On the Dynamics of Communication and Cooperation in Artificial Societies

A.E. Eiben M.C. Schut* N. Vink

Artificial Intelligence Section
Department of Computer Science
Vrije Universiteit
De Boelelaan 1081,
1081 HV Amsterdam (NL)
`{gusz,schut}@cs.vu.nl` `nvink@illyan.nl`

Abstract

This paper compares two fundamental building blocks of complex interaction-based systems: communication and cooperation. We investigate the effectiveness of communication in an environment where the need for cooperation is scalable as well as the available resources. Several aspects of communication are considered: firstly, we compare a centralised with a decentralised communication protocol; secondly, we compare a population that always communicates with one where the entities can (evolutionary) learn to communicate. This work is part of a larger project whose main goal is to investigate the emergence of cooperation and communication in response of (scalable) environmental challenges. Our application context is an artificial society, i.e., a simulation of a societal system that was inspired by the classical SUGARSCAPE that embodies a bottom-up approach to investigate complex effects that do not necessarily have complex causes.

Keywords: communication, cooperation, artificial societies, evolutionary learning.

---

*Corresponding author, tel: +31.20.598.7668, fax: +31.20.598.7653.

1 Introduction

The building blocks of a complex system with many active entities are *communication* and *cooperation*. Communication is used for information exchange beetween the entities and cooperation is necessary if the entities want to achieve goals that are beyond their own reach. If we are to design complex artificial systems, we have to know the effects of deciding for available ways to communicate with respect to a given problem environment. This paper presented a first step towards such design questions.

We investigate the communication and cooperation properties of a complex system with many interacting entities among three different dimensions.

- Firstly, we compare i) a *centralised* communication protocol – where individuals multicast messages that can be received by any individual – and ii) a *decentralised* communication protocol – where information is transferred directly between agents without a third party (messageboard, or alike).
- Secondly, we vary the available *resources* and the *cooperation threshold*; the resources are needed by the agents to survive, the cooperation threshold determines how strong the pressure is on the agents to cooperate.
- Thirdly, we parameterise the communication protocols (e.g., probability that an agent talks or listens to other agents) and we empirically compare the implications of i) fixing these parameters and ii) letting the agent learn these parameters themselves.

Our agents are equiped with a hardwired mechanism for communication, and learn (by evolution) to use this mechanism. Our notion of cooperation is rooted in the environment. It is interesting to mention that our approach is complementary to some of the classics. Namely, we study the emergence of communication under fixed properties of cooperation (hard-coding its mechanics), while many studies focus on the emergence of cooperation under fixed properties of communication (see for instance [2], which assumes there is none).

Our experiments are conducted in a straightforward artificial society. This society consists of a collective of agents that lives off harvesting sugar resources in the environment. In some situations, agents may be forced to harvest sugar together with other agents. Each agent is able to communicate information about the amount of sugar at its location and may also receive such information from other agents.

Our main research objective can now be specified as follows: To study the development of the agents communication attitude (talk/listen gene distributions) and cooperative behaviour (eating together) under varying levels of cooperation (maximum amount of sugar they can eat alone).

This paper is organised as follows. In Section 2 we explain the concept of an artificial society, the communication protocols that we researched and some earlier work that we did on this topic. Section 3 describes the system that we designed for carrying out the experiments. Section 4 contains the setup of the experiments and presents the obtained results. Section 5 analyses the results. We conclude and present future work in Section 6.

2 Background

2.1 Artificial Societies

Our research can be positioned in a broader context, that of artificial societies. We let artificial societies be agent-based models of social processes [11]. This definition brings with it some notion of agents (the “people” of the artificial society), simulation (models are computationally executed to explore societal phenomena) and social structures (the macroscopic behaviour of a group of interacting individuals).

Epstein and Axtel [11] let an artificial society consist of 1) agents, 2) an environment or space, and 3) rules. An agent then has internal states and behavioral rules, which each can be fixed or flexible. Interactions and changes of internal states depend on rules of behaviour for the agents and the space. Environments can be abstractly defined (e.g., a communication network) or more resemble our own natural environment (e.g., a lattice of resource-bearing sites). The environment is a medium separate from agents, on which the agents operate and with which they interact. Rules can be defined to describe the behaviour of agents and the environment on different interaction levels, i.e., agent-environment (e.g., agents looking for and consuming food), environment-environment (e.g., growing resources), and agent-agent (e.g., combat and trade).

2.2 Communication

As mentioned, our research compares a centralised with a dentralised communication protocol.

Centralised Communication In the centralised approach, communication between agents is supported by a centralised component; this is a component that is accessible by all agents, e.g., communication through a messageboard. A messageboard enables the agents to communicate by facilitating the storage and retrieval of communicated information. All agents wanting to communicate can access the board to post a message to it, or read messages posted on it.

