Salzburg Interest Group on Integrated Systems

siGis-001-2001 March 6, 2001

Evolution of Wandering Behavior in a Multi Agent System: An Experiment

2001 WSES International Conference on Neural Networks and Applications
February 11-15, 2001, Puerto De La Cruz, Tenerife, Spain

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
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

Evolution of Wandering Behavior in a Multi Agent System: An Experiment

Roland Schwaiger and Konrad Lang
Department of Computer Science,
University of Salzburg,
Jakob–Haringer–Strasse 2, A–5020 Salzburg
AUSTRIA
{rschwaig}, {klang}@cosy.sbg.ac.at

Abstract: - In this article we discuss our approach to the evolution of *wandering behavior* in a multi agent system (MAS). Our discussion covers the various aspects of the system setup, the performed experiments and the interpretation of the results observed. Utilizing a genetic algorithm (GA) and multi layer perceptrons (ANN) we show how *wandering behavior* is developed provided a single fitness criterion. Finally we conclude by reviewing the proposed experiment and point out some future directions of research.

Key–Words: - Multi–Layer Perceptrons, Multi Agent System, Genetic Algorithm, Wandering Behavior, Artificial Neural Networks

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 a predefined goal 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 movement patterns during their course of wandering influenced by other agents?

Insights and analyses of the raised questions, give us the means to evolve populations of artificial individuals which show complex behaviors, s.a. swarming. The system itself, based on a coarsly defined criterion, 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 predefined goal.

We are especially interested in simple criteria, which induce so-called *wandering behavior* in our MAS. By wandering we mean moving in a topologically defined environment. *Wandering behavior* implicitly covers other behaviors, s.a. collision avoidance and movement patterns with other agents. Our goal is to investigate closely some of the byproducts of wandering and wandering itself. Henceforth two distinct aspects, namely the quantitative and the qualitative level ([1]) of our simulation will be discussed. On the one hand the quantitative level will give insights in the overall performance of the MAS with respect to the selected criterion. On the other hand, the qualitative level will provide explanations for the individual behaviors of the agents.

Since our experiments are intended to simulate mass behavior we choose to facilitate reactive agents [1]. 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. [2]).

Wandering governing information is drawn from positional information of agents within a neighborhood of a single agent. We will give the means to compactly represent the response of an agent where evolved behavior may be interpreted easily at the individual level by a so-called *neuro controller output*.

Although our view of agenthood is inspired by artificial life where the term "agent" is hardly used we prefer to use this term to denote the artificial beings in our simulations. We may describe the agents in our simulation, from a distributed artificial intelligence perspective (DAI, e.g. [3]), as: reactive (they sense agents in their neighborhood and react (wander) upon them), autonomous (they control their wandering behavior), 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), temporarily continous (per generation since we investigate a generational based evolutionary system contrary to a steady state system), not learning (they do learn evolutionarily but do not learn during their lifetime), mobile (they wander from one location to the next location in their environment), not flexible characters (their actions are scripted).

This work is organized in the following way: in section 2 related work is reviewed and discussed. 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. Section 4 discusses the results of the conducted experiments and finally section 5 gives the conclusions drawn from our experiments.

2 Related Work

Miglino et al. [4], 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 was characterized by sharply stepped periods in contrast to periods of stasis and (c) simulated and real robots behaved quite similar.

Reynolds ([5], [6]) researched 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 implicitly develop basic rules for wandering in the world.

Baray [7] developed a system in which agents were able to extend their life spans by coordinating their actions via undirected communication. A homogeneous population of agents populates the world, where each agent has a health value. Further mobile predators are part of the environment contrary to our system, where the whole population is homogeneous. All of the agents behaviors are based on predefined rules.

In [8], Mataric's research considers social interactions leading to purposive group behavior. She stated that analyzing and predicting the behavior of a single situated agent is an unsolved problem in robotics and AI. 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 the agents in a particular population are homogeneous. As we will see in our system such a collision avoidance behavior evolves automatically but has, at the time of writing, to be interpreted by a human observer.

Werner and Dyer [9] created a simulated word, called BioLand, where they simulated the evolution of herding behavior in prey animals. Additionally a population of predators were put in the simulation. An evolutionary pathway was seen to this herding, from aggregation, to staying nearby other animals for mating opportunities, to using herding for safety and food finding. They modeled the behavior of the biots by means of artificial networks.

3 Experimental Setup

We conduct our experiments in an infinite two dimensional world where the range of the x and y dimension, e.g. $x = 10, y = 10$, are given, respectively. The region is not constrained, i.e. the agents move on a torus (e.g. [10]).

The life time for an agent is measured in discrete time steps, e.g. $N_t = 100$, independent of the system time of the executing hardware device. The world is populated with a predefined number of agents, e.g. $N_a = 10$, grouped in populations, e.g. $N_p = 100$, which live their lifes according to the following phases: (1) If an experiment is started, the positions of the agents $a_i^0 \in \mathcal{A}^0$ and their neuro controllers are initialized, otherwise the evolved neuro controllers from the previous generation are imported. (2) In each step during their life span the agents $a_i^p \in \mathcal{A}^p$, $p = 1, \ldots, N_p$, perform a move according to the output of their neuro controller. (3) At the end of the life span of each agent the fitness is determined. Evolutionary mechanisms produce the offsprings for the next generation which starts its life cycle as described above.

