World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:2, 2009

Enhancement Approaches for Supporting Default Hierarchies Formation for Robot Behaviors

Saeed Mohammed Baneamoon and Rosalina Abdul Salam

Abstract—Robotic system is an important area in artificial intelligence that aims at developing the performance techniques of the robot and making it more efficient and more effective in choosing its correct behavior. In this paper the distributed learning classifier system is used for designing a simulated control system for robot to perform complex behaviors. A set of enhanced approaches that support default hierarchies formation is suggested and compared with each other in order to make the simulated robot more effective in mapping the input to the correct output behavior.

Keywords—Learning Classifier System; Default Hierarchies; Robot Behaviors.

Open Science Index, Computer and Information Engineering Vol:3, No:2, 2009 waset.org/Publication/12460

I. Introduction

ONE of the important challenges in the development of robot in real world is achieving complex behaviors with high efficiency and accuracy [2].

This motivated designing a simulated system and training it to optimize its complex tasks in simulated environment to helps the designer to avoid problems that may appear when designing the system directly in real world [2].

In this study, we adopted type of genetic-based machine learning called learning classifier system; in which a distributed classifier system is used for designing a simulated control system for robot.

LCS is a production rules system whose learning process is guided by two learning mechanism. Bucket brigade algorithm (BBA) used to change the strengths of classifiers in classifier pool in order to classify the matching classifiers depending on their usefulness. Second genetic algorithm (GA) used to inject new classifiers to the classifier pool by its operators [3].

The architecture of LCS makes it possible for it to be used as main components in designing the simulated control system for robot to determine which behavior must be used [2], [12].

This paper is organized as follows: Section 2 presents the introduction of learning classifier system. In section 3, robot behavior is reviewed. The default hierarchy is described section 4. Section 5 describes the simulated system. Section 6 discusses the suggested approaches for supporting default hierarchies. Section 7 presents experimental results. Finally, conclusion of present research topic is presented.

II. Introduction To Learning Classifier System

Learning classifier system (LCS) was first introduced by Holland (1976). A LCS has three main components: the performance system, the apportionment of credit system and the rule discovery system [1].

The performance system consists of message list and classifier store. The information about the external environment is detected through the detectors and placed on the message list in the performance system and after a set of internal process an action is determined by its effectors. On the other hand, classifier store has a fixed size with a set of rules; each rule in the classifier store has the form of *if-then* statement. The *if-then* rules are encoded as strings with fixed length from the ternary alphabet {0,1,#} where # denotes "don't care" that makes the classifier more general because it is match either 0 or 1 [8], [10].

The Apportionment of Credit System (AOC) is determining the best classifier in matching pool through bucket brigade algorithm BBA that modified the strength of the classifiers in matching pool depending on their usefulness [3].

In AOC the processes auction, clearinghouse and taxation are achieved.

In auction process; all classifiers in matching pool are competed with each other. The bid of this classifier is calculated and the classifier with the highest bid is selected as the winning classifier [2], [3].

On the other hand in taxation process, two types of taxes are performed: Tax_{bid} and Tax_{life}. a fixed rate Tax_{bid} is applied to all classifiers in matching pool except the winner classifier while Tax_{life}, is also fixed rate applied to all don't match classifiers [2], [3].

Finally in clearinghouse process; the strength of classifiers is changed by distributing rewards and penalties. The rewards and penalties is used to decrease the strength of current winner by amount of its bid value and to increase its strength by reward value [2], [3].

In the rule discovery system the genetic algorithm (GA) is used to inject new rules through search method in space of rules by using GA operators; reproduction, selection, crossover and mutation. Finding good rule depends on some fitness function [5], [8].

There are two approaches to use GA in LCS, Michigan approach and Pittsburgh approach. In Michigan approach each classifier in the population represents a single individual. On the other hand in Pittsburgh approach a set of classifiers in the population represents a single individual [10].

Saeed Mohammed Baneamoon is with School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia (e-mail: saeed@cs.usm.my).

Rosalina Abdul Salam is with School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia. (Phone: + 604-6532486; fax: +604-6573335; e-mail: rosalina@cs.usm.my).

