Salzburg Interest Group on Integrated Systems

siGis-00x-1999 April 12, 2000

Evolution of Wandering Behavior in a Multi Agent System by Means of a Multi–Objective Fitness Function: An Experiment

Draft

Roland Schwaiger
rschwaig@cosy.sbg.ac.at

Konrad Lang
klang@cosy.sbg.ac.at

Department of Computer Science
University of Salzburg

Correspondence to:

Universität Salzburg
Institut für Computerwissenschaften und Systemanalyse
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

Evolution of Wandering Behavior in a Multi Agent System by Means of a Multi–Objective Fitness Function: An Experiment

Roland Schwaiger and Konrad Lang
Department of Computer Science
University of Salzburg
A–5020 Salzburg, Austria
{rschwaig}, {klang}@cosy.sbg.ac.at

Abstract

In this article we discuss our approach to evolution of wandering behavior in a multi agent system (MAS). Our discussion will cover the various aspects of the system setup, the performed experiment and the interpretation of the results observed. Utilizing a genetic algorithm (GA) with a multi–objective fitness function and multi layer perceptrons we will show how wandering behavior will develop provided a limited number of fitness criteria. Finally we will conclude by reviewing the proposed experiment and point some future directions of research.

Contents

List of Figures

1 Introduction

In this work we give an overview of our approach to evolution of wandering behavior in a multi agent system (MAS) with respect to predefined goals to reach. We draw our motivation for this work from various questions, some of these are:

- What is the maximum number of places an agent may visit in a world populated with other agents?
- Do the agents evolve strategies to navigate in a multi agent neighborhood?
- Will evolution lead to a steering behavior which avoids collisions with other agents?
- Are there any interaction patterns between agents during their course?
- ...

Insights and analyses of the raised questions, give us the means to evolve populations of artificial individuals which show complex behaviors. The system itselfe, based on coarsly defined criteria, determines the course of evolution. Behavior on the MAS level as well as on the individual level will appear as a result and byproduct of achieving the goal.

We are especially interested in simple criteria, which induce socalled wandering behavior in our MAS. By wandering we mean moving in a topologically defined environment. The movement of an agent emerges through the application of selection criteria govering the development of neuro controllers in an evolutionary setup. Wandering behavior implicitely covers other behaviors, s.a. collision avoidance and interaction patterns with other agents. Our goal is to investigate closely some of the byproducts of wandering and wandering itselfe. Henceforth two distinct aspects, namely the quantitavive and the qualitative level ([Fer96] of our simulation will be researched. On the one hand the quantitative level will give insights in the overall performance of the system with respect to the selected criteria. On the other hand, the qualitative level will provide explanations for the individual behaviors of the agents. As a detail aside, we will show that or MAS evolves obstacle avoidance rules at the individual level, as proposed by Mataric [Mat93].

Since our experiments are intended to simulate mass behavior we choose to facilitate reactive agents [Fer96]. This choice will provide the means to analyze the global behavior at the individual level more easily than in a setup with agents with internal states, s.a. the believe desire intention (BDI) approach (e.g. []).

Performance of our system heavily relies on the interaction between the various agents in the world, since no informational cues can be drawn from the space in which the agents move. Wandering governing information is drawn from interaction and observation of agents within a neighborhood of a single agent. We will give the means to compactly represent the neuro controller output where evolved behavior may be interpreted easily at the individual level.

Although our view of agenthood is inspired by artificial life ([Int99]) where the term "agent" is hardly used we preferre to use this term to denote the artificial beeings in our simulations. Utilizing the summary in section 6.3, we may describe the agents in our simulation, from a distributed artificial intelligence perspective (DAI, e.g. []), as

1. Reactive, they sense objects (agents) in their neighborhood and react (com municate and wander) upon them,
2. autonomous, they control their communication and wander behavior,
3. indirect goal–oriented, they react in response to their environment but h ave encoded the means to act goal–oriented with respect to the generation goal,
4. temporally continous, per generation since we investigate a generational b ased evolutionary system contrary to a steady state system,
5. communicative, they communicate with other agents about agents,
6. not learning, they do learn evolutionaryly but do not learn during their l ifetime,
7. mobile, they wander from one location to the next location in their enviro nment,

8. not flexible characters, their actions are scripted (first communicate the n select direction to move).

This work is organized in the following way, In section ?? related publications are reviewed and discussed. Parallels and differences are pointed out. Section 3 outlines the system design with various details on the system components, s.a. agent design, the neuro controller, the encoding of the neuro controller and the evolution of the agents. Finally in section 4 we discuss a number of conducted experiments. The appendix (section 6) includes various topics, specially section 6.3 which gives an overview of the literature concerning the definition of agenthood. Section 6.4 restates the definition of naming games, proposed by L. Steels (e.g. [SM97]) which motivated the defininition of one selection criteria for the evolutionary cycle of our system.

2 Related Work

Miglino et al. [MNT96], evolved artificial neural networks (ANN) to control wandering behavior of small robots. The task was to touch as many squares in a grid as possible. They observed that (a) evolution was an effective means to program control, (b) progress characterized by sharply stepped periods in contrast to periods of stasis and (c) simulated and real robots behaved quite similar.

Reynolds ([Rey87], [Rey91]) gave a solution for the problem of navigating around the world in a life–like and improvisational manner. His proposed solution follows a procedural approach, where a set of basic rules generates the wanted behavior. He defines wandering as a type of random steering and exploration as an exhaustively coverage of a region of space. In that spirit, the agents in our MAS explore their environment. As we will see below, our agents implicitely develop basic rules for wandering in the world but these rules are encoded in a so-called neuro controller output.

Baray [Bar98] developed a system in which agents were able to extend their life spans by coordinating their actions via undirected communication. In his work, he proposed a method through which one can measure the effectiveness of a group's behavior. In his experiments, Baray uses a toroidal world. A homogeneous population of agents populates the world, where each agent has a health value. There are areas in the world which increase or decrease the health. Further mobile predators are part of the environment. All of the agents behaviors are based on rules.

In [Mat93] M.J. Mataric's research considers social interactions leading to purposive group behavior. She states that analyzing and predicting the behavior of a single situated agent is an unsolved problem in robotics and AI. Interactions between individual agents need not be complex to produce complex global consequences. She gives various types of local interactions: collision avoidance, following, dispersion, aggregation, homing and flocking. Inspired by simple basic avoiding behaviors in insects she devises the following avoidance behavior:

```
IF another agent is on the right THEN
     turn left
ELSE
    turn right
ENDIF
```

This behavior takes advantage of the fact that agents are homogeneous. Interestingly our agents evolve exactly the same collision avoidance rule. Contrary to Mataric we do not define the behavior a priori.

Werner and Dyer [WD93] have created a simulated word, called BioLand, where they simulated the evolution of herding behavior in prey animals. Additionally a population of predators where put in the simulation. An evolutionary pathway is seen to this herding, from aggregation, to staying nearby other animals for mating opportunities, to using herding for safety and food finding. They model the behavior of the biots by means of a neural network. Each biot has sensors with sensor units associated with it. The same for motor neurons. Additionally each biot contains three hidden units and higher order gating (axoaxonal) connections. The architecture is allowed to evolve over time, where crossover (one on the average) and mutation (10% chance) are applied at the bit level. Rarely networks with hidden layers are evolved with this encoding thus separate genome which encodes axoaxonal - multiplicative connections that gate neuron-to-neuron connections - connections.

They placed 8000 prey and 8000 predator animals into a 1000 by 1000 toroidal environment. Each genome can encode up to 25 regular and 25 axoaxonal connections. The maximum length of the genome was 400 bits, they say that there are 2^{400} total possible biot neural architectures. They used as steady state approach.

3 Experimental Setup

We conduct our experiments in a finite two dimensional world where the range of the x and y dimension are given, respectively. The region is not constrained, i.e. the agents move on a torus, in other words the world can be written as (e.g. [Dug66])

$$\Pi \cong R/\equiv,$$

where $R = [0,1] \times [0,1]$ is the unitsquare and the relation $\equiv$ is defined as

$$\equiv \quad : \quad \begin{cases} \forall t \in [0,1] & : \quad (t,0) \equiv (t,1); \\ \forall s \in [0,1] & : \quad (0,s) \equiv (1,s); \\ \forall u,v \in R & : \quad (u,v) \equiv (u,v). \end{cases}$$

Time is measured in discrete time steps independent of the system time of the executing hardware device. The world is populated with a predefined number of agents $N_{\mathcal{A}} = 49$ [1], grouped in a population, which live their lifes according to the following phases:

- If the system is started for the first time the positions of the agents $a_i \in \mathcal{A}$ and their neuro controllers are initialized, otherwise the evolved neuro controllers from the previous generation are imported.
- In each step during their life span the agents perform a move according to the output of their neuro controller. An agent may encounter another agent during its course or not. If another agent is met then the agents are engaged in naming games as proposed by L. Steels (e.g. [SM97])[2].
- At the end of the life span of one agent the fitness is determined by means of a multi-objective fitness function (see appendix 6.1)[3]. The various selection criteria, s.a. number of places visited ($\mathcal{C}_{\mathcal{PV}}$) or number of distinct communication partners ($\mathcal{C}_{\mathcal{CP}}$), are used for multi–objective tournament selection for the mating pool. Evolutionary mechanisms produce the offsprings for the next generation which starts its life cycle as described above.

To summarize the overall process:

1. Has the evolutionary process been started for the first time?

   Yes Initialize the position and the neuro controller for each agent in the population randomly.

   No Initialize the position randomly and the neuro controller for each agent in the population with the results of the evolutionary mechanisms of the previous generation.

2. For each simulation step and each agent

   Move The agent moves based on the output of its neuro controller.

   Perform Interaction If an another agent is within the range of the agent sensor then an interaction is initiated, where a communication partner is selected randomly in a certain neighborhood of the agent.

3. After the agents have lived their lifes the following evolutionary mechanisms are applied:

   Determine Fitness The fitness for each agent is determined according to some multi–objective fitness function.

[1] Note: Next experiment dynamic number of agents

[2] Another experiment will utilize short term memory, where lexicon entries are refreshed or forgotten, respectively. Further a long term memory for frequently used words can be envisioned.

[3] In a reinforcment learning experiment the success value has an influence on the adaptation of the connection weights in the neuro controller.

Selection The agents for the mating pool are selected according to some selection method.

Crossover The agents in the mating pool are used for crossover.

Mutation The genotype of the neuro controller is mutated according to a given mutation probability.

4. Store the evolutionarily modified neuro controllers for the next generation. If the maximum number of generations is reached stop the evolutionary process otherwise continue with step 1.

