Evolution of Self-Interested Agents: An Experimental Study

Naoki Yamada* and Chiaki Sakama†

Department of Computer and Communication Sciences
Wakayama University, Sakaedani, Wakayama 640-8510, Japan
† sakama@sys.wakayama-u.ac.jp

Abstract. In this paper, we perform an experimental study to examine the evolution of self-interested agents in cooperative agent societies. To this end, we realize a multiagent system in which agents initially behave altruistically by sharing information of food. After generations of a genetic algorithm, we observe the emergence of selfish agents who do not share food information. The experimental results show the process of evolving self-interested agents in resource-restrictive environments, which is observed in nature and in human society.

Keywords: evolution, genetic algorithm, multiagent system, self-interested agents

1 Introduction

In nature, animals communicate in many ways to share information. For example, ants inform each other about the location of food using scent, and sheep alert others about predator attacks by making a bleating sound. On the other hand, animals compete with one another for limited resources, such as food, space and mates. In his book "*The Selfish Gene*" [1], Richard Dawkins says: "Any altruistic system is inherently unstable, because it is open to abuse by selfish individuals, ready to exploit it." Animals are inherently self-interested and often behave dishonestly to have their own benefit. In [6], Searcy and Nowicki argue that "The predominant view nowadays, however, is that selection acts largely at the level of the individual, so that behavior evolves toward what is best for the individual performing the behavior, and not toward what is best for the group. If behavior is commonly selfish, in this sense, then it is not always obvious why animals should exchange information cooperatively. Instead, one might expect many instances in which signalers would attempt to profit individually by conveying dishonest information." Some studies are devoted to modelling evolution of selfish or dishonest behaviors of animals. Wade and Breden [8] provide a population generic model and examine necessary conditions for the spread of genes that determine selfish and cheating behaviors. Sober [7] provides a simple model which explains that lying and credulity are behaviors that evolved by natural selection. Rowell *et al.* [5] develop a game-theoretic model of animal communication in which animals effectively use deceptive signals as strategies.

* Current Address: Shima Seiki MFG., LTD., 85 Sakata Wakayama 641-8511, Japan.

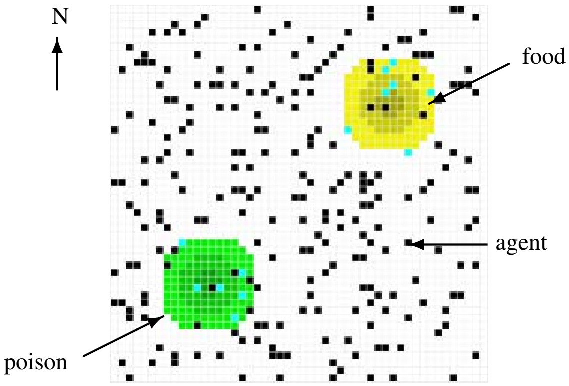

Fig. 1. An environment

Recent studies show the evolution of selfishness in robot communication. Floreano *et al.* [2, 4] show that robots which compete for food learn to conceal food information. In their studies, a group of robots has the task of finding a food source in a field. Once a robot found the food, it stays nearby and emits blue light. This informs other robots of the location and results in overcrowding around the food. After a few generations, robots become more secretive and learn to conceal food information for their own survival. The study shows the possibility of designing artificial agents which would acquire selfish or dishonest attitudes in their environment.

In this study, we consider an environment similar to [2, 4] and observe how self-interested behaviors evolve in an artificial society. In contrast to [2, 4], we do not use robots but realize software agents who can communicate and move in an environment. We implement a multiagent system in which agents initially behave cooperatively by sharing information of food. After generations of a *genetic algorithm*, we observe the evolution of self-interested agents who do not share food information. We analyze experimental results and see the adaptation of agents in a resource-restrictive environment. The rest of this paper is organized as follows. Section 2 presents an agent society which is considered in this paper. Section 3 provides experimental results and considerations. Section 4 discusses related issues and Section 5 concludes the paper.

