Multi-Agent Robot Simulation for Evolutionary Learning of Cooperative Behavior

Yoichiro Maeda
Faculty of Information Science and Technology
Osaka Electro-Communication University
18-8, Hatsu-cho, Neyagawa,
Osaka, 572-8530, Japan
maeda@mdlab.osakac.ac.jp

Abstract

In this research, the evolutionary algorithm is applied to behavior learning of an individual agent in the multi-agent robot system. Each agent robot is given two behavior duties both collision avoidance from the other agent and target (food point) reaching for recovering self-energy. In this paper, we carried out the evolutionary simulation of the cooperative behavior creating an environmental map for the above-mentioned multi-agent robots. Each agent robot has two conflicted task, that is, local individual behaviors for self-preservation or self-protection and global group behaviors for the cooperative task, and has the additional algorithm of the group evolution which the parameters of the best agent are copied to a dead agent, that is, an agent lost its energy. We also report simulation results performed with the evolutionary behavior learning simulator which we developed for multi-agent robots with the map creation task.

Introduction

Main purpose for the research of multi-agent robot system which constructed with multiple autonomous robots is to distribute system requirements to same autonomous agents and to perform an intelligent behavior cooperating each other according to the state. Recently, researches of the emergence are grown up in the artificial life and several researches of the multi-agent robot are influenced by them.

For example, G.Weiß(Weiß1993) is researching to implement learning functions to the multi-agent robot system. This approach is regarded as a distributed classifier system because agents which have individual behavior rules cooperatively perform a task. M.J.Mataric (Mataric 1995) proposed the generation mechanism of group behavior by the experiments used five heterogeneous agent robots. T.Unemi (Unemi 1993) confirmed to generate the behavior for territory dividing of the group by applying the reinforcement learning to multi agents.

However, in most distributed multi-agent systems, the local behaviors (obstacle avoidance, power consumption, etc.) and global behaviors (cooperative task completion) are at odds in each agent. Therefore, we think both rules of local and global behaviors are influenced to the emergence generating as their mutual relation.

In the first stage of our research, we applied an evolutionary algorithm to the behavior learning in an individual agents for multi-agent robots (Maeda 1996) (Maeda 1997). Multi-agent robots in the simulation were respectively assigned two behaviors both the collision avoidance motion for moving without collision with an another agent and the target reaching motion for recovering its self-energy in a food point. It was confirmed in the simulation that each agent has abilities of behavior learning and group evolution creating better agents. As a result, we reported the agent group with different behavior characteristics is comparatively apt to cause the emergence.

In this paper, as the second stage of our research, we simulated the group evolution for the agents which have a behavior purpose of generating environmental map. And we also report results of the simulation for confirming the effectiveness of the proposed method.

Multi-Agent Robot Behavior

Multi-Agent Robot Model

Each agent individually keeps its self-energy and behavior gains of each agent are controlled by its self-energy. And each agent moves according to behavior gains. Self-energy and behavior gains are decided as shown in Figure 1. An agent has to stop its activity if he keeps no self-energy. Each self-energy is decided according to both local self-preservation tasks of energy consumption by its locomotion, rewards for target reaching (TR) and punishments for collision avoidance (CA) and global cooperative tasks of rewards for the map creation. The food point gain is decided by this self-energy, the collision avoidance gain by minimum distance to the nearest agent. Furthermore, the map search gain, wall avoidance gain and steering gain are obtained in advance. An agent robot moves towards the direction of locomotion which is calculated according to these behavior gains.

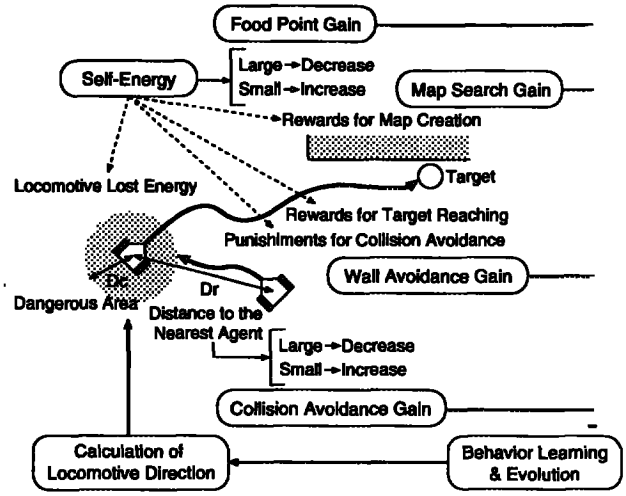

Figure 1: Self-energy and behavior gain

Cooperative Behavior

We explain about the map creation method as the cooperative behavior of agents. In this simulation, some polygonal obstacles are set up in a virtual field and agents and food points for the energy supply are set up in the other place.

