ORIGINAL ARTICLE

Craig Mautner · Richard K. Belew

Evolving robot morphology and control

Received and accepted: January 19, 1999

Abstract Most robotic approaches begin with a fixed robot hardware design and then experiment with control structures. We take a different approach that considers both the robot hardware and the control structure as variables in the evolutionary process. This paper reports the results of experiments which explore the placement of sensors and effectors around the perimeter of a simulated agent's body, and the neural network (NNet) that controls them.

Introduction

Evolutionary algorithms have been used in the design of robot controllers for some time and with great success. Almost all work on evolving autonomous agents and robots has focused on the evolution of the control structure, often a neural network (NNet). This work has taken as a basic assumption that the agent body is immutable. This is a consequence of the flexibility of software compared with hardware; in the past it has not been feasible to explore the space of robot morphology. In contrast to this, the course of natural evolution shows a history of body, nervous system and environment all evolving simultaneously in cooperation with, and in response to, each other. The research described below investigates the interaction between coevolved body and control structures.

The most successful agents we know of are those found in real life. These agents are well adapted to their environment and can handle many small and large surprises in their world and to themselves without failures. Because of this, we look to biology for much of our inspiration for this work. However, in contrast to some biorobotic models, we are not trying to reflect the results of our work back to the biology that inspires it.

The methods by which we investigate this is to specify the agent's body and NNet using a grammar. Grammars offer modularity in terms of encompassing detailed structures at various levels of granularity. That is, a grammar can provide a compact representation for complicated and repeated structures. By using grammars, we are able to build hierarchical solutions to problems based on the solutions found at lower levels. These grammars are then evolved using common genetic algorithm techniques based on the performance of the instantiated agent.

Many aspects of this research have been investigated in isolation by others. Examples of evolved robots that have implemented NNet controllers include six-legged walking controllers,[1,2] maze following,[3] predator–prey behavior,[4] and food tracking.[5]

An early experiment applying grammatical models to the construction of feed-forward NNets was reported by Kitano.[6] Other researchers who have used grammars to develop NNets are Gruau et al.[1] and Lucas.[7]

Very little earlier work on evolving morphology exists. Menczer and Belew[8] investigated the nature of sensor usage by providing their agents with an evolvable NNet connected to sensors and effectors. Mark et al.[9] explored eye types and positioning. Sims[10] demonstrated a simulation where the complete morphology of the individuals was involved. Sims created an artificial world in which each agent was grown from a genome that defined both the physical structure and the control structure. Eggenberger[11] has developed an evolutionary system that simulates the growth of a body based on differential gene expression. Lee et al.[12] have also worked on evolving both control structures and body plans.

C. Mautner · R.K. Belew (✉)
Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093-0114, USA
Tel. +1-858-539-0548; Fax +1-858-539-0549
e-mail: rik@cs.ucsd.edu

This work was presented in part at the Fourth International Symposium on Artificial Life and Robotics, Oita, Japan, January 19–22, 1999

Background

We have carried out several experiments based on the paradigms listed above. The experiments begin with a population of genome strings which are converted to individual grammars. The grammars are used to generate agent bodies and their associated NNet control systems through a development process that transforms a single undivided cell (the gamete) into a body consisting of several interconnected cells.

The body/controller pair is then evaluated in the context of an environment, and the genomes for the more successful individuals are preferentially selected for reproduction in the next generation. The process of cell division, neural network extraction, and fitness evaluation are described in the following sections.

Production rules

The grammar consists of an alphabet of terminal and nonterminal symbols, a set of production rules, and a single starting symbol from the set of nonterminals. In our experiments, the terminals are the hexadecimal characters {0–9a–f}, and the nonterminals are a subset of the uppercase characters {A–Z}.

Each production rule is made up of a left side and a right side. The left side is a single nonterminal symbol; the right side is a specification for the creation of one or two cells. A nonterminal may appear as the left side of more than one production rule.

The specification of each cell defines it to be either a single nonterminal symbol or a terminal cell which is a list of directed, weighted edges from the sides of the cell to each other (Fig. 1). The production rule also contains the orientation of the cell (whether it is horizontally or vertically flipped). Specifying a cell to be a nonterminal allows it to be subject to the application of further production rules.

A production rule also specifies whether the nonterminal produces one or two cells, and if two, the relative position (i.e., above, below, to the left of, or to the right of) of each.