3.1 Our Agent

The agent, as shown in figure 1, is depicted as a triangle with one tip colored. The colored tip indicates the heading direction of the agent. The agent may move according to a finite set of directions, i.e. $\{FRONT, BACK, LEFT, RIGHT\}$ with respect to its local coordinate system. The new position and heading direction of the agent after one move is shown in figure 2. After one step the agent changes its location on the underlying grid and its heading direction accordingly. If there is no way for the agent to perform the move – e.g. if there is another agent on the wanted location – a random direction is generated. Again, it may happen that the wanted locations is occupied. After several unsuccessful tries (four) the agent does not move at all and waits for its next chance to move.

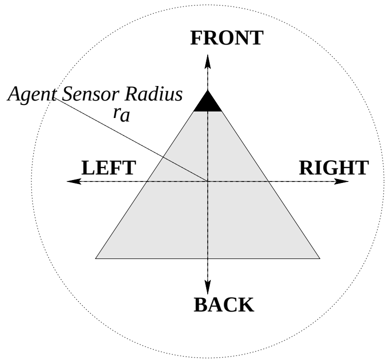


Figure 1: The agent with its sensors and possible moving directions. The agent sensor radius r_a determines the neighborhood for the potential communication partners and the object sensor radius r_o indicates the neighborhood for the potential objects to communicate about. In our case objects are identical to agents thus $r_a = r_o$. The moving directions FRONT, BACK, LEFT and RIGHT are given according to the local coordinate system of the agent.

The agent has the ability to detect other agents (objects) in its agent neighborhood (objects neighborhood) through an agent (object) sensor, where the range of the sensor is depicted by a circle in figure 1 with radius $r_a(r_o)$. In our case $r_a = r_o$. Another agent is selected for communication, where selection probability is normaly distributed over the sensor range[4].

3.2 The Neuro Controller

A neuro controller gives the possible directions for the agent to move through the world. For the motivation and justification to use an artificial neural network to controll the agent see [?]. As seen in figure 3.(a), the neuro controller consists of a multiple layer perceptron as proposed by [ME86] with a winner–takes–all layer at the output. Hence exactly one output unit responds to the input. The input and hidden units transfer their weighted inputs by a logistic function.

Input layer The inputs for the neuro controller are polar coordinates indicating the position of a randomly chosen agent withing the sensor range of the agent sensor[5][6] as shown in figure 3.(b).

[4] What about a Gaussian distributed sensor detection or some others?

[5] Note: Do I need my position as input?

[6] Note: At the moment of writing the radius for agent and object sensor is static but with respect to coevolution of the agent plan it seems reasonable to evolve the radius as well.

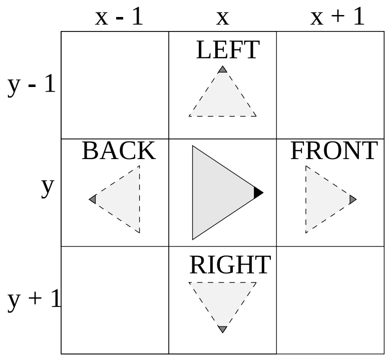


Figure 2: The possible directions of one move of the agent and the new heading direction according to the output of the neuro controller .

If the coordinates of agent a_i are given by $\vec{a_i} = (x_i, y_i)$ and the coordinates of agent a_j are given by $\vec{a_j} = (x_j, y_j)$ then the coordinates of the agent a_j have to be transformed into the local coordinate system of agent a_i.

First the origin of the world coordinate system O is transformed to the origin O' with respect to the translation vector $\vec{t}$. The coordinates of the origin O' are identical to the coordinates of agent a_i with repspect to the world coordinate system. Thus the coordinates of agent a_j with respect to the new origin are

$$\vec{a'_j} = \begin{pmatrix} x'_j \\ y'_j \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_j \\ y_j \end{pmatrix} - \vec{t}$$

where $\vec{t} = \vec{a_i}$. Second the basis of the world coordinate system (e_1, e_2) has to be transformed into the basis of agents a_i local coordinate system (e'_1, e'_2). Then the coordinates of agent a_j with respect to the local coordinate system of a_i are

$$\vec{a''_j} = \begin{pmatrix} x''_j \\ y''_j \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x'_j \\ y'_j \end{pmatrix}$$

where θ denotes the rotation angle of agent a_i with respect to the viewing direction in the world coordinate system.

Combining the two equations above yields

$$\vec{a''_j} = \begin{pmatrix} x''_j \\ y''_j \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_j \\ y_j \end{pmatrix} - \vec{t}$$

Considering the four possible viewing directions of agent a_i and substituting $\vec{t}$ by $\vec{a_i}$, we get

$$\begin{aligned}
\theta = 0 \quad & x''_j = x_j - x_i \\
& y''_j = y_j - y_i \\
\theta = \frac{\pi}{2} \quad & x''_j = y_j - x_i \\
& y''_j = -x_j - y_i \\
\theta = \pi \quad & x''_j = -x_j - x_i \\
& y''_j = -y_j - y_i \\
\theta = \frac{3\pi}{2} \quad & x''_j = -y_j - x_i \\
& y''_j = x_j - y_i
\end{aligned}$$

In order to calculate the polar coordinate of agent a_j we write

$$
\begin{aligned}
r' &= \sqrt{(x_j'')^2 + (y_j'')^2} \\
\phi' &= \arctan\left(\frac{y_j''}{x_j''}\right)
\end{aligned}
$$

The range of angle is $\phi' \in (-1, 1)$ and the radius $r' \in (0, r_a]$, where r_a denotes the agent sensor radius. The radius r' and the angle ϕ' are transformed in $(0, 1]$ and $(0, 1)$, respectively, in order to balance the influence of the angle and the radius on the output of the neuro controller, i.e.

$$
r = \frac{r'}{r_a}
$$

and

$$
\phi = \frac{\phi'}{2} + \frac{1}{2}
$$

If no other agent is within the sensor range of the agent sensor then the neuro controller receives the input $(0, 0)$.

Hidden Layer(s) The topology of the neuro controller especially the number of hidden units and their incoming and outgoing connections, is not static. Through the encoding of the neuro controller these parameters are determined evolutionaryly.

Output Layer Four possible outcomes are represented at the output layer - FRONT, BACK, LEFT and RIGHT - where the actual outputs observed at the output layer indicate the steering direction. The output layer is constructed as a winner–takes–all layer ([Hay94] , i.e. the output neuron with the maximum output is the only responding neuron[7].

In figure 4 the output of the neuro controller is shown. For each possible position of an agent a_j within r_a of a_i the output of the neuro controller indicates the steering and heading direction of the agent in its local coordinate system. For example assume that agent a_j was found at position $(x+1, y+1)$ and selected for communication then the output of the neuro controller is RIGHT.

Evolution gives us the means to dynamically change and adopt the topology, the weights and biases of the neuro controller to the problem domain.

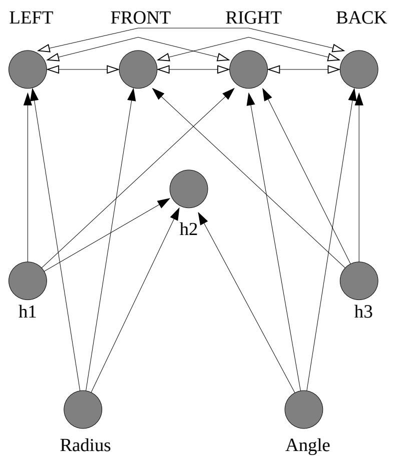
3.3 Encoding of the Neuro Controller

The ANN's genetic blueprint is based on a direct encoding suggested by Miller et al. [MTH89]. With this approach each connection is represented in a binary adjacency matrix called *Miller–Matrix* (MM) describing ANN architecture. Contrary to the original crossover operator exchanging rows and columns of the MM, standard 2–point crossover operating on a linearized MM is used where the triangle matrix of fixed 0s (feed–forward architecture) is not included.

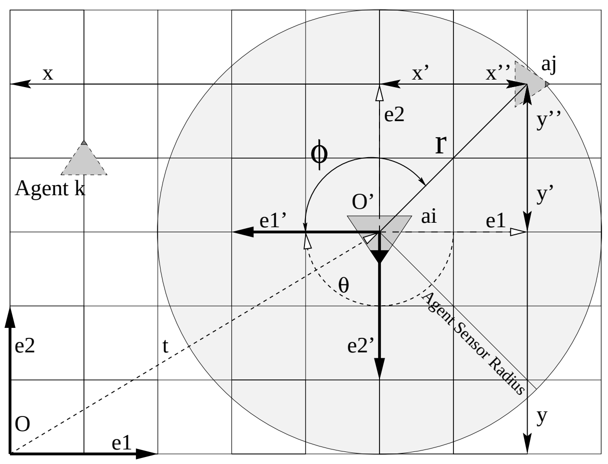
The main enhancement in the *Modified–Miller–Matrix (MMM)* direct encoding scheme [HMS95] are the *Neuron Markers* which are encoded in a separate section on the ANN chromosome - the four other sections contain steering parameters (stand still parameter, which is not used in this context), all possible connections, biases and weights respectively as well as junk parts between the neuro marker section, the connectivity section and the biases section. Technically, each neuron marker (one bit) is transferred to the main diagonal (i, i) of the MMM adjacency matrix, indicating the absence or presence of a particular neuron i and its connections (figure 5). As a consequence, most ANN chromosomes contain non–coding regions [8],

[7]For example if Forward, Back and Left are on and Right is off, then the agent will move to the left.

[8]Non–coding DNA regions or *Junk DNA* can be found in abundance in biological systems, i.e. *Eukaryotes*.



(a) Neuro Controller Topology



(b) Neuro Controller Input

Figure 3: (a) A possibly evolved neuro controller topology with two input neurons (radius r, angle ϕ), a variable number of hidden units (maximum four) and four output units (FRONT, BACK, LEFT and RIGHT) organized in a winner-takes-all layer. (b) The polar coordinate input (r, ϕ) of the neuro controller derived from the coordinates of agent a_j with respect to the local coordinate system of agent a_i. For derivations see text.

as all connections associated with a specific neuron become obsolete, if the corresponding neuron marker is zero. The non–coding regions (*Introns*) reduce the disruptive effects of crossover [Lev91] [Wu96] [May97].

The maximum number of hidden neurons (neuron markers) has to be set in advance with this encoding scheme, hence, it could be labeled as *Evolutionary Pruning*, since the system imposes an upper bound on the complexity of the network.