For the map creation, at first, virtual field is divided to 3 × 3 square areas with a constant width in the length and breadth (See Figure 2). In each area, the non-search rate (= non-search area / searched area) which is shown as square numbers in this figure. The angle of a vector sum between the vector toward the center in a area with the highest non-search rate and the vector of the locomotive direction is the steering angle for the map search. When a agent sensed an non-search area in the locomotion, the searched attribute is set from the non-search area to the searched area in the point. Total performance of the map creation behavior by all agents is evaluated by the map creation rate (= total area already searched / total search area) which is calculated at every time.

Behavior Learning of Agent

Behavior Rules

Rules for collision avoidance (CA) are described with simplified fuzzy reasoning method to express the behavior like human. Steering angles for collision avoidance are decided by the distance and direction between itself and the nearest agent with simplified fuzzy rules as shown in Figure 3 (Maeda 1996).

Rules for target reaching (TR) are setting so that an agent robot moves toward two goals, the nearest food point (Food Point : FP) and the nearest non-search area (Search Point : SP). The weight for two goals is decided by each gains. Agent robot never go toward the previous food point. It assumed that each

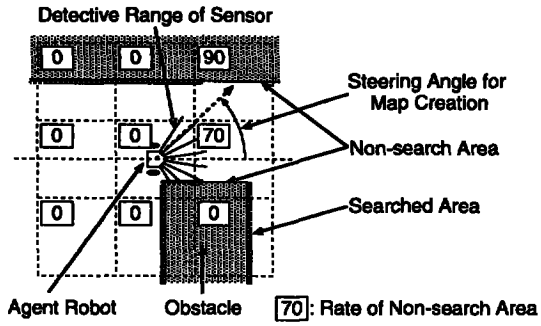

Figure 2: Map creation method of agent

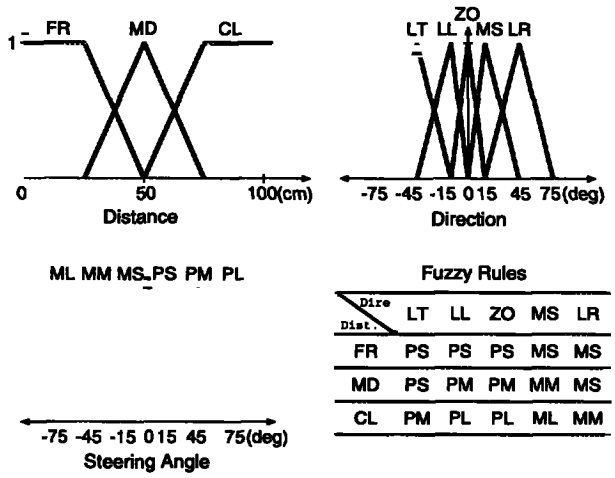

Figure 3: Simplified fuzzy rule for collision avoidance

agent has eight sonar sensors in front at intervals of 15 degrees. By this sensors, agent robots can recognize the position of the other agent and the wall.

Each agent has several parameters, for example, self-energy E_{all}, food point gain G_{fp}, collision avoidance gain G_{col}, fitness value F_{fit} and so on. Self-energy is the energy for action and an agent with no self-energy is regarded as a dead agent. Food point gain shows the strenth of going to the nearest food point and collision avoidance gain the strenth of avoiding the collision for the other agents or the wall. These gains decide the weight of behavior strategy. Fitness shows the degree of adaptation for the environment, that is, the amount of energy and rewards.

Behavior Decision

Absolute positions of all food points in a field are given to each agent. They can also obtain the information of own surrounding environment, for example, positions of the other agents and walls by its sensors. Based on these information, several gains and behaviors are decided.

