Available online at www.worldscientificnews.com

World Scientific News

WSN 36 (2016) 12-26 EISSN 2392-2192

# Reinforcement-Based Learning for Process Classification Task

**Lubna Zaghlul Bashir**

Technology University, Baghdad, Iraq

E-mail address: lubna_zaghlul@yahoo.com

**ABSTRACT**

In this work, we present a reinforcement-based learning algorithm that includes the automatic classification of both sensors and actions. The classification process is prior to any application of reinforcement learning. If categories are not at the adequate abstraction level, the problem could be not learnable. The classification process is usually done by the programmer and is not considered as part of the learning process. However, in complex tasks, environments, or agents, this manual process could become extremely difficult. To solve this inconvenience, we propose to include the classification into the learning process. We apply an algorithm to automatically learn to achieve a task through reinforcement learning that works without needing a previous classification process. The system is called Fish or Ship (FOS) assigned the task of inducing classification rules for classification task described in terms of 6 attributes. The task is to categorize an object that has one or more of the following features: Sail, Solid, Big, Swim, Eye, Fins into one of the following: fish, or ship. First results of the application of this algorithm are shown Reinforcement learning techniques were used to implement classification task with interesting properties such as provides guidance to the system and shortening the number of cycles required to learn.

***Keywords*:** Reinforcement Learning; Reward; Classification; Bucket Brigade Algorithm

## 1. INTRODUCTION

The word learning can cover many aspects of processes in cognitive agents. The famous definitions of the word are:

“Learning denotes changes in a system that ... enable a system to do the same task more efficiently the next time.”

“Learning is constructing or modifying representations of what is being experienced”. [1].

The study of adaptation involves the study of both the adaptive system and its environment, it is the study of how systems can generate procedures enabling them to adjust efficiently to their environment. if adaptability is not to be arbitrarily restricted at the outside, the adapting system must be able to generate any method or procedure capable of an effective definition [2].

Reinforcement learning is a framework to learn from delayed reward/punishment for a model of both animal and robot learning. To make it more practical to design an intelligent machine, it would be better to be able to combine with human knowledge. Reinforcement learning (RL) is one of the key functions of animal learning in which the learner discovers better strategy to avoid harmful stimuli and to obtain pleasant sensation through its own experience. At the same time, it is also a suitable model for intelligent machines to adapt against un expectable environment. A number of researchers have proposed different types of basic algorithms to realize this type of learning from delayed reward/punishment by machines, and a lot of orientations to extend this framework have also been proposed to apply them to more complicated practical problems [3].

## 2. REINFORCEMENT LEARNING

Reinforcement learning allows an agent to learn sequential decision tasks by trial and error interaction with the environment. The agent is able to perform one of a number of actions in order to interact with the environment. The environment is represented by a number of parameters which collectively define the current state and the task of the learning process is to determine the appropriate action to apply for any given state. The various techniques that comprise reinforcement learning are used to derive a matrix of state action pairs that determine behavior which is known as a policy.

Reinforcement learning differs from supervised learning which provides the learning process with information on the correct action to select for a particular state. Instead of information on the appropriate action to perform, the reinforcement learning process receives a reward value as it operates in the environment. This reward is matched with information on the current state and used to create a policy which provides progressively better action selection as the agent gains experience in the environment [4].

### 2. 1. The Reinforcement Learning framework

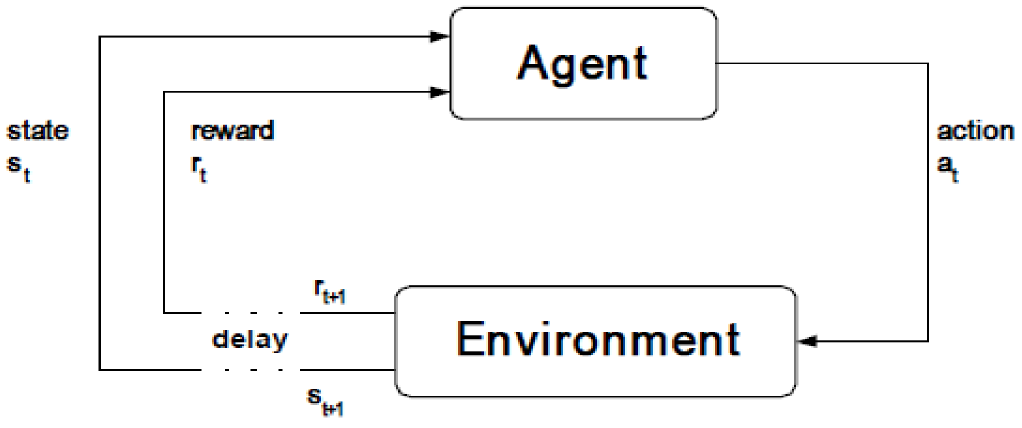


**Figure 1.** Reinforcement Learning System.

- The agent does not know anything about the environment but St
- The agent's goal is to maximize the amount of rewards it receives in the long run i.e., the payoff.
- What happened a long time ago can affect the reward, i.e., rewards can be delayed.figure.1 illustrate reinforcement learning system [5].