In order to utilize the MMM encoding scheme for our tasks we have to incorporate further information about the ANN. Since evolutionary learning is the prime learning feedback for the system, the weights and the biases of the neurons have to be encoded. As seen in figure 5, the information about biases is placed on the chromosome right after the connection bits and the weight bits are placed right after the biases bits. Contrary to the encoding of the connections, each bias and each weight is encoded with 16 bits.

In addition to the modifications of the MM encoding scheme, the GA does not operate on the matrix representation, but on the concatenation of the rows of the non–zero part of the MMM's lower triangle

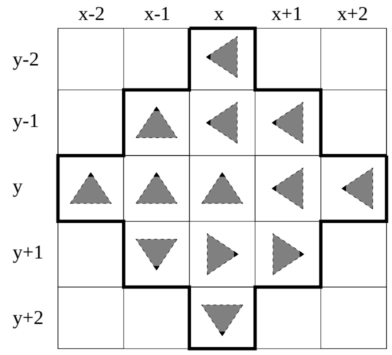


Figure 4: The output of the neuro controller is shown with respect to the local coordinate system of the agent. In our example the agent sensor radius $r_a = 2$. If an agent is detected at position $(x + 1, y + 1)$ then the neuro controller outputs RIGHT.

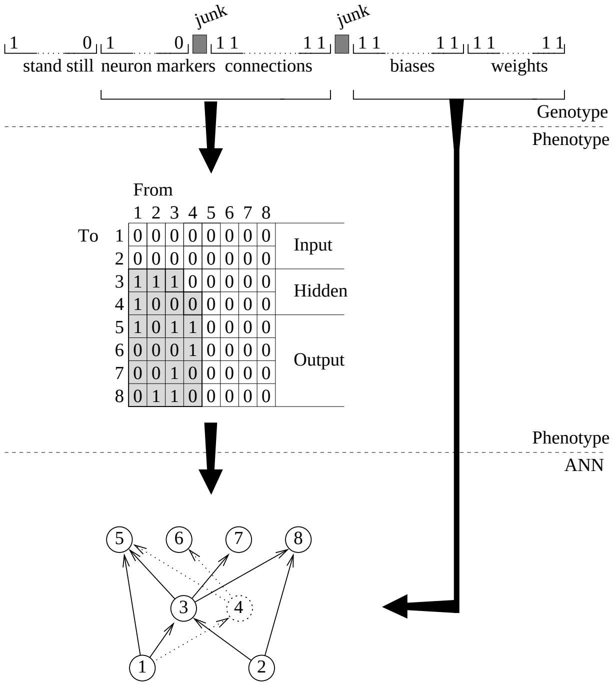


Figure 5: The encoding of the neuro controller following the direct encoding scheme. The first section of the bit-string encodes the socalled stand-still probability, the next section holds the MMM as described in the text with possibly an amount of junk between the neuron markers and the connection encoding. In the next section, which may also be preceded by junk information, the biases are encoded. Finally the weights of the connections comprise the last section.

matrix representing the possible connections. All MMM entries clamped to zero need not be processed by the GA, hence, they are not encoded in the genotype.

3.3.1 Size of the Genotype

What is the size of one genotype in Bit, one might ask. As seen in figure 5 the genotype consists of various sections of encoded information, i.e.

Stand Still $g_1 = 1 \times 16$

Neuron Markers $g_2' = |h|$ then $g_2 = g_2' +$ junk, where

$$\text{junk} = \begin{cases} (16 - g_2' \mod 16), & \text{if} \quad g_2' \mod 16 \neq 0; \\ 0, & \text{otherwise.} \end{cases}$$

Connections $g_3' = |h| \times |i| + \frac{(|h|-1)\times|h|}{2} + |o| \times |i| + |h| \times |o|$ then $g_3 = g_3' +$ junk, where

$$\text{junk} = \begin{cases} (16 - g_3' \mod 16), & \text{if} \quad g_3' \mod 16 \neq 0; \\ 0, & \text{otherwise.} \end{cases}$$

Biases $g_4 = (|h| + |o|) \times 16)$

Weights $g_5 = (g_3' \times 16)$

where $|h|$ denotes the number of hidden units, $|i|$ denotes the number of input units, $|o|$ denotes the number of output units and g_i denote the different sections of the genotype.

3.4 Evolution of the Agents

3.4.1 Random Bitstrings

In the initial population the genotypes are initialized with random values. Our software system gives the possibility to initalize the bitstrings with the same random seed in order to reproduce the results of previous experiments.

3.4.2 Genotype to Phenotype Mapping

As shown in figure 5 the genotype, i.e. bitstring, is transformed into the ANN. According to the values of the neuron markers, hidden units and their incoming and outgoing connections are omitted or integrated in the topology. The connection weights and biases are decoded from the two byte representation in the bitstring into fixed point numbers according to the following transformation:

1. Use the first byte (no matter if you use Litte Endian or Big Endian representation) for the non–fraction part of the number. The possible values are in $[0, 255]$, let's denote the number by p_n.
2. Use the second byte for the fraction part of the number. The possible values are in $[0, 0.9921875]$[9], let's denote the number by p_f.
3. Add the two numbers $p' = p_n + p_f$, $p' \in [0, 255.9921875]$.
4. In order to represent negative numbers as well transform p' by subtracting 128, i.e. $p = p' - 128$. Hence $p \in [-128, 127.9921875]$.

3.4.3 Fitness Evaluation and Selection

In our experiments we use two decision variables, number of places visited $\mathcal{C}_{\mathcal{PV}}$ and number of distinct communication partners $\mathcal{C}_{\mathcal{CP}}$, to determine the performance of an individual. We faced the problem of incorporating the various decision variables into one global fitness function. One possible solution is a weighted sum incorporating all decision variables, but how do we have to choose the weights?

As an alternative we applied the definition in section 6.1. Arrange the measures of the decision criteria (simple fitness values) in a vector and let two agents, randomly chosen from the generation at hand, play games against each other. A game in this context means that according to the simple fitness values the performance of the agents are evaluated and compared. There are three possible outcomes per game and per agent: win, loose, draw. If the outcome for a_i is win or draw then it gets a reward of 1.

After all the single fitness measures have been evaluated, an overall result is determined by cumulating the rewards. Again, there are three outcomes: overall win, overall loose, overall draw. The agent which has won overall is selected for the mating pool. In the case of an overall draw, we have various possibilities how to select an agent for the mating pool, e.g.

- choose one of the agents randomly, or
- weight the rewards of an agent according to the order of the decision criteria in the decision vector.

Selection scheme 1. The rewards are summed up and in the case of an overall draw, an agent is chosen randomly for the mating pool. Let x_1, y_1 denote the number of places visited for a_i, a_j, respectively. Further denote the number of communication partners by x_2, y_2 for a_i, a_j, respectively.

	a_i	a_j
1. $\mathcal{C}_{\mathcal{PV}}$	$\begin{cases} 1, & if \quad x_1 \geq y_1; \\ 0 & otherwise. \end{cases}$	$\begin{cases} 1, & if \quad y_1 \geq x_1; \\ 0 & otherwise. \end{cases}$
2. $\mathcal{C}_{\mathcal{CP}}$	$\begin{cases} 1, & if \quad x_2 \geq y_2; \\ 0 & otherwise. \end{cases}$	$\begin{cases} 1, & if \quad y_2 \geq x_2; \\ 0 & otherwise. \end{cases}$
Sum of wins	s_i	s_j

[9] $\sum_{i=0}^{n} q = \frac{1-q^{n+1}}{1-q}$. We start the calculation at the first digit right after the fixed point. i.e. we have to subtract 1 from the formula. This yields $\frac{1-(1/2)^8}{1/2} - 1$.

```
If s_i > s_j then a_i is selected;
If s_j > s_i then a_j is selected;
If s_j = s_i then random(a_i,a_j) is selected;
```

Selection scheme 2. The weighted rewards are summed up and an agent is chosen according to the weighted rewards. Let w_{PV} denote the weight for the reward of $\mathcal{C}_{\mathcal{PV}}$ and w_{CP} the weight for reward of $\mathcal{C}_{\mathcal{CP}}$. Choosing the weights according to 2^n — where n is the number of selection criteria and n gets assigned to the highest priority criterion an 1 to the lowest — resolves the problem of taking into account the order of the criteria.

		a_i	a_j
1.	$\mathcal{C}_{\mathcal{PV}}$	$\begin{cases} w_{PV}, & if \quad x_1 \geq y_1; \\ 0 & otherwise. \end{cases}$	$\begin{cases} w_{PV}, & if \quad y_1 \geq x_1; \\ 0 & otherwise. \end{cases}$
2.	$\mathcal{C}_{\mathcal{CP}}$	$\begin{cases} w_{CP}, & if \quad x_2 \geq y_2; \\ 0 & otherwise. \end{cases}$	$\begin{cases} w_{CP}, & if \quad y_2 \geq x_2; \\ 0 & otherwise. \end{cases}$
	Sum of wins	s_i	s_j

```
If s_i > s_j then a_i is selected;
If s_j > s_i then a_j is selected;
If s_j = s_i then random(a_i,a_j) is selected;
```

In the second selection scheme a draw over different selection criteria is resolved by the weights of the selection criteria. In the example $\mathcal{C}_{\mathcal{PV}}$ has a higher priority and therefore weight than $\mathcal{C}_{\mathcal{CP}}$. This modified tournament selection is played twice the number of agents in one population.

3.4.4 Mutation & Crossover

After selection, mutation — mutation rate equals $1/l$, where l denotes the length of the bitstring — and two point crossover are applied to the genotypes and the next generation is produced which starts the evolutionary cycle again[10].

4 Experiments

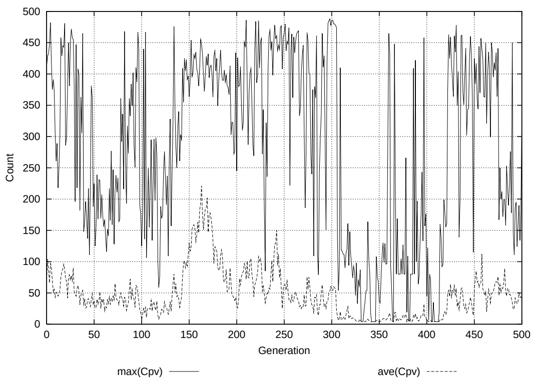
In this section five performed experiments with the setup defined in section **??** will be discussed. Each experiment consists of 100 generations, 500 simulation steps and 49 agents in each generation. The radius of the agent sensor is set to 10, where the dimensions of the world are 80×80, connected in a toroidal manner. The neuro controller has 2 input neurons, four output units and up to four hidden units, where the number of hidden units and their connections evolve over time. We use a tanh function as transfer function and let the neuro controller learn through evolution.

