Proceedings 2003 IEEE International Symposium on
Computational Intelligence in Robotics and Automation
July 16-20, 2003, Kobe, Japan

Behavior Selection Architecture for Tangible Agent*

Kyung-Joong Kim, Sung-Bae Cho
Dept. of Computer Science
Yonsei University, South Korea
uribyul@candy.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Sang-Rok Oh
Intelligent System Control Research Center
KIST, South Korea
sroh@amadeus.kist.re.kr

Abstract

Tangible agent (TA) is a new medium that communicates the senses of sight, hearing, touch, smell, and taste of human to computer. Intelligent behavior is a key property to realize the agent because it must interact with human to grasp all the information. We have adopted behavior-based approach for high-level behaviors such as navigation of office, conversation with human, and cleaning of room. Behavior-based method can control unexpected situation without prior knowledge and generate high-level behavior with behavior selection. However, TA requires improvements of behavior selection architecture for better communication with human. In this paper, we propose an intelligent behavior selection architecture that contains inferencing, learning and planning capability to TA. In this paper, overview of technical details and experimental results on physical device (Khepera robot) are presented. Preliminary results show the possibility of the proposed behavior selection architecture for TA.

1 Introduction

Traditional user interface for cyberspace is restricted to the monitor connected to computer but tangible medium expands the concept of interface by including physical devices such as desk, robot, blackboard, and electric light [1]. Tangible agent (TA) connects human to bits by delivering the five senses of human to cyberspace. Among many candidates for TA, physical robot and artificial life character are easily applicable because human can feel friendship from them. For autonomous and seamless interaction, intelligent behavior generation for TA has many issues including inferencing user's goals, automatically constructing behavior selection architecture (BSA), and optimizing behavior. Behavior-based approach have been adopted for behavior generation of TA because the method can react without hesitation in unexpected situation. This property is necessary for TA that does jobs in office.

There are many ASM's (Action Selection Mechanisms) to combine behaviors for generating high-level behaviors including spreading activation network, subsumption architecture and hierarchical ASM [2]. The ASM is essential in behavior-based methods because it selects appropriate one among candidate behaviors and coordinates them. Usually, ASM's can not insert goals into the model in explicit or implicit manner. Behavior network, one of ASM's, can contain goals of robot in implicit manner and propagates activation of behavior in two directions (forward and backward) through the network for dynamic selection [3]. The network is adopted for TA because the ASM is situated between traditional artificial intelligence approach and pure behavior-based approach, and allows incorporation of intelligent properties naturally.

To deal with many issues mentioned before, behavior network needs improvements in inferencing, learning, and planning capability. Behavior network can have many goals that need to activate in different environments and status. User can insert prior knowledge of goal activation into behavior network in the design stage of behavior network but it is difficult to capture and represent the knowledge. There are some computational methodologies to represent knowledge into graph model with inference capability such as Bayesian network, fuzzy concept network and fuzzy cognitive map [4, 5, 6]. Above all, Bayesian network is used practically to inference goals of software users in Microsoft Excel [7]. Maes and Mataric proposed learning algorithm of behavior network respectively but their methods are based on statistical or temporal learning [8, 9]. These algorithms need user's prior knowledge but learning classifier system (LCS) evolves structure of behavior network without expert. Bagchi proposed modified behavior network with planning capability but their method inserts planning in implicit manner [10]. Explicit planning with behavior search tree is adopted in this paper.

Behavior selection architecture for TA is composed of three components such as inference using Bayesian network, structure learning using LCS, and explicit planning using tree search. Traditional behavior-based approach has realized animal level intelligence with reactivity. However, the animal can recognize their status, learn new capability, and plan how to capture prey with reactive behavior. We attempt to implement better agent with animal level intelligence that has reactive and cognitive properties. Experiments on Khepera mobile robot show the possibility of the proposed BSA for TA.

