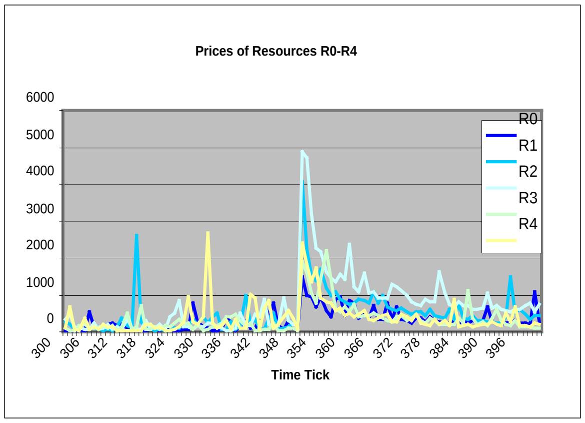


Fig. 6. Prices of Resources R0-R4 (before and after the 'shock')

In another experiment we carried out, figures 7 and 8, a different shock was injected into the system. This time, at time tick 350 the number of traders was decreased by 30%. The system was left to run and here we show what happened up to tick 2000 as the rest is more or less similar.

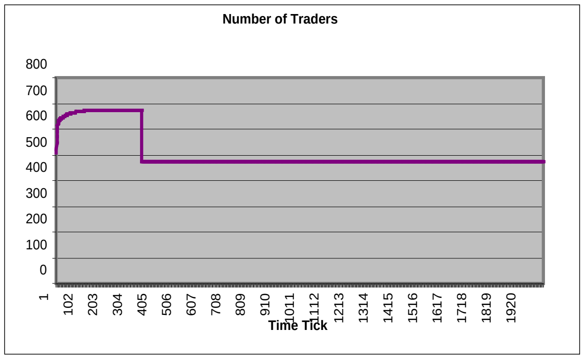


Fig. 7. Number of Traders (before and after the 'shock')

6 Conclusions

All living things contain a code. So do computer programs, languages, designs and artwork. The code consists of all the information that makes replication possible. In a competitive environment, programs are pitched against each other,

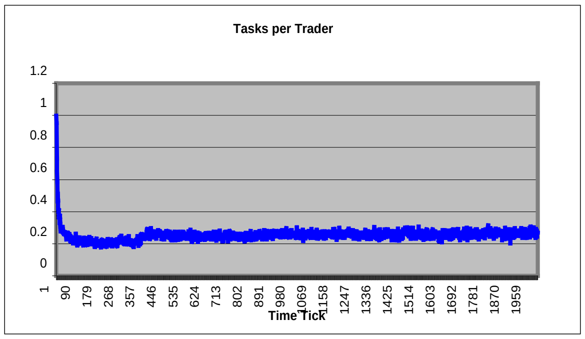


Fig. 8. Tasks per Trader (before and after the 'shock')

in a way similar to individuals in an ecosystem. The interaction brings a payoff u to the program, or language, or design.

The population dynamics with conservation (11) is crucially dependent on the conservation subsidy a per strategy, and on b, which depends on q, the total population, and $|K_e|$, the number of endangered strategies. Conservation maintains a greater pool of strategies than the ecosystem without conservation (12). This makes it possible that the fitness of any single non-endangered strategy could increase when the environment changes adversely for that strategy, via the mutual-benefit feedback loop with an endangered strategy. *The price to pay for this is an overall decrease of the payoff values of the non-endangered strategies.*

In the animal and plant kingdoms, the number of endangered species seems much smaller that the number of non-endangered ones [11], although there is great uncertainty on the numerical data. In this situation, equations (11)-(13) seem to indicate that it is possible to conserve the endangered species, if the effort is spread over all other species. However, replicator dynamics are not such good models of sexual reproduction and mutation, so that it is difficult to reach conclusions.

In the case of languages, artificial and computer, and information systems, the number of endangered types is of the same order as the number of non-endangered ones. In this case, (11)-(15) show that a conservation effort will decrease the payoff of the non-endangered types so much, and their dynamics affected to such an extent, that they also could become extinct if the environment changes.

If conservation is successful, i.e. without lowering too much the payoff for the non-endangered types, it has to go on forever, because the non-endangered types still get a good payoff, so that they continue to thrive, and continue to endanger the endangered types. This is not sustainable if the number of endangered ones is of the same order as the number of non-endangered ones. In other words, *one should not try to control the pure Darwinian evolution in a population of competing agents by artificially maintaining a diversity of agents.* If the number of endangered species is much smaller than the others, they will have little influence on the dynamics of the system, and whether the others sustain them or not will make little difference again.

We have illustrated the evolution of a population of agents using trading agents. We have also shown how robust this population was against perturbations of the payoff.

In short, we have proposed replicator dynamics as a model for the evolution of populations of software agents. We have shown what happens if the utility of some types in increased (conservation), if some types of agents do not interact with each other (niches and islands), and if some types of agents change their utility in time (individual learning). In each of these three cases the adaptation of the population is artificially modified. It is up to the systems analyst to decide which situation applies is a practical case. Our replicator dynamics then allow us to predict what will happen to the different types of agents.

7 Acknowledgements

Partly funded by European Community, under the Future and Emerging Technologies arm of the IST Programme, FET-UIE scheme, Ecology and Evolution of Interacting Infohabitants, IST-1999-10304.

References

1. Gary S. Becker. Altruism, egoism, and genetic fitness: Economics and sociobiology. In *The Economic Approach to Human Behavior*, pages 282–294. University of Chicago Press, Chicago, 1976.
2. T. Borgers and R. Sarin. Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 77:1–14, 1997.
3. Andrea Cavagna, Juan P. Garrahan, Irene Giardina, and David Sherrington. Thermal model for adaptive competition in a market. *Physical Review Letters*, 83:4429–4432, 1999.
4. Philippe De Wilde. *Neural Network Models, second expanded edition.* Springer Verlag, London, 1997.
5. Philippe De Wilde. How soft games can be played. In H.-J. Zimmermann, editor, *EUFIT '99, 7th European Congress on Intelligent Techniques & Soft Computing*, pages FSD–6–12698, Aachen, September 1999. Verlag Mainz.
6. Philippe De Wilde, Hyacinth S. Nwana, and Lyndon C. Lee. Stability, fairness and scalability of multi-agent systems. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 3(2):84–91, 1999.
7. Drew Fudenberg and David K. Levine. *The Theory of Learning in Games.* MIT Press, Cambridge, Massachusetts, 1998.
8. B. A. Huberman, editor. *The Ecology of Computation.* North-Holland, Amsterdam, 1988.
9. Nicholas J. Jennings. On agent-based software engineering. *Artificial Intelligence*, 117:277–296, 2000.
10. John Pezzey. *Economic analysis of sustainable growth and sustainable development.* World Bank, Washington DC, 1989.

11. Andy Purvis and Andy Hector. Getting the measure of biodiversity. *Nature*, 405:212–219, 20.
12. John Maynard Smith. *Evolutionary Genetics.* Oxford University Press, Oxford, 1998.
13. S. M. Stanley. *Macroevolution.* W. H. Freeman, San Fransisco, 1979.
14. Jörgen W. Weibull. *Evolutionary Game Theory.* MIT Press, Cambridge, Massachusetts, 1995.