For every simulation the following data are collected: maximum number of places visited, average number of places visited, maximum number of distinct communication partners and average number of communication partners. Furthermore, for every single agent the bitstring defining its neuro controller, the output of its neuro controller and the movements during the experiment were recorded.

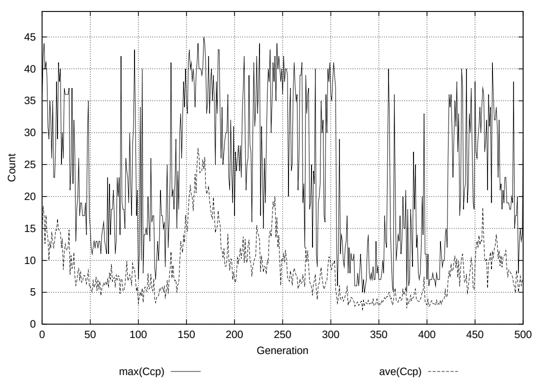
Based on this data the following issues are discussed for each experiment:

- Graph of evolution of selected criteria, s.a. maximum number of places visited, average number of places visited, maximum number of distinct communication partners and average number of communication partner.
- Evolution of wandering behavior. Contrary to other experiments, e.g. [MNT96], no default direction for the step an agent will perform if no other agent is within sensor range is defined. The neuro controller gets (0,0) as its input in this situation. Starting with randomly created bitstrings in generation 0 the evolution of implicit rules which optimize the agents behavior with respect to the selection criteria is observed and discussed. This discussion is based on two generations, i.e. initial(0) and last generation(100).

[10] Another method would be to learn the agent by means of reinforcment learning after each step.



(a) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$.



(b) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{CP}}$.

Figure 6: Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$ and $\mathcal{C}_{\mathcal{CP}}$ for experiment 0.

- Evolution of neuro controller topology. The topology of one typical neuro controllers from the last generation is presented.

4.1 Experiment 0

4.2 Experiment 1

The criteria selected for this experiment are the number of places visited $\mathcal{C}_{\mathcal{PV}}$ and the number of distinct communication partners $\mathcal{C}_{\mathcal{CP}}$.

The average and maximum number of places visited by the agents is shown in figure 8. A rapid increase in the average number of places visited is achieved for the first 25 generations. Then the progress slows down and stays around 450 places visited by the agents. As seen, the performance of the system is not affected by the randomly chosen starting points in a new generation. Thus, a starting position invariant wandering behavior is implemented by the neuro controller.

In the initial phase of the experiment about 75% of all agents start with an initial direction $\neq$ FORWARD, i.e. these agents either perform left or right rotation or flip back and forward. The average number of places visited in these cases are four for the rotational case and two for the flip case. If an agent has FORWARD as its initial direction it will move forward, according to its heading direction. Agents just implementing this FORWARD behavior visit 80 distinct locations. Hence they are superior to rotating or flipping agents with respect to the fitness evaluation and will be selected for the mating pool for the next generation. A basic "rule" that can be derived from the evolved neuro controllers with respect to criterion $\mathcal{C}_{\mathcal{PV}}$:

RULE 1:

```
IF no agents are within sensor range THEN
  Make a step with direction FORWARD
END IF
```

While this basic "rule" evolves and becomes common for all agents, the population diversity, as defined in section 6.2, decreases, as seen in figure 9.

A value close to 1.0 in figure 9 indicates a widespread population by means of topologically different neuro controllers. If diversity drops, the variability of neuro controller topologies decreases and the population is dominated by a limited number of neuro controller topologies. In the 100th generation just two topolgies survived, shown in figure **??**, which yielded the same output.

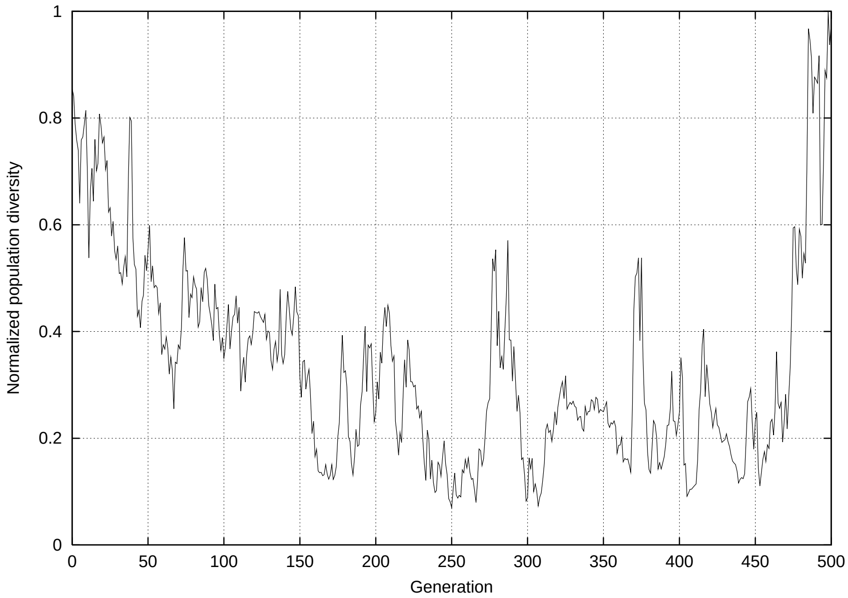


Figure 7: Normalized population diversity for experiment 0.

In figure 10 the density plots of places visited by agent in generation 0 and 100 are shown. As can be seen from figure 10.(a), agents implement different wandering behaviors. Some perform a left or right rotation, shown by four isolated crosses. Others implement a behavior according to **RULE 1** derived above. The third minor class of agents show a behavior, which is dominant in generation 100. It may be described as follows:

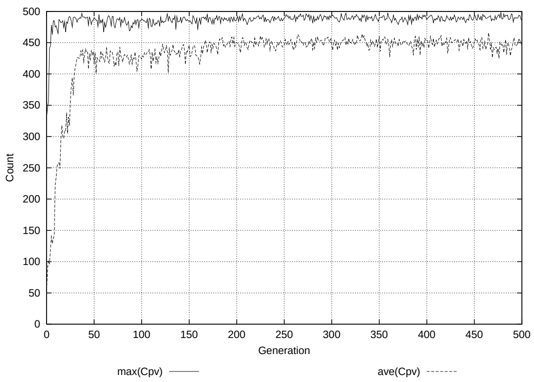
RULE 2:

```
IF no agents are within sensor range THEN
  Make a step with direction FORWARD
ELSE
  Make a step with a direction drawn from {FORWARD, BACK, LEFT, RIGHT}
ENDIF
```

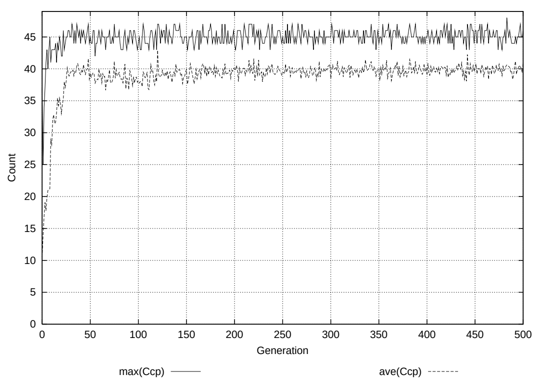
Although the single agents implement a consistent response in certain regions of their controller output, the variability of neuro controller outputs is high compared to generation 100. As mentioned above the normalized population diversity in generation 0 is close to 1.0. This may serve as an explanation for the variability although we have seen that in generation 100 different topologies encode the same behavior. Analyzing the neuro controller output in generation 100 we may rewrite **RULE 2** stated above as follows:

RULE 3:

```
IF no agents are within sensor range THEN
  Make a step with direction FORWARD
ELSEIF an agent is within sensor range AND
       the agent is not in the left upper hemisphere of the sensor THEN
  Make a step with direction FORWARD
ELSE
  Make a step with direction LEFT
ENDIF
```



(a) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$.



(b) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{CP}}$.

Figure 8: Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$ and $\mathcal{C}_{\mathcal{CP}}$ for experiment 1.

If we analyze the wandering behavior of two agents in generation 100 we realize that there are a number of interaction patterns when two agents meet. Three of the interaction patterns are shown in figure 11, where the two dominating neuro controller outputs of generation 100 are shown in figure 12.

Interaction pattern 1. The agents a_i and a_j move towards each other where each agent senses the other agent in the upper left hemisphere of the agent sensor. As seen from the output of the sensor, both agents perform a move to the right. In the next steps they will move away from each other, since the output of the controller dictates a forward move for detected agents in the back. Thus we learn that agent a_i helps agent a_j (and vice–versa) to leave its orbit on the torus and move into other, possibly unknown regions.

Interaction pattern 2. The agents a_i and a_j meet each other on orthogonal orbits. In this case agent a_i has agent a_j on its right side, i.e. the neuro controller does respond with forward. Thus a_i stays on its orbit. Agent a_j on the other side observes a_i on its left side. Hence it will change its direction and turn right. The two agents will perform a parallel obsevation of the environment.

Interaction pattern 3. Agent a_i and a_j sense each other in such regions of the agent sensors which yield a forward output. It follows that they stay on the same orbit as before they have met.

As the interaction patterns in figure 11 and their analysis show, the agents influence on the course of another agent encountered varies from high, i.e. change direction, to low, i.e. no change. As it was outlined above, a major reason for the succesful evolution of wandering behavior is the evolved forward drive, if no other agent is within sensor range and the orbit changing influence in interaction with other agents.

In figure 13 various topologies of the evolved neuro controllers in generation 100 are shown. The topologies figure 13.(a), figure 13.(b) and figure 13.(d) mostly correspond to the output in figure 12.(a) and the topology in figure 13.(c) corresponds to the output in figure 12.(b). Although the ANNs 13.(a), 13.(b) and 13.(d) show a wide topological variety they yield quite similar results. It is noteworthy that all of the networks shown have an isolated output unit which is responsible for the BACK response. Since no incoming signals can be integrated and transformed into an output value, the output unit will answer with $\frac{1}{1+e^{-\theta}}$, where θ denotes the bias. Hence it will never win over the other units. One possible explanation for this evolutionary result is that a backward move does not either alter the orbit nor changes the location of the agent siginificantly as the forward output does.