2 Behavior Selection Architecture

Behavior-based approach is to realize intelligence without internal representation. This property makes robot

*Acknowledgement: This research was supported by KIST (Korea Institute of Science and Technology).

react immediately to unexpected situation such as "navigation on unknown planet." Robot does not have complex internal representation to process input signal and it is possible to be reactive. For the higher behavior, it is essential to combine the reactive behaviors using ASM. Using inference, learning and planning algorithms, it is possible to implement animal level intelligence for TA. Designing ASM is not an easy task because there are many variables to consider and knowledge about environment is not enough. Inference module uses computational model such as Bayesian network, fuzzy concept network and fuzzy cognitive map to represent prior knowledge and estimate unknown variables. ASM is not adequate to insert knowledge for inference and cannot select behaviors properly when the problem contains uncertainty. Learning module can determine structure of ASM automatically and change the part of structure adaptively to the environments. Planning optimizes the sequence of behaviors for solving the task. Figure 1 is an overview of the proposed method. Intelligent behavior is generated using behavior network with inference, learning and planning capability and it is used for TA including humanoid, mobile robot, and artificial life character.

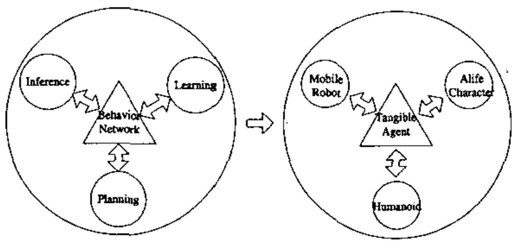


Figure 1: Overview of the proposed method.

2.1 Behavior Network

Competition of behaviors is the basic characteristics of behavior network. Each behavior attempts to get higher activation level than other behaviors from activation spreading in forward and backward. Among candidate behaviors, one that has the highest activation level is selected and has control of robot. Activation level a of behavior is calculated as follows.

- Forward propagation: Activation a is updated as the value added by environmental sensors that are precondition of the behavior. Precondition is the sensor that is likely to be true when the behavior is executed. n means the number of sensors. a_s is the activation level of the sensor.

$$\Delta a_1 = \sum_{i=1}^{n} f(a_{s_i}) \tag{1}$$

$$f(a_{s_i}) = \begin{cases} \phi \times a_{s_i}, & s_i \in \text{precondition} \\ 0, & s_i \notin \text{precondition} \end{cases} \tag{2}$$

- Backward propagation: Activation a is updated as the value added by goals that are directly connected to the behavior. If the execution of the behavior is desirable for the goal, positive goal-behavior link is activated. Otherwise, negative goal-behavior link is activated. n means the number of goals. a_g is the activation level of the goal.

$$\Delta a_2 = \sum_{i=1}^{n} f(a_{g_i}) \tag{3}$$

$$f(a_{g_i}) = \begin{cases} \gamma \times a_{g_i}, & g_i \in \text{positive link} \\ -\delta \times a_{g_i}, & g_i \in \text{negative link} \end{cases} \tag{4}$$

- Internal spreading: Activation a is updated as the value added by other behaviors that are directly connected. If the execution of behavior B is desirable for behavior A, predecessor link from A to B and successor link from B to A are active. If the execution of behavior B is not desirable for behavior A, conflictor link from A to B is active. Here, n is the number of behaviors, and a_b is the activation level of the behavior.

$$\Delta a_3 = \sum_{i=1}^{n} f(a_{b_i}) \tag{5}$$

$$f(a_{b_i}) = \begin{cases} a_{b_i}, & \text{predecessor link from } b_i \\ \phi/\gamma \times a_{b_i}, & \text{successor link from } b_i \\ -\delta/\gamma \times a_{b_i}, & \text{conflictor link from } b_i \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Finally, the activation of a is updated as follows.

$$a' = a + \Delta a_1 + \Delta a_2 + \Delta a_3 \tag{7}$$

If the activation level a' is larger than threshold θ and precondition of the behavior is true, the behavior becomes candidate to be selected. Among candidate behaviors, the highest activation behavior is selected for execution, threshold θ is reduced by 10% and the activation update procedure is repeated until there are candidate behaviors.