Cell division

As described in the previous section, each cell is either labeled with a nonterminal symbol or is a terminal cell. Initally the gamete is labeled with the starting nonterminal symbol of the grammar, and cell differentiation proceeds by selecting and applying the rules of the grammar. For each cell labeled as a nonterminal, a rule is found whose left side matches the nonterminal. The cell is then replaced with the one or two cells specified by the right side of the rule. If there is no matching rule, the cell is replaced with a terminal cell with random weights and edges.

The process of replacing nonterminals with terminals and nonterminals continues until there are only terminals left or a maximum depth of replacements has occurred. The rules are applied only a limited number of times to keep rules of the form A → A from generating an infinite regress. In our experiments, the maximum number of cell divisions was set as six. This permits a body to have at most 64 cells (one cell divided in half six times produces $2^6 = 64$ cells). A derivation that continues beyond the sixth rule application will replace the cell by a terminal cell with random weights and edges.

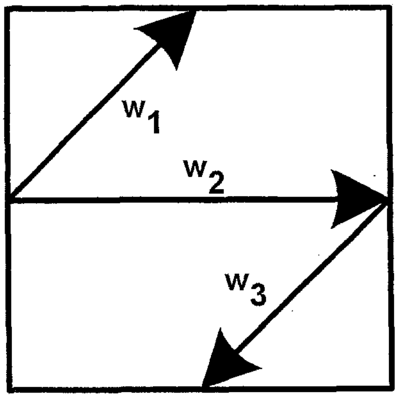


Fig. 1. Terminal cell example

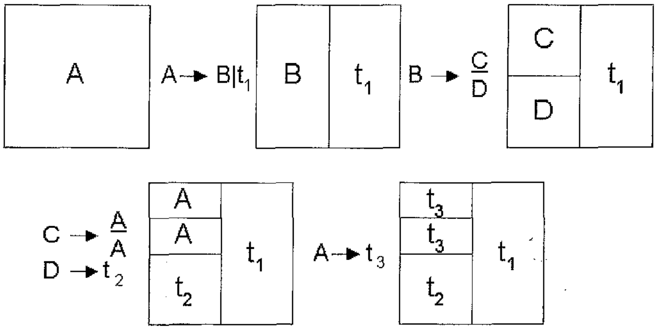


Fig. 2. Development process

A derivation that takes four generations is shown in Fig. 2. In this example the gamete is labeled with the starting symbol, A. The production rule A → B|t_1 indicates that the nonterminal A is converted into two cells. The first cell, a nonterminal B, is to be placed to the left of the second, which is the terminal cell, t_1.

Neural network interpretation

Once the cell division is complete, the body consists of a set of cells that have within them directed, weighted edges. The cells and edges are interpreted as sensors, effectors, and the neural networks that connect them.

Each side of a cell has two values associated with it, an input value and an output value. The input value is dependent on the output values of all adjacent cells. The output value of a side is dependent on the input value of all sides and the internal weighted edges between the sides. Both the input and output values are described in detail below.

Network edges within a cell

There are three types of edges that can exist within the cell: corner, cross, and tonic. Corner edges run from one side to an adjacent side. Cross edges run from one side to the opposite side, and tonic edges are interpreted as bias weights. The three types are shown in Fig. 3. The edge from I_1 is of the corner type with weight w_1, I_2 is of the cross type with weight w_2, and T_3 is of the tonic type providing activation of value T_3.

The output of a cell is the squashed sum of the weighted inputs to that cell. The weighting is only done across those edges that are defined for that cell. In Fig. 3, there are two edges and one tonic defined for the cell pictured. The output of the cell on the right edge is the squashed sum of the products I_1w_1, I_2w_2, and the tonic weight T_3. For all of our experiments, the squashing function is the hyperbolic tangent function: $\tanh(x)$.

Edges between cells

When one or more cells are adjacent to the side of a cell, the outputs of the incident cell are combined to form the inputs to the adjacent cell. This is demonstrated in Fig. 4. In this case, the inputs labeled I_1, I_2, and I_3 are adjacent to the output labeled O_4. Similarly, the input labeled I_4 is adjacent to the outputs labeled O_1, O_2, and O_3. A neural network layer is formed by setting the input value on an edge of a cell equal to the squashed sum of the outputs of the cells adjacent to that edge.