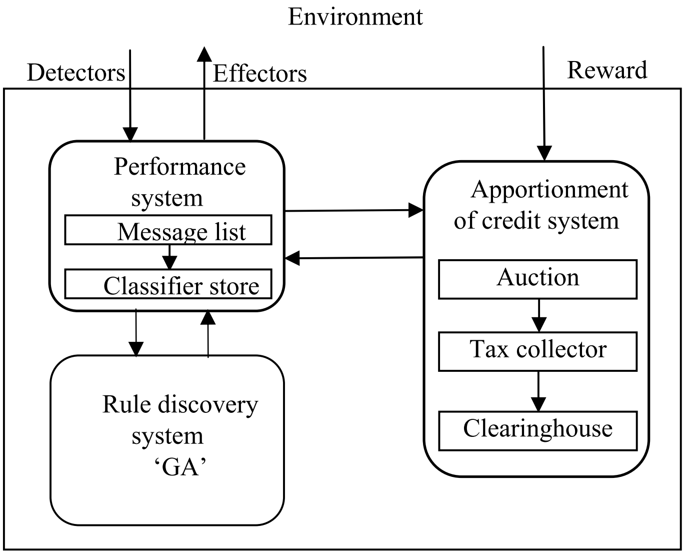

Fig.1 The structure of learning classifier system

III. Robot Behaviors

Behavior is the interaction between robot and its external environment. The robot senses information from the environment and acts on it by its sensors. There are two main classifications of behaviors, Stimulus – Response (S-R) and Dynamic behavior. In (S-R) behavior, the detectors are connected in a direct way with the effectors. On the other hand in dynamic behavior, the internal state is built between detectors and effectors [2].

Regarding the structure, the behaviors that do not partition into simpler behaviors are called basic behaviors, while the behaviors that can be divided into simpler behaviors are called complex behaviors [2].

IV. Default Hierarchies

Default hierarchy is defined as a rule sets that represent knowledge of learning classifier system about its environmental states. Default hierarchy makes the interaction with the environment more efficient and the system performance can be improved, on the other hand the learning process of the system will be graceful [6], [11].

Default rules in default hierarchy are covering most of the possible environmental states, because using of # symbol "don't care" that match either 0 or 1 makes the classifier more general. That is rules with many #'s have the ability to cover the specific and general conditions [6], [11].

In default hierarchy a rule set structure make extension of the set of correct solution, allowing small rule sets to perform small search space and the addition of exception rules can be reduce the error.[1]

The models of mapping the rules set are: a quasi-homomorphic model and an equivalent-homomorphic model. The set of rule in a quasi-homomorphic model is more general and less than the set of rule in an equivalent-homomorphic model. An example of default hierarchy is given in Fig.2.[2].

non-hierarchical set (homomorphic set)	hierarchical set (quasi-homomorphic set)
00;00→00	00;00→00
01;01→11	**;**→11
10;10→11	
11;11→11	

Fig. 2 The hierarchical set models [2]

The first approach is non-hierarchical set (homomorphic set) that represent the rule set with the complete map contains four rules, while the second approach is hierarchical set (quasi-homomorphic set) that represent the rule set with the map of two rules in which the second is the most general, covering sixteen messages. The last three rules in the Homomorphic set are the exception to the second rule in quasi-homomorphic set, each of them covering just one message of those sixteen messages [2], [6].

In LCS the default hierarchy has effective role of the processes of AOC [6].

V. Simulated System

In this research the simulated control system for the simulated robot is designed by using a distributed LCS system, which consists of a set of five classifier systems that is organized in three levels with hierarchical architecture [4].

The lower level in the simulated control system consists of three LCSs, each one of them has independent basic behavior. By these LCSs the information about the external environment is detected and competition between their behaviors occurs. On the other hand, levels two and three each of them consist of one LCS that achieves coordination between behaviors and determines final action of the simulated robot [4].