## 3. LEARNING CLASSIFIER SYSTEMS

Classifier systems are rule based systems that learn by adjusting rule strengths from environmental feedback and by discovering better rules using genetic algorithms. In the following a simplified classifier system is used where all possible message action pairs are explicitly stored and classifiers have one condition and one action. Following their notation, a classifier i is described by (ci; ai), where ci and ai are respectively the condition and action parts of the classifier. St(ci; ai) gives the strength of classifier i at time step t. [6,7].

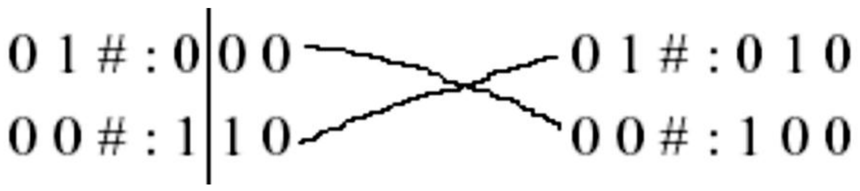
All classifiers are initialized to some default strength. At each time step of problem solving, an input message is received from the environment and matched with the classifier rules to form a match set, M. One of these classifiers is chosen to fire and, based on its action, a feedback may be received from the environment. Then the strengths of the classifier rules are adjusted. This cycle is repeated for a given number of time steps. A series of cycles constitute a trial of the classifier system. In the bucket brigade algorithm (BBA) for credit allocation, when a classifier is chosen to fire, its strength is increased by the environmental feedback. But before that, a fraction α of its strength is removed and added to the strength of the classifier that fired in the last time cycle. So, if (i) the firing of classifier i at time step t results in an external feedback R and (ii) classifier j fires at the next time step, the equation (1) gives the strength update of classifier i: [8-10].

$$St + 1(ci; ai) = (1-\alpha)\_St(ci; ai) + \alpha*(R + St + 1(cj; aj)) \qquad (1)$$

Genetic Algorithms are used to search for new plausible classifiers or rules by recombining good classifiers to produce new ones [11]. A situation may arise when the environmental message string may not find any matching classifier. The system should be robust enough to deal with such situations. This is dealt by unleashing the power of Genetic Algorithms. The tripartite process of reproduction, crossover and mutation is used to produce temporary classifiers. The fitness function for these classifiers is in accordance with the message sent.

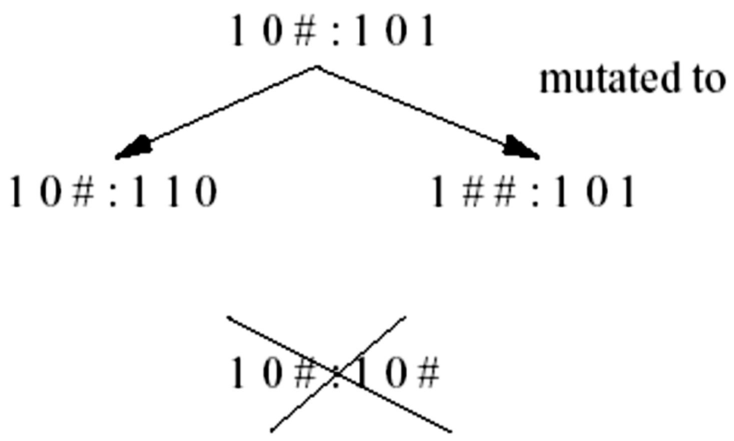
### 3. 1. Crossover

The GA crosses the classifiers in the normal way For example: Classifiers to be used in crossover are selected by Roulette Wheel Selection.



### 3. 2. Mutation

Mutation must now randomly select 1 of {0, 1, #} for the condition part of the classifier and 1 of {0, 1} for the action part. The action part is not allowed to have a wild card in it. For example: [11,12].



### 3. 3. Basic Operation of LCS

The cycle of an LCS, which is repeated sequentially is as follows:

1. The input interface generates messages which are posted into the message list.
2. The messages are matched against the conditions of all classifiers and if the matching rules don't advocate the same action, conflict resolution occurs and an action is chosen.
3. The message list is emptied and the encoded actions are posted to the message list.
4. Finally we activate the output interface from the message list and perform the actions they describe. We get rewards back from the environment and update the predictions of our classifiers [12-14].