Figures 7 and 9–11 illustrate some final agents' terminal cell divisions and the neural networks that result. In these diagrams, the semicircles pointing outwards are effectors, and those pointing inwards are sensors.

Sensors and effectors

Sensors and effectors are defined by the edges of the terminal cells. Any directed edge that originates from a cell side that is on the perimeter of the body becomes a sensor or input node. Any directed edge that terminates on a cell side that is on the perimeter of the body becomes an effector or output node.

Sensor nodes detect signals of the environment. They provide the input that is propagated through the NNet of the body. Effector nodes are the outputs of the NNet.

Effector nodes provide propulsion to the agent's body. The force of this propulsion is proportional to their output activation. The direction of propulsion of the agent is a result of summing all of these forces based on their position on the body. The vector sum of all effector outputs is broken into two forces. The first is a pressure, which acts through the center of the body and translates the body through the environment. The second force is torque, which acts perpendicular to a line through the center of the body and causes the body to rotate. The net effect is shown in Fig. 5.

Application in a simple environment

In order to judge the effectiveness of a given body plan, and hence the effectiveness of the genome from which the body plan developed, the mature agent is placed within an environment. For our experiments, the environment consists of a 500 × 500 world that has at its center a source of reward which produces a detectable signal that falls off as the inverse square of the distance.

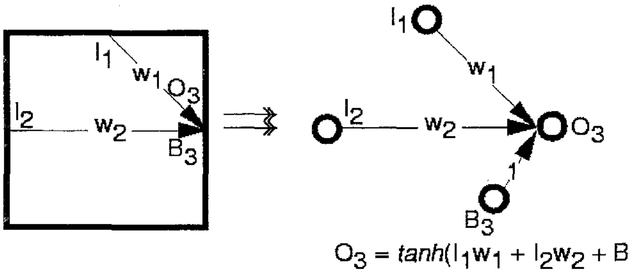
All agents' bodies are set to be 20 units on each edge. The body has a mouth placed at its top. The fitness is measured from the mouth. The perfect agent would turn its mouth toward the center of the world and approach the center in as few time steps as possible.

The agents are placed at a random location in the world and then allowed to wander freely (or just to sit in the case of many agents) for 30 time steps. After 30 time steps the agent is moved to another random location. This is repeated for ten placements for a total of 300 time steps. All agents in a given generation are started from the same set of locations. If an agent gets within a body length of the center of the world, then it is moved to the next location.

The fitness, f_i, at time step i is shown below, where d is the distance from the agent's mouth to the center of the world.

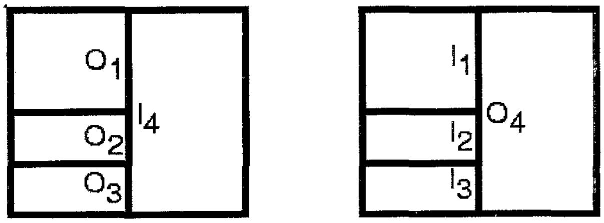
$$f_i = \begin{cases} \frac{1}{d^2} & \text{if } d \geq \text{agent body length} \\ 1 & \text{otherwise} \end{cases}$$

The fitness, F, of an individual is the sum of the fitnesses at each time step over its lifetime.