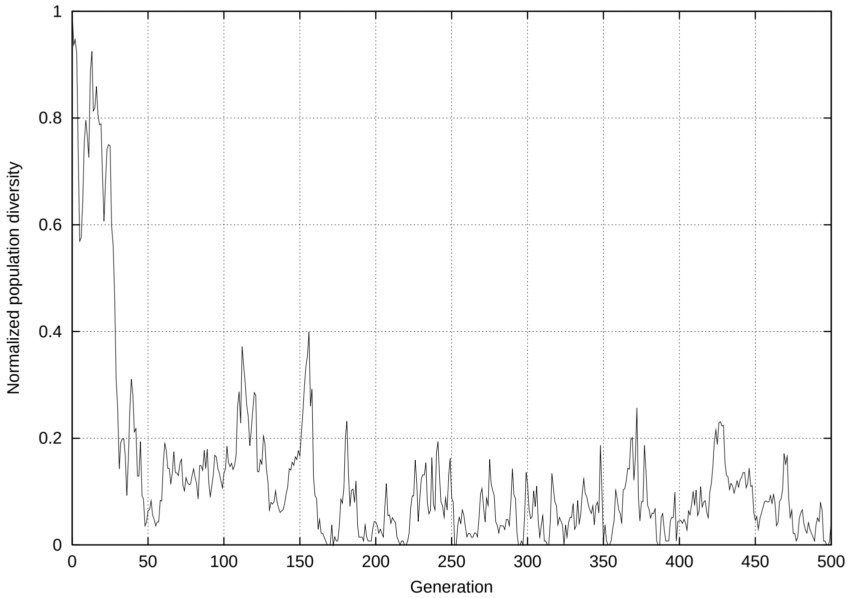


Figure 9: Normalized population diversity for experiment 1.

4.3 Experiment 2

4.4 Experiment 3

4.5 Experiment 4

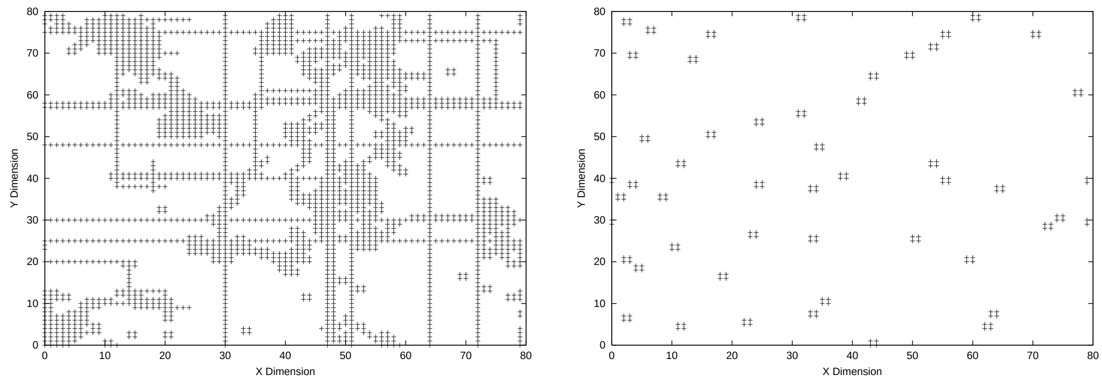
4.6 Experiment 5

	w_{PV}	w_{CP}	$\mathcal{C}^*_{\mathcal{PV}}$	$\mathcal{C}^*_{\mathcal{CP}}$	$\overline{\mathcal{C}}^*_{\mathcal{PV}}$	$\overline{\mathcal{C}}^*_{\mathcal{CP}}$
Experiment 0	0	0	488	45	221	27.41
Experiment 1	1	0	498	48	466	42.92
Experiment 2	0	1	482	41	117	17.51
Experiment 3	1	1	494	47	450	41.39
Experiment 4	2	1	497	48	454	41.78
Experiment 5	1	2	488	43	60	10.76

Table 1: Results for the various experiments described above. $\mathcal{C}^*_{\mathcal{PV}}$ denotes the maximum number of distinct places visited by one agent for all generations. $\mathcal{C}^*_{\mathcal{CP}}$ denotes the maximum number of distinct communication partners for one agent for all generations. $\overline{\mathcal{C}}^*_{\mathcal{PV}}$ denotes the maximum average number of distinct places visited by all agents for all generations. $\overline{\mathcal{C}}^*_{\mathcal{CP}}$ denotes the maximum average number of distinct communication partners for all agents for all generations.

5 Conclusion

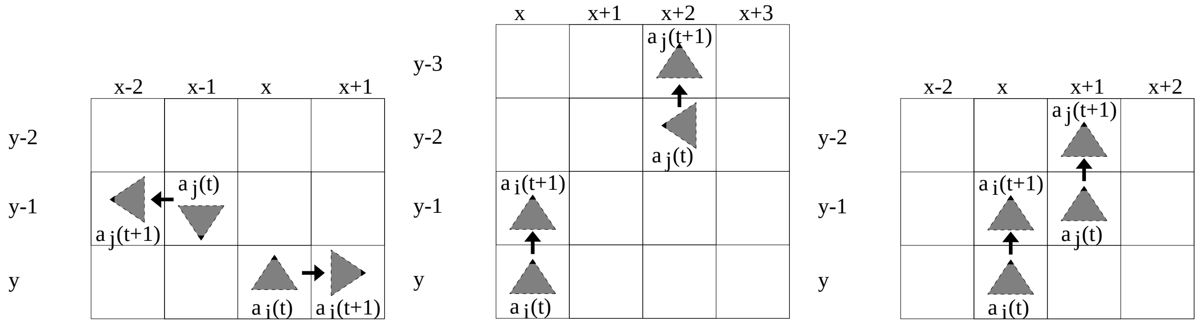
We have described an experiment for evolving wandering behavior in a multi agent system. Our discussion started with the system setup where we showed how to integrate a genetic algorithm with a multi–objective fitness function and a multi layer perceptron into an evolutionary system. We were concerned with the simplicity of the various system components. Thus we even encoded the network topologies in the genotype in order to automatically determine another parameter of the system. As we saw in our experiment various network topologies developed through the encoding of the so–called neuro markers. Surprisingly they yielded



(a) Density plot of places visited by agents in generation 0.

(b) Density plot of places visited by agents in generation 500.

Figure 10: Density plots of places visited by agents in generation 0 and 500 for experiment 1.



(a) Interaction pattern 1.

(b) Interaction pattern 2.

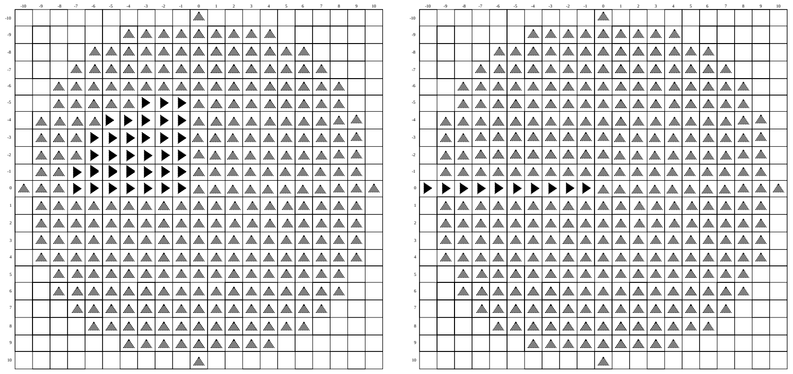
(c) Interaction pattern 3.

Figure 11: Various interaction patterns for agent a_i and a_j in generation 100 of experiment 1.

comparable outputs where an implicit pruning of the output layer was performed. This pruning was not explicitely encoded into the system nor demanded by the fitness criteria.

The evolved wandering behavior is invariant regarding the starting position of the agents in the simulation which was shown by discussion of interaction patterns. We saw that the system developed simple rules during the course of evolution for the wandering behavior of the agents. The first simple rule may be stated as “if there is no agent in sensor range, then move forward”. As a second rule other agents were incorporated into the simple rule. The second rule may be restated as “apply rule one and if there is an agent on the left side, then turn right.”

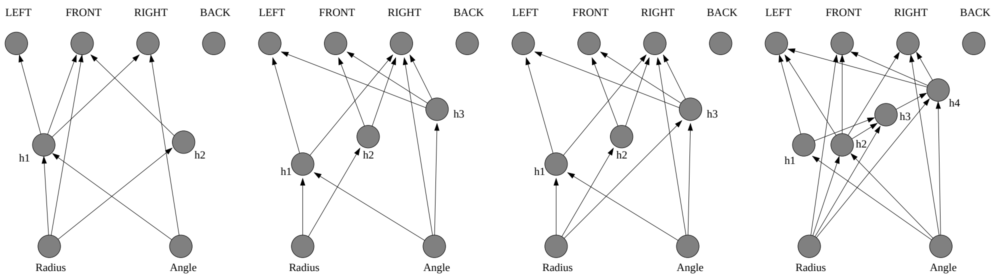
Promising extensions of the proposed experiment cover the extension of the number of selection criteria and their influence on the evolution of the wandering behaviors of the single agents. Especially topological complexity may catch our extension, since the interpretation of the encoded behavior may be performed more straightforward. Another interesting question concerns the number of agents in the simulation and the extension of the system into an open environment were the number of agents is dynamic.



(a) Example of neuro controller output 1.

(b) Example of neuro controller output 2.

Figure 12: Dominating neuro controller outputs in generation 100 of experiment 1.



(a) Example of neuro controller topology with two hidden units.

(b) Example of neuro controller topology with three hidden units.

(c) Example of neuro controller topology with three hidden units.

(d) Example of neuro controller topology with four hidden units.

Figure 13: Some examples of neuro controller topologies in generation 100 of experiment 1.