A classifier system consists of three main components as illustrated in Figure 2 [15]:

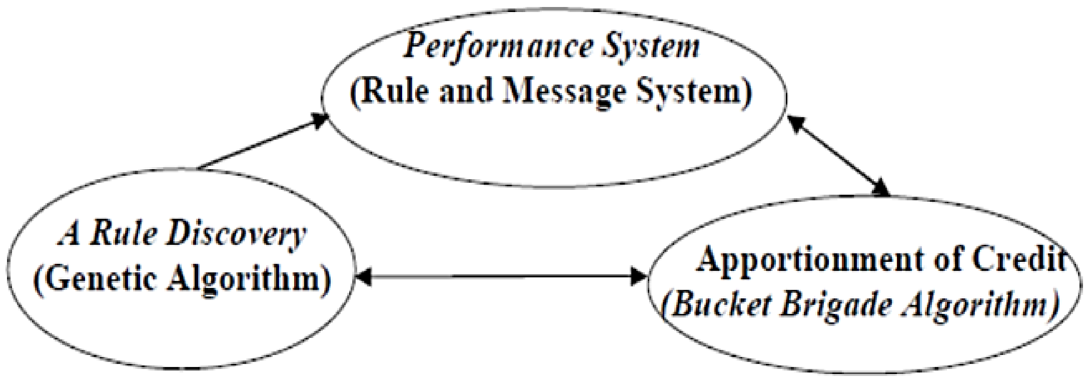


**Figure 2.** Learning classifier system component.

The basic structure of the rule and message system is presented in Figure 3 [16].

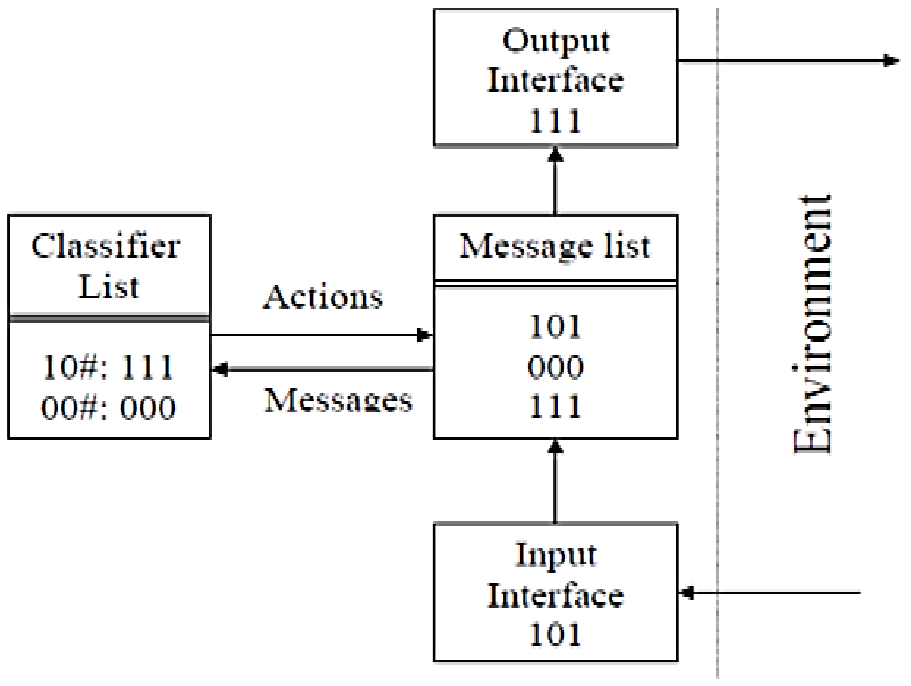


**Figure 3.** Rule and Message System.

## 4. REINFORCEMENT LEARNING AND CATEGORIZATION

A general description of a reinforcement learning situation includes the following elements:

- **A definition of the task**: The only strictly needed information about the task is a signal, called the reinforcement signal that becomes active when the task is achieved. This signal is. If more information about the task is available (as for instance, correct or incorrect actions in some situations, or necessary preconditions for achieving the task), it can be added to the reinforcement signal to help the learner.
- **An environment in which to accomplish the task**: Depending on the kind of environment (static, dynamic,...) the resolution of the task can be less or more difficult.
- **An agent that must complete the task**: This agent can execute actions. The solution of a task consists in determining adequate actions to activate the reinforcement signal as frequently as possible. The first attempts to formalize the framework of reinforcement learning were at a high level of abstraction, and the resulting formalization was not as general as the previous description. The main assumption of this formalization is to consider the environment as a state machine controlled by the agent, so that the interaction between the environment and the agent is arranged in a two step loop: In the first step, the agent perceives the state of the environment and, in the second one, it performs an action that produces a change in this state. In this formalization we have:
- **States:** Representing relevant situations for the achieving of a task to which the agent should respond with the correct decision. The agent can directly perceive the state of the

system, or at least obtain sensor readings that depend probabilistically (in a constant way) on the state.