$$O_3 = tanh(I_1w_1 + I_2w_2 + B_3)$$

Fig. 3. NNet edges formed from within cell connections



$$I_4 = tanh(O_1 + O_2 + O_3) \qquad I_1 = I_2 = I_3 = tanh(O_4)$$

Fig. 4. NNet edges formed from connections between adjacent cells

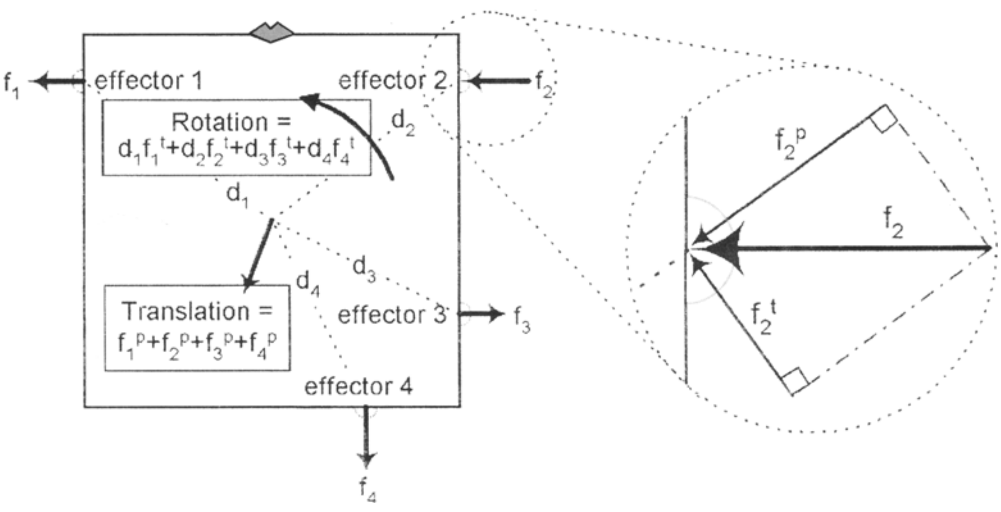


Fig. 5. Conversion of effector outputs to rotation and translation

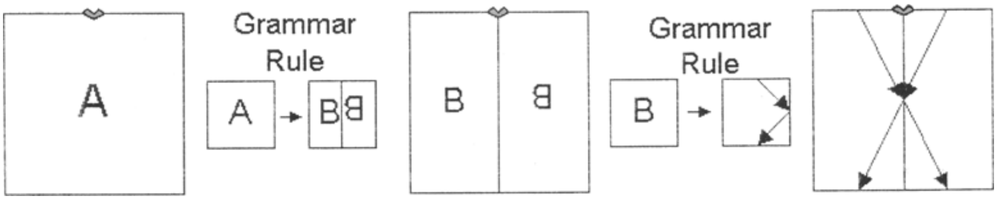


Fig. 6. Grammar for generating a Braitenberg vehicle

The Braitenberg vehicle

A classic design in the field of robotic control is the Braitenberg vehicle 2b.[13] This agent has two sensors on the front and two effectors on the back. The agent's body is bilaterally symmetric, with each sensor connected via a positive weight to the effector on the opposite side. The effect of this connectivity is to steer the agent to the side with the stronger sensor. The effectiveness of this design has been demonstrated in a number of robots.

A simple grammar that generates a complete Braitenberg body is shown in Fig. 6. The grammar consists of two rules. The first rule rewrites the undifferentiated cell body (start symbol A) into two nonterminal B cells, one of which is horizontally flipped relative to the other. The second rule converts a B nonterminal cells into a terminal cell with two edges defined. Figure 7 shows Braitenberg's vehicle 2b alongside the one generated by this hand-designed grammar. The parsimonious nature of the grammar that generates the Braitenberg vehicle under our system shows the representational adequacy of the grammar system.

Preliminary results

Using the experimental platform described above, we ran several experiments to determine the effectiveness of our approach. Three such results are described in the graph in Fig. 8. The results show the average of the best agent over ten different runs with the same starting conditions. The first experiment is our hand-designed Braitenberg agent which is run in the environment for 100 generations. Its performance is shown by the short line around a fitness of 26. This experiment provides a baseline with which we may compare other results.

The top line of the graph begins with the same Braitenberg agents as in the first experiment, but in this experiment they are allowed to evolve. Their performance shows a significant improvement when compared with the original run of Braitenberg agents. All ten populations converged by the end of the run, although they each converged on different solutions. One commonality of all solutions was the preservation of the original *bilateral symmetry* that was part of the Braitenberg population prototype. However, in about half of the final populations the connections were positive and crossed as in the Braitenberg, but surprisingly the remaining populations converged on solutions that had negatively weighted connections straight down the body rather than across it. Most of the solutions also added two negative bias weights to the front of the agent which acted as tractor effectors constantly pulling the body forward. This left new sensors and effectors on the side free to steer the body toward the goal. A typical solution is shown in Fig. 9.

The final experiment (middle line) initialized the population with random genomes. In this case, the results varied widely. Some runs were able to find steadily improving solutions within a few hundred generations, while others took quite a bit longer. Some solutions found were Braitenberg-like in that they discovered bilateral symmetry (Fig. 10). Other solutions produced bizarre body types that performed very well (Fig. 11).