3.1 Our Agent

The agent, as shown in figure 1, is depicted as a triangle with one tip colored, indicating the heading direction. The agent may move according to a finite set of directions, i.e. $\{FRONT, BACK, LEFT, RIGHT\}$, with respect to its local coordinate system. No initial direction is defined for the agent. After one step the agent changes its location on the underlying grid and its heading direction accordingly.

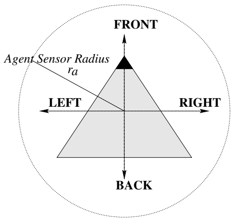


Figure 1: The agent with its sensors and possible moving directions. The agent sensor radius r_a determines the neighborhood where other agents are sensed.

The agent has the ability to detect other agents in its agent neighborhood through a so-called agent sensor, where the range of the sensor is depicted by a circle in figure 1 with sensor radius r_a, e.g. $r_a = 3$.

3.2 The Neuro Controller (NC)

A neuro controller gives the possible directions for the agent to move through the world. As seen in figure 2, the neuro controller consists of a multiple layer perceptron as proposed by [11] with a winner–takes–all ([12]) 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.

The inputs for the neuro controller are polar coordinates indicating the position of a randomly chosen agent a_j within r_a of a_i, with respect to the local coordinate system of a_i. 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 evolutionarily.

Four possible outcomes are represented at the output layer where the actual outputs observed at the output layer indicate the steering direction.

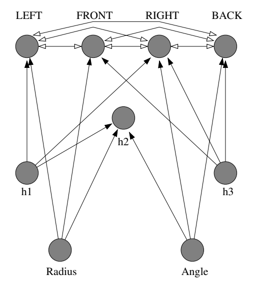


Figure 2: A possibly evolved neuro controller topology with two input neurons (radius r, angle ϕ), a variable number of hidden units (maximum four) and four output units organized in a winner-takes-all layer.

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. [13]. 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 [14] are the *Neuron Markers* which are encoded in a separate section on the ANN chromosome - the four other sections contain steering parameters, 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 3). As a consequence, most ANN chromosomes contain non–coding regions [1], 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 [15] [16] [17].

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.

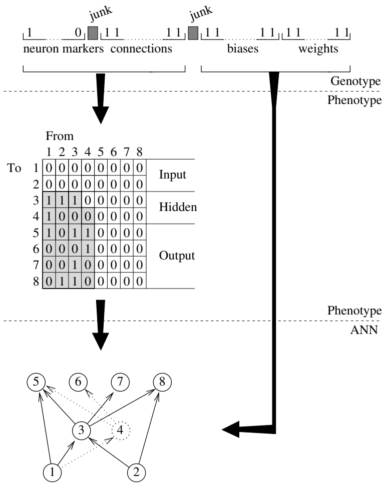


Figure 3: The encoding of the neuro controller following the direct encoding scheme.

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 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.

4 Experiment

In this section the averaged results of 10 experiments with the setup defined in section 3 will be discussed: $x = 10, y = 10, N_t = 100, N_p = 100, N_a = 100, r_a = 3$. For the GA we used a standard 2-point crossover and mutation probability $1/l$, where l denotes the length of the genotype, and tournament selection.

The average and maximum number of places visited ($\mathcal{C}_{\mathcal{PV}}$) by the agents is shown in figure 4. 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 average 55 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 *wan-*

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

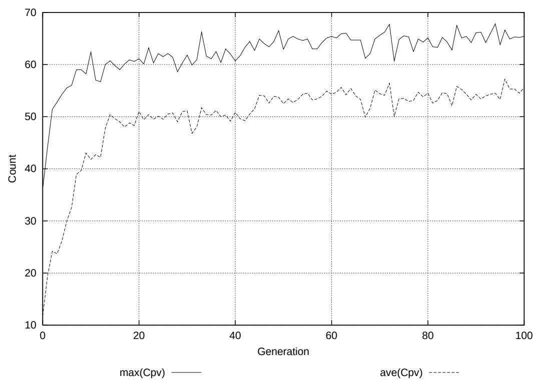


Figure 4: Graph of evolution of the agents according to the criterion $\mathcal{C}_{\mathcal{PV}}$, i.e. number of places visited.

dering behavior is resulting from the neuro controller output.

In the initial phase of the experiment the majority 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 10 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 $\mathcal{C}_{\mathcal{PV}}$:

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

While this basic "rule" evolves and becomes common for all agents, the normalized population diversity, as defined by

Definition 1 *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_k denotes the Modified Miller Matrix of a_k.

Definition 2 *The population diversity D_i of a set of agents $\mathcal{A}^i$ in 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 3 *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.