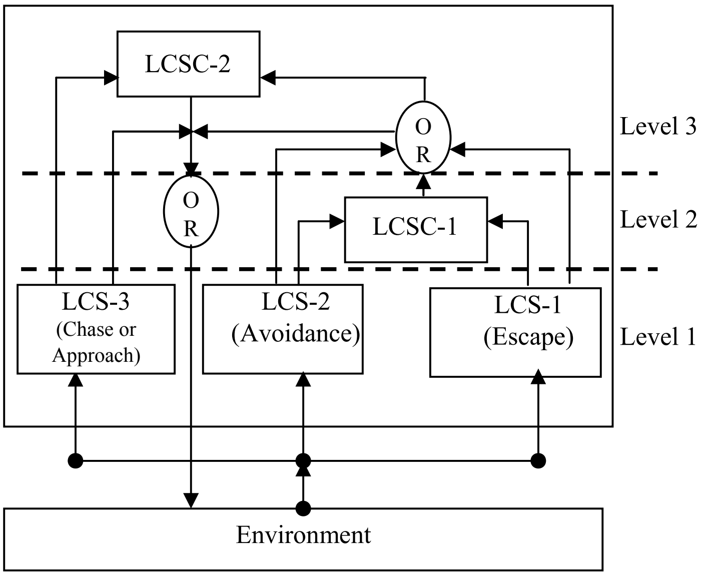

Fig. 3 The architecture of simulated control system [4]

The simulated environment consists of simulated autonomous robot that perceives the following objects: a moving object, which moves toward the goal and its initial position, will be random. The goal, obstacles and the lair will have a fixed position. In addition there is an emergency which could only be heard when the distance between a moving object and the goal becomes less than or equal to the known fixed distance [4].

In the simulated system, the complex behavior that must execute by the simulated robot consist of four basic behaviors, these behaviors are: escaping, avoiding, chasing and approaching behavior [4].

The rules in all LCSs are represented by using standard ternary alphabet strings {0,1,#} for conditions parts. While binary representation {0,1} is used for the actions parts, the environmental message, input message and output message with fixed length [13].

The environmental message is with 12 bits length. While the length of input messages for all LCSs in the lower level is 4-bits; and the length of their output messages are 4-bits.On the other hand the length of input messages for all LCSs in levels two and three is always 8-bits; and the length of their output messages is 4-bits [13].

The set of rules for all LCSs in the simulated system has the fixed size that consists of 164 rules [13].

On the other hand, in the beginning of experiments each rule in all set of rules has the initial strength equal to 50. The classifier bid coefficient C_{bid} is 0.45.

The GA parameters are set as follow: the probability of selection is 0.5, the probability of crossover is 0.5, and the probability of mutation is 0.008.

VI. Suggested Approaches For Default Hierarchies

In this study are suggested a set of enhanced approaches that deal with the competition between the exception rules and default ones in order to avoid failure of the simulated system which conduces to enhancement of the performance of the simulated robot related with the complex behaviors. These enhanced approaches are investigated through enhancement the algorithm of the performance system component of LCS.

The basic execution cycle of the performance system is as follows [2]:

Step 1: Read messages by detectors and place them on the current message list.
Step 2: Determine the matching classifiers by comparing all messages to all conditions.
Step 3: For each match generate a message for the new message list.
Step 4: Replace the current message list by the new message list.
Step 5: Process the new message list through the effectors to produce system output.
Step 6: Return to step 1.

In all enhanced approaches, the algorithm of the performance system is modified.

A. The first suggested approach is investigated by adding the second and the forth steps to the execution cycle in the performance system as follow:

Step 1: Read messages by detectors and place them on the current message list.
Step 2: Compute the mean of specificity for all default hierarchies (those rules with '#').
Step 3: Determine the matching classifiers by comparing all messages to all conditions.
Step 4: Add to matching pool only the matching classifiers that achieve the condition below:
If mclassifier & (σ < (Mean (σ) / nclassifier)
where
mclassifier: match classifier.
nclassifier : number of all classifiers.
σ : Specificity, which is defined as follow:

$$\sigma = \frac{\text{number of non - \# position}}{\text{length of classifier}}$$

Step 5: For each match generate a message for the new message list.
Step 6: Replace the current message list by the new message list.
Step 7: Process the new message list through the effectors to produce system output.
Step 8: Return to step 1.

B. The second enhanced approach differs from the first enhanced approach in computing the mean of specificity. In the execution cycle, the step 2 in the first enhanced approach is modified for this enhancement as follow:

Step 2: Compute the mean of specificity for all classifiers in the classifier store (those rules with or without '#').