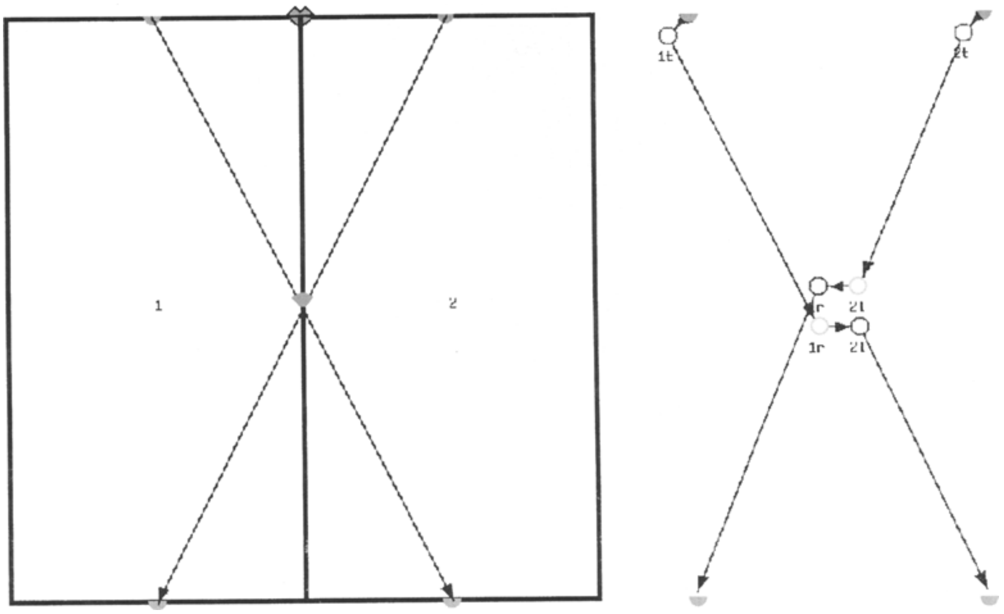


Fig. 7. Braitenberg vehicle with cell and NNet representations

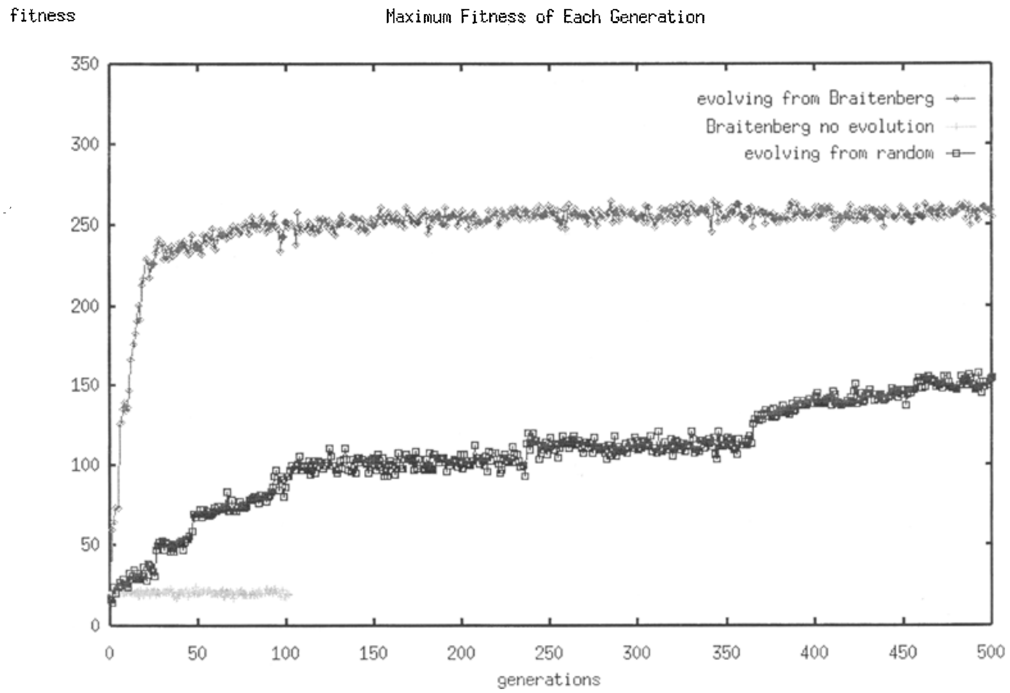


Fig. 8. Average of fitnesses (best of each generation)

Summary

The conclusion that we can draw from these results is that even in this very simple world a wide range of potential solutions exist. Many of these solutions are superior to the known good a priori Braitenberg vehicle solution. It was shown that we can evolve agents whose body and control structures are tightly coupled in order to solve the simple environment presented here. The fact that the system found so many solutions is encouraging as we apply it to more complex environments.