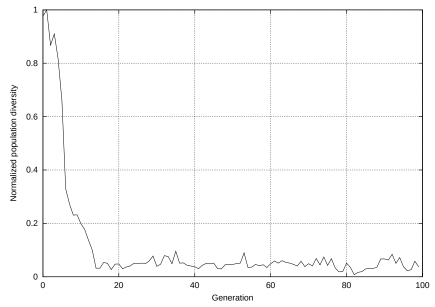
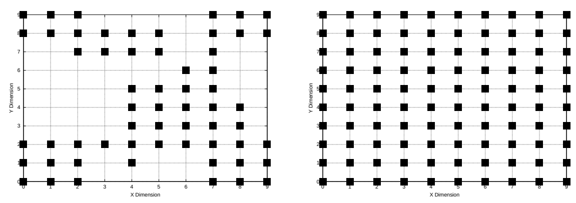


Figure 5: Normalized population diversity for the whole simulation.

decreases, as seen in figure 5.

A value close to 1.0 in figure 5 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.



(a) Generation 0. (b) Generation 100.

Figure 6: Density plots of places visited by agents in generation 0 and 100. A square denotes a place visited by at least one agent

In figure 6 the density plots of places visited by agents in generation 0 and 100 are shown. As can be seen from figure 6(a) agents implement different wandering behaviors. Some perform a left or right rotation, others implement a behavior according to the simple rule derived above. A minor class of agents show a behavior to maximize the number of different place the agent visits, which is dominant in generation 100 as can be seen from figure 6(b).

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 of neuro controller topologies although we have seen that in generation 100 different topologies encode the same behavior. In generation 100 we observed a changed behavior which can be stated as the following rule:

```
IF no agents are in sensor range THEN
  Make a step with direction FORWARD
ELSEIF an agent is within sensor range
AND the agent is not on the right THEN
```

```
  Make a step with direction FORWARD
ELSE
  Make a step with direction LEFT
ENDIF
```

If we analyze the *wandering behavior* of two agents in generation 100 we realize that there are a number of movement patterns if two agents sense each other. Three of the movement patterns are discussed, where the dominating neuro controller output is shown in figure 7.

Movement pattern 1. The agents a_i and a_j move towards each other where each agent senses the other agent in the right hemisphere of the agent sensor. As seen from the output of the sensor, both agents perform a move to the left. 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.

Movement 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 left 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 right side. Hence it will change its direction and turn left. The two agents will perform a parallel observation of the environment.

Movement 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.

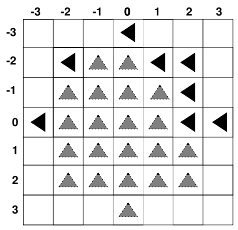
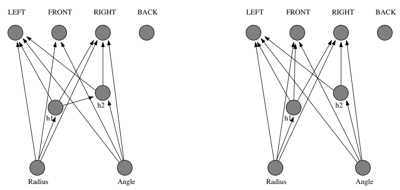


Figure 7: Dominating neuro controller output in generation 100.

As the movement patterns show, the agents influences on the course of another encountered agent vary from high to low. As it was outlined above, a major reason for the successful 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 8 two topologies of the evolved neuro controllers in generation 100 are shown. The topologies in



(a) NC 1. (b) NC 2.

Figure 8: Some examples of evolved neuro controller topologies with a varying number of hidden units in generation 100.

figure 8(a/b) correspond to the output in figure 7. It is noteworthy that all of the networks shown have an isolated output unit which is responsible for the BACK response. 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 significantly as the FORWARD output does.

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 and a multi layer perceptron into an evolutionary MAS. 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 comparable outputs where an implicit pruning of the output layer was performed. This pruning was not explicitly encoded into the system nor demanded by the fitness criterion.

The evolved wandering behavior is invariant of the starting position of the agents in the simulation. This was shown by discussion of movement 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 right side, then turn left.”

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 with respect to graceful degradation of the observed behaviors and the extension of the system into an open environment were the number of agents is dynamic.

Bibliography

[1] 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.

[2] Afsaneh Haddadi. *Communication and cooperation in agent systems: a pragmatic theory*, volume 1056 of *Lecture Notes in Artificial Intelligence and Lecture Notes in Computer Science*. Springer-Verlag Inc., New York, NY, USA, 1996.

[3] Munindar P. Singh. *Multiagent Systems. A Theoretical Framework for Intentions, Know-How, and Communication*, volume 799 of *Lecture Notes in Artificial Intelligence and Lecture Notes in Computer Science*. Springer-Verlag Inc., New York, NY, USA, 1994.

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

[5] 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.

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

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

[8] M.J. Mataric. Designing emergent behaviors: From local interactins to collective intelligence. In Meyer et al. [18], pages 433 – 441.

[9] G.M. Werner and M.G. Dyer. Evolution of herding behavior in artificial animals. In Meyer et al. [18], pages 393 – 399.

[10] J. D. Foley, A. van Dam, St. K. Feiner, and J. F. Hughes. *Computer Graphics. Principles and Practice*. Addison-Wesley, 1990.

[11] D. E. Rumelhart and J. L. McClelland. Parallel distributed processing. exploration in the microstrcuture of cognition. volume 1: Foundation. In *Parallel Distributed Processing vol 1 and 2*. MIT Press, 1986.

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

[13] 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.

[14] 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.

[15] 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.

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

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

[18] 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.