- **Actions:** Devised as a list of options from which the agent picks the most appropriate for each state. At each moment the agent only performs one action from the list.
- **Reinforcement function**: It is a function that maps transition between states to a real numbers. In general this function is zero except for some transitions that are considered the goals of the current task (for instance, reaching a specific position, grasping an object,...).
- **Policy:** A mapping from states to actions. Each policy is evaluated according to a criterion that depends on the reinforcement function. Reinforcement learning algorithms aim to find the optimal policy for the task at hand.
- **Model of the agent-environment interaction**: Including the transition probabilities from one state to another under the execution of each action [17,18]. Figure 4, illistrare learning classifier system.

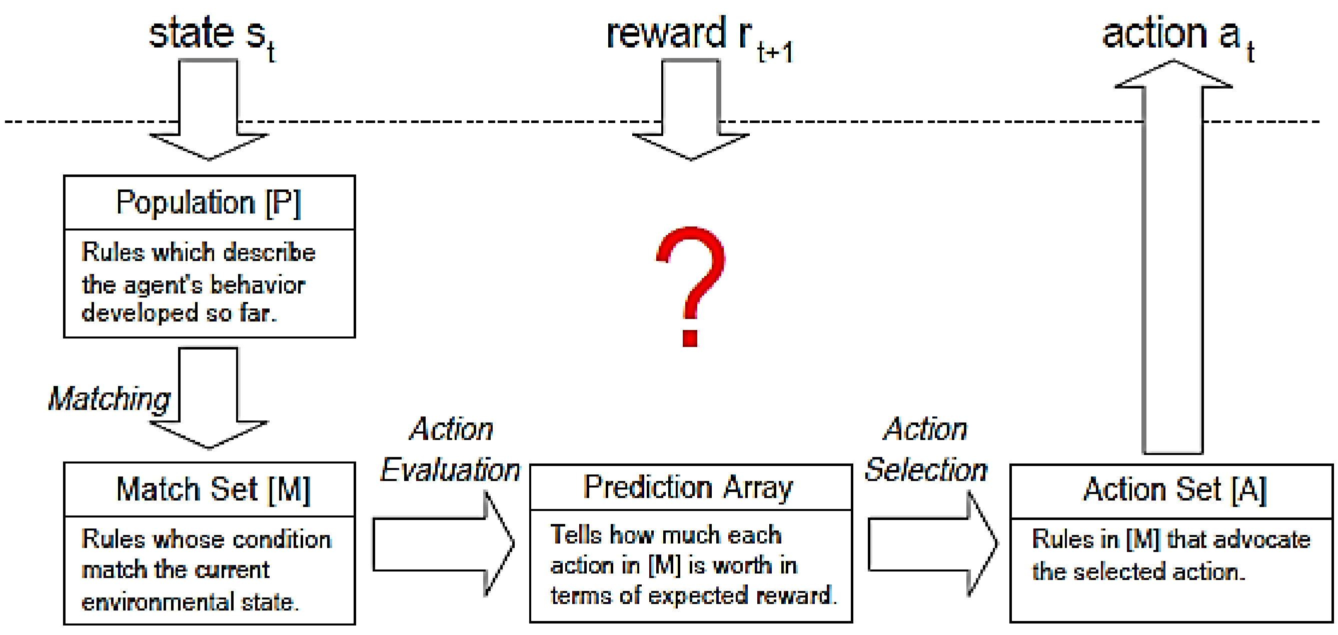

state $s_t$
reward $r_{t+1}$
action $a_t$
Population [P]
Rules which describe the agent's behavior developed so far.
?
Matching
Action Evaluation
Action Selection
Match Set [M]
Rules whose condition match the current environmental state.
Prediction Array
Tells how much each action in [M] is worth in terms of expected reward.
Action Set [A]
Rules in [M] that advocate the selected action.

**Figure 4.** Learning Classifier System.

## 5. FISH OR SHIP CLASSIFICATION TASK (FOS): THE CASE STUDY

The system is called fish or ship (FOS) assigned the task of inducing classification rules for categorization task described in terms of 6 attributes. The task is to categorize an object that has one or more of the following features: Sail, solid, big, swim, eyes, fins color into one of the following: fish, or ship.

### 5. 1. Coding (LCS – FOS) Conditions

The learning classifier system (LCS - FOS) receives 6 – bit messages from environment mapping it to 64 states from 0 to 63 of six bits only. The six bit represents as follows:

- First bit represented is object has Sail (1 means the object has Sail, 0 means that object do not has Sail).
- Second bit represented is object solid (1 means the object is solid, 0 means that object is not solid).
- Third bit represent is object big (1 means the object is big, 0 means that object is small).
- Four-bit represent is object has swim (1 means the object has swim, 0 means that object don't swim).
- Five bit represent is object has eyes (1 means object has eyes, 0 means object don't has eyes).
- Six-bit represent is object has fins (1 means object has fins, 0 means object don't has fins).

### 5. 2. Coding (LCS – FOS) Actions

LCS – FOS has an action consisting of only one bit, LCS – FOS action has the form and meaning as follows:

- 1 means the object is Ship.
- 0 means the object is Fish.