2 Agent Society

2.1 Agents

We set the *environment* as a two-dimensional grid of 50×50 cells where 250 *agents* are living. Each agent stays at one cell and two different agents cannot stay at a cell at the same time. The environment contains 132 cells of food and 132 cells of poison. The spacial constraints on the food allow a maximum of 132 agents to be fed simultaneously. The maximum amounts of food or poison in each cell are initially given. The range of maximal values is from 3 to 1 in order of the depth of a color (Figure 1). If the amounts of food or poison in each cell become less than its maximum value by consumption,

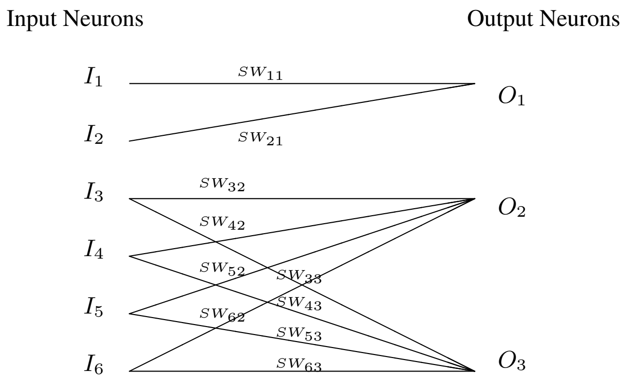

Fig. 2. Neural Network

they are automatically supplemented in every fixed period. Agents act synchronously in discrete time steps. A *generation* consists of 200 time steps. Each agent can move to neighbor cells (horizontally or vertically adjacent cells), send a signal, and obtain food or poison at every step. If an agent stays at a cell where food or poison is located, then it is counted as one at a step. The numbers of food or poison which an agent obtains in one generation are counted.

Each agent can send a signal when they find food or poison. An agent can recognize signals on the $360°$ field and decides a direction to move at the next step based on the amount and direction of signals. Each agent has a simple *neural network* which consists of 6 input neurons $I_i\ (1 \leq i \leq 6)$ and 3 output neurons $O_j\ (1 \leq j \leq 3)$ through 10 synaptic weights SW_{ij} representing the strength of connections between the input neuron I_i and the output neuron O_j (Figure 2).[1]

The value $v(I_i)$ of the input neuron $I_i\ (i = 1, 2)$ is decided by the location of an agent as follows.

$$v(I_1) = \begin{cases} 1 & \text{if there is food on the cell where an agent is staying;} \\ 0 & \text{otherwise} \end{cases}$$

$$v(I_2) = \begin{cases} 1 & \text{if there is poison on the cell where an agent is staying;} \\ 0 & \text{otherwise} \end{cases}$$

The values of I_3, I_4, I_5 and I_6 are decided by the amount of signals which an agent perceives from each direction as follows.

$$v(I_3) = \frac{S_w}{S}, \quad v(I_4) = \frac{S_e}{S}, \quad v(I_5) = \frac{S_n}{S}, \quad v(I_6) = \frac{S_s}{S}. \tag{1}$$

Here, S_w, S_e, S_n and S_s respectively represent the amount of signals from the four sections (west, east, north and south) of $90°$ each and $S = S_w + S_e + S_n + S_s$ (Figure 3).

[1] In [2, 4], a similar but more complicated neural network is used which consists of 11 input neurons connected to 3 output neurons through 33 synaptic weights.

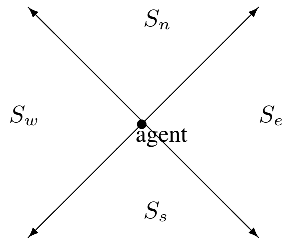

Fig. 3. Signals from four sections

Each agent has a sequence of 80-bit genes $\langle\, g_i \,\rangle_{1 \leq i \leq 80}$ as a binary digit. The values of synaptic weights are calculated by