6 Appendix

6.1 Pareto–Optimal Solutions

In their paper Zitzler and Thiele [ZT98] state that many real–world problems comprise competing objectives. Pareto–optimal solutions are solution where no other solutions are superior when considering all objectives. Defining the concept of pareto–optimality mathematically yields:

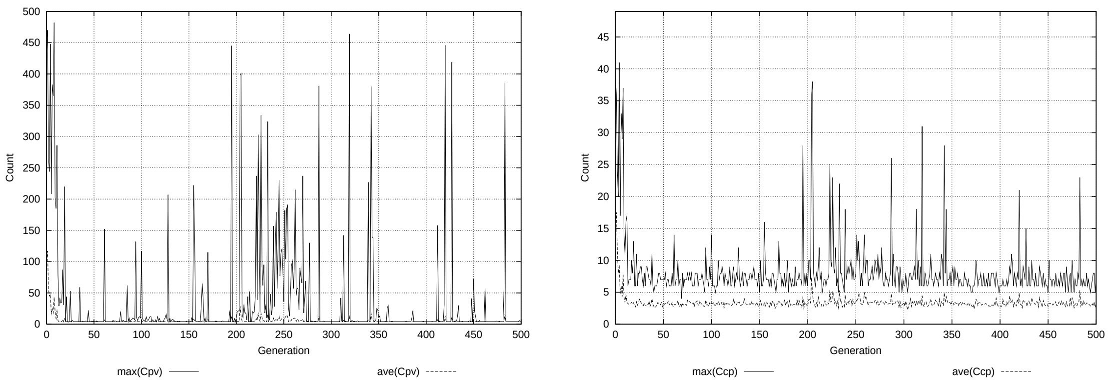
Definition 1 *Given a multiobjective maximization problem with m parameters (decision variables) and n objectives:*

$$Maximize \quad \vec{y} = f(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \ldots, f_n(\vec{x})),$$

where $\vec{x} = (x_1, x_2, \ldots, x_m) \in X$ *and* $\vec{y} = (y_1, y_2, \ldots, y_m) \in Y$ *are tuple. A decision vector* $\vec{a} \in X$ *is said to dominate a decision vector* $b \in X$ *(also written as* $a \succ b$*)* $\Longleftrightarrow$

$$\forall i \in 1, 2, \ldots, n : f_i(\vec{a}) \geq f_i(\vec{b}) \wedge \exists j \in 1, 2, \ldots, n : f_j(a) > f_j(b)$$

All decision vectors which are not dominated by any other decision vector are called nondominated or Pareto–optimal.



(a) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$.

(b) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{CP}}$.

Figure 14: Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$ and $\mathcal{C}_{\mathcal{CP}}$ for experiment 2.

Applying the definition to our system means,

1. we have two decision variables $\mathcal{C}_{\mathcal{PV}}$ and $\mathcal{C}_{\mathcal{CP}}$,
2. two objectives $f_1(\mathcal{C}_{\mathcal{PV}}, \mathcal{C}_{\mathcal{CP}}) = \mathcal{C}_{\mathcal{PV}}$ and $f_2(\mathcal{C}_{\mathcal{PV}}, \mathcal{C}_{\mathcal{CP}}) = \mathcal{C}_{\mathcal{CP}}$ (Projection),

6.2 Diversity

Definition 2 *The agent diversity d_{ij} between two agents a_i, a_j is defined as*

$$d_{ij} = \sum_n \sum_m |M_i(n,m) - M_j(n,m)|,$$

where M_i denotes the Modified Miller Matrix of agent a_i.

Definition 3 *The population diversity D_i of a set of agents $\mathcal{A}_\rangle = \{\dashv_\rangle\}$ in a particular generation i is defined as*

$$D_i = \sum_{a_v} \sum_{a_u : a_u \neq a_v} d_{uv},$$

where d_{uv} denotes the agent diversity.

Definition 4 *The normalized population diversity $||D||$ is defined as*

$$||D|| = \frac{D_i}{\max_j D_j},$$

where D_i, D_j denote the population diversity.

6.3 Agents

In this section a definition of an autonomous agent is reviewed from [FG97]. Further key discrimination aspects of DAI systems are restated from [Wei95].

In their paper Franklin and Graesser [FG97] give an overview of the term agent. They draw from a variety of definitions and examples the essentials and come up with a categorization for autonomous agents. They put forward their understanding of autonomous agency in the following definition

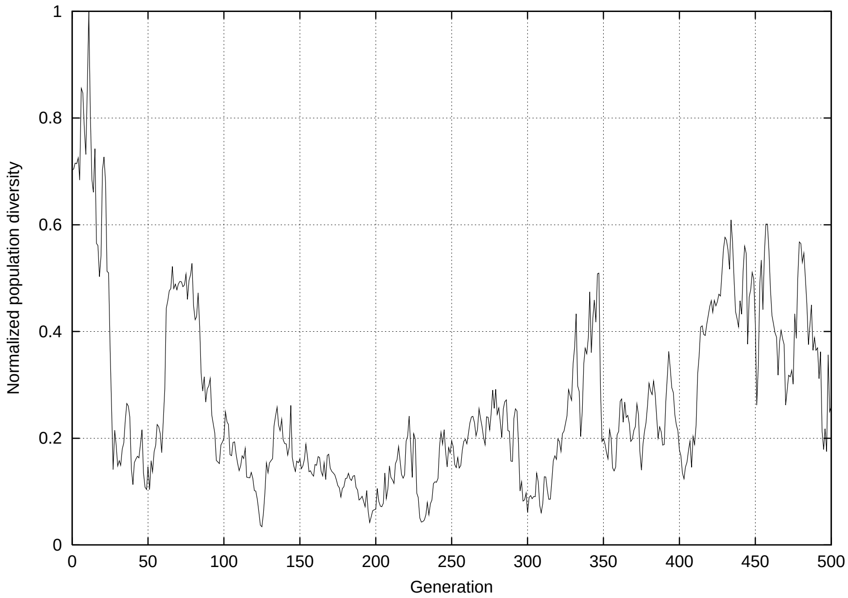


Figure 15: Normalized population diversity for experiment 2.

Definition 5 *An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.*

In order to classify agents according to their properties the authors extracted aspects from the discussed definitions and summarized them accordingly in table 6.3. As they stated, this is just one possibility to

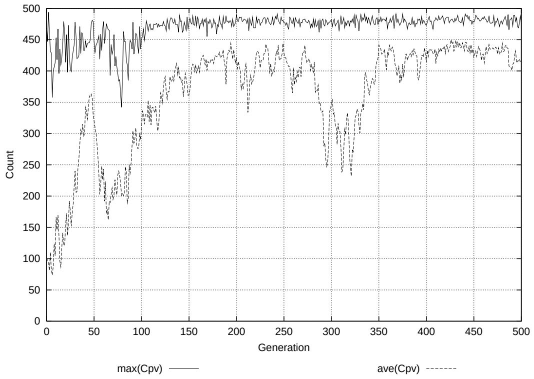
Property	Other Names	Meaning
reactive	sensing and acting	responds in a timely fashion to changes in the environment
autonomous		exercises control over its own actions
goal-oriented	pro-active, purposeful	does not simply act in response to the environment
temporally continous		is a continously running process
communicative	socially able	communicates with other agents, perhaps including people
learning	adaptive	changes its behavior based on its previous experience
mobile		able to transport itself from one machine to another
flexible		actions are not scripted
character		believable “personality” and emotional state

Table 2: Agent properties, where various classes of agents can be defined.

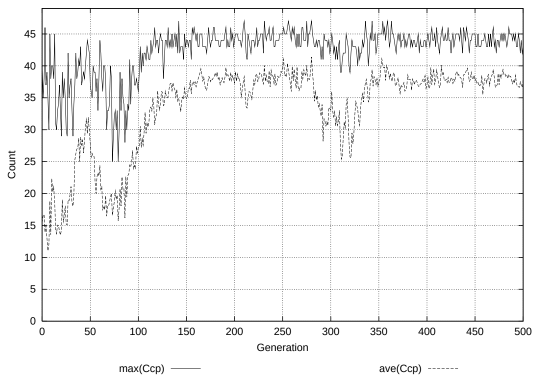
classify agents. Another interesting classification scheme was given by them from a biological point of view which is not shown in our paper.

Weiß [Wei95] describes three key discrimination aspects of distributed artificial intelligence (DAI) systems

1. the environment occupied by the multi-agent system (MAS), i.e.
   (a) the availability of environmental resources (ranging from restricted to ample),
   (b) the environmental diversity (ranging from poor to rich),
   (c) the environmental uncertainty and predictability (ranging from predictable to unpredictable), and



(a) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$.



(b) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{CP}}$.

Figure 16: Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$ and $\mathcal{C}_{\mathcal{CP}}$ for experiment 3.

   (d) the environmental dynamics and status (ranging from fixed to able),

2. agent-agent and agent-environment interaction, i.e.
   (a) the frequency of interaction (ranging from low to high),
   (b) the persistence of interaction (ranging from short-term to long-term),
   (c) the level of interaction (ranging from signal passing to knowledge exchange),
   (d) the pattern of interaction (ranging from unstructured to structured),
   (e) the variability of interaction (ranging from fixed to changeable),
   (f) the type of interaction (ranging from competitive to cooperative),
   (g) the purpose of interaction (ranging from random to goal-directed),

3. the agents, i.e.
   (a) the number of agents involved in the MAS (ranging from two upward),
   (b) the number of goals an agent has (ranging from one upward),
   (c) the compatibility of the goals (ranging from contradicting to complimentary),
   (d) the uniformity of the agents (ranging from homogenous to heterrogeneous), and
   (e) the properties of the inidividual agent.

He derives the following kernel aspects of agengcy

1. perceptual, cognitive and effectual skills,
2. communicative and social abilities and
3. autonomy.

Besides these properties the following are considered as beeing essential

1. reactivity (i.e., the ability to respond to environmental changes in reasonable time),
2. situatedness (i.e., the ability to continuously interact with - or to be embedded in - its environment),
3. pro-activeness and deliberation (i.e., the ability to act in a foreseeing, goal- or plan-oriented manner),

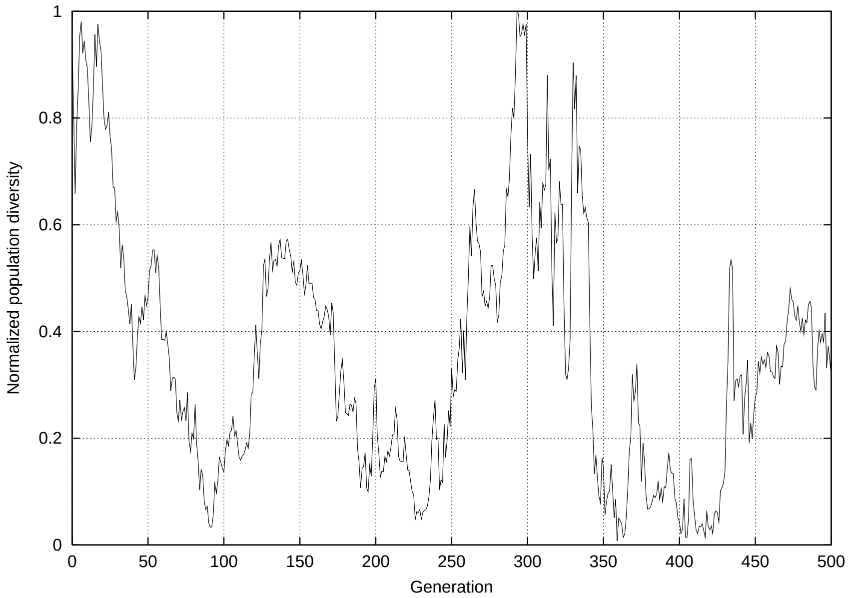


Figure 17: Normalized population diversity for experiment 3.