Decentralised Communication In the decentralised approach, there is no central support for communication. Agents that wish to communicate to other agents need to manually find agents to communicate to and exchange the information with them. Many decentralised communication protocols (e.g., gossip, epidemic-based) have been proposed and researched recently. We have implemented the newscast model [12] in the experiments described below that we explain in more detail.

Newscast Communication

The newscast computing model is a fully distributed information propagation protocol for large-scale peer-to-peer computing [12]. The main idea of newscast is that each agent maintains a cache of information items holding the information for and from the agent; the cache also contains the names of all agents that are ”friends” with the agent. The cache of names, i.e., IDs and addresses, is used each time a communication is initiated by the agent. Each agent can listen and receive the messages from other agents that have it in their cache. At fixed time intervals, the agent updates the information in its cache and the list of names.

The used metaphore for illustrating the newscast model is the concept of a *news agency*. This agency regularly asks all agents for news. Additionally, the agency provides each agent with news about the other agents in the society.

Each agent has a *correspondent module* that maintains a cache of $c > 0$ newsitems, where c is fixed. A news item contains a timestamp, the agent ID and the message itself (location + sugar amount). Agents regularly exchange their caches by following this procedure (where the local agent is the agent who initiates an exchange with a peer agent):

1. Request a fresh news item from the local agent and merge the item into the cache.
2. Randomly select a peer correspondent by considering its ID as found in the cache.
3. Send and receive each other's caches. Merge received items into the local cache.
4. Since the cache now contains $2c + 1$ cache items, the oldest ones are thrown away to keep the c freshest ones (breaking ties randomly).

2.3 Scalable Environments

As mentioned, the work presented here is part of a larger research project investigating the (communication) responses of an artificial society to scalable environmental challenges (here: cooperation). As such, this work extends earlier work [9] where we also compared the centralised messageboard protocol with the decentralised newscast protocol, but where we fixed the cooperation threshold and available resources on single values. Other work [5] investigated the message removal methods for the messageboard protocol. We investigated the following settings: 1) removal after fixed number of iterations, 2) removal after the message has been listened to, and 3) removal after cooperation took place resulting from the message. No significant differences between methods 1) and 2) were found, whereas method 3) consistently resulted in early extinction of the society. In the work reported here, we used removal method 2).

In related work [4] we investigated the agent's learning capabilities to develop *physical* and *mental* properties. We researched both lifetime and evolutionary learning. The results indicate that the evolutionary approach is able to sustain larger and more stable agent populations as well as maintain a higher degree of individual success comparated to the lifetime learning approach. Furthermore, quite unexpectedly, the method used for mental development has a strong effect on the development of the physical features within the very same environment: the individuals bodies evolve to completely different segments of the physical feature space under the two regimes.

In [8] we extended 1) the environment to include a heterogeneous set of resources, and 2) the agent's reproduction mechanism as to where and when reproduction took place. For the first extension, the environment presented here knows only one type of resource (defined in terms of the benefit to the agent utilising this resource), whereas in [8] we researched three different types of resources. For the second extension, agents could reproduce 1) only with their neighbours or with anyone in the environment and 2) during their lifetime or only at the end of their lifetime. The results show that indicate that utilizing reproduction at the end of an agents lifetime and local reproduction (only with neighbours) afforded the agent collective a significantly higher level of performance in its cooperative task.

Finally, in [9] we also present a preliminary theoretical model on the relationship between the decay of value of information that agents act upon and the rate at which agents exchange information with each other. The environments that we investigate turn out to have a very rapid decay of information value: a communicated message may long have lost its value once its reaches its listeners. This puts the results found in these environments in the following perspective: the used communication protocol must be able to remove outdated information fast enough. We later indeed find that the newscast protocol (due to its decentralised nature) is not able to remove outdated information fast enough.

Because our decentralised communication protocol (messageboard) removes information once it has been listened to, this gives it an advantage in these environments already from the outset.

3 System Description

The system in which we conduct our experiments consists of a simulated environment that represents an artificial society (called VUSCAPE) and a set of agents that populates this society.

3.1 JAWAS

The JAWAS[1] simulation platform is comparable with existing social simulation software such as Ascape [13], Repast [6] and Swarm [7]. All settings in JAWAS can be specified either in configuration files or via command line arguments. This enables the user to automate experiments, which substantially speeds up the time needed for, for example, investigating effects of varying experimental parameters (often requiring a large number of runs). Data are automatically saved at specified locations, enabling detailed experimental logging. Thorough statistically based experimental research on complex systems that depend on numerous parameters requires a large number of runs. Facilitating this is one of the main design objectives of JAWAS. It is very easy to add, replace or delete code when changes or extensions to the model need to be implemented. No direct connection is necessary between the program and the graphical user interface. Initial exploration can use the graphical user interface, and for automated experimentation, the system can be run solely with configuration files or from the command line.