$$SW_{ij} = \frac{D_{ij} \times 2}{255} - 1 \qquad (2)$$

where D_{ij} represents the decimal number corresponding to 8-bit genes as follows: $D_{11} = (g_1 \ldots g_8)_{10}$, $D_{21} = (g_9 \ldots g_{16})_{10}$, $D_{32} = (g_{17} \ldots g_{24})_{10}$, $D_{33} = (g_{25} \ldots g_{32})_{10}$, $D_{42} = (g_{33} \ldots g_{40})_{10}$, $D_{43} = (g_{41} \ldots g_{48})_{10}$, $D_{52} = (g_{49} \ldots g_{56})_{10}$, $D_{53} = (g_{57} \ldots g_{64})_{10}$, $D_{62} = (g_{65} \ldots g_{72})_{10}$, $D_{63} = (g_{73} \ldots g_{80})_{10}$. For instance, when the 8-bit gene is 00000000 (resp. 11111111), it becomes $D_{ij} = 0$ and $SW_{ij} = -1$ (resp. $D_{ij} = 255$ and $SW_{ij} = 1$) by (2). Thus, a synaptic weight takes a value of $-1 \leq SW_{ij} \leq 1$.

The values of output neurons are computed using the values of input neurons and synaptic weights as follows:

$$O_1 = \tanh\,(\sum_{k=1}^{2} (v(I_k) \times SW_{k\,1})),$$

$$O_n = \tanh\,(\sum_{k=3}^{6} (v(I_k) \times SW_{k\,n})) \;\; (n = 2, 3).$$

O_j is expressed using the hyperbolic function and takes a value between -1 and 1. The values of output neurons are used for deciding action of an agent.

2.2 Action Rules

An agent can take two different actions at each step: sending a signal or moving to neighbor cells. First, an agent sends a signal if the value of the output neuron is $O_1 > 0$. This may happen when an agent finds food ($I_1 > 0$) or poison ($I_2 > 0$), but whether $O_1 > 0$ or not depends on the values of the synaptic weights. If $O_1 \leq 0$, then an agent does not send any signal even if it obtains food or poison. Next, a move of an agent is decided by the values of output neurons O_2 and O_3. A move of an agent is expressed by (dx, dy) where dx (resp. dy) represents a movement in the x-axis (resp. y-axis) direction. Each movement is defined by O_2 and O_3 as follows:

$$dx = \begin{cases} -1 & (O_2 < -\frac{1}{3}) \\ 0 & (-\frac{1}{3} \leq O_2 \leq \frac{1}{3}) \\ 1 & (\frac{1}{3} < O_2) \end{cases} \qquad dy = \begin{cases} 1 & (O_3 < -\frac{1}{3}) \\ 0 & (-\frac{1}{3} \leq O_3 \leq \frac{1}{3}) \\ -1 & (\frac{1}{3} < O_3) \end{cases} \qquad (3)$$

where the values of O_2 and O_3 are divided into three.

We consider a few agents who may not act properly in an environment. This is realized by making an agent move randomly at the probability of 0.2, regardless of the values of output neurons.

The movement of an agent depends on the values of its synaptic weights. As stated before, the synaptic weights of an agent are calculated by a sequence of 80-bit genes of the agent. The initial genes are randomly generated, and then evolve under a fitness condition using the *genetic algorithm*. The fitness of a gene is computed by

$$f = F - P \tag{4}$$

where F and P respectively represent the numbers of food and poison which an agent obtained in one generation. By definition, the fitness increases if an agent obtains more food, while the fitness decreases if an agent obtains more poison. In each generation, agents having higher fitnesses are selected, and their genes are modified (by *crossover* and *mutation*) to form a new generation. The selection is made using the fitness proportionate selection in which the i-th individual is selected based on the probability:

$$p_i = \frac{f_i}{\sum_{k=1}^{n} f_k}$$

where $n = 250$ is the total number of agents and f_i is the fitness of the i-th individual ($1 \leq i \leq 250$). At the end of each generation, the 250 agents are ranked based on their fitness and the best 20% are selected. From these selected agents, two individuals are randomly chosen and paired to perform crossovers and mutations to create a new generation of 250 individuals. Here we use the *uniform crossover* which evaluates each bit in the parent strings for exchange with a probability of 0.5. We also set mutation with a rate of 0.01. The individual which has the highest fitness is also retained in the next generation (*elitism*).