2.2 Inference

Consider a domain U of n discrete variables $x_1,\cdots,x_n$, where each x_i has a finite number of states. A Bayesian network for U represents a joint probability distribution over U by encoding (1) assertions of conditional independence and (2) a collection of probability distributions. Specifically, a Bayesian network B can be selected as the pair (B_S,Θ), where B_S is the structure of the network, and Θ is a set of parameters that encode local probability distributions [11]. The joint probability for any desired assignment of values $(y_1, ..., y_n)$ to the tuple of network variables $(Y_1, ..., Y_n)$ can be computed by the equation

$$P(y_1, ..., y_n) = \prod_{i=1}^{n} P(y_i|Parents(Y_i)) \tag{8}$$

where $Parents(Y_i)$ denote the set of immediate predecessors of Y_i in the network.

Behavior network has one or more goals to achieve in the environments. Coordination of their activation can not be fixed in design stage of behavior network because there is uncertainty. Bayesian network is adopted to infer activation of goal from some observations from the environments. Structure of Bayesian network is automatically learned from the data that are collected from random navigation. Observed variables are sensor information that can be collected from the robot sensors including distance, light and velocity. From these data, it is possible to estimate unknown variables including area information, emergency level and cooperation with other agents. Equation (3) is modified as follows.

$$\Delta a_2 = \sum_{i=1}^{n}\sum_{j=1}^{m} f(a_{g_i}) \times r_{i,j} \times P(v_j|\text{observation}) \quad (9)$$

m is the number of unknown variables. $r_{i,j}$ is the relevance of goal i with respect to variable j. This value is determined manually. $P(v_j|\text{observation})$ is calculated using Bayesian network and equation (8).

2.3 Learning

LCS, a kind of machine learning technique, was introduced by Holland and Reitman in 1978 [12]. LCS has two different learning procedures. One is to learn via mixing given rules (credit assignment system) and the other is to learn via creating useful rules as possible (rule discovery system). It is very appropriate to be adapted to a changing environment. Classifier system consists of several rules, so-called classifiers. One classifier has one or more condition parts and one action part. The condition of a classifier consists of ternary elements {0, 1, #} and the action part consists of {0, 1}. The character '#' means "don't care" and can take either '0' or '1'.

In the competition of classifiers, the value of strength gives a measure of the rule's past performance. That is, the higher a classifier's strength the better it has performed. In addition, each classifier has the value of specificity, which is the number of non-# symbols in the condition part. LCS consists of three modules as shown in Figure 2.

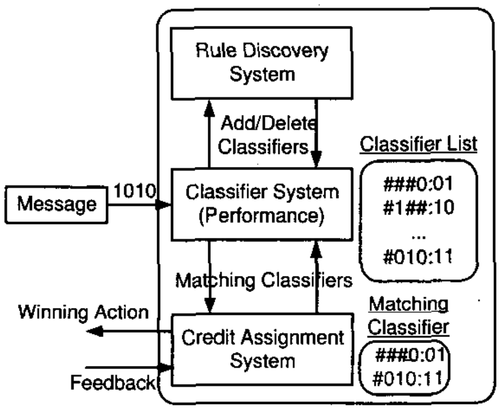


Figure 2: The structure of LCS.

- Classifier System: The system compares input messages with the condition part of all classifiers in the classifier list and performs matching. This acts by bit-to-bit. The matching classifiers enter competition for the right to post their output messages, and only the winner of the competition actually posts messages. The measure of each classifier is the value of bid as follows:

$$bid = c \times specificity \times strength$$

where c: constant less than 1, *specificity*: specificity of the classifier condition, condition's length minus the number of '#' symbols, and *strength*: the measure of confidence in a classifier. When feedback comes from its environment, the strength of the winner is recomputed.

