Integration of Multiple Neural Networks Evolved on Cellular Automata by Action Selection Mechanism

Kyong-Joong Kim

Department of Computer Science
Yonsei University
134 Shinchon-dong, Sudaemoon-ku
Seoul 120-749, Korea
Fax: +82-2-365-2579
E-mail : uribyul@candy.yonsei.ac.kr

Sung-Bae Cho

Department of Computer Science
Yonsei University
134 Shinchon-dong, Sudaemoon-ku
Seoul 120-749, Korea
Fax: +82-2-365-2579
E-mail : sbcho@candy.yonsei.ac.kr

Abstract

There has been extensive research of developing the controller for a mobile robot. Especially, several researchers have constructed the mobile robot controller that can avoid obstacles, evade predators, or catch moving prey by evolutionary algorithms such as genetic algorithm and genetic programming. In this line of research, we have also presented a method of applying CAM-Brain, evolved neural networks based on cellular automata (CA), to control a mobile robot. However, this approach has a limitation to make the robot to perform appropriate behavior in complex environments. In this paper, we have attempted to solve this problem by combining several modules evolved to do a simple behavior by Maes's Action Selection Mechanism. Experimental results show that this approach has potential to develop a sophisticated neural controller for complex environments.

1. Introduction

There are many studies of constructing mobile robot controller with different approaches such as evolving neural network by genetic algorithm [1], using genetic programming [2], combining fuzzy controller with genetic algorithm [3] and programming [4]. In previous work [5], we presented CAM-Brain, evolved neural networks based on cellular automata [5,6], and applied it to controlling a mobile robot.

However, the controller composed of one module has a difficulty to make the robot to perform complex behavior. To overcome this shortcoming, some researchers combine several modules evolved or programmed to do a simple behavior such as "going straight," "avoiding obstacles," "seeking object," and so on. They expect the controller combined with several modules can do complex behaviors [4,7,8].

In this paper, we also attempt to combine several neural networks for solving this problem. Each neural network can be evolved or programmed. Evolved neural network is based on CAM-Brain model, and a programmed module controls the robot directly. We apply Pattie Maes's Action Selection Mechanism (MASM) [10] to combine modules and control a mobile robot in simulated environments. The rest of this paper introduces CAM-Brain model and basic behaviors, and presents the integration method in detail. The detailed description of simulation follows, and the results of simulation is given.

2. Neural Networks Evolved on CA

2.1 CAM-Brain

CAM-Brain is a model based on CA which can show complicated behavior by combining simple rules, and developed by its own chromosome : One chromosome is mapped to exactly one neural network module. Therefore, with genetic algorithm working on this chromosome, it is possible to evolve and adapt the structure of the neural network for a specific task. Figure 1 shows the evolution process of CAM-Brain. It is the basic idea of CAM-Brain that brain-like system can be constructed by combining many neural network modules of various functions [5,6]. This section illustrates a design of CA-space for developing a neural network module.

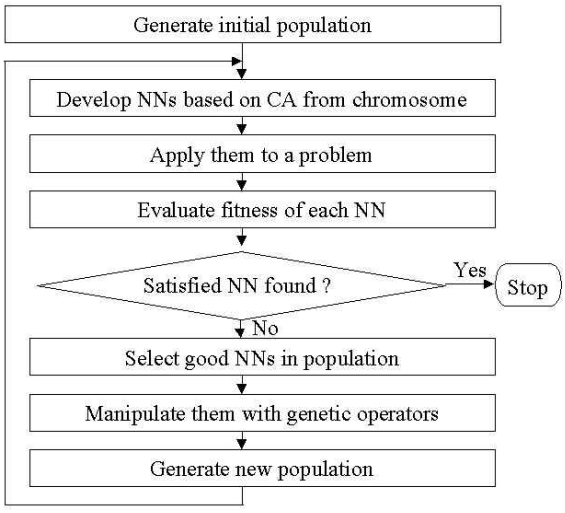


Figure 1: The evolution process of CAM-Brain.

2.2 CoDi Model

CAM-Brain's neural network structure composed of blank, neuron, axon and dendrite is grown inside 2-D or 3-D CA-space by state, neighborhoods and rules encoded by chromosome. If cell state is blank, it represents empty space and cannot transmit any signals. Neuron cell collects signals from surrounding axon cells. Axon cell sends signals received from neurons to the neighborhood cells. Dendrite cell collects signals from neighborhood cells and passes them to the connected neuron in the end [5,6].