Our current work involves applying these techniques to real 'robots, including Kheperas and LEGO-bots whose body shapes and sensor and effector placement will be determined by the evolved grammars.

References

1. Gruau F, Whitley D, Pyeatt L (1995) Cellular encoding applied to neurocontrol. In: Eshelman L (ed) International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo

Fig. 9. Agent and NNét evolved from Braitenberg

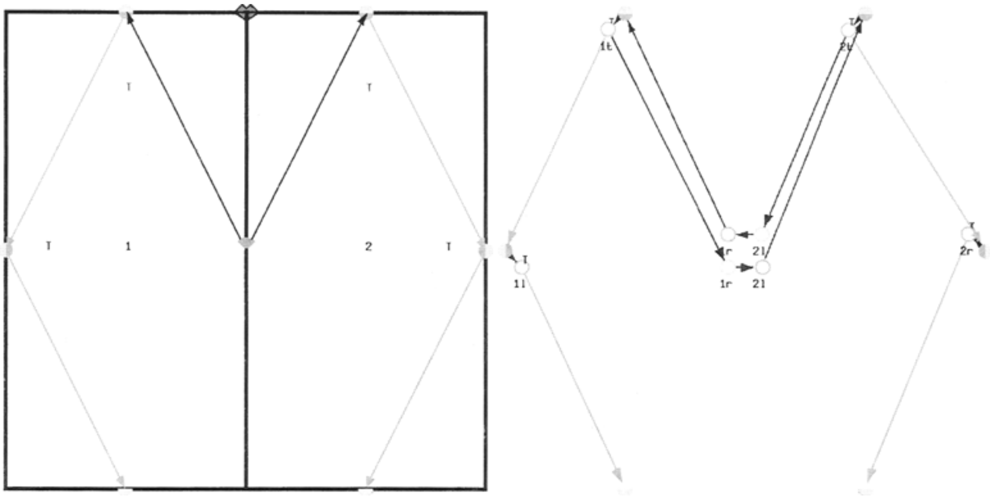


Fig. 10. Agent and NNet evolved from a random genome (example 1)