- Rule Discovery System: The system uses genetic algorithm to discover new rules. Genetic algorithm is stochastic algorithm that has been used both as optimization and as rule discovery mechanism. They work modifying a population of solutions (in LCS, solution is a classifier). Solutions are properly coded and a fitness function is defined to relate solutions to performance. The value from this function is a measure of the solution's quality. The fitness of a classifier is set by its usefulness calculated with a credit apportionment system instead of a fitness function.

In general, while classifier system and credit apportionment system are interested in the classifiers with better performances, rule discovery system is interested in the classifiers with worse performances. If the only classifiers with high scores survive, the system cannot discover new classifiers.

Our goal is to improve the robot's ability to solve problems using LCS in the behavior network. Firstly the rules of LCS, the basic elements, are defined for learning links in the network. The rules include the state nodes, the extent of the problems, and links between the nodes. Each action node is not included in the rules. When the rule is fixed, or the network structure is decided, it calculates the activation level and selects the action node with the highest activation value. Figure 3 shows the structure of the rule defined. The extent of the problems and state nodes are determined through the robot's current states from environment.

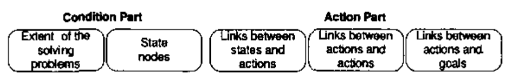


Figure 3: The structure of a rule.

While behavior network includes three kinds of links between the nodes, we redefine the excitatory links for LCS. Excitatory links indicate predecessor links and successor links because they have similar meaning. In addition, the classifier lists are initialized in accordance with the purpose of tasks, not randomly. That is, the initial structure of network is converted into rules.

2.4 Planning

The behavior network selects behaviors to achieve goals of robot and planning module constructs behavior search tree to find optimal behavior sequence for a specific task. Optimality of behavior sequence is measured by the prior knowledge of the environment and problem. We have to choose an appropriate level to search behavior sequences and decides the measure of optimal behavior sequence. Construction of behavior search tree is repeated until robot achieves goals.

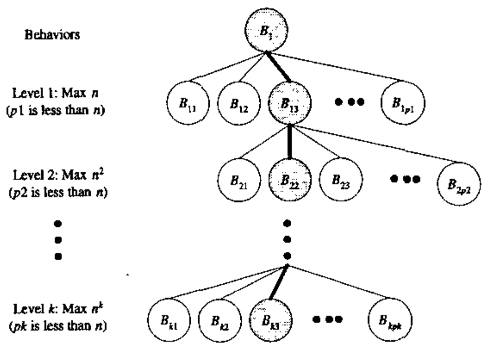


Figure 4: Behavior search tree selects at most n behaviors at each level and expands its structure until the k-th level. Finally, maximum number of behavior sequences is n^k.

Figure 4 shows the behavior search tree that can find an optimal behavior sequence by expanding its structure until the appropriate level. In figure 4 the optimal behavior sequences are $B_1 \Rightarrow B_{13} \Rightarrow B_{22} \Rightarrow \cdots \Rightarrow B_{k3}$. By repeating this procedure until the goals are achieved, we can get optimal behavior sequence step by step. Figure 5 shows the flowchart to find the globally optimal sequences of behaviors after making the behavior search tree selecting locally appropriate behaviors in behavior network.

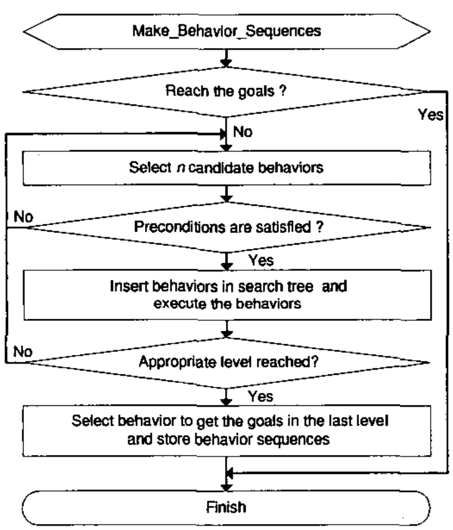


Figure 5: Flow chart making the behavior sequences.