### 5. 3. Representation of (LCS – FOS)

Performance system of the LCS-FOS consists of a message list and classifier store. The classifier list of LCS-FOS contain a set of rules called classifiers, which represents the knowledge and controller of the system at execution time. Condition part of classifier consists of (6 bit), and action part consists of (1 bit). The size of classifier store for LCS-FOS will be (64) Rules and all classifiers have the same strength value at the beginning.

**Example**

The representation of the rule "If object has Sail & solid & big & has not swim & has not eyes & not fins then the object is ship ":

| **Sail** | **solid** | **big** | **swim** | **eyes** | **fins** | **/** | **Ship** |
|---|---|---|---|---|---|---|---|
| **1** | **1** | **1** | **0** | **0** | **0** | **/** | **1** |

### 5. 4. The Implemented of the (LCS – FOS) System

The characteristics of the enhanced version of the classifier learning system developed and implemented for the LCS – FOS are given below.

#### 5. 4. 1. Classifier List of (LCS – FOS)

For the sake of simplicity, each classifier had one condition and one action. The condition was represented as a fixed string of characters defined over the alphabets {1, 0, #}. A '1' indicates a feature that must be present, '0' represents a feature that must be absent and

'#' indicates a feature that may or may not be present. The action part was also represented as a string of characters over the same three alphabets as the condition part.

### 5. 4. 2. Message List of (LCS – FOS)

The message list was made up of training examples and messages generated by the classifiers. The training examples were not removed from the message list until the end of the learning session. They were 'fed' into the system again after each generation. This departs from the standard practice of generating a completely new message list at each generation. The procedure implemented here ensured that newly produced classifiers had a chance of bidding for and classifying the training examples no matter when they were generated, and it also ensured that messages which were not recognized at previous generations were not lost.

### 5. 4. 3. Initial Classifier and Classifier Strength (LCS – FOS)

An example pattern for each class was randomly chosen, and used to form part of the initial classifiers. Some features were randomly converted to don't cares and pass through. Pairs were then randomly selected from these initial classifiers and 'mated' to produce the remaining members of a fixed-size population of classifiers. Each newly generated classifier is assigned a value which is modified by the bucket-brigade to reflect is usefulness. This value is the same initially for each classifier.

### 5. 4. 4. Bucket Brigade of (LCS – FOS)

The bucket-brigade algorithm was modified so that external payoffs (rewards or punishment), when received, are paid to the classifier responsible for the action that earned the payoff and to all those that had taken part (in previous generations) in helping to arrive at that goal.

It should be noted that in the standard classifier system, payoffs are added to the strengths of classifiers active at the time the payoff is received. The new measure was introduced to overcome some of the problems encountered during the initial stages of the experiments. It was discovered that some bad classifiers were rewarded simply because they happened to be active when payoff was received. It was designed to prevent this from happening. The measure was also designed to speed up the passage of reward or punishment to classifiers responsible for previous actions that led to the receipt of the payoff. In a standard implementation of the bucket-brigade algorithm, a particular sequence of actions will have to be repeated several times before payoffs could flow through to the classifiers that were active at the early stages.

### 5. 4. 5. Replacement of Classifiers of (LCS – FOS)

In addition to the strength of a classifier which is modified by the bucket-brigade, another metric called no of wins was introduced to determine when a classifier is to be removed from the fixed-size classifier list so as to have room for new ones. The metric is a simple count of the number of times a classifier succeeded in bidding for the right to place messages onto the message list. Whenever new classifiers are to be introduced into the system, those whose strengths fell below the fixed value assigned to each newly generated classifier and those whose no-of-wins is zero are replaced.

Standard classifier systems typically employ only the strength of a classifier to determine the 'life' span of the particular classifier. Therefor crowding replacement were used in FOS problem to choose the classifiers that die to make room for new offspring. Crowding algorithm is shown below.

```
For 1 to crowding factor do
        x: = find worst of a random set
         If this is not more similar to offspring then past x then
                Set worst most similar to x
        End if
End for
    Replace worst most similar with offspring
```

### 5. 4. 6. The Reinforcement Learning of (FOS) System.

The way that is used to provide system with reinforcement learning is by allowing the system to control the reinforcement task by using reinforcement algorithm.

Reinforcement algorithm is preferable to providing it by a human trainer who observes the performance of the system and provides positive or negative reinforcement according to action to be taken.

The reinforcement is provided in many ways by using reward and punishment that is by rewarding positive action and punishing negative action, or by only using reward to provide it.

In FOS System we using a reward algorithm only to provide reinforcement and the amount of reward is (10) to reward good action and (0) for bad action for LCS, following procedure describe function of reinforcement learning using reward only:

```
If (classifier output = desired output) then
Add (Reward = 10) to classifier strength responsible for sending action
Else
Add (Reward = 0) to classifier strength responsible for sending action
```

### 5 .4. 7. The FOS System Cycle

The using of FOS starts with insertion environment messages to the FOS system. Each environment message consists of 6–bit. These environment messages are received from detectors of the LCS transfer them to its performance system and executed them sequentially only one message in each cycle.