The growth phase organizes neural structure and makes the signal trails among neurons. First, a chromosome is randomly made and the states of all cells are initialized as blank. At this point some of the cells are specified as neuron with some probability. A neuron cell sends axon and dendrite growth signals to the direction decided by chromosome. Axon growth signal is sent to two directions and dendrite growth signal is sent to the other remaining directions. Next the blank cell received growth signal changes to axon or dendrite cell according to the type of growth signal. It sends the signals received from other cells to the direction determined by chromosome. Finally, repeating this process, the neural network is obtained when the state of every cell changes no longer.

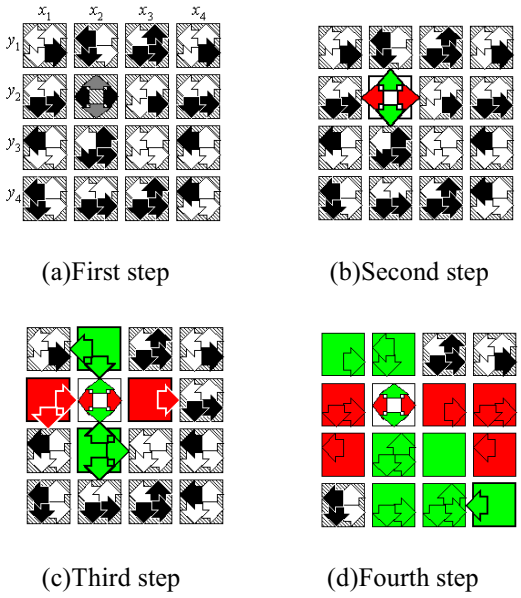


Figure 2: Growth phase : (a) Black arrow represents a neuron is located in (x_2,y_2). (b) The neuron sends growth signals. (c) The cell state is decided according to the type of growth signals. (d) Propagating growth signals, blank cells become axon or dendrite.

Figure 2 shows the growing process in 4×4 2-D CA-space. In this figure, the cell which has oblique lines is blank cell, and the blank arrows show the direction of signaling decided by chromosome. Figure 2(a) shows the process of sending a neuron in blank cells, where a neuron is located in (x_2,y_2). Figure 2(b) shows that the neuron cell sends growth signal to surrounding cells. Figure 2(c) shows the cell state is changed by growth signal. Figure 2(d) shows that blank cells grow into axon or dendrite. In a neuron, the dendrite collects signals and sends to the neuron, and the axon distributes signals originated from the neuron.

Signaling phase transmits the signal from input to output cells continuously. The trails of signaling are transmitted with evolved structure at the growth phase. Each cell plays a different role according to the type of cells. If the cell type is neuron, it gets the signal from connected dendrite cells and gives the signal to neighborhood axon cells when the sum of signals is greater than threshold. If the cell type is dendrite, it collects data from the faced cells and eventually passes them to the neuron body. The position of input and output cells in CA-space is decided in advance. At first, if input cells produce the signal, it is sent to the faced axon cells, which distribute that signal. Then, neighborhood dendrite cells belonged to other neurons collect and send this signal to the connected neurons. The neurons that have received the signal from dendrite cells send it to axon cells. Finally, dendrite cells of output neuron receive and send this signal to the output neurons. Output value can be obtained from output neurons. During signaling phase, the fitness is evaluated by the output in this process. Figure 3 shows the process of signaling after neuron, axon and dendrite have been made.

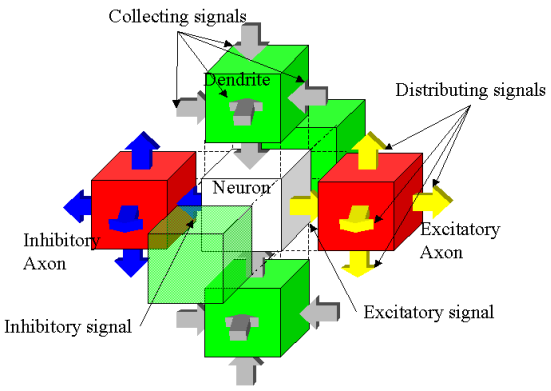


Figure 3: Signaling phase.

2.3 Evolution of CAM-Brain

In general, simple genetic algorithm generates the population of individuals and evolves them with genetic operators such as selection, mutation, and crossover [9]. We have used the genetic algorithm to search the optimal neural network. At first, a half of the population that has better fitness value is selected to produce new population. Two individuals in the new population are randomly selected and parts of them are exchanged by one-point crossover. The

crossover is occurred at the same position in the chromosomes to maintain the same length in chromosomes. Mutation is operated in the segment of chromosome. The genetic algorithm generates a new population from the fittest individuals on the given problem.