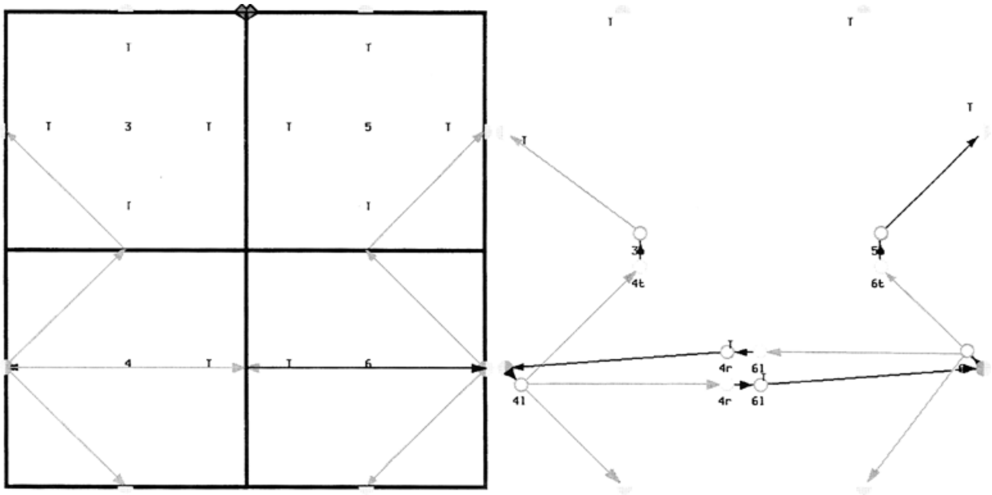
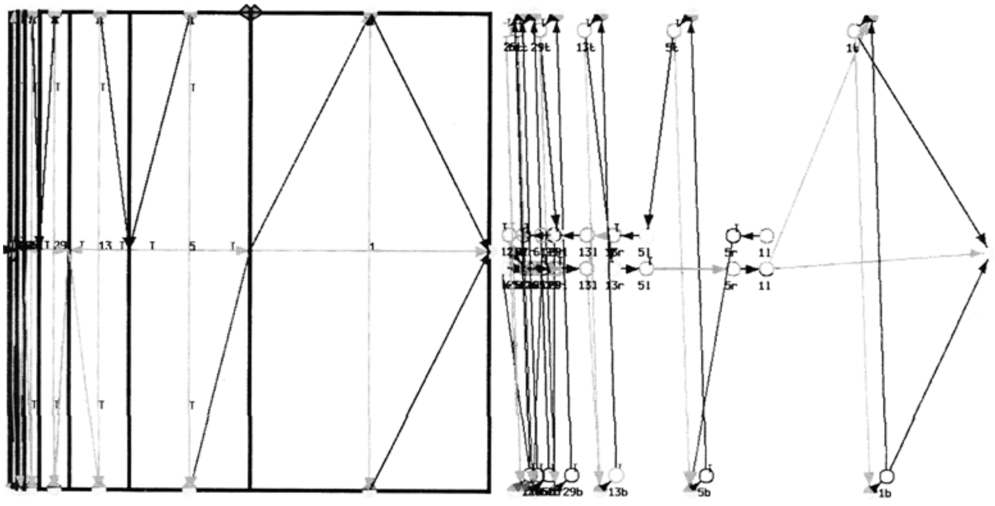


Fig. 11. Agent and NNet evolved from a random genome (example 2)



2. Kodjabachian J, Meyer JA (1997) Evolution and development of neural networks controlling locomotion, gradient following, and obstacle avoidance in artificial insects. http://www.biologie.ens.fr/fr/animatlab/perso/kodjaba/kodjaba.html
3. Floreano D, Mondada F (1994) Automatic creation of an autonomous agent: genetic evolution of a neural network driven robot. In: Meyer J-A, Cliff D, Husbands P, Wilson S (eds) From animals to animats, vol III. MIT Press, Cambridge
4. Floreano D, Nolfi S (1997) Adaptive behavior in competing co-evolving species. In: Husbands P, Harvey I (eds) 4th European Conference on Artificial Life. MIT Press, Cambridge
5. Angeline P, Saunders G, Pollack J (1994) An evolutionary algorithm that constructs recurrent networks. IEEE Trans Neural Networks 5:54–65
6. Kitano H (1990) Designing neural networks using genetic algorithms with graph generation system. Complex Syst 4(4)
7. Lucas SM (1995) Growing adaptive neural networks with graph grammars. In: Proceedings of the European Symposium on Artificial Neural Networks (ESANN '95), pp 235–240. http://esewww.essex.ac.uk/sml/papers.html
8. Menczer F, Belew RK (1994) Evolving sensors in environments of controlled complexity. In: Brooks R (ed) Proceedings of the 4th Conference on Artificial Life
9. Mark A, Polani D, Uthmann T (1998) A framework for sensor evolution in a population of Braitenberg vehicle-like agents. In: Adami C, Belew RK, Kitano H, Taylor CE (eds) Artificial life, vol VI. MIT Press, Cambridge, pp 428–432
10. Sims K (1994) Evolving 3d morphology and behavior by competition. In: Brooks R, Maes P (eds) Proceedings of the International Conference on Artificial Life IV. MIT Press, Cambridge
11. Eggenberger P (1997) Evolving morphologies of simulated 3d organisms based on differential gene expression. In: Husbands P, Harvey I (eds) 4th European Conference on Artificial Life. MIT Press, Cambridge, pp 205–213
12. Lee W, Hallam JC, Lund HH (1996) Hybrid gp/ga approach for co-evolving controllers and robot bodies to achieve fitness-specified tasks. In: Proceeding of the IEEE 3rd International Conference on Evolutionary Programming. IEEE Press, New York, pp 384–389
13. Braitenberg V (1984) Vehicles: Experiments in synthetic psychology. MIT Press, Cambridge