3.2 Simulated Environment

The simulated environment is an artificial society called VUSCAPE [5], which is based on SUGARSCAPE [11]. This artificial society concerns a two dimensional grid, wrapped around the edges, where each position corresponds with an area which can contain multiple agents and some amount of sugar. Agents move through the world by vertically or horizontally jumping to another location (cf. moving in SUGARSCAPE). The agents live off the sugar, determining their level of fitness; if an agent's fitness reaches zero, it dies. The major differences between VUScape and SUGARSCAPE concern the implementations of cooperation, communication, explorative behaviour, increased grid-point inhabitance, randomised sugar distribution, and randomised age initialisation. We investigated the effects of these differences experimentally in [5].

Cooperation

Each agent can only harvest a maximum amount of sugar on its own. This amount is set by the maxSugarHarvest (MSH) parameter. If an agent is at a location at which the amount of sugar is over this threshold, it needs other agents to harvest the sugar. If there are more agents at such a location, these agents harvest the sugar together and the sugar is evenly distributed over these agents. In the experiments described below, the cooperation threshold is the same for all agents.

In addition to the MSH parameter, we scale the necessity to cooperate by varying the number of available resources in the environment, called the maximum sugar size (MSS) in VUSCAPE. Based on the settings of MSH (implements the earlier mentioned *cooperation threshold*) and MSS (implements

---

[1] JAWAS: Java Artificial Worlds and Agent Societies, can be downloaded from http://www.cs.vu.nl/ci/eci/.

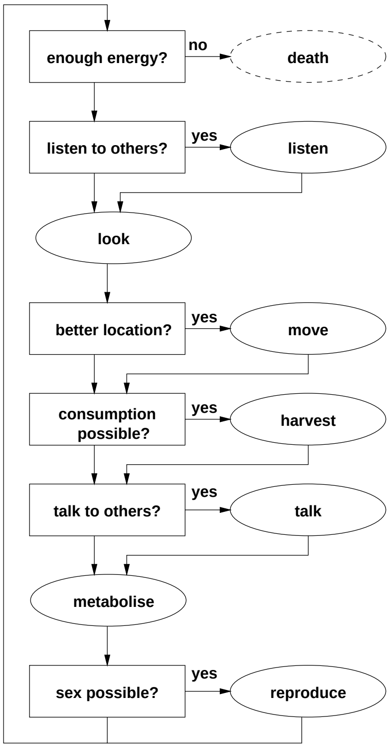


Figure 1: The agent control loop.

the earlier mentioned *available resources*), we can create easier and more difficult environments for the agents to survive.

3.3 Agents

Our agents were based on the classical SUGARSCAPE agent design: prominent features include metabolism, gender, child bearing, death, vision, allow sex and replacement. An agent was able to detect agents and resources for a number of grid-cells determined by its *vision* parameter. It was able to move for a number of grid-cells determined by its *move* parameter.

The control loop of the agent is shown in Figure 1:

- An agent gathers information about the distribution of sugar in the world. This is done by means of listening (to other agents) and looking (at the directly surrounding locations and the current location). Upon completion of this stage, the agent has at its disposal an array of locations and amounts of sugar on these locations.

- Based on this array, the agent makes a decision about its next action. In particular, it chooses a location and moves to this location. This location chosen is always the one containing the largest amount of sugar. If there are more than one such location, the agent chooses a random one.

- Having arrived at the sugar, this sugar is harvested in case the amount is under the cooperation threshold. If the amount is above the cooperation threshold, the agent cooperates immediately if there are more agents at the location. Otherwise, it multicasts (with some probability) to other agents that it needs help.

- If possible, the agent reproduces and generates offspring. For this, it is (at least) necessary that there is another agent of the opposite sex at the location.

Communication

Agents are endowed with talk and listen capabilities. The talk feature determines whether the agent performs a communicative action itself, namely informing other agents of: 1) the amount of sugar that is on its location, and 2) the coordinates of its location. The listen feature is used in the observation and decision making processes of the agent. By listening, the agent receives information from other agents about amounts of sugar at the locations of those agents.

The listen and talk genes express probabilities and are formally real valued numbers between 0 and 1. They are called *listen preference* and *talk preference*, respectively. These probabilities are used in the control loop of the agent - steps “listen to others?” and “talk to others?” in Figure 1. In these steps, a random real is drawn and if the resultant value is under the (talk or listen) preference, the agent listens or talks to others.

Evolution

Agents underwent an evolutionary process of selection and variation. Agents with a high fitness were selected for and variation of agents was accomplished by *crossover* of agent genotypes. The agent genes involved are the talk and listen preferences and the initial amount of sugar. Crossover happens by reproduction of two agents; this is not subject to individual decisions, nor is there any mate selection. If 1) two agents are on the same location or next to each other, 2) the genders differ, 3) their sugar levels are above the reproduction threshold, 4) they are both in the fertile age range, 5) there is a vacant cell in the vicinity for placing the offspring, agents will always mate and generate offspring.