3. Action Selection Mechanism

Maes's Action Selection Mechanism (MASM) was proposed originally as an improvement over previous approaches to action selection in the field of AI. In particular, it was proposed as an improvement on traditional planning systems and reactive systems [8].

3.1 Description

MASM is a distributed, non-hierarchical network. There are two waves of input to the network : from the sensors of the external environment (mostly external stimuli) and from the motivations or goals (mostly derived from internal stimuli such as body temperature). The central idea of Maes's scheme for action selection is that the different types of links encode various relationships.

MASM is composed of nodes, internal links and external links. Each node has a set of preconditions. The preconditions are logical conditions about the environment that are required to be true in order for the node to be executable. The add list consists of conditions about the environment that the node is likely to make true. The delete list consists of conditions that are likely to be made false by the execution of the node. The final two components of the node are the activation level and the code that gets run if the node is executed. Figure 4 shows a node in MASM. The internal links are specified in table 1. The external links providing input to the network are specified in table 2. Table 1,2 contain descriptions of links.

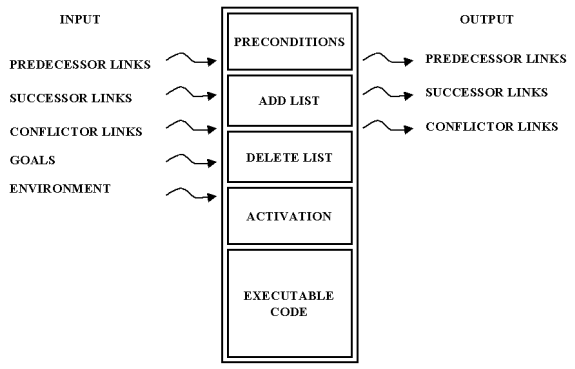


Figure 4: A node in MASM.

3.2 Procedure

The procedure used to select a node to execute at each time step is as follows:

1. Calculate the excitation coming in from the environment and the motivations.
2. Spread excitation along the predecessor, successor, and conflictor links.
3. Normalize the node activations so that the average activation becomes equal to the constant π.
4. Check to see whether any nodes are executable and, if so, choose the one with the highest activation, execute it, and finish.
5. If no node is executable, reduce the global threshold and repeat the cycle.

A node is executable if all its preconditions are true and if its activation is greater than the global threshold. If more than one node is executable after a cycle, the one with the highest activation is chosen. Figure 5 shows the procedure of MASM.

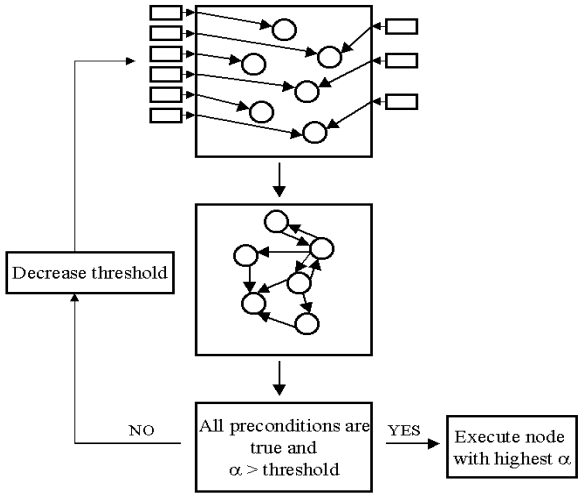


Figure 5: The procedure of MASM (α=activation).

3.3 Parameter

Several global parameters are used to tune the performance of the ASM to a particular environment. The mean activation value after each cycle (π) is used in normalization. The initial value of the global threshold (θ), which is reduced by an amount (e.g., 10%) after each cycle if there is no node to be executable. A constant (ϕ) determining the weight of environmental sensor inputs, as well as the weight of successor links. A constant (δ) determining the weights of protected goal inputs and conflictor links. The different inputs to a node are multiplied by the following :

(1) environmental sensors by ϕ
(2) goals by γ
(3) protected goals by δ
(4) successor links by ϕ/γ
(5) predecessor links by (γ/γ=1)
(6) conflictor links δ/γ

Table 1: Internal links

Predecessor link	If proposition X is false and proposition X is a precondition of node A and proposition X is in the add list of node B, then there is an active predecessor link from A to B.
Successor link	If proposition X is true and proposition X is in the add list of node A and proposition X is a precondition of node B and the node A is executable, then there is an active successor link from A to B.
Conflictor link	If proposition X is true and proposition X is a precondition of node A and proposition X is in the delete list of node B, then there is an active conflictor link from A to B.