3 Experiments

We perform experiments in three different situations: (i) the field contains food only; (ii) the field contains poison only; and (iii) the field contains both food and poison. In each case, the location of food and poison are fixed; food are located in the north-east corner of the field and poison are located in the south-west corner of the field (Figure 1).

Initially, each agent is assigned a sequence of 80-bit genes which is randomly generated. The initial location of 250 agents is also decided randomly and the evolution of agents is observed in 300 generations. The fitness value of an agent depends on its initial location. To reduce such biases, a group of agents in each generation is produced based on the average of fitness values in 20 trials in which each trial consists of 200 steps of transactions by the same agents. With these settings, we observe the following changes over generations: (i) the number of signals in the field, and (ii) the average values of the synaptic weights among all agents. A snapshot of an experiment is shown in Figure 4 where some (but not all) agents send signals around food.

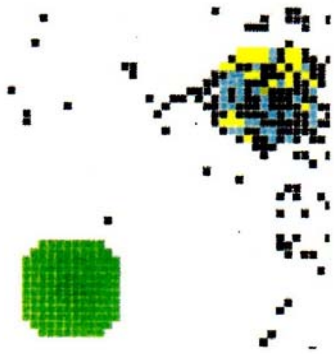

Fig. 4. Snapshot of an experiment

3.1 Food and Signal

We first observe how the number of signals by obtaining food changes in generations. When the field contains food only, the number of signals increases at first and arrives at the peak around the 20th generation (Figure 5). Then, the number of signals suddenly decreases and keeps low values in subsequent generations, mostly less than 5000. This phenomenon is explained as follows. At the initial stage some agents at the location of food send signals which attract other agents. Then the number of agents monotonically increases around the food and the number of signals increases accordingly. In several generations, however, it results in the crowd around food. Those agents who actively send signals around the food cannot obtain food as before (because they cannot move neighbors where other agents stay), which results in the decrease of fitness values of those agents. In contrast, those agents which do not actively send signals would have relatively high fitness values. (Note that some agents would not send signals at food due to the negative synaptic weights given initially.) Then, the probability of selecting agents who actively send signals around the food reduces, which results in the decrease of signaling agents in the next generation. This is observed by Figure 6 in which the average synaptic weight SW_{11} increases at first, while it decreases after the peak around the 20th generation. After the 50th generation, SW_{11} mostly takes negative values which indicates that most agents do not send signals around the food. However, agents who send signals around the food do not die out. This is because reduction of signals has the effect of solving overpopulation around the food, which results in weakening the selection pressure on secretive agents.

When the field contains both food and poison, the number of signals also decreases in Figure 5 but the values sharply oscillate compared with the case of food only. This is because agents around the poison also send signals, which eliminates the effect of decrease of signals in the field. The average synaptic weight SW_{11} also oscillates between positive and negative values in Figure 6.