Reproduction takes place by crossover applied to two parents yielding the child, followed by a mutation operator on the child. The value of a gene in a child is the inherited value (from the wealthiest parent) plus a random number drawn from a Gaussian distribution with zero mean and fixed standard deviation σ. The child receives from each of the parents half of their sugar. The child inherits each of the values for vision, age of death, metabolism, and child bearing independently from one of the parents without change. After mating, each agent has a so-called recovery period, which is the number of cycles after mating that an agent cannot mate.

To illustrate the process of reproduction, consider the following example (without mutation). Two agents are next to each other one with 24 sugar units, a listen preference of 0.7 and a talk preference of 0.55; the other has 16 sugar units, a listen preference of 0.6 and a talk preference of 0.5. A child of these two agents gets its listen and talk preferences from the first agent (0.7 and 0.55 respectively). Its initial sugar amount is the sum of half of the sugar amounts of each of the parents, thus 12 from the first parent and 8 from the second parent - its initial sugar amount is thus 20.

4 Experiments

4.1 Setup

The experimental setup is shown in Table 1. We varied four parameters:

maxSugarHarvest (cooperation)	**maxSugarSize (resources)**	**Communication protocol**	**Communication parameters**
$[0 \ldots 10]$	$[1 \ldots 10]$	messageboard newscast	evolutionary fixed

Table 1: An overview of the experimental setup.

- The cooperation threshold - in VUSCAPE this is implemented by the maxSugarHarvest (MSH) (maximum amount of sugar that agent can harvest on its own). We varied the MSH from 0 to 10 in single increments.
- The number of resources - in VUSCAPE this is implemented by the maxSugarSize (MSS) (amount of sugar that is ditributed in the world). We varied the MSS from 1 to 10 in single increments.
- The communication protocol - this was either the centralised *messageboard* protocol or the decentralised *newscast* protocol.
- The communication parameters - these could be either *fixed* (agents always listen and talk) or *evolutionary* (agents learn to use their communication capabilities by evolution).

This makes a total of $11 \times 10 \times 2 \times 2 = 440$ experimental runs. Each run was done 50 times.

An overview of all experimental parameter values is given in Table 2. Additionally, talk and listen features are inherited from the parent with the most sugar. The mutation sigma is 0.1.

4.2 Results

The obtained results have been included in Figures 2, 3 and 4.

Figure 2 shows the results concerning the *performance* of the research populations[2]. Each datapoint corresponds to an environment with a certain need for cooperation. This need is expressed by the number of available resources (maximum sugarsize MSS) and the cooperation threshold (maximum sugar harvest MSH). Each line should be interpreted as follows: in environments on and under the line, all runs resulted in extinction of the agent population. In environments above the line, the agent population size stabilised at some value. (Note that Figure 2 also contains the results of a benchmark experiment that we conducted without communication.)

Figure 3 and 4 show the results of the experimental runs where the communication parameters were evolutionary. Each graph shows, for some given MSH and MSS, the final talk and listen preferences of these runs. In other words, every datapoint represents the average talk preference and average listen preference of all agents of the population at iteration 2000.

Because we used $10 \times 11 = 110$ different combinations of MSH and MSS, it is infeasible to give separate graphs for each environment; we therefore defined *low* (MSH $\in [0-2]$), *medium* (MSH $\in [3-5]$) and *high* (MSH $\in [6-10]$) cooperation thresholds. The ranges have not been chosen

---

[2] This graph is very dense presentation of many experimental runs. Each of the 5 graph lines (no communication, messageboard (fixed), newscast (fixed), messageboard (evolutionary) and newscast (evolutionary)) was extracted from the set of experimental runs covering the complete MSH $\times$ MSS range. These graphs can be found in [15].

Experiment		**Agent**	
numberOfRuns	10	maxSugarHarvest (msh)	**exp**
Scape		singleStep	false
height	50	initAgeZero	false
width	50	minVision	1
runLength	2000	maxVision	1
reseedSugar	true	minSugarMetabolism	1
initialPopulation	400	maxSugarMetabolism	1
sugarSeed.uniqueCell	false	minDeathAge	60
sugarDistributionUnif	3	maxDeathAge	100
sugarGrowBackRate	1.0	sexRecoveryPeriod	0
numberOfSeeds	1000	minReproductionSugar	0
sugarDistributionType	uniform	allowSex	true
maxSugarSize (mss)	**exp**	minInitialSugar	50
Cell		maxInitialSugar	100
allowMultipleAgents	true	preferNearestCell	false

Table 2: An overview of the experimental parameters. Parameters indicated with **exp** are varied in the experiments.

arbitrarily, but are related to the percentage of seeds that can be eaten by the agents when they are alone on a cell. They are chosen such that when the cooperation threshold is low, and when the most sugar is available (MSS = 10), the agents can harvest less than 25% of the seeds without the help of other agents. When the cooperation threshold is medium, the agents can harvest only about 50% of the seeds without the help of other agents. When the cooperation threshold is high, the agents can harvest nearly all seeds without help. Note that the above mentioned percentages are not stable during the simulation; they only apply to the initial situation where all seeds are at their maximum size. (For space restrictions, we did not include the graphs for `mss` = 2, 4, 7, and 9; these can be found in [15].)