First of all, the steering angles for going toward the nearest food point (ΔS_{fp}), for avoiding the nearest other agent (ΔS_{col}), for going toward the area with the highest non-search rate (ΔS_{sp}) and for avoiding the nearest wall (ΔS_{wall}) are calculated. ΔS_{fp} is easily obtained by the vector for facing to the absolute position of the nearest food point. ΔS_{col} is decided by the above-mentioned fuzzy rules. ΔS_{sp} is determined by the method mentioned in section 2.2. ΔS_{wall} is decided by the following equations:

$$\begin{aligned} \Delta S_{wall} &= (-\tfrac{1}{2} + \tfrac{3.5 - El_{min}}{12})(\tfrac{S_{lim} \times \pi}{Dist + S_{lim}}) \quad [El_{min} < 3.5] \\ &= (\tfrac{1}{2} + \tfrac{3.5 - El_{min}}{12})(\tfrac{S_{lim} \times \pi}{Dist + S_{lim}}) \quad [El_{min} \geq 3.5] \end{aligned} \tag{1}$$

where

ΔS_{wall}	: Steering angle for avoiding the nearest wall
El_{min}	: Sonar number detected the nearest object
S_{lim}	: Detective range of sonar
$Dist$	: Distance value detected by the sonar.

The final locomotive direction (the steering angle θ) of an agent decided by these steering angles and several gains, that is, the food point gain, collision avoidance gain, wall avoidance gain and the map search gain according to the following equation:

$$\theta = \theta_0 + G_s \times \frac{\Delta S_{fp} \times G_{fp} + \Delta S_{col} \times G_{col} + \Delta S_{wall} \times G_{wall} + \Delta S_{sp} \times G_{sp}}{G_{fp} + G_{col} + G_{wall} + G_{sp}} \tag{2}$$

where

θ_0	: Absolute direction of an agent in the last position
G_s	: Steering gain
ΔS_{fp}	: Steering angle for the nearest food point
G_{fp}	: Food point gain
ΔS_{col}	: Steering angle for the collision avoidance
G_{col}	: Collision avoidance gain
G_{wall}	: Wall avoidance gain
ΔS_{sp}	: Steering angle for the map search
G_{sp}	: Map search gain.

Among these gains, G_s, G_{wall} and G_{sp} are constant values decided in advance and G_{fp} and G_{col} are discussed in the next section.

Behavior Learning

Each agent has the learning mechanism for the behavior controlled by its self-energy and behavior gains given by the results of two conflicted behaviors CA and TR. This mechanism is a kind of the reinforcement learning because the learning process of an agent progresses giving rewards to itself according to results of its behavior. Rewards for several behaviors are given by the following equations. In the equations, + means a reward and − means a punishment.

An agent robot loses its self-energy according to the locomotive amount as follows:

$$E_{loss} = -\Delta E_{loss} \times T \tag{3}$$

where

E_{loss}	: Total energy loss for the locomotion
ΔE_{loss}	: Energy loss per unit time (constant value)
T	: Passing time.

When an agent robot has reached a food point, he is given the reward within the limits of his maximum energy as the equation. On the other hand, energies of the food point are decreased the same amount. A food point vanishes if its all energy are lost and a new food point appears in a random place.

$$\begin{aligned} E_{tr} &= +E_{max} - E_{now} \quad (E_{now} + E_{rewmax} > E_{max}) \\ &= +E_{rewmax} \quad (E_{now} + E_{rewmax} \leq E_{max}) \end{aligned} \tag{4}$$

where

E_{tr}	: Reward for the target reaching
E_{max}	: Initial self-energy (Maximum energy)
E_{now}	: Current energy of an agent
E_{rewmax}	: Maximum reward given in a food point.

An agent A is given the following punishment if the nearest agent B exists within the constant dangerous area Dc centering around the agent A. On the other hand, this agent A obtains the reward only when he could avoid the agent B out of the dangerous area as well:

$$E_{ca} = -\frac{G_p \times E_{max} \times (D_c - D_r)}{D_c} \quad \begin{matrix} [D_c > D_r\text{: punishment}] \\ [D_c < D_r and D_r < 2D_c\text{: Reward}] \end{matrix} \tag{5}$$

where

E_{ca}	: Punishment (or Reward) for the collision avoidance
G_p	: Punishment (or Reward) gain
D_c	: Radius of dangerous area
D_r	: Distance from agent A to the nearest agent B.

Furthermore, when an agent successed the map creation it obtains the following reward.

$$\begin{aligned} E_{sp} &= +E_{max} - E_{now} \quad (E_{now} + E_{spmax} > E_{max}) \\ &= +E_{spmax} \quad (E_{now} + E_{spmax} \leq E_{max}) \end{aligned} \tag{6}$$

where

E_{sp}	: Reward for map creation
E_{spmax}	: Maximum reward for map creation.

After all, the total self-energy of an agent robot is calculated with the following equation:

$$E_{all} = E_o + E_{loss} + E_{tr} + E_{ca} + Esp \tag{7}$$

where

E_{all}	: Total self-energy
E_o	: Total self-energy in last position.