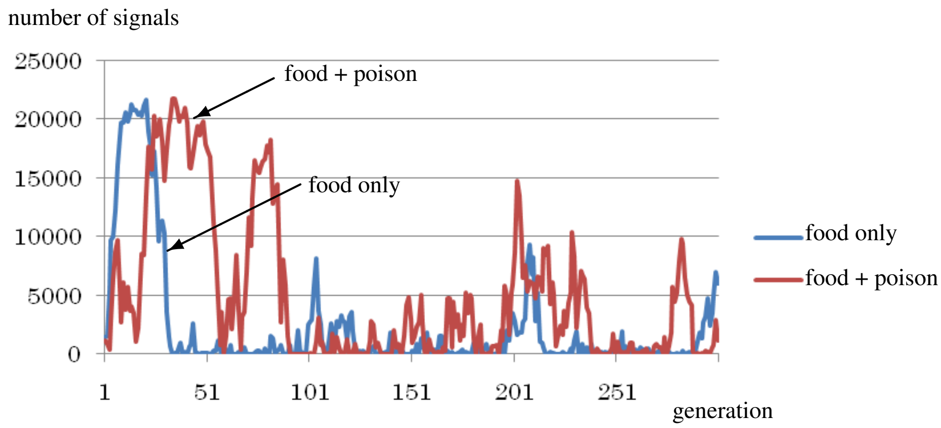

Fig. 5. Number of signals by obtaining food

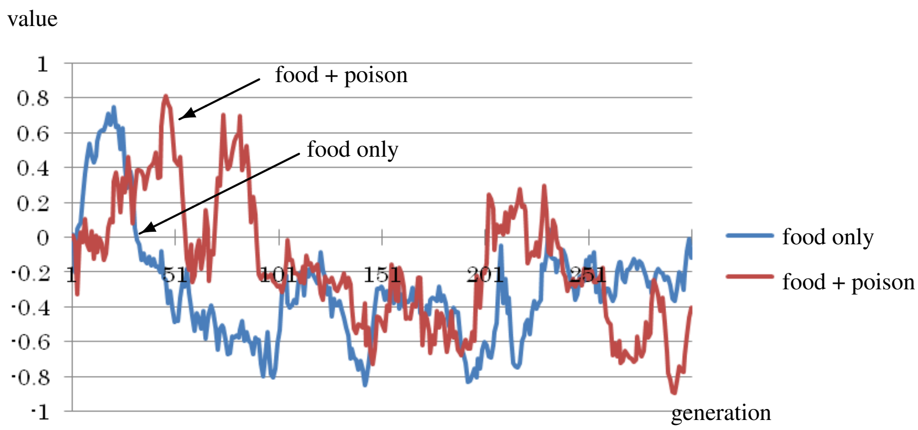

Fig. 6. Evolution of SW_{11}

3.2 Poison and Signal

We next observe how the number of signals by obtaining poison changes in generations. When the field contains poison only, the number of signals is relatively small through generations (Figure 7). The reason is that in this case agents who obtained poison have low fitness values by the equation (4), which results in the evolution that agents are directed away from signals. When the field contains both food and poison, the number of signals randomly oscillates. Such a chaotic behavior is due to the mixture of signals from food and poison. The average synaptic weight SW_{21} also oscillates between positive and negative values (Figure 8). This means that in case of poison stopping signals does not imply any advantage for an agent, which results in no particular evolution of the synaptic weight SW_{21}.