C. The third enhanced approach differs from the first enhanced approach in condition statement that determines the classifiers injected to the matching pool. In the execution cycle the step 4 in the first enhanced approach is modified for this enhancement as follow:

Step 4: Add to matching pool only the matching classifiers that achieved the condition below:
If mclassifier & (σ < (Mean (σ)* X)
where
X: is parameter, in the experiment its value is 1.25.

D. The fourth enhanced approach differs from the first enhanced approach in computing the mean of specificity and also in the condition statement that determines the classifiers injected to the matching pool. In the execution cycle the steps 2 and 4 in the first enhanced approach is modified for this enhancement as follow:

Step 2: Compute the mean of specificity for all classifiers in the classifier store (those rules with or without '#').
Step 4: Add to matching pool only the matching classifiers that achieve the condition below:
If mclassifier & (σ < (Mean (σ)* X)

E. The fifth enhanced approach differs from the first enhanced approach in the steps 2 and 4 of the execution cycle that is modified for this enhancement as follow:

Step 2: For all classifiers in the classifier store (those rules with or without '#') is compute the following:

−Mean (σ)

−VAR(σ)

−STDEV(σ)

Step 4: Add to matching pool only the matching classifiers that achieve the condition below:

if mclassifier & (σ <= ((Mean(σ) + STDEV(σ)) / (Mean(σ) + STDEV(σ))))

VII. Experimental Results

In this study a set of experiments is executed and the effect of the simulated robot's performance is analyzed.

In the experiments analysis, the performance of the simulated robot is measured as the ratio of the number of correct moves to the total number of moves performed from the beginning of the simulation as shown in "(1)" [2].

$$\text{Performance} = \frac{\text{number of correct moves}}{\text{total number of moves}} \le 1 \qquad (1)$$

The results that show in Fig.4, Fig.5, Fig.6, Fig.7 and Fig.8 indicate the relationship between the number of iteration and performance of the simulated system for the five suggested enhanced approaches that support default hierarchies formation.

The results of the simulated robot's performance are analyzed for the five enhanced approaches that support default hierarchies formation and compared with each other. From the results, it is observed that the fifth enhanced approach is more efficient than the others enhanced approaches.