This flowchart is terminated if a mobile robot attains the goals. The selection of behaviors follows Maes's action selection mechanism using the activation value of behaviors in behavior network. Therefore, the selected behaviors have to satisfy the preconditions of that behavior as well as large activation value. The behaviors selected by these procedures must be put in the behavior search tree. Finally, planning module processes the behaviors in the behavior search tree and finds optimal behavior in the last level of behavior search tree. Behavior sequences are made by linking parent behaviors (nodes) of optimal behavior.

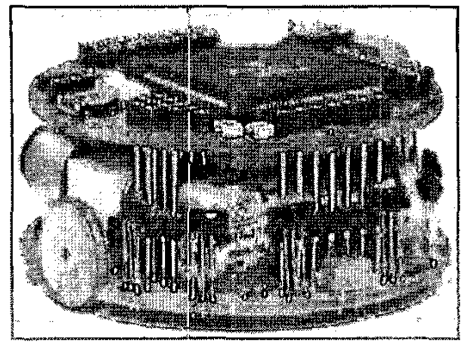


Figure 6: Mobile robot, Khepera.

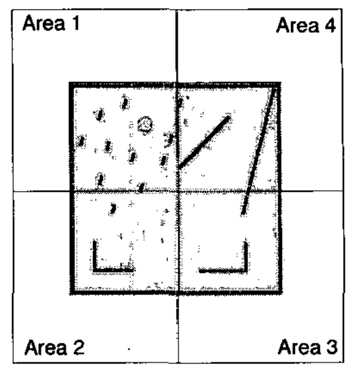


Figure 7: Experimental environments for behavior network with inference.

3 Experimental Results

Khepera was originally designed for research and education in the framework of a Swiss Research Priority Program (see Figure 6). It allows confrontation to the real world of algorithms developed in simulation for trajectory execution, obstacle avoidance, pre-processing of sensory information, and hypothesis test on behavior processing. Khepera robot has two wheels. Eight infrared proximity sensors are placed around the robot.

3.1 Inference

There are four different areas in Figure 7 (Area1, Area2, Area3, Area4). Area1 is start position that has many small obstacles. Area2 contains two light sources and Area3 contains one light source. Area 4 has simple obstacles. If robot can classify area using observed information, behavior network can generate more appropriate behavior sequences. Robot uses three behaviors evolved on CAM-Brain. They are "Avoiding obstacles," "Going straight" and "Following light." Figure 8 shows behavior network for this experiment. There are two different links among behaviors. Predecessor link is represented with solid line and successor link is represented with dashed line. There are five different environmental sensors that can be inferred from original raw sensor data. Structure of Bayesian network is learned from the

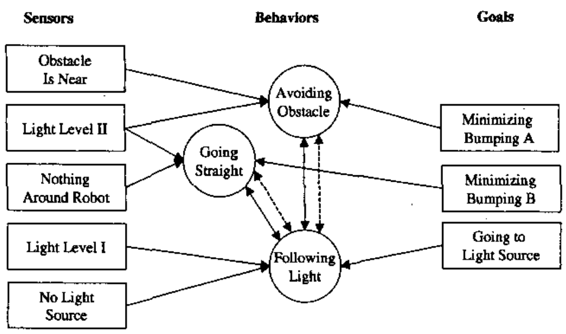
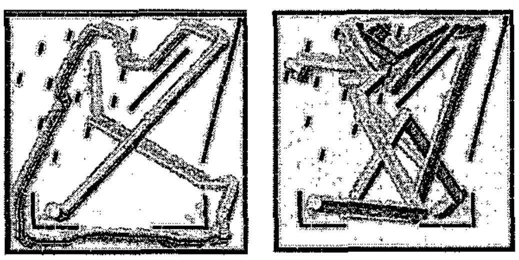


Figure 8: Behavior network with inference.

data that are collected from the environment to discriminate area information. Figure 9 shows experimental results in the environment. In (a), robot navigates area with the behavior sequence that is determined using the behavior network. The network selects one behavior at one time and executes it. In (c), robot navigates area with the combination of behavior network and Bayesian network learned. Bayesian network determines the conditional probability of area1, area2, area3, and area4 with observed sensors. In (a) robot does not pass area3 but in (c), robot passes area3.