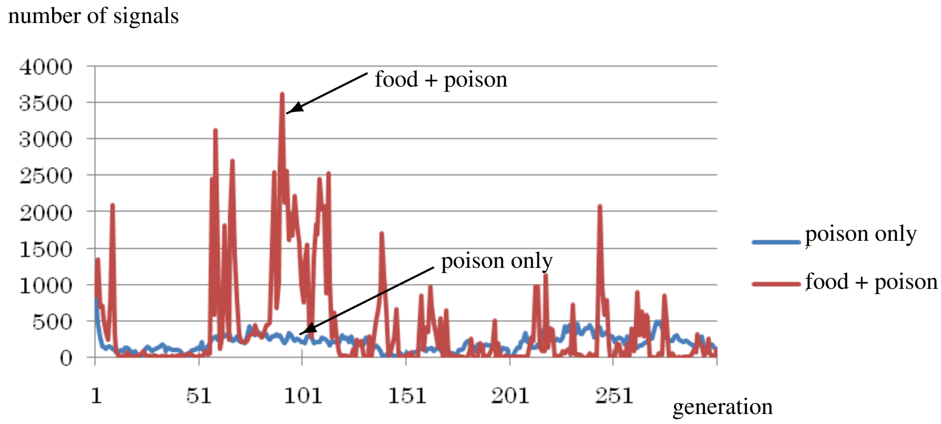

Fig. 7. Number of signals by obtaining poison

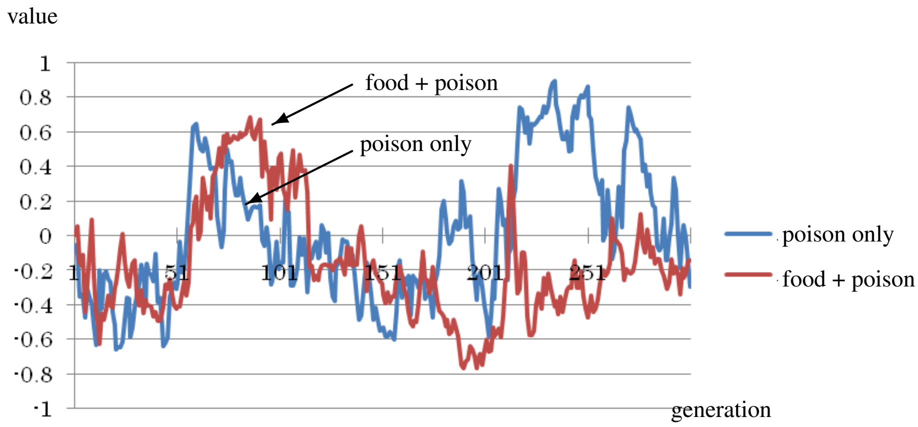

Fig. 8. Evolution of SW_{21}

3.3 Movement

We finally observe how each agent develops genes controlling its movement in generations. Figure 9 shows the evolution of average synaptic weights SW_{x2} and SW_{x3} ($x = 3, 4, 5, 6$) which control output neurons O_2 and O_3, respectively. In the figure, (a) shows the case of food only, (b) shows the case of poison only, and (c) shows the case of food plus poison. Observing (a), synaptic weights are classified into three different classes. The first class takes positive values, the second class takes negative values, and the third class oscillates between positive and negative values. More precisely, SW_{42} and SW_{63} take positive values, while SW_{32} and SW_{53} take negative values. Others oscillate between positive and negative values. The synaptic weight SW_{42} depends on the signals from the east and the synaptic weight SW_{32} depends on the signals from the west (see (1)). Both SW_{42} and SW_{32} control movements in the x-axis direction. The