In the performance system of the LCS is performed matching process for each environment message with the condition part of all classifiers in classifier store.

All classifiers that matched with environment message are sent to the AOC system and use reinforcement learning to reward the winner, then a system is calls GA to inject new rule which may increase the performance of the system.

Figure 5 illustrate FOS cycle.

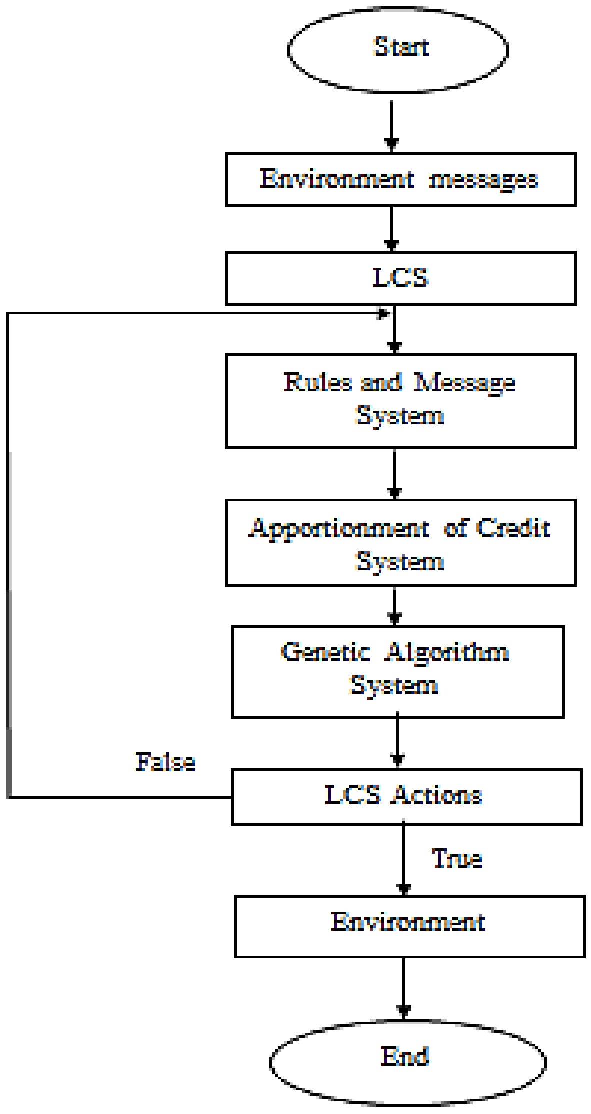


**Fig. 5.** LCS -FOS System Cycle.

## 6. EXPERIMENTAL RESULTS

The whole project execute using Pascal language, Executing the (FOS) code, The system was able to learn rules for the given task using only a few training examples and starting with classifiers that were randomly generated.

The system responds by presenting the initial report for LCS-FOS. The classifier system run for 1000 iterations, termination with the snapshot report for LCS-FOS, the initial and last report display in Appendix A. The correct rules have achieved high strength values; by contrast, the bad rules have strength and bid values near zero.

The classifier system eliminates the bad rule quickly thereby achieving near perfect performance. The performance of the system after 1000 iteration illustrated in Table 1 and data chart illustrated in Figure 6.

**Table 1.** The FOS System Performance After 1000 Iterations.

| Number of iteration | Proportion correct |
|---|---|
| 100 | 0.92000 |
| 200 | 0.97000 |
| 300 | 0.98667 |
| 400 | 0.99000 |
| 500 | 0.99200 |
| 600 | 0.99333 |
| 700 | 0.99429 |
| 800 | 0.99500 |
| 900 | 0.99556 |
| 1000 | 0.99600 |