(a) (b)

Figure 9: Experimental results of behavior network with inference: (a) behavior network, (b) behavior network with inference.

3.2 Learning

Before learning, the robot always begin to move by going straight as shown in Figure 10. Therefore, it is impossible to reach the goal in the shortest path whenever the initial position of the robot changes. The goal of robot is the left upper corner and robot's initial position is set randomly in the experiments. Figure 10 shows the results of learning when the initial position of robot is the same.

The learning mode performance measures how well the LCS is learning to perform the correct behavior in an environment. In order to evaluate, we use the simple measure of learning performance, the ratio of the number of correct responses during epoch to the epoch length, as follows [13]:

$$\frac{\textit{Number of correct responses during epoch}}{\textit{Epoch length}}$$

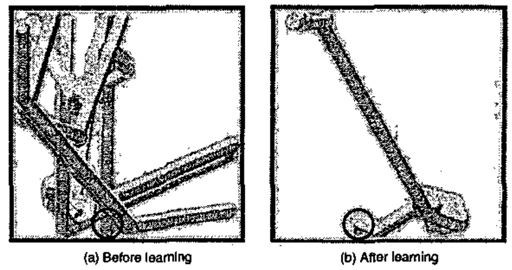


Figure 10: Results of action selection in initial network (left) and learned network (right).

Figure 11 shows the average number of correct responses at every epoch. It proves that the robot can select appropriate behaviors via learning.

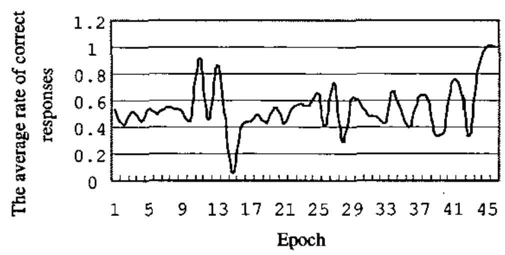


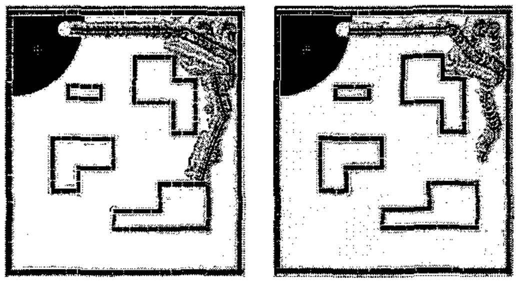
Figure 11: The average rate of correct responses with respect to epochs.

3.3 Planning

We compare behavior sequences in behavior network with behavior network with planning capability. Figure 12 (a) shows a simulation result of behavior network. Robot in this environment executes 2586 times of behaviors selection up to recharge battery. Robot's goal is to reach the battery recharge area that is colored as black to recharge battery. The behavior sequences defined such as 1 is "Recharging battery," 2 is "Following light," 3 is "Avoiding obstacles," and 4 is "Going straight," are shown in figure 13 (a). Figure 12 (b) shows the simulation result in the behavior network with planning capability. Planning is up to 8 levels locally and the behavior of the greatest light value in each level 8 is selected. This occurs total 1376 times of behavior selection up to reach the goal, reach battery area. The behavior sequences of figure 12 (b) are shown in figure 13 (b).