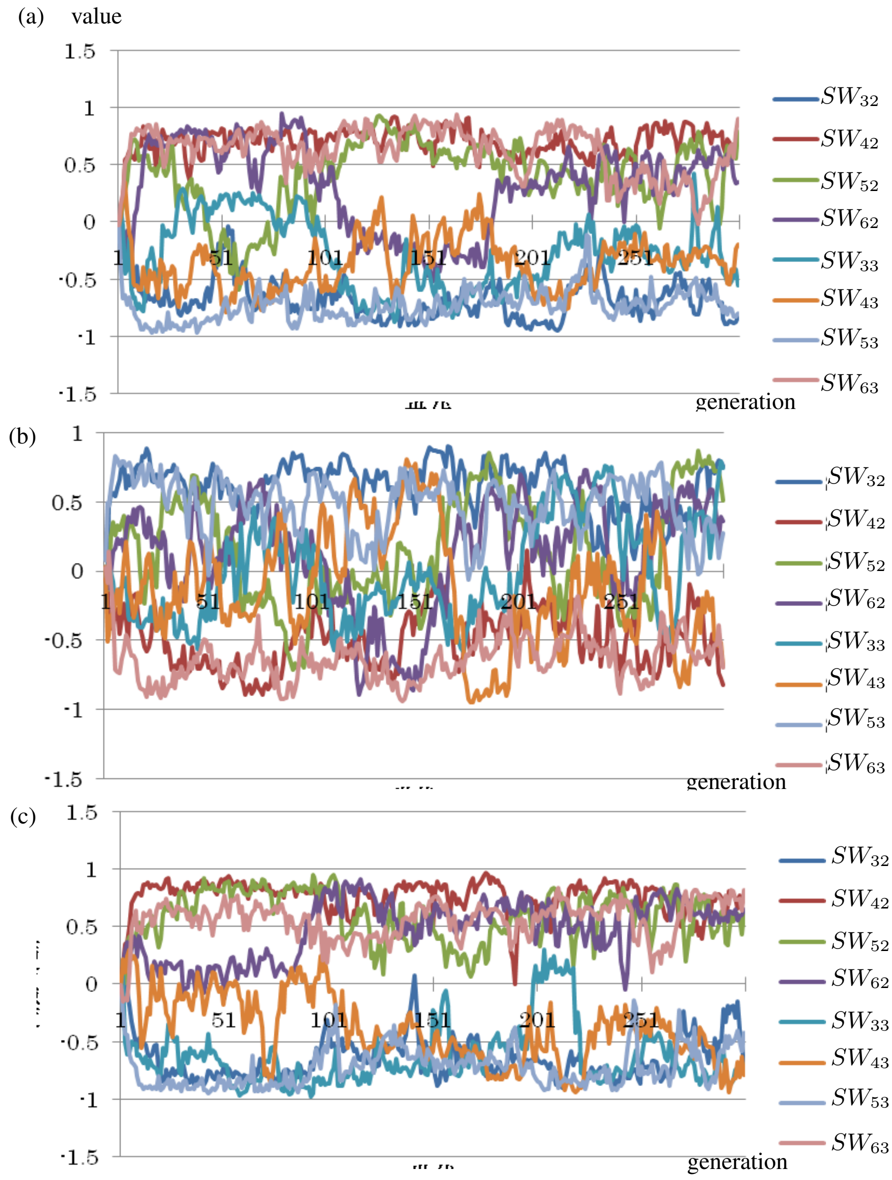

Fig. 9. Evolution of SW_{x2} and SW_{x3}

synaptic weight SW_{53} depends on the signals from the north and the synaptic weight SW_{63} depends on the signals from the south. Both SW_{53} and SW_{63} control movements in the y-axis direction. As a result, synaptic weights which control movements in the x-axis direction from signals from the east and the west have evolved, and synaptic weights which control movements in the y-axis direction from signals from the north and the south have evolved. Moreover, positive SW_{42} contributes to making O_2 positive (hence movement $dx = +1$ in (3)) and negative SW_{53} contributes to making O_3 negative (hence movement $dy = +1$ in (3)). This leads agents toward the food location in the north-east section of the field.

In case of (b), synaptic weights are also classified into three different classes. In contrast to (a), SW_{42} and SW_{63} take negative values, while SW_{32} and SW_{53} take positive values. Thus, the positive-negative patterns are the reverse of those of (a). This is explained that in case of (a) agents are evolved to be attracted to food, while in case of (b) agents are evolved to be kept away from poison.

In (c), synaptic weights, which are oscillated in (a) and (b), converge on either positive or negative values. S_{52} and S_{62} take positive values while S_{33} and S_{43} take negative values. This indicates when there are both food and poison, those synaptic weights play roles for deciding movements. When signals from the north and the south, the positive weights S_{52} and S_{62} make O_2 positive, which will lead an agent to proceed forward the x-axis direction (the east) (cf. (3)). When signals from the east and the west, the negative weights S_{33} and S_{43} make O_3 negative, which will lead an agent to proceed forward the y-axis direction (the north) (cf. (3)). In each case, the weights lead agents close to food and apart from poison.

4 Comparison with Floreano $et\ al.$'s Study

Floreano *et al.* [2, 4] simulate evolution and natural selection in robot learning. In their experiments, robots are randomly placed in an arena containing a food source and a poison source that both emit red light. The food and poison sources are placed at two opposite corners of the arena. The robots earn points for how much time they spent near food as opposed to poison. The robots could produce information by emitting blue light, which other robots could perceive. Each robot has a neural network which consists of 11 input neurons that are connected to a robot's sensors and 3 output neurons that control movement of the robot and the emission of blue light. Each input neuron is connected to every output neuron in terms of 33 synapses whose strength are controlled by a 8-bit gene. Each robot has $33 \times 8 = 264$ bits genome that determines its behavior. With this setting, groups of 10 robots compete for food in separate arenas. After 100 rounds, the robots with the highest scores are selected for the next round. As robots become more efficient at finding and remaining near the food, the concentration of blue light near food also increases. Thus, blue light plays an inadvertent cue providing information on the food location. However, spacial constraints around the food source allow a maximum of 8 robots of 10 to feed simultaneously and result in higher robots density and increased competition and interference near the food. By the 50th generation, robots are selected to decrease the rate of blue light emission. Thus, selection is acting toward suppressing information on the food location.