5 Analysis

We analyse the results according to how we presented them in the graphs. Firstly, we analyse Figure 2 that says something about the *performance* of the populations. Secondly, we analyse Figures 3 and 4 to find out about the *evolution of communication* (what values did the talk and listen preferences evolve to?) in our experiments.

5.1 Performance

Figure 2 shows the performances of the populations[3] under the investigated communication protocols. Each line in the graph represents a *extinction border* and should be read as such: on all datapoints

---

[3] Our notion of *performance* only takes into account the *population size*. In [15] we present a more elaborate *welfare function*, but extensive testing (reported in [15]) demonstrated no significant difference between the population size and the population welfare for the environments that we investigate here. Hence, this paper only reports on the population sizes.

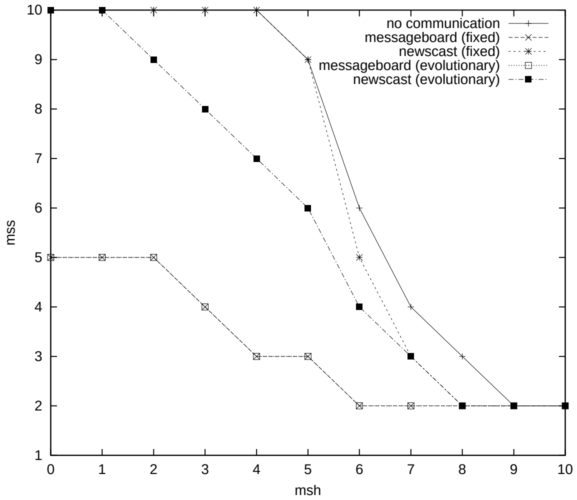


Figure 2: Summary of the performance results of our comparison of protocols. Each datapoint corresponds to an environment with a certain need for cooperation. This need is expressed by the number of available resources (maximum sugarsize mss) and the cooperation threshold (maximum sugar harvest). Each line should be interpreted as follows: in environments on and under the line, all runs resulted in extinction of the agent population.

under and on the graphline the population died out during the runs; on all environments above the line, the population stabilised at some value.

From the data from which we generated Figure 2 (covering the complete MSH × MSS spectrum), we drew the following conclusions. (Note that these conclusions can only in part be observed in Figure 2; we included them here because of their relevance to the findings we make later.) We found that overall the messageboard protocol with fixed communication parameters performs much better than the newscast protocol with fixed communication parameters. Also for the evolutionary variant, we saw that the messageboard protocol outperforms the newscast protocol. However, the difference is smaller here, because the performance of the newscast protocol with evolutionary parameters performs better that with fixed parameters. In general, we saw that the messageboard performs better in the fixed variant than in the evolutionary variant, whereas the newscast protocol performs better in the evolutionary variant than in the fixed variant.

In Figure 2 the graphs show that the messageboard (with evolution) performs best, followed by the newscast protocol with evolution, then by pure newscast and finally no communication, which performs worst. In more detail:

- *Communication gives an advantage to the agents that use it.*
  In Figure 2 we see that communication improves the life expectancy of agents. All protocol variants we used extend the set of environments in which agent populations are viable. There are no environments in which agents using communication go extinct while agents which do not use communication survive; the set of environments in which agents not using communication survive is a true subset of the set of environments in which agents using communication survive.

- *In environments in which agents can survive without using communication, the added value of communication is relatively small.*
  When we only look at environments in which non-communicating societies survive, we see that

adding communication leads to a relatively small increase in the ‘performance’ of the agent population. We contemplate that there exists some minimum population size for the population to survive; this may logically follow from the fact that the probability of reproduction depends on the population size. If the population size is too small, the probability of finding a partner for reproduction may also be too small. Based on our results, we expect the minimum population size necessary for survival to be somewhere close to 500.

5.2 Evolution of Communication

Messageboard (Figure 3)

- *Talk preference evolves to higher values when the need for cooperation increases.*
  In figure 3(a) we see that when the need for cooperation is high (`mss` ≤ 6), the runs in which the agent population survive result in higher average talk preference than when the need for cooperation is lower[4] (e.g. `mss` = 10). We observe the same effect in figure 3(b). The effect is not visible in figure 3(c); this is possibly because the selective pressure on higher talk preference is insignificant when the cooperation threshold is too high.

- *Listen preference evolves to higher values when the need for cooperation increases.*
  What we observed for talk preference also holds for listen preference. When looking at figure 3(c) (high cooperation threshold) we see that as the value for `mss` increases, the surviving runs result in higher average listen preferences. The effect is also visible when comparing the corresponding graphs in figures 3(a), 3(b) and 3(c) to each other; when the cooperation threshold is low, the surviving runs result in higher average listen preferences than when the cooperation threshold is higher.