Table 2: External links

From sensors of the environment	If proposition X about the environment is true and proposition X is a precondition of node A, then there is an active link from the sensor of the proposition X to node A.
From goals	If goal Y has an activation greater than zero and goal Y is in the add list of node A, then there is an active link from the goal Y to node A.
From protected goals	If goal Y has an activation greater than zero and goal Y is in the delete list of node A, then there is an active link from the goal Y to node A.

4. Experimental Results

In order to show that highly complex behaviors can be made by combining lower simple behaviors, we have conducted the simulation with Khepera robot.

4.1 Khepera

Khepera robot contains 8 infrared sensors to detect by reflection the proximity of objects in front of it, behind it, and to the right and the left sides of it, and to measure the level of ambient light all around the robot. Also, the robot has two motors to control the left and right wheels. Khepera simulator also features the ability to drive a real Khepera robot, so that we can very easily transfer the simulation results to the real robot [2,3,4]. We have modified the Khepera simulator for the purpose of our experiments.

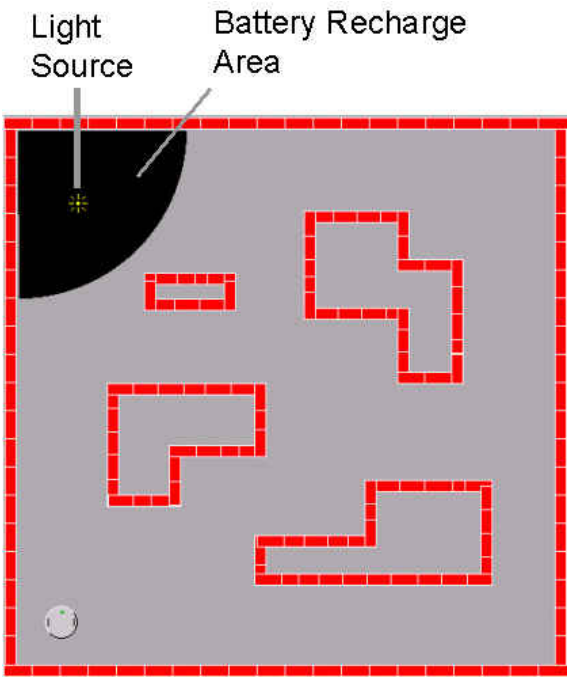
4.2 Basic Behaviors

In this paper, four basic behaviors are defined as follows.

- Recharge Battery : If a robot arrives at battery recharge area, battery is recharged. This module enables the robot to operate as long as possible.
- Follow Light : The robot goes to stronger light. This module must be operated to go to the battery recharge area because the light source exists in that area.
- Avoid obstacle : If the obstacles exist around the robot, it avoids them without bumping against them.
- Go Straight : If there is nothing around the robot, it goes ahead. This module takes it to move continuously without stopping.

Basic behaviors are programmed or evolved on CAM-Brain. Battery Recharge and Go Straight modules are programmed. Avoid obstacles and Follow Light are evolved on CAM-Brain. Figure 6 shows simulation environments.

Figure 6 : Simulation environments.



4.3 Module Integration

In this section we apply the action selection mechanism to the robot control environment. Our environment requires 5 states, such as “In battery recharge area,” “Obstacles are close,” “Near battery recharge area,” “Light is low,” and “Nothing around the robot.” They are set as follows :

- In battery recharge area : If robot is in battery recharge area, the state is true.

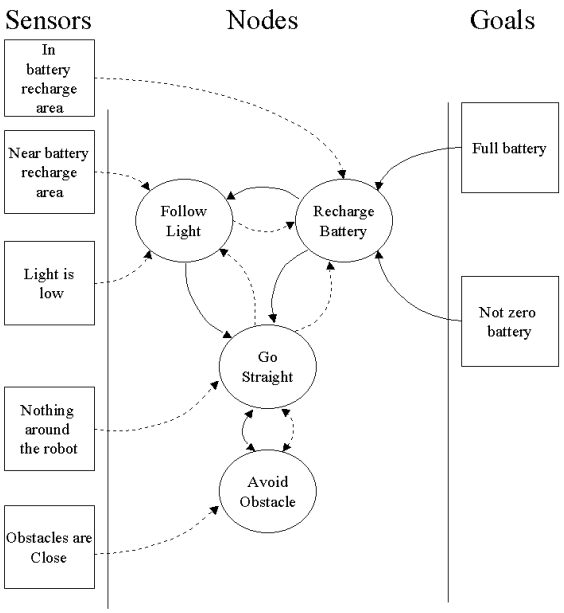


Figure 7: MASM model : Solid lines denote goal or predecessor connections, and dashed lines denote sensor or successor connections.