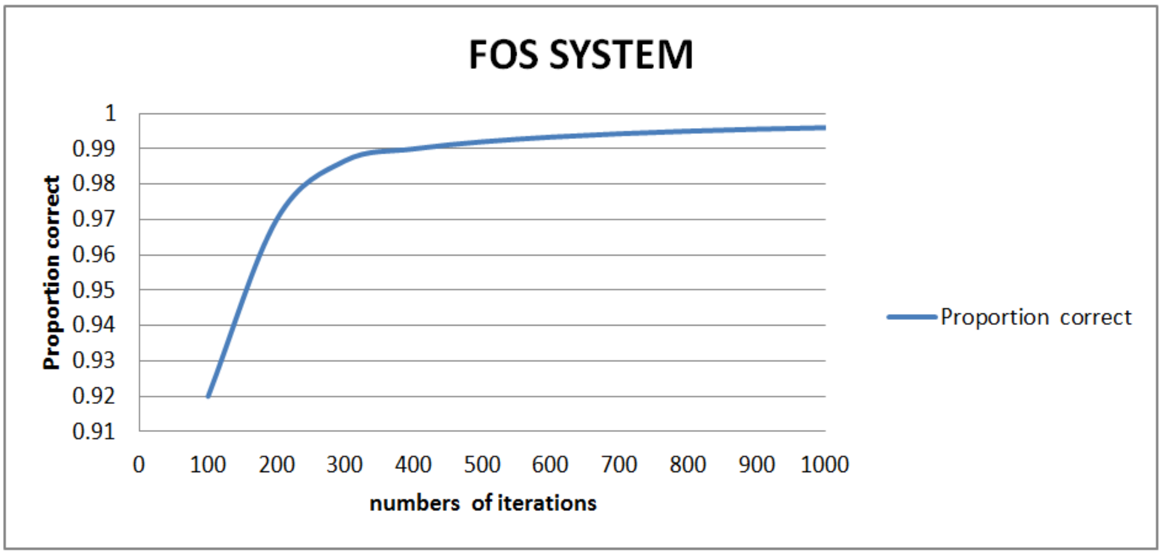


**Fig. 6.** The FOS system Performance after 1000 iterations.

## 7. CONCLUSIONS

1. The system was able to learn rules for the given task using only a few training examples and starting with classifiers that were randomly generated.
2. The parallel implementation of the algorithm would speed up the training process.

3. Reinforcement learning techniques were used to implement classification task with interesting properties such as provides guidance to the system and shortening the number of cycles required to learn.
4. A genetic based approach – an enhanced version of the Holland classifier learning system – was able to evolve learning classification rules.
5. The use of Genetic Algorithms to search for new plausible rules, the method should be able to cope with changing conditions.
6. The system was able to learn rules for the given task using only a few training examples and starting with classifiers that were randomly generated.

## References

[1] Luger and Stubblefield. “Artificial Intelligence” Structures and Strategies for complex Problem solving, Benjamin / Cummings, Menlo Park, CA, 1998.

[2] Goldberg, David E. “Genetic Algorithm in Search, Optimization, and Machine Learning”, Addison Wesley Longmont, International Student Edition 1989.

[3] Unemi, Tatsuo,” Scaling up Reinforcement Learning with Human Knowledge as an Intrinsic Behavior”, Dept. of Information Technology, University of Zurich, Switzerland unemi@ifi.unizh.ch, Dept. of Information Systems Science, Soka University, Japan unemi@iss.soka.ac.jp, 1999.

[4] Crook Stamati, ”Evolving expert systems for autonomous agent control using reinforcement learning”, M.Sc. Thesis, Evolutionary and Adaptive Systems. School of Cognitive and Computing Sciences. Sussex University. stamati.crook@redware.com, September 2003.

[5] Pier Luca Lanzi, ”An Introduction to Learning Classifier Systems”, Artificial Intelligence and Robotics Laboratory, Department di Electronicae Information, Politecnico di Milano, pierluca.lanzi@polimi.it, 2002.

[6] Sen, Sandip and Weiss, Gerhard, ”Learning in Multiagent Systems”, 1999.

[7] Bacardit Jaume, Ester Bernad´o-Mansilla, and Martin V. Butz, ”Learning Classifier Systems: Looking Back and Glimpsing Ahead”. ASAP research group, School of Computer Science, Jubilee Campus, Nottingham, NG8 1BB and Multidisciplinary Centre for Integrative Biology, School of Biosciences, Sutton Bonington, LE12 5RD, University of Nottingham, UK, 2009.

[8] Bull Larry, “Learning Classifier Systems: A Brief Introduction”, Faculty of Computing, Engineering & Mathematical Sciences University of the West of England, Bristol BS16 1QY, U.K. Larry, 2004.

[9] HuntJohn,"learningclassifiersnystems", Jaydee Technology Ltd, Harthamn Park, Corsham, Wiltshire, SN13ORP, 2002.

[10] Robert Elliott Smith, Max Kun Jiang, Jaume Bacardit, Michael Stout, Natalio Krasnogor, Jonathan D. Hirst, ”A learning classifier system with mutual-information-based fitness”, UK Engineering and Physical Sciences Research Council (EPSRC), 2010.

[11] Odetayo Michael O., “On Genetic Algorithms in Machine Learning And Optimisation”, PhD Thesis, University of Strathclyde, Glasgow, U.K. 1990.

[12] Brownlee Jason, ”Learning Classifier Systems”, Technical Report 070514A,Complex Intelligent Systems Laboratory, Centre for Information Technology Research, Faculty of Information and Communication Technologies, Swinburne University of Technology, Melbourne, Australia, jbrownlee@ict.swin.edu.au, 2007.

[13] Kovacs Tim, ”Advanced topics in machine learnin strength or accuracy Fitness Evaluation in Learning Classifier Systems”, UNIVERSITY OF BRISTOL, 2002.