- *Listen preference evolves to higher values than talk preference.*
  In almost all graphs in figure 3 we see that listen preference evolves to higher values than talk preference. In other words, in almost all graphs we see that most points are below the diagonal from $(0,0)$ to $(1,1)$.

Newscast (Figure 4)

- *Listen preference evolves to* lower *values when the need for cooperation increases.*
  The results for the newscast protocol show that the average listen preference evolves to *lower* values when the need for cooperation increases. This effect is most clearly observable in figure 4(b). Supposedly there exists some optimal level of use of the newscast protocol which the agents should not exceed. These results indicate that this level of use involves an average listen preference of less than 0.5.

- *Talk preference does not evolve consistently.*
  We observe no effects for the evolution of average talk preference in the results of the newscast protocol. The resulting average talk preferences of the surviving agent populations seem to span the entire possible range for average talk preference. The evolution of talk preference observed may be due to genetic drift. Such genetic drift may explain some observation made in Figure 4(c). In the bottom right corners of the graphs in this figure, we see many experiments in which

---

[4] We assume that the need for cooperation is lower when more sugar is available. Although more sugar also means larger seeds, and thus less seeds that can be harvested by single agents, we assume that the need for cooperation in order to survive is lower, because when agents harvest, they harvest more sugar.

nearly all agents talk whenever possible, while hardly any agents listen. We can also observe such results for the messageboard protocol, but they are much more prevalent in the results for the newscast protocol.

- *Listen preference evolves to higher values than talk preference.*
  As for the messageboard protocol, in the results of the newscast protocol we see that listen preference generally evolves to higher values than talk preference. Still, the effect is weaker than in the results of the messageboard protocol because of the earlier mentioned findings.

Summary

In both experiments, we observed that *the listen preference evolves consistently to higher values than talk preference*. This indicates that, for some reason, listening is more ‘important’ to the agents than talking. Although we think that this is because of the induced distribution of talk and listen events, we do not speculate further about what may be this reason - this is subject to future study.

6 Conclusions

The building blocks of a complex system with active entities are communication and cooperation. Communication is used for information exchange beetween the entities and cooperation is necessary if the entities want to achieve goals that are beyond their own reach. If we are to design complex artificial systems, we have to know the effects of deciding for available ways to communicate with respect to a given problem environment. This paper presented a first step towards such design questions.

In a straightforward artificial society, we compared a centralised communication protocol (messageboard) with a decentralised protocol (newscast). The environments we investigated were scalable as we varied the need for cooperation (by including actions to be carried out by multiple agents) and the available resources. In addition we considered hardwired agents (that always communicate) with learning agents (that evolutionary learn to communicate).

The results show that the performance of the messageboard protocol is much better than that of the newscast protocol in the environments that we examined. We observe this with both types of agents. Preliminary further investigation of this result indicates that the ratio between the speed at which information is distributed and the speed at which information looses its value is essential for the success of a given communication protocol [9]. Surprising still, with the newscast protocol, the learning agents outperformed the hardwired agents. Also, with the learning agents, we consistently observe that agents develop a higher preference for listening (receiving information from others) that talking (communicating information to others).

In the long run, we aim to design Emergent Collective Intelligence (ECI). The end goal of ECI research is to combine and exceed achievements in multi-agent systems [1], swarm intelligence [3], and evolutionary computation [10] research via developing synthetic methodologies such that groups of computationally complex agents produce desired emergent collective behaviors resulting from the bottom-up development of certain individual properties and social interactions. Forthcoming results will be published on different scientific forums; for locating them conveniently one can visit: http://www.cs.vu.nl/ci/eci.

References

[1] Avouris, N. and Gasser, L., *Distributed Artificial Intelligence: Theory and Praxis*. Kluwer Academic Publishers. Dordrecht, 1992.

[2] Axelrod, R *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton University Press, New Jersey.

[3] Bonabeau, E., Dorigo, M., Theraulaz, G., *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, Oxford, 1998.

[4] Buresch, T., *Evolutionary versus lifetime learning of agents controllers on a gridworld*, Technical report, Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands, 2004. Available online at http://www.cs.vu.nl/ci/eci/.

[5] Buzing, P.C., Eiben, A.E., and Schut, M.C. *Evolving Agent Societies with VUScape*. In: Banzhaf, W., Christaller, T., Dittrich, P., Kim, J.T., and Ziegler, J., editors. *Advances in Artificial Life, Proceedings of the 7th European Conference on Artificial Life (ECAL 2003)*, Lecture Notes in Artificial Intelligence, volume 2801, Springer Verlag, 2003, pp. 434-441.

[6] Collier, N. *Repast: An extensible framework for agent simulation*. Technical report, Social Science Research Computing, University of Chicago, Chicago, 2000.