- Obstacles are close : If the maximum value of distance sensors is larger than 700, the state is true.
- Near battery recharge area : If distance from robot to light source is less than 800, the state is true.
- Light is low: If the minimum value of light sensors is larger than 400 , the state is true.
- Nothing around the robot : If the maximum value of distance sensors is less than 700, the state is true.

We set 2 goals. Such as "Full battery," "Not zero battery". Because robot's battery decrease when robot moves, so robot will maintain high battery value to live long. They are set as follows :

- Full battery :

$$c = \frac{m-n}{m} \quad (1)$$

c : Value of "Full battery"
m: Maximum battery
n : Robot's battery

- Not zero battery : If battery is less than half of the maximum battery, the state is true.

5 states are binary-valued and 2 goals are continuous values. Our MASM model is composed of 4 nodes, 5 states, 2 goals and their relationships. Figure 7 shows our MASM model.

Each node has preconditions. Node must fulfill all preconditions to be executed. Table 3 describes preconditions of the nodes. Each node has one or two preconditions.

Table 3 : Preconditions of nodes

Node	Preconditions
Recharge Battery	In battery recharge area
Follow Light	Light is low, Near battery recharge area
Avoid Obstacle	Obstacles are close
Go Straight	Nothing around the robot

The relationships among nodes are decided by successor links or predecessor links. If predecessor link exist from A node to B node, successor link from B node to A node exists. If node A and node B are connected , they will exchange their activation values while activation spreads. Table 4 describes add lists of nodes.

Table 4 : Add lists of nodes

Node	Add lists
Recharge Battery	Full Battery, Not zero battery
Follow Light	In battery recharge area
Avoid Obstacle	Nothing around the robot
Go Straight	Obstacles are close, In battery recharge area, Near battery recharge area

We have to set the values of π, θ, γ, ϕ, and δ in MASM model. In this model $\pi = 4.5$, $\theta = 3.0$, $\gamma = 0.8$, $\phi = 1.2$, and $\delta = 1.0$.

4.4 Result Analysis

Robot moves 4942 times. Figure 8 shows robot's trajectory. We can analyze robot's behavior : When robot is in battery recharge area, state "In battery recharge area" is true, and recharge battery module can be selected. Because "In battery recharge area" is a precondition of recharge battery module. When battery is low, robot's goals increase and follow light module is selected more frequently than other behavior module.

Robot selects avoid obstacle and follow light module continuously before battery recharge area. This makes robot go battery recharge area without bumping obstacles. By selecting lower level behavior, robot can make high level behavior which we can not make easily. Figure 9 shows that robot selects follow light module before battery recharge area. Figure 10 shows that robot selects avoid obstacle module before battery recharge area.

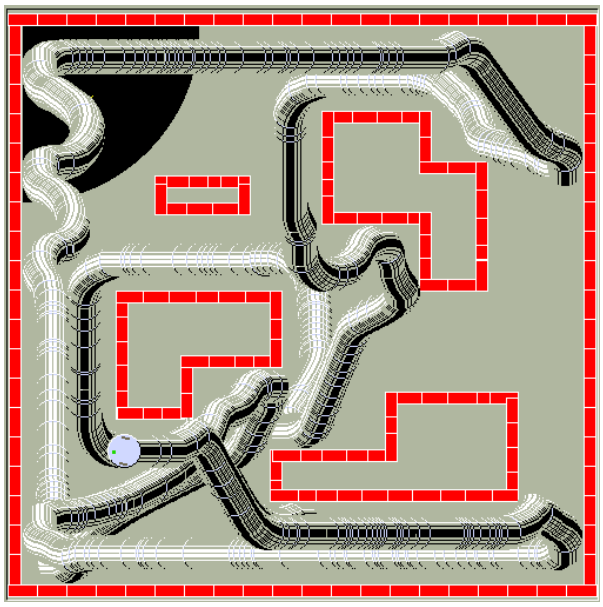


Figure 8 : Robot's trajectory.