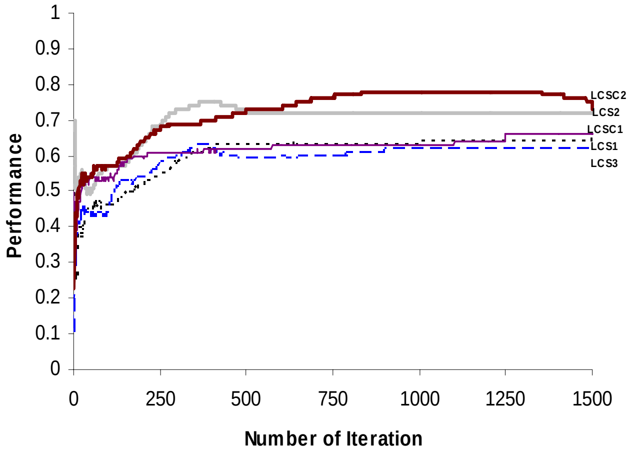

Fig. 4 Proportion of correct trials – first enhanced approach

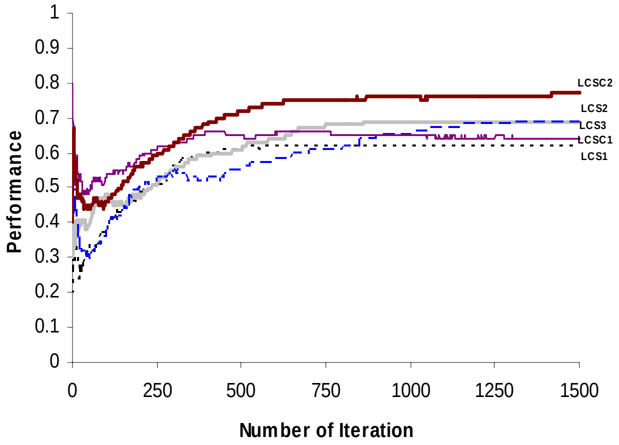

Fig. 5 Proportion of correct trials – second enhanced approach

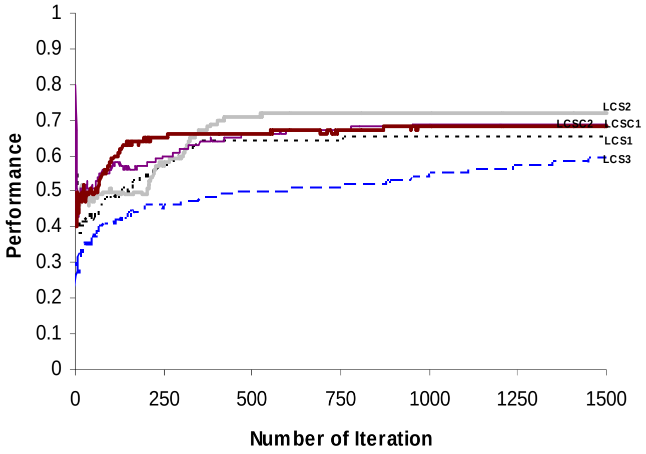

Fig. 6 Proportion of correct trials – third enhanced approach

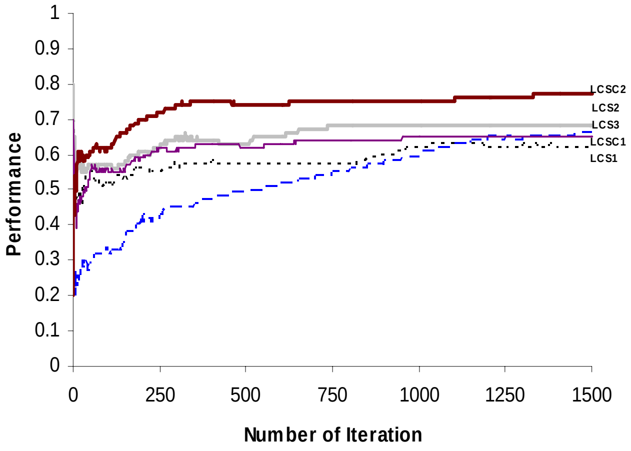

Fig. 7 Proportion of correct trials–fourth enhanced approach

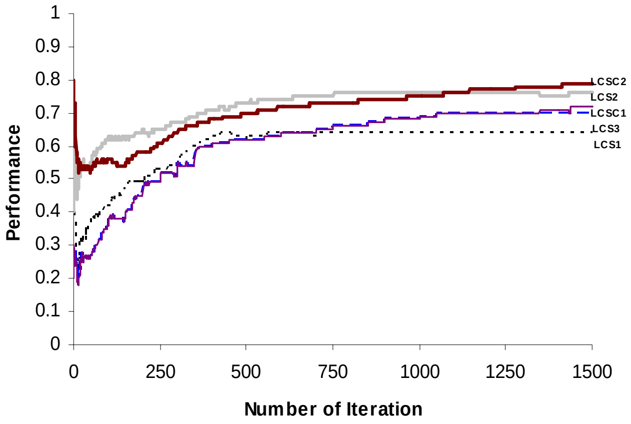

Fig. 8 Proportion of correct trials – fifth enhanced approach

VIII. Conclusion

This paper focuses on the improvement of the simulated robot to perform complex behaviors in a complex environment by using distributed LCS in designing simulated control system for the simulated robot.

In this study the execution cycle of the performance system in learning classifier is modified in order to support default hierarchies formation.

A set of approaches that support default hierarchies formation is suggested to avoid instability of the simulated system

The results from the experiments showed that applying the fifth enhanced approach that supports default hierarchies formation in the simulated system improves the performance of the simulated robot and makes it more effective in choosing the appropriate behavior.

Acknowledgment

Thanks and gratitude goes to those who provided support and encouragement.

References

[1] D. E. Goldberg, " Genetic Algorithms in Search, Optimization ", and Machine Learning, *Addison-Wesley, Reading, Mass*, 1989.

[2] M. Dorigo and M. Colombetti, " Robot shaping: An experiment in behavior engineering ", *The MIT Press*, Cambridge, Massachusetts, London, England, 1998.

[3] P-L Lanzi, W. Stolzmann, and S. W. Wilson, " Learning Classifier Systems: From Foundations to Applications", *Springer*, 2000.