4. rationality (i.e., the ability to always behave in a way which is suitable or even optimal for goal attainment),

5. mobility (i.e., the ability to change the physical position,

6. introspection (i.e., the ability to examine and self-reflect ist own thoughts, ideas plans, etc.,

7. veracity (i.e., the ability of communicating false information),

8. benevolence (i.e., the property of always doing what is asked to do).

Further properties may describe the state the agent is in, s.a.

1. belief, knowledge, etc., which describe information or cognitive states,

2. intention, commitment, plan, etc., which describe deliberative or conative states,

3. desire, goal, choice, preference, etc., which describe motivational or affective states.

Each of the properties given above may be thought of as a dimension in a space describing a multi-agent system.

Weiß gives a various categories of learning, s.a.

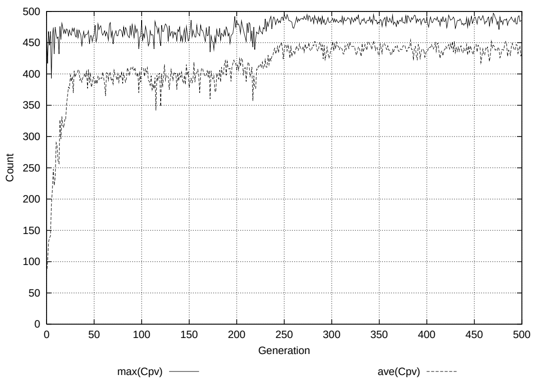
1. single-agent or isolated learning, where learning does not rely on the presence of multiple agents,

2. multi-agent or interactive learning, where learning relies on or even requires the presence of multiple agents. Other expressions for multi-agent learning are mutual learning, cooperative learning, collaborative learning, co-learning, shared learning, team learning, social learning, pluralistic learning and organizational learning.

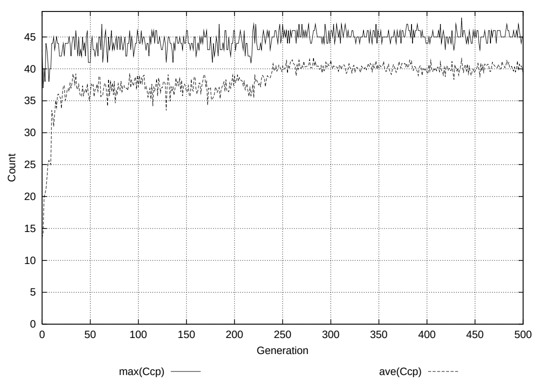
Furthermore he summarizes various forms of learning

1. The learning method

   (a) rote learning, i.e. direct implantation of knowledge and skills without requiring further inference or transformation from the learner,



(a) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$.



(b) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{CP}}$.

Figure 18: Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$ and $\mathcal{C}_{\mathcal{CP}}$ for experiment 4.

   (b) learning from instruction and by advice taking
   (c) learning from examples and by practice
   (d) learning from analogy
   (e) learning from discovery

2. The learning feedback
   (a) supervised learning
   (b) reinforcment learning
   (c) unsupervised learning

As he notes, it is important to see that different agents do not have to learn on the basis of the same learning method or feedback.

In our context it is misleading to apply the definition of an agent give in [FG97] in a one to one manner. Our agents do not pursuit in their own agenda but in a population agenda since the overall goal is determined by the global fitness function for the whole population. Our agents are, according to the classification given in table 6.3,

1. Reactive, they sense objects (agents) in their neighborhood and react (communicate and wander) upon them,
2. autonomous, they control their communication and wander behavior,
3. indirect goal–oriented, they react in response to their environment but have encoded the means to act goal–oriented with respect to the generation goal,
4. temporally continous, per generation since we investigate a generational based evolutionary system contrary to a steady state system,
5. communicative, they communicate with other agents about agents,
6. not learning, they do learn evolutionaryly but do not learn during their lifetime,
7. mobile, they wander from one location to the next location in their environment,
8. not flexible characters, their actions are scripted (first communicate then select direction to move).

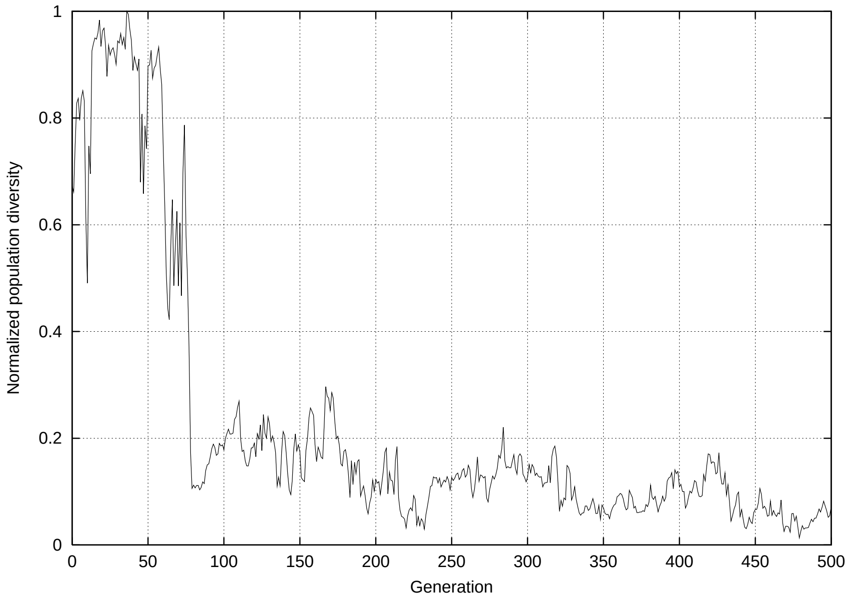


Figure 19: Normalized population diversity for experiment 4.

6.4 Naming Games

In our system one of the selection criteria used for fitness evaluation is based on naming games proposed by L. Steels (e.g. [SM97]). We utilize naming games to determine the number of distinct communication partners met during wandering in the world.

As defined in [SM97], let a_i denote a single *agent*, where $\mathcal{A}$ denotes the set of all agents. The number of agents, i.e. the cardinality of $\mathcal{A}$ is denoted by $N_{\mathcal{A}}$. Each $a_i \in \mathcal{A}$ has associated a set $\mathcal{O}_i = \{o_{0_i}, \ldots, o_{k_i}\}$ of *objects*, with $o_j \in \mathcal{O}$, where $\mathcal{O}_i \subset \mathcal{O}$. The size of the set $\mathcal{O}$ is denoted by $N_{\mathcal{O}}$ and the size of set $\mathcal{O}_i$ is denoted by $N_{\mathcal{O}_i}$. Some or all the objects are shared between the agents.

Definition 6 *A* **word** $w \in \mathcal{W}$ *is a sequence of letters drawn from a finite alphabet. The agents are all assumed to share the same alphabet.*

Definition 7 *A* **lexicon** L *is a relation between objects and words, a number which denotes the number of times the relation has been used and a number which stands for the number of times the relation has been used sucessfully, thus* $L = \mathcal{O} \times \mathcal{W} \times N_u \times N_s$*.*

Definition 8 [11] *A word* $w_i \in \mathcal{W}$ *is more* **succesful** *than a word* $w_j \in \mathcal{W}$ *iff* $(o, w_i, n_{u_i}, n_{s_i}) \in L$, $(o, w_j, n_{u_j}, n_{s_j}) \in L$, $n_{u_i} \geq n_{u_j} \geq 0$ *and either* $n_{s_i}/n_{u_i} > n_{s_j}/n_{u_j}$ *or* $n_{s_i}/n_{u_i} = n_{s_j}/n_{u_j}$ *and* $n_{u_i} > n_{u_j}$*.*

Each agent $a_i \in \mathcal{A}$ has his own lexicon $L_i = \mathcal{O}_i \times \mathcal{W}_i \times N_{u_i} \times N_{s_i}$ which is initially empty. This definitions leaves open the possibility of *homony* – a single word can be associated with several objects – and *synonymy* – a single object can be associated with several words.

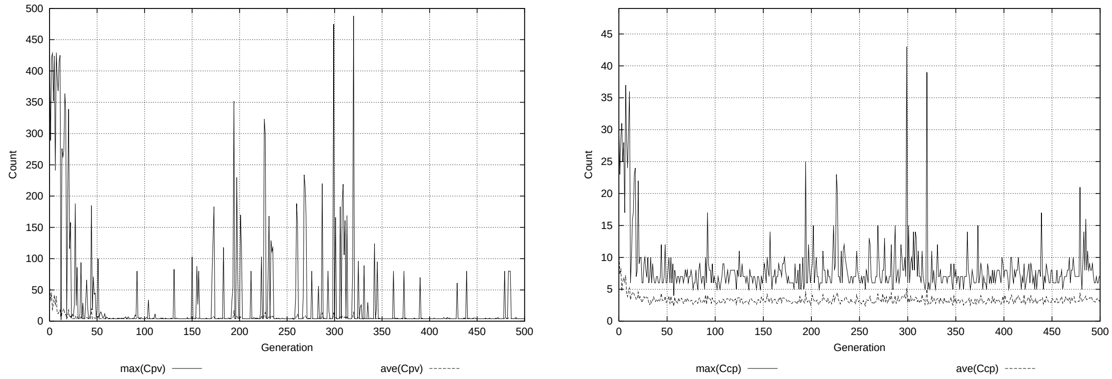
Definition 9 *An* **agent** $a_i \in \mathcal{A}$ *is a tupel* $a_i = (L_i, \mathcal{O}_i)$*.*

Definition 10 *An* **interaction** *between two agents is defined by the following procedure:*

1. *The speaker* a_s *randomly selects a topic* $o \in \mathcal{O}_s$ *out of his set of objects. The speaker draws the attention of the hearer* a_h *to the topic (e.g. pointing or prior linguistic exchange).*

2. *The speaker* a_s *encodes the topic* o *through a word* w*. The word chosen is the most successful word* w *where* $(o, w, n_u, n_s) \in L_s$*. If several words are equally successful, a random word is selected from the most successful set.*

[11] Improvement: Define order relation



(a) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$.

(b) Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{CP}}$.

Figure 20: Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$ and $\mathcal{C}_{\mathcal{CP}}$ for experiment 5.

3. *The hearer* a_h *decodes the word* w *to derive a set of objects* $\mathcal{O}'_h \subset \mathcal{O}_h$ *such that* $o_h \in \mathcal{O}'_h$ *iff* $(o_h, w, n_u, n_s) \in L_h$.