5. Concluding Remarks

In this paper, we apply Maes's Action Selection Mechanism to robot control environment. Robot selects basic behaviors from behavior networks. MASM gets signals from environments and internal motivations, and spreads the activation among internal links. We have four basic behaviors which are evolved on CAM-Brain or programmed. MASM model can be used to control mobile robot by modeling simulation environments. Robot achieves goals by selecting basic behaviors, shows MASM has potential to combine several neural network module and control robot.

Acknowledgements

This research was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Science and Technology.

References

[1] D. Floreano and F. Mondana, "Evolution of homing navigation in a real mobile robot," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 26, No 3, pp. 396-407, June, 1996.

[2] P. Nordin and W. Banzhaf, "Real time control of a Khepera robot using genetic programming," *Cybernetics and Control*, Vol. 26, No. 3, pp. 533-561, 1997.

[3] S.-B. Cho and S.-I. Lee, "Evolutionary learning of fuzzy controller for a mobile robot," *Proc. Int. Conf on Soft Computing, pp*. 745-748, Iizuka, Japan, 1996.

[4] M.J. Mataric, "Designing and understanding adaptive group behavior," *Adaptive Behavior*, Vol. 4, No.1, pp. 51-80, 1995.

[5] G.-B. Song and S.-B. Cho, "Incremental evolution of CAM-Brain," *Proc. AROB99*, pp. 297-300, Beppu, Japan, 1999.

[6] F. Gers, H. de Garis and M. Korkin, "CoDi-1Bit: A simplified cellular automata based neural model," *Proc. Conf. on Artificial Evolution*, Nimes, France, October, 1997.

[7] W. Banzhaf, P. Nordin, and M. Olmer, "Generating adaptive behavior using function regression within genetic programming and a real robot," *Int. Conf. on Genetic Programming*, pp. 35-43, 1997.

[8] T. Tyrrell, "An evaluation of Maes's bottom-up mechanism for behavior selection," *Adaptive Behavior*, Vol. 2, pp. 307- 348, 1994.

[9] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, 1989.

[10] P. Maes, "How to do the right thing," *Connection Science Journal*, Vol 1, No. 3, pp 291-323, 1989.

[11] G.-B. Song and S.-B. Cho, "Rule-based integration of multiple neural networks evolved based on cellular automata," *IEEE Fuzzy Systems Proceedings*, pp. 791-796 , Seoul, Korea, 1999.

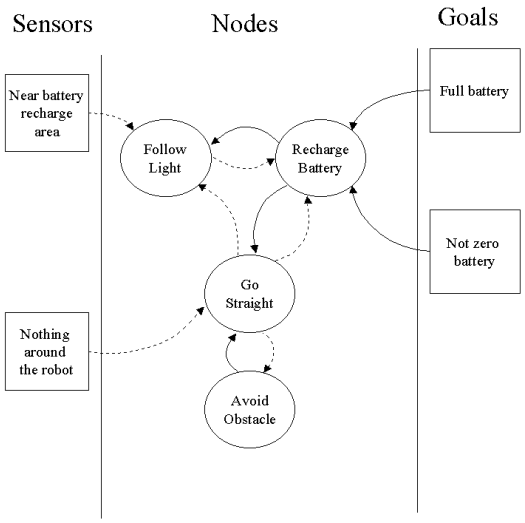


Figure 9: battery = 500, Follow Light is selected.

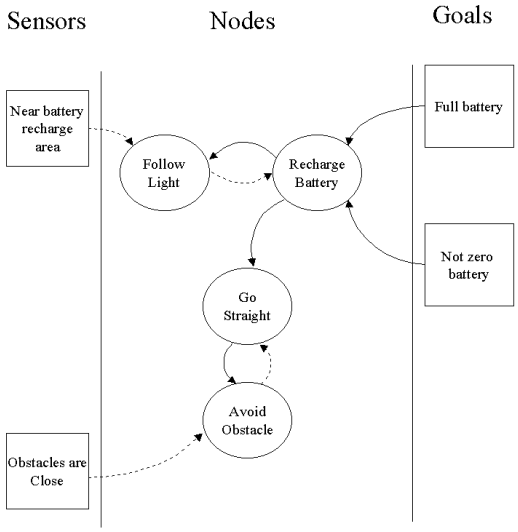


Figure 10: battery = 487, Avoid Obstacle is selected.