According to the amount of the energy E_{all} an agent robot keeps, the food point gain and the collision avoidance gain are modified as follows:

$$\begin{aligned} G_{fp} &= G_{fp0} + G_{fpup} \quad (E_{all} < E_{low}) \\ &= G_{fp0} - G_{fpdown} \quad (E_{all} \geq E_{low}) \end{aligned} \tag{8}$$

where

G_{fp0} : Food point gain in last position
G_{fpup} : Increasing amount of food point gain
G_{fpdown} : Decreasing amount of food point gain
E_{high} : Upper threshold of energy in the gain tuning
E_{low} : Lower threshold of energy in the gain tuning.

$$\begin{aligned} G_{col} &= G_{col0} - \frac{D_c - D_r}{D_c} \times G_{coldown} \quad (D_c > D_r) \\ &= G_{col0} + \frac{2D_c - D_r}{D_c} \times G_{colup} \quad (D_c < D_r and D_r < 2D_c) \end{aligned} \tag{9}$$

where

G_{col0} : Collision avoidance gain in last position
G_{colup} : Increasing amount of collision avoidance gain
$G_{coldown}$: Decreasing amount of collision avoidance gain.

By modifying these two gains in the online simulation, it is performed the behavior learning, that is, the weighting between TR and CA.

Group Evolution

In this section, the evolutionary mechanism of agent robots is discussed. We are able to regard the total self-energies and rewards as the evaluation value for the robustness of the agent in changing of its environment as well as the creature evolution. Therefore, agent groups perform evolutionary selection based on the following rule : If $E_{all} < 0$, then the agent dies and the agent with the largest self-energy in the group is multiplied.

In this simulation, only when an agent is dead, parameters of the agent with the highest fitness value (defined below) among all the agents. The fitness value of an agent is decided with the following equation (10).

$$F_{fit} = E_{ave} + P_{ave} \tag{10}$$

F_{fit} : Fitness value of an agent
E_{ave} : Weighting calculated average of self-energy in last 10 times
P_{ave} : Weighting calculated average of rewards and punishments in last 10 times.

Our algorithm with the above-mentioned behavior learning and group evolution is shown in Figure 4. Each agent decides the desired direction with input information and parameters of agents are modified according to locomotive condition. An agent without its self-energy evolves and that with some self-energy learns. In the simulation, these processes are repeated until 3000 generations.

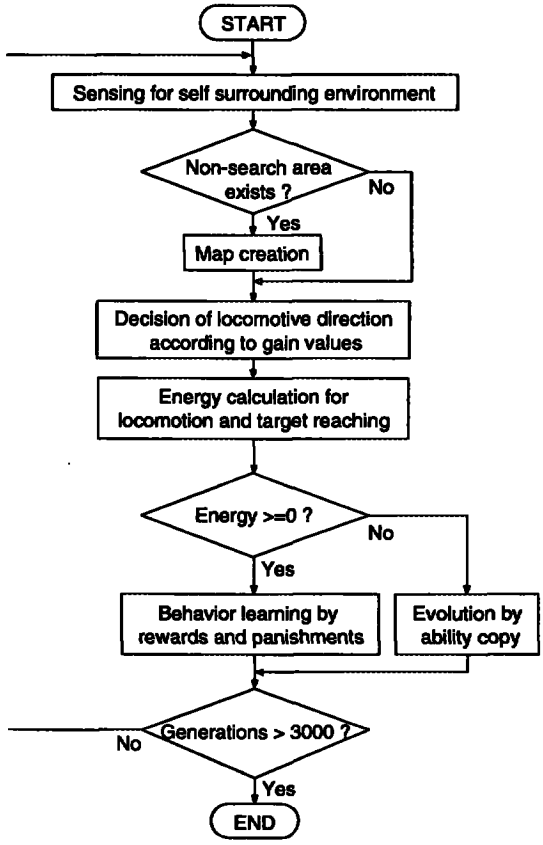

Figure 4: Flowchart of this method

Simulation

Evolutionary Simulator

We developed the evolutionary behavior learning simulator (See Figure 5-Figure 7) produced with C language, X library and Motif window system on Sun Sparc Station (SS-5) to prove the effectiveness of our algorithm. The window of this simulator mainly has four simulation areas : a field area that agents move around (in upper left side), a bar graph area displayed self-energies, collision avoidance gains and food point gains (in upper right side), a time-scale graph area displayed fitness values of each agent and a map creation rate (in lower side) and slider knobs for user's selection of parameters (in right side). When an agent performs map creation (recognition of the non-search area), the wall of the obstacle changes from pink to black. The buttons and slider knobs in right side were developed by using Motif widget. User can control Start, End, Pause and Restart of the simulation by using the buttons and can change almost parameters in the simulation in real time by using slider knobs.