As addressed in the introduction, our experimental setup is similar to [2, 4], while there are some important differences as follows. First, in their experiment robots emit blue light randomly while it provides inadvertent social information on the food location. Once robots evolve the ability to find food and stay nearby, their increasing density near the food source translates into higher blue density near the food and a source of information for other robots in the arena. On the other hand, in our experiment, agents are initially set to send signals when they find food or poison. Floreano *et al.* observe how unintentional communication develops useful information, which generates self-centered behaviors of robots. By contrast, we observe that agents who behave cooperatively at first also turn to become self-interested in evolution. Second, neural nets used in [2, 4] have 11 input neurons and 3 output neurons.[2] One of the input neurons is devoted to the sensing of food and another one is to the sensing of poison. Other 8 neurons are used for encoding the 360° visual input image, which is divided into four sections of 90° each. For each section, one neuron is used for perceiving blue light, and the other neuron is used for perceiving red light (food or poison). The activation of output neurons is computed as the sum of all inputs multiplied by the 8-bit synaptic weight of the connection and passed through the continuous hyperbolic function. Two of the 3 output neurons are used for controlling the two tracks, where the output value of each neuron gave the direction of rotation and velocity of one of the two tracks. The third output neuron determines whether to emit blue light. Thus the total length of the genetic string of an individual is: (8 bits) $\times$ (11 input neurons) $\times$ (3 output neurons) = 264 bits. In our multiagent model, each agent has a simpler neural network: 6 input neurons and 3 output neurons through 10 synaptic weights, and a 80-bit genetic string in total. Thus, an agent has less than one-third of genes compared with a robot of [2, 4]. With this simplified neurons, we demonstrate an evolution similar to [2, 4].

Thirdly, in the experimental setup, Floreano *et al.* consider the single environment in which both food and poison are located. By contrast, we set up three different environments: the first one contains food only, the second one contains poison only, and the third one contains both food and poison. We observe that such different settings affect the results of evolution of agents. Moreover, we analyze the evolution of synaptic weights in generations which are not reported in [2, 4].

5 Summary

We experimentally realized a multiagent system to observe the evolution of self-interested agents to survive in a resource-restrictive environment. The results show that at the initial stage agents act altruistically to inform others of the location of food, while the increased population around the food results in the increase of self-interested agents who act egoistically to hide food information. Agents react to signals by other agents to obtain useful information, while once they successfully obtain food they evolve into agents who do not always send signals cooperatively. The evolution of self-interested nature of agents from simple action rules would explain a reason for the emergence of selfish behaviors of animals in resource-limited environments in nature.

In this study, self-interested agents appear around food, while further evolution might generate dishonest agents who intentionally send "false" signals to other agents in order to keep them away from food (or even lead them to poison). Such deceptive signals exist in nature [3]. Further refinement of social models is needed to realize the evolution of agents who may act dishonestly.

[2] More precisely, 10 input neurons are used in [2] while 11 input neurons are used in [4]. The role of the additional input neuron is not clearly stated, however.

References

1. Dawkins, R. *The Selfish Gene*. Oxford University Press, 1976.
2. Floreano, D., Mitri, S., Magnenat, S. and Keller, L. Evolutionary conditions for the emergence of communication in robots. *Current Biology* 17(6), pp. 514–519, 2007.
3. Hasson, O. Cheating Signals. *Journal of Theoretical Biology* 167(3), pp. 223–238, 1994.
4. Mitri, S., Floreano, D. and Keller, L. The evolution of information suppression in communicating robots with conflicting interests. In: *Proceedings of National Academy of Sciences 106(37)*, pp. 15786–15790, 2009.
5. Rowell, J. T., Ellner, S. P. and Reeve, H. K. Why animals lie: how dishonesty and belief can coexist in a signaling system. *American Naturalist* 168(6), E180–E204, 2006.
6. Searcy, W. A. and Nowicki, S. *The evolution of animal communication: reliability and deception in signaling systems*. Princeton University Press, Princeton, NJ, 2005.
7. Sober, E. The primacy of truth-telling and the evolution of lying. In: *From a Biological Point of View Essays in Evolutionary Philosophy*, Cambridge Studies in Philosophy and Biology, pp. 71–92, 1994.
8. Wade M. J. and Breden, F. The evolution of cheating and selfish behavior. *Behavioral Ecology and Sociobiology* 7, pp. 167–172, 1980.