[7] Daniels, M. *Integrating simulation technologies with swarm*. In: Proceedings of the Workshop on Agent Simulation: Applications, Models, and Tools, University of Chicago, Chicago, 1999.

[8] Eiben, A.E., Nitschke, G., and Schut, M.C., *Comparison of Reproduction Schemes in an Artificial Society for Cooperative Gathering*. In: Proceedings of the AISB Socially Inspired Computing - Engineering with Social Metaphors Symposium. Technical report. University of Hertforshire. 2004.

[9] Eiben, A.E., Schut, M.C. and Toma, T., *Comparing Multicast and Newscast Communication in Evolving Agent Societies*. In: Beyer, H.-G., et al, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005)*, ACM Press, 2005, pp. 75-82.

[10] Eiben, A. E. and Smith, J. E., *Introduction to Evolutionary Computing*, Springer-Verlag: Berlin, 2003.

[11] Epstein, J., and Axtell, R. *Growing Artificial Societies*, Brookings Institute Press, 1996.

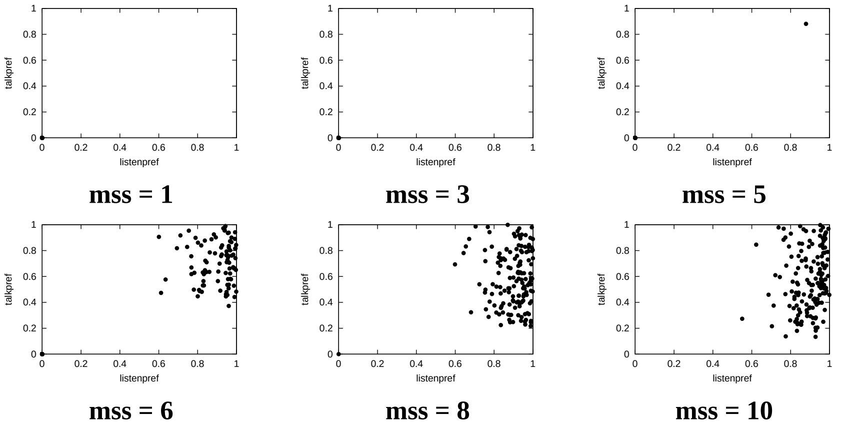
[12] Jelasity, M., and Van Steen, M. *Large-Scale Newscast Computing on the Internet*. Technical Report IR-503, Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands, 2002.

[13] Parker, M. *Ascape: Abstracting complexity*, Technical report, Center on Social and Economic Dynamics, The Brookings Institution, Washington D.C., 2000.

[14] Toma, T. *Communication in Artificial Societies - Effects of Different Communication Protocols in an Artificial Environment*, Technical report, Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands, 2003. Available online at http://www.cs.vu.nl/ci/eci/.

[15] Vink, N. *Exploring Communication Dynamics in VUSCAPE*, Technical report, Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands, 2004. Available online at http://www.cs.vu.nl/ci/eci/.