Simulation Results

In this research, we performed several simulations with different initial parameters which all agents have the same initial parameters. Conditions of the simulation we report in this paper are shown in Table 1. The simulation used 12 agents. Initial condition #1 shows in case of the agent group which makes more of individual behaviors than cooperative behaviors (for example, target reaching or collision avoidance). In this case, map search gain is set to zero. Initial condition #2 shows in case of the agent group which makes more of cooperative behaviors than individual behaviors. The rate of energy supply is a little high in this case. Initial condition #3 shows in case of the agent group which makes too much of cooperative behavior (map creation). In this case, parameters are set so that each agent is biased against all behaviors but map creation.

Table 1: Initial parameters of agent robot

Initial Condition	#1	#2	#3
Agents	20	20	20
Energy	100	100	100
E_{max}	200	200	200
E_{high}	0.95	0.95	0.80
E_{low}	0.25	0.35	0.15
G_{fp}	5.0	5.0	5.0
G_{fpmax}	10.0	10.0	10.0
G_{col}	5.0	5.0	5.0
G_{colmax}	10.0	10.0	10.0
G_{sp}	0.0	5.0	10.0
G_{wall}	5.0	5.0	5.0
G_{fpup}	0.40	0.40	0.40
G_{fpdown}	0.02	0.02	0.02
G_{colup}	0.50	0.50	0.50
$G_{coldown}$	0.20	0.20	0.05

Simulation results performed in above-mentioned initial conditions are shown in Figure 5-Figure 7.

- In case of initial condition #1

  When the map search was not performed immediately after starting simulation, each agent does not almost move toward the food point because the food point gain decreases by the energy obtained by map creation. However, when each agent finished to create maps near the initial position, the food point gain begins to increase. In this time, agents formed a dumpling group and map creation behavior was not stable. Map creation rate almost continued 0.5 since a certain time. In several simulations with different initial position of agents, we also confimed that environmental map was created slowly.

- In case of initial condition #2

  After simulation started, each agent performed as well as the above-mentioned behavior, but soon began to search the non-search area. Then we observed map creation behaviors by several agents cooperatively. The changing speed of map creation rate is relatively high and the map is almost perfectly created in about 1500 generations.

- In case of initial condition #3

  Immediately after starting simulation, the changing speed of map creation rate was a little high, but weights of the wall avoidance gain and so on relatively become small. Therefore, agents which can not avoid obstacles wandered about to the non-search area on the opposite side of the wall and the behavior which the other agents are gathering toward the same place was observed. The changing speed of map creation rate is not generally so much high, non-search areas were remained about 10 after over 3000 generations. In this case, we observed that fitness values of all agents becomes very low because agents make too much of map creation behaviors and neglected individual behaviors for the self-preservation.

Comparing simulation results in this paper, we could confirm the agent group makes much of individual behaviors (case #1) can not increase the map creation rate soon, but it is easy to survive because of high food point gains and final map creation rate is relatively high. The agent group makes too much of cooperative behaviors (case #3) could not finally achieve the cooperative behavior because of limitation of the other behaviors. Furthermore, the agent group makes much of cooperative behaviors (case #2) showed the map creation rate and its changing speed are stablely high even if the amount of agents is little. By these results, we could confirm that agent group with homogeneous behavior characteristics has a tendency to emerge the relatively high level cooperative behaviors by setting the balanced parameters of both cooperative and individual behaviors.

Conclusions

In the multi-agent robot system with two behavior duties both the collision avoidance and the target reaching motion, the learning method of behavior based on the self-energy and the behavior gain of each agent was discussed in this paper. Furthermore, group evolution mechanism that parameters of an agent with no self-energies are reset and change to parameters of an agent with the highest evaluated value, was also proposed in this paper.

For this research, we developed the evolutionary behavior learning simulator for the autonomous multi-agent robot system. It was confirmed in the simulation that the agent group with homogeneous behavior characteristics is comparatively apt to cause emergence behaviors by setting the balanced parameters of both cooperative and individual behaviors. In the future, we will make an effort mainly to propose more available

mechanism for group evolution of agents and confirm higher level emergence behaviors.