4 Conclusion and Future Works

In this paper, we have proposed a framework to design intelligent behavior for tangible agent. Recently, researchers attempt to realize tangible agent for seamless interface with digital space, but the generation of intelligent behavior is not solved properly. We attempt to attack these problems with hybrid of reactive behavior-based approach and other intelligent tools including inference, learning, and planning. Our goal is to show the



(a) (b)

Figure 12: The simulation result using behavior network. (a) Behavior network, (b) Behavior network with planning

usefulness of our approach in many different tangible agent including humanoid, software agent, artificial life character, and mobile robot. In experiment with mobile robot, we can prove that our approach can be used for improvement of behavior-based system.

References

[1] H. Ishii and B. Ullmer, "Tangible bits: Towards seamless interfaces between people, bits and atoms," *Proceedings of Conference on Human Factors in Computing Systems (CHI'97)*, pp. 234-241, March 1997.

[2] P. Pirjanian, "Behavior coordination mechanisms-state-of-the-art," *Tech-report IRIS-99-375*, Institute for Robotics and Intelligent Systems, School of University of Southern California, Oct 1999.

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

[4] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann Publishers, San Mateo, CA., 1998.

[5] S.-M. Chen and J.-Y. Wang, "Document retrieval using knowledge-based fuzzy information retrieval techniques," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 5, pp. 793-803, 1995.

[6] C. D. Stylios and P. P. Groumpos, "Fuzzy cognitive maps: A model for intelligent supervisory control systems," *Computers in Industry*, vol. 39, no. 3, pp. 229-238, July 1999.

[7] E. Horvitz, J. Breese, D. Heckerman, D. Hovel and K. Rommelse, "The lumiere project: Bayesian user modeling for inferring the goals and needs of software users," *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 256-265, 1998.

[8] P. Maes, "Learning behavior networks from experience," *Proceedings of the First European Conference on Artificial Life*, 1991.

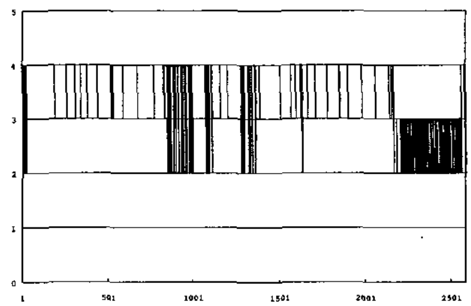
[9] M. Nicolescu and M. J. Mataric, "Learning and interacting in human-robot domains," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 31, no. 5, pp. 419-430, 2001.

[10] S. Bagchi, G. Biswas ans K. Kawamura, "Task planning under uncertainty using a spreading activation network," *IEEE Trans. on Systems, Man and Cybernetics Part A*, vol. 30, no. 6, pp. 639-650, Nov 2000.

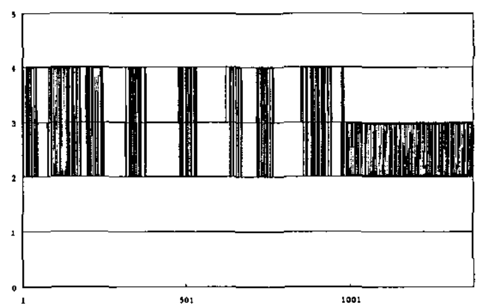
[11] D. M. Chickering, D. Heckerman, and C. Meek, "A Bayesian approach to learning Bayesian networks with local structure," *Microsoft Technical Report MSR-TR-97-7*, 1997.

[12] L. Booker, D. E. Goldberg, and J. H. Holland, "Classifier systems and genetic algorithms," *Artificial Intelligence*, vol. 40, pp. 235-282, 1989.

[13] A. Richards, "Zeroth-order shape optimization utilizing a learning classifier system," *Ph. D. Dissertation*, Mechanical Engineering Department, Stanford University, 1995.



(a)



(b)

Figure 13: Behavior sequence analysis for behavior network with planning The x-axis is the number of behavior selections and y-axis shows behavior indexes.