[4] S. M. Baneamoon, R. Abdul Salam, A. Z. Hj. Talib, " Learning Process Enhancement for Robot Behaviors", *International Journal of Intelligent Technology*, Volume 2 Number 3, ISSN 1305-6417, 2007, pp. 172-177.

[5] D. E. Goldberg. " Sizing populations for serial and parallel genetic algorithms " – *Third international conference on genetic algorithms* , George Mason University, United States 1989, pp. 70 – 79

[6] M. Dorigo " New Perspective about Default Hierarchies Formation in Learning Classifier System", *Proceedings of the 2nd Congress of the Italian Association for Artificial Intelligence on Trends in Artificial Intelligence*, Lecture Notes in Artificial Intelligence, Vol. 549, Springer-Verlag London, UK 1991, pp. 218 – 227.

[7] M. Dorigo, " Message-Based Bucket Brigade: An Algorithm for the Apportionment of Credit Problem ", *Lecture Notes in Artificial Intelligence* 482, Springer, Berlin , 1991, pp. 235-244.

[8] M. Dorigo, " Genetic-Based Machine Learning and Behavior-Based Robotics: A New Synthesis ". 0018-9472/93, *IEEE TRANSACTIONS ON SYSTEM. MAN.* And CYBERNETICS, Vol. 23, No.1, January/ February 1993, pp. 141 - 154.

[9] M. Studely & L. Bull, " X-TCS: Accuracy-based Learning Classifier System Robotics ", In Proceedings of the IEEE Congress on Evolutionary Computation, Volume 3, 2005, pp. 2099 – 2106..

[10] P. Musilek, Sa Li, and L. Wyard-Scot, " Enhanced Learning Classifier System for Robot Navigation ", IROS 2005, IEEE/RSJ *International Conference on Intelligent Robots and Systems*, Alberta, Canada, 2-6 Aug. 2005, pp 3390- 3395.

[11] R. E. Smith and D. E. Goldberg. "Reinforcement Learning with Classifier System " – *AI, Simulation, and Planning in High Autonomy System ., Proceedings.* , IEEE, 1990, pp. 184 – 192.

[12] S. J. Bay, " Learning Classifier Systems for Single and Multiple Mobile Robots in Unstructured Environments", *Mobile Robots* X. Philadelphia, PA, Nov. 1995, pp. 88-99.

[13] S. M. Baneamoon, R. Abdul Salam, " Bucket Brigade Algorithm Enhancement for Robot Behaviors ", *International Conference on Robotics, Vision, Information and Signal Processing* (ROVISP 2007), Penang, Malaysia, 28-30 November 2007, pp. 930-934.

[14] A. C. Schultz, " Learning robot behaviors using genetic algorithms ", *Proceedings of the First World Automation Congress* (WAC '94), TSIP Press: Albuquerque, New Mexico, August 1994, pp. 607-612.

Saeed Mohammed Baneamoon received the B. Sc. and the M.Sc. degrees in computer engineering from Technical University, Sofia, Bulgaria in 1996, the M.Sc. degree in computer science from University of Technology, Baghdad, Iraq in 2003 and is currently a Ph.D. candidate in School of Computer Sciences, Universiti Sains Malaysia, USM, Penang, Malaysia. His research interest focuses on the field of Artificial Intelligence and Robotics.

Rosalina Abdul Salam is an Associate Professor at the School of Computer Sciences, Universiti Sains Malaysia and a member of Artificial Intelligence Research Group. She received her Bachelors degree in Computer Science in 1992 from Leeds Metropolitan University, United Kingdom. She was a system analyst in Intel Penang, from 1992 to 1995. She returned to United Kingdom to further her studies. She received her Masters degree in Software Engineering from Sheffield University, United Kingdom in 1997. She completed her PhD in 2001 from Hull University in the area of computer vision.

She has published more than 70 papers in journals and conferences. She is a member of International Computational Intelligence Society and World Enformatika Society. Recently she joined the editorial board of the International Journal of Computational Intelligence and the International Journal of Signal Processing.

Presently, she is continuing her teaching, graduate supervisions and her research. Her current research area is in the area of artificial intelligence, image processing and bioinformatics applications. The most recent project that she is working is on underwater images and cellular images.