References

Weiß, G. 1993. Learning to Coordinate Action in Multi-Agent Systems. *Proc. of the 13th International Joint Conference on Artificial Intelligence.* 331–316.

Mataric, M. J. 1995. Designing and Understanding Adaptive Group Behavior. *Adaptive Behavior.* 4(1):51–80.

Unemi, T. 1993. Collective Behavior of Reinforcement Learning Agent. *Proc. of IEEE/Nagoya Univ. WWW on Learning and Adaptive System.* 92–97.

Maeda, Y. 1996. Evolutionary Simulation for Group Behavior of Multi-Agent Robot. *Proc. of 4th International Conference on Soft Computing (IIZUKA'96).* 1:61–64.

Maeda, Y. 1997. Behavior Learning and Group Evolution for Autonomous Multi-Agent Robot. *Proc. of 6th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'97).* III:1355–1360.

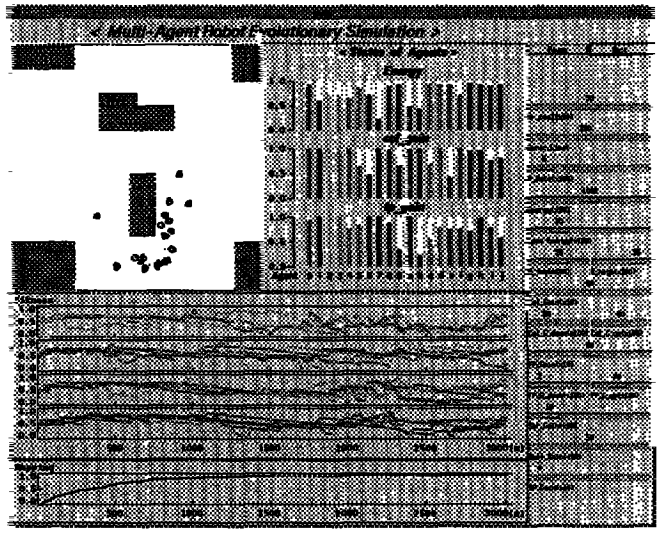

Figure 5: Simulation results in case of #1

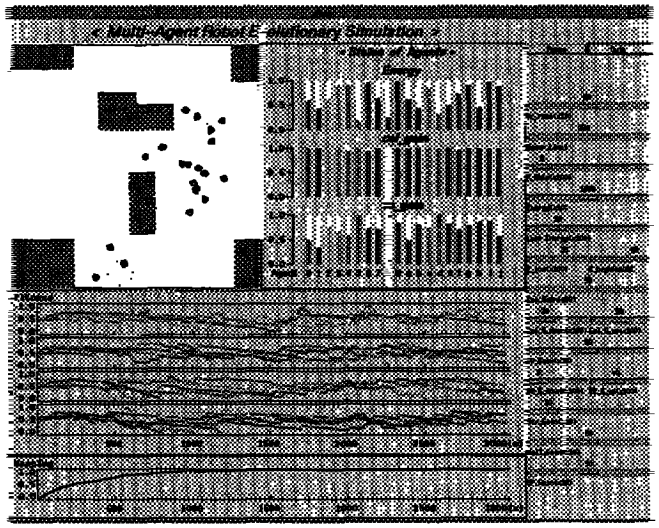

Figure 6: Simulation results in case of #2

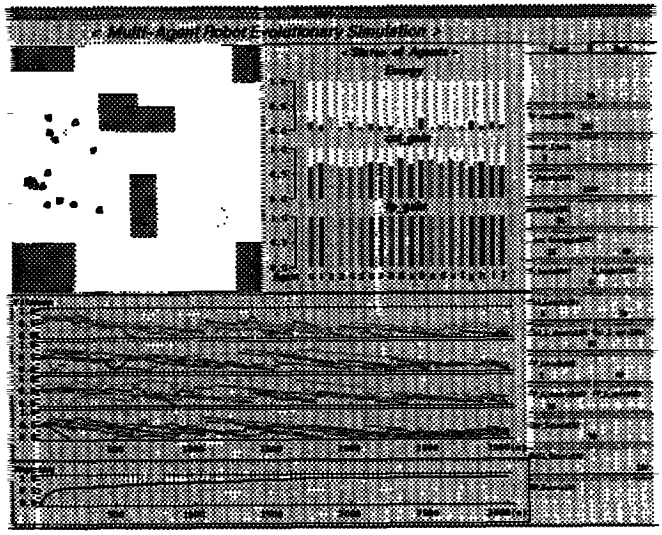

Figure 7: Simulation results in case of #3