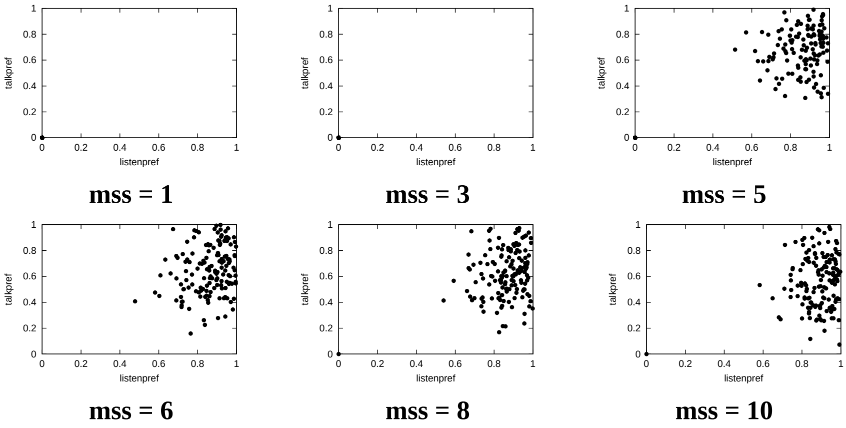
(a) **`maxSugarHarvest`** $\in [0 - 2]$



mss = 1 **mss = 3** **mss = 5**

mss = 6 **mss = 8** **mss = 10**

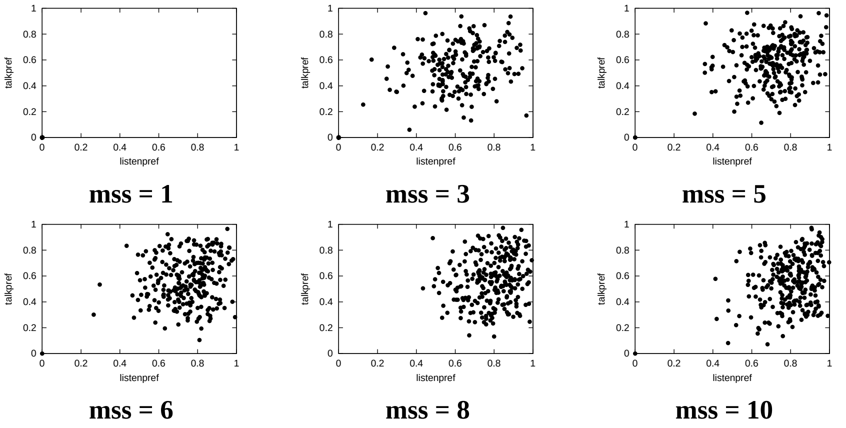
(b) **`maxSugarHarvest`** $\in [3 - 5]$



mss = 1 **mss = 3** **mss = 5**

mss = 6 **mss = 8** **mss = 10**

(c) **`maxSugarHarvest`** $\in [6 - 10]$

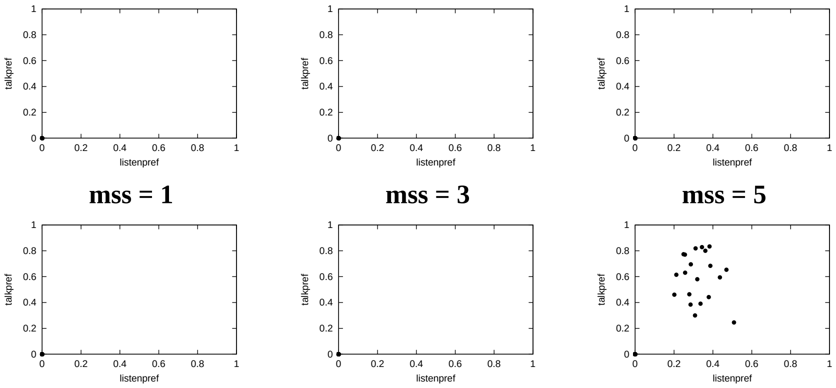


mss = 1 **mss = 3** **mss = 5**

mss = 6 **mss = 8** **mss = 10**

Figure 3: Messageboard results (**mss** = maxSugarSize)

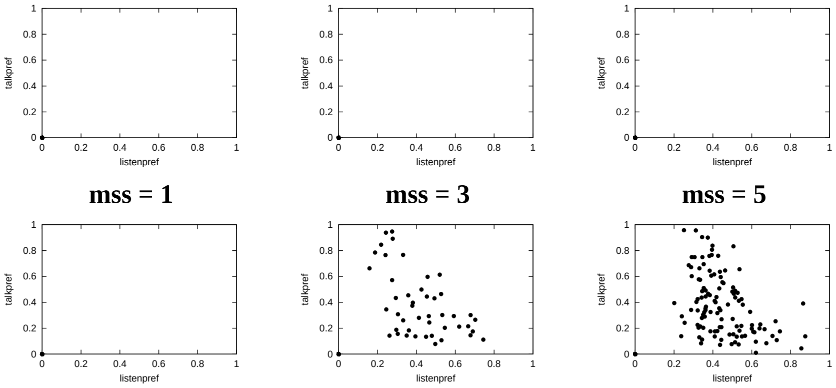
(a) **`maxSugarHarvest`** $\in [0-2]$



mss = 1 **mss = 3** **mss = 5**

mss = 6 **mss = 8** **mss = 10**

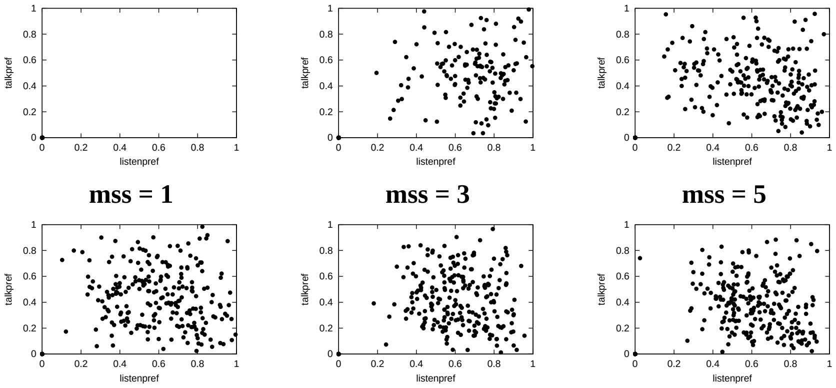
(b) **`maxSugarHarvest`** $\in [3-5]$



mss = 1 **mss = 3** **mss = 5**

mss = 6 **mss = 8** **mss = 10**

(c) **`maxSugarHarvest`** $\in [6-10]$



mss = 1 **mss = 3** **mss = 5**

mss = 6 **mss = 8** **mss = 10**

Figure 4: Newscast results (**mss** = maxSugarSize)