4. *An interaction is successful iff* $o \in \mathcal{O}'_h$.

Definition 11 *A* **naming game** $N = (s, h, o)$ *is an interaction between two agents: a* **speaker** $a_s \in \mathcal{A}$ *and a* **hearer** $a_h \in \mathcal{A}$ *about a* **topic** $o \in \mathcal{O}_s$.

Definition 12 *An* **agent** $a_i \in \mathcal{A}$ *at time* t *is a tupel* $a_{i,t} = (L_{i,t}, \mathcal{O}_{i,t})$. *Where time* t *denotes a discrete point on the time axis where two agents play a naming game.*

Definition 13 *An* **adaptive naming game** *is a naming game over the time between two agents: a* **speaker** $a_{s,t} \in \mathcal{A}$ *and a* **hearer** $a_{h,t} \in \mathcal{A}$ *about a* **topic** $o \in \mathcal{O}_{s,t}$.

If the speaker $a_{s,t}$ uses a word w where $(o, w, n_{s,t}, n_{u,t}) \in L_{s,t}$ there are two possible outcomes of an adaptive naming game:

Success The naming game ends in success, then $(o, w, n_{s,t}+1, n_{u,t}+1) \in L_{s,t+1}$ and $(o, w, n_{s,t}+1, n_{u,t}+1) \in L_{h,t+1}$.

Failure The naming game did not end in success. A word is used by speaker or hearer but some failure has occured, then $(o, w, n_{u,t}+1, n_{s,t}) \in L_{s,t+1}$ and $(o, w, n_{u,t}+1, n_{s,t}) \in L_{h,t+1}$. The other well defined changes from time t to time $t+1$ are as follows:

1. *Missing object* (step 1 fails) The object chosen by the speaker is not shared by the hearer, i.e. $o \notin \mathcal{O}_{h,t}$. The hearer might acquire the object with an object propagation probability p_o. If the object is successfully acquired then $\mathcal{O}_{h,t+1} = \mathcal{O}_{h,t} \bigcup \{o\}$.
2. *Speaker does not have a word* (step 2 fails) There is no association in the lexicon between a word and the object chosen by the speaker, i.e. $\nexists(o, w, n_{u,t}, n_{s,t}) \in L_{s,t}$. The speaker might create a word w' for the object and insert it into the lexicon with a word creation probability p_c, i.e. $L_{s,t+1} = L_{s,t} \bigcup (o, w', 1, 0)$.
3. *Word not in hearer lexicon* (step 3 fails) There is no association in the lexicon between the word chosen by the speaker and an arbitrary object, i.e. $\nexists(o, w, n_{u,t}, n_{s,t}) \in L_{h,t}$ or $\mathcal{O}'_h = \emptyset$. The hearer might use the word w for the object and insert it into the lexicon with a word absorption probability p_a, i.e. $L_{h,t+1} = L_{h,t} \bigcup (o, w, 1, 0)$.

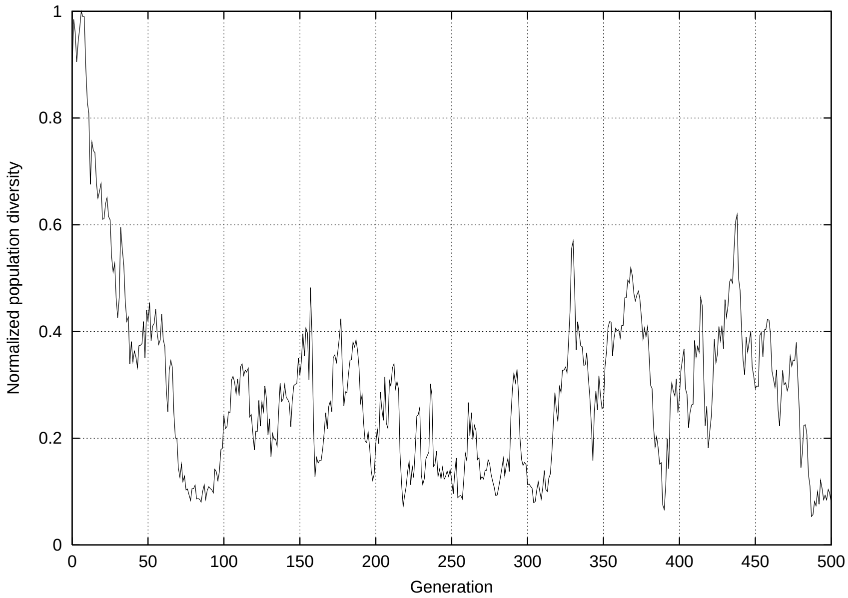


Figure 21: Normalized population diversity for experiment 5.

4. *Word is not decoded by the hearer* (step 4 fails) There is no association in the lexicon between the word and the object chosen by the speaker, i.e. $o \notin \mathcal{O}'_h$. The hearer might aquire the word w for the object and insert it into the lexicon with a word absorption probability p_a, i.e. $L_{h,t+1} = L_{h,t} \bigcup (o, w, 1, 0)$.

If there is no explicit change as defined above, then the definition of an agent a_i remains the same, i.e. $a_{i,t+1} = (L_{i,t}, \mathcal{O}_{i,t})$.

7 Acknowledgements

References

[Bar98] C. Baray. Effects of population size upon emergent group behavior. *Complexity International*, 6, 1998.

[Dug66] James Dugundji. *Topology*. Allyn and Bacon, 1966.

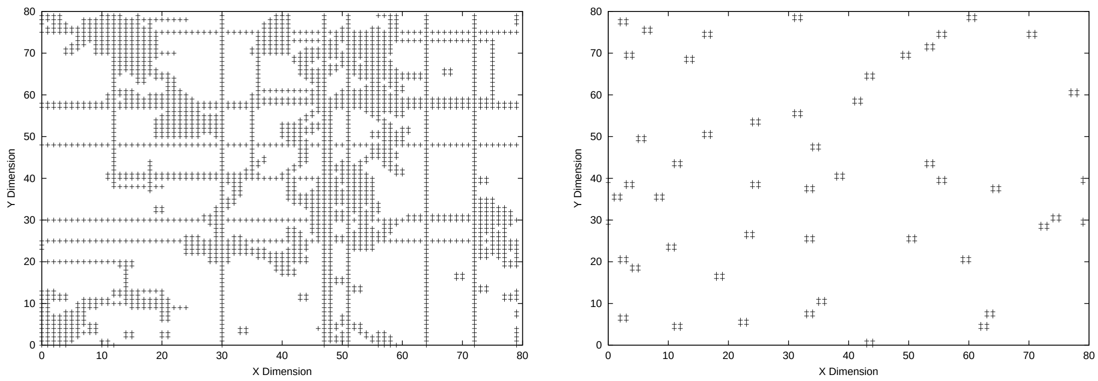
[Fer96] Jacques Ferber. Reactive DAI: Principles and applications. In Greg O'Hare and Nick Jennings, editors, *Foundations of Distributed Artificial Intelligence*, chapter 11, pages 287 – 314. John Wiley and Sons, 1996.

[FG97] Stan Franklin and Art Graesser. Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. In Jörg P. Müller, Michael J. Wooldridge, and Nicholas R. Jennings, editors, *Intelligent Agents III*, pages 21–35. Springer, 1997.

[Hay94] Simon Haykin. *Neural Networks. A Comprehensive Foundation*. MacMillan, 1994.

[HMS95] Reinhold Huber, Helmut A. Mayer, and Roland Schwaiger. netGEN - A Parallel System Generating Problem-Ada pted Topologies of Artificial Neural Networks by means of Genetic Al gorithms. In *Beiträge zum 7. Fachgruppentreffen Maschinelles Lern en der GI-Fachgruppe 1.1.3, Forschungsbericht Nr. 580, Dortmund*, August 1995.

[Int99] Internet. What is Artificial Life?, 1999.



(a) Density plot of places visited by agents in generation 0.

(b) Density plot of places visited by agents in generation 500.

Figure 22: Density plots of places visited by agents in generation 0 and 500 for experiment 5.

[Lev91] James R. Levenick. Inserting Introns Improves Genetic Algorithm Sucess Rate: Taking a Cue from Biology. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of the Fourth International Conference on Genet ic Algorithms*, pages 123–127. University of California, San Diego, Morgan Kaufmann, 1991.

[Mat93] M.J. Mataric. Designing emergent behaviors: From local interactins to collective intelligence. In Meyer et al. [MRW93], pages 433 – 441.

[May97] Helmut A. Mayer. *ptGAs - Genetic Algorithms Using Promoter/Terminator Sequences - Evolution of Number, Size, and Location of Parameters and Parts of the Represe ntation*. PhD thesis, University of Salzburg, 1997.

[ME86] J. L. McClelland and H. Elman. In *Parallel Distributed Processing vol 1 and 2*. MIT Press, 1986.

[MNT96] O. Miglino, K. Nafasi, and Ch. E. Taylor. Selection for Wandering Behavior in a Small Robot. *Artificial Life*, 2(1):101–116, 1996.

[MRW93] J.-A. Meyer, H.L. Roitblat, and S.W. Wilson, editors. *From animals to animats 2: proceeding of the Second International Conference on Simulation of Adaptive Behavior*. MIT Press, 1993.

[MTH89] Geoffrey F. Miller, Peter M. Todd, and Shailesh U. Hegde. Designing Neural Networks using Genetic Algorithms. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Geneti c Algorithms*, pages 379–384, San Mateo, California, 1989. Philips Laboratories, Morgan Kaufmann Publishers, Inc.

[Rey87] C. W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. In M. C. Stone, editor, *SIGGRAPH '87 Conference Proceeding*, volume 21, pages 25–34, 1987.

[Rey91] C. W. Reynolds. Steering Behaviors for Autonomous Characters. In *Game Developers Conference 1999*, pages –, 1991.

[SM97] Luc Steels and Angus McIntyre. Spatially Distributed Naming Games. Technical report, VUB AI Laboratory, 1997.

[WD93] G.M. Werner and M.G. Dyer. Evolution of herding behavior in artificial animals. In Meyer et al. [MRW93], pages 393 – 399.

[Wei95] Gerhard Weiß. Adaptation and Learning in Multi-Agent Systems: Some Remarks and a Bibliography. In Gerhard Weiß and Sandip Sen, editors, *Adaptation and Learning in Multi-Agent Systems*, pages 1–21. Springer, 1995.

[Wu96] Annie Siahung Wu. *Non–Coding DNA and Floating Building Blocks for the Genetic Al gortihm.* PhD thesis, University of Michigan, 1996.

[ZT98] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN V*, pages 292–301. Springer, September 1998.