[14] Ryan J. Urbanowicz and Jason H. Moore, ”Learning Classifier Systems: A Complet Introduction, Review, and Roadmap”, Department of Genetics, Dartmouth College, Hanover, NH 03755, USA, 2009.

[15] Zhou Qing Qing and Purvis Martin, “A Market-Based Rule Learning System” a Guang Dong Data Communication Bureau China Telecom  Dongyuanheng Rd., Yuexiunan, 2004.

[16] Nagasaka Ichiro & Kikuchi Makoto & Kitamura. Shinzo, “A Formal Analysis of Classifier System and Interface Between Learning System and Environment”. Department of Computer and Systems Engineering, Kobe University, 2002.

[17] Porta Josep M., ”Reinforcement-Based Learning with Automatic Categorization”, Institut  de Robòtica I Informàtica Industrial (CSIC, UPC), Gran Capità 2-4, 08034, Barcelona (SPAIN), jporta@iri.upc.es, 1999.

[18] Kelly Ian Darrell, ”The Development of Shared Experience Learning in a Group of Mobile Robots.”, Thesis submitted for the Degree of Doctor of Philosophy Department of Cybernetics, University of Reading, 1997.

( Received 01 January 2016; accepted 12 January 2016 )

## Appendix – A

```
-----------------------------------------
Fish or Ship System (FOS)
-----------------------------------------
population parameters
---------------------
number of classifiers    =     16
numberof positions       =      6
bid coefficient          =  0.1000
bid spread               =  0.0750
bidding tax              =  0.0100
existence tax            =  0.0200
generality probability   =  0.5000
bid specificity base     =  0.2500
bid specificity mult.    =  0.1250
edid specificity base    =  0.2500
ebid specificity mult.   =  0.1250
```

```
total number of bits   =      3

apportionment of credit parameters
----------------------------------
bucket brigade flag   =     false

reinforcement parameters
------------------------
 reinforcement reward    =    10.0

Timekeeper parameters
---------------------
Initial iteration           =      0
Initial block               =      0
Report period               =    1000
Console report period       =      50
Plot report period          =      50
Genetic algorithm period    =      50

genetic algorithm parameters
-----------------------------
proportion of select/gen    =  0.8000
Number of pairs to select   =      6
p mutation                  =  0.0030
p crossover                 =  1.0000
crowding factor             =      3
crowding population         =      3
snapshot report
---------------
[block:iteration]  -  [0:0]
current statius
---------------------------
signal            =    000111
desired output     =  0
classifier output  =  0
enviromental message:      000

no.     strength       bid      ebid   M   classifier
-------------------------------------------------------

   1     10.00      0.00      0.00       111000:[1]
   2     10.00      0.00      0.00       110000:[1]
   3     10.00      0.00      0.00       011000:[1]
   4     10.00      0.00      0.00       101000:[1]
   5     10.00      0.00      0.00       100000:[1]
   6     10.00      0.00      0.00       010000:[1]
   7     10.00      0.00      0.00       001000:[1]
   8     10.00      0.00      0.00       000111:[0]
   9     10.00      0.00      0.00       000011:[0]
  10     10.00      0.00      0.00       000110:[0]
  11    10.00    0.00    0.00      000101:[0]
  12    10.00    0.00    0.00      000001:[0]
  13    10.00    0.00    0.00      000010:[0]
  14    10.00    0.00    0.00      000100:[0]
  15    10.00    0.00    0.00      ######:[1]
  16    10.00    0.00    0.00      ######:[0]

new winner[1] : old winner[1]

Initial Report for LCS – FOS
[block: iteration]  -  [0:1000]
current  status
----------------
signal     =    000111
desired output      =      0
classifier output    =      0
environmental message:     000111
no.  strength   bid    ebid    M       classifier
-------------------------------------------------------
   1    30.99    0.00    0.00             0011#0:[0]
   2    30.99    0.00    0.00             0101#1:[0]
   3     0.00    0.00    0.00             #101##:[0]
   4    18.56    1.91    1.95      x      0#0111:[0]
   5     0.00    0.00    0.00             100000:[1]
   6     0.90    0.00    0.00             11#1#1:[0]
   7     2.37    0.21    0.11      x      0011#1:[0]
   8     0.28    0.00    0.00             11#1#1:[0]
   9    30.99    0.00    0.00             0111#0:[0]
  10    30.99    0.00    0.00             0101#1:[0]
  11     0.31    0.00    0.00             11#1#1:[0]
  12     0.35    0.03    0.12      x      0#01#1:[0]
  13     0.28    0.03   -0.04      x      0011#1:[0]
  14     1.66    0.00    0.00             0111#0:[0]
  15    30.99    0.00    0.00             0111#0:[0]
  16    76.93    7.69    7.76      x      000111:[0]

New winner [16]: old winner [16]
Last Report for LCS – FOS
```