Compare Between Flat and Hierarchical Architecture In Learning Classifier System for Robotics Controller

Lubna Zaghlul Bashir

Abstract—A control system is an interconnection of components forming a system configuration that will provide a desired system response. A network of different Classifier Systems can implement the control system of an agent. The issue of architecture is therefore the problem of designing the network that best fits some various classes of robotics behaviors. The case study is system called (FTS),The system Seek For Food And Avoid Tree (FTS) built of three-classifier subsystem work together, each classifier system learns a simple behavior, the system as a whole has as its learning goal the control of robot activities. two levels hierarchical architecture was used. The hierarchical organization allows distinguish between two different learning activities: the learning of basic behavior and the learning of switch behavior. two classifier systems in hierarchical model, learn basic behavioral,(chase / avoid), they are used, to learn the simulated robot, single step movement in every direction in the environment, where the switch classifier systems learn to control the activities of basic classifier systems, they are used to learn, to choose between basic behavior using suppression as a composition mechanism to chose between two basic behaviors which represent complex behavior. The Seek for food and avoid Tree (FTS) system is repeated again using flat architecture by built of two learning classifier systems only (instead of three classifier systems) Organized in one level architecture(instead of two level architecture),In this work we compare between the flat architecture and the hierarchal architecture ,difference in performance between these two architectures is examined. Experimentally show that the use of hierarchal architecture can help to control the complexity of learning. our study shows that learning classifier system are a feasible tool to build control system .to achieve this goal we found very helpful to decompose the desired overall task into a set of simpler interacting classifiers organized in a hierarchy. these interacting behaviors were implemented as a set of classifier systems using a distributed architecture in which each classifier system runs on a different set of processors. this choice was sufficient to achieve adequate levels of performance for a variety of tasks. Results using hierarchal architecture show improvement over the flat architecture by reduce the learning complexity ,reduce time need to perform task, and signal accuracy.

Keywords — Flat Architecture, Hierarchal Architecture, Robotics Behavior, Classifier System.

1.INTRODUCTION TO LEARNING CLASSIFIER SYSTEMS

Classifier system is using as a machine learning system that learns syntactically simple string rules (called classifier) to guide its performance in an arbitrary environment .a classifier system consist of three main components:

A. Performance System (Rule and Message System).
B. Apportionment of Credit System (Bucket Brigade Algorithm).
C. Genetic algorithm (Rule Discovery).[1],[2,[3]

A. The Performance System.

The performance system is composed of:

1.Classifier list

The classifier list is the system's long term memory. It is made up of a population of classifiers. A classifier is made up of one or more conditions (known as the condition part) and one action (called the action part). The condition part specifies the set of messages to which a classifier is sensitive, and the action part indicates the message it will broadcast or send out when its condition part is satisfied. Thus, a classifier list consists of one or more classifiers of the form:

C1, C2, ..., Cn/a

Where C1,C2,…Cn {n>=1}are the conditions making up the condition part and 'a' is the action part, conditions are connected by AND operator. Each Ci is a string of fixed length K over a fixed alphabet. In most practical systems, the string is defined over three alphabets: {1,0, #}. The '#' is a don't care (wild card) symbol that can match any of the chosen symbols. A classifier posts one or more messages onto the message list when it is activated. The action part of a classifier is used to form the message it sends out when it is activated. It is also a string of fixed length K defined over the alphabet {1,0}.[1],[4].

1.Message list

The message list acts as the system's short-term memory and as the medium for communication between classifiers, and the output interface. It is made up of external messages (input observations) and internal messages (messages from classifiers). A message is represented by a string of fixed length K (same length as that for a condition) over the same set of alphabets {1, 0} as the action. [4],[5],[6].

1. Input interface (detectors)

This receives the input messages from the environment and transforms them into fixed length strings to be placed on the message list [5].

2. Output interface (effectors)

Messages placed on the message list by classifiers are processed through the output interface in order to communicate with the system's environment. [5]

B. Apportionment of Credit System (Bucket-Brigade Algorithm)

The bucket-brigade algorithm is designed to solve the credit assignment problem for classifier systems and to determine the worth of each classifier. The credit assignment problem is that of deciding which of a set of early active classifiers should receive credit for setting the stage for later successful actions. To this end, a numerical quantity (called strength) is assigned to each classifier. This strength is adjusted continually by the algorithm to reflect the classifier's past usefulness. Each classifier whose condition part is satisfied by one or more messages makes a bid to post a message onto the message list. Only the highest bidders are allowed to become active and hence post messages. A classifier's bid depends on its strength and the specificity of its condition. The specificity measures the relevance of a classifier's condition to a particular message. Formally, a classifier's bid is defined as:

Bid = Cbid*strength*specificity.

where specificity = number of non-#s/Total Length of condition part.

Cbid = a constant less than 1.

The winning classifiers place their messages on the message list and their strengths are reduced by the amount of their bids. [7],[8].

C. Genetic Algorithm (Rule discovery).

There exist three major classes of genetic operators:

- Selection: this operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce.
- Crossover: this operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. For example, the strings 10000100 and 11111111 could be crossed over after the third locus in each to produce the two offspring **10011111** and **11100100**.
- Mutation: this operator randomly flips some of the bits in a chromosome. For example, the string 00000100 might be mutated in its second position to yield 01000100. [2],[9],[10].

2 Architectures Types of Classifier Systems

There are different ways in which behavioral modules can be put together to build a learning system, which is briefly summarized below [11],[12]

1. Monolithic architectures, built by one CS directly connected to the agent's sensors.
2. Distributed architectures, built by many CSs; in this case we distinguish between two subclasses:
   - Flat architectures, built by more than one CS, in which all CSs are at "level 1", i.e. directly connected to the agent's sensors.
   - Hierarchical architectures built by a hierarchy of levels.

2.1. Monolithic Architectures

The simplest choice is, of course, the monolithic architecture, with only one Classifier System in charge of controlling the whole behavior Fig. 1. If the target behavior is made up of several basic responses, there is a further choice to be made: the state of all sensors can be wrapped up in a single message Fig. (1-a), or distributed into a set of independent messages Fig. (1-b). We call the latter case monolithic architecture with distributed input. The idea is that inputs relevant to different responses can go into distinct messages; in such a way, input messages are shorter, and the overall learning effort can be reduced [12].

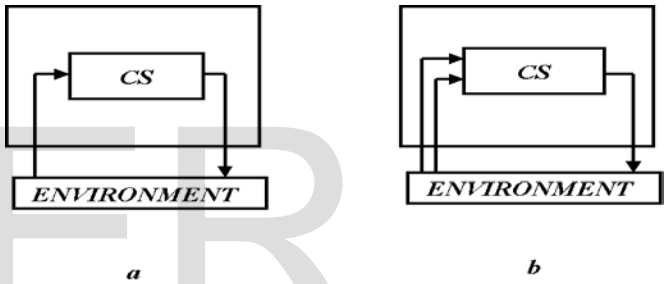

Fig.1 . Monolithic architecture

2.2.Flat Architectures

A distributed architecture is made up of more than one Classifier System. If all Classifier Systems are directly connected to the agent's sensors, then we use the term flat architecture Fig. 2. The idea is that distinct Classifier Systems implement the different basic responses that make up a complex behavior pattern. There is a further issue, here, regarding the way in which the agent's response is built up from the moves proposed by the distinct Classifier Systems. If such moves are independent, they can be realized by different effectors at the same time Fig. (2-a) those moves that are non-independent, however, have to be integrated into a single response before they are realized Fig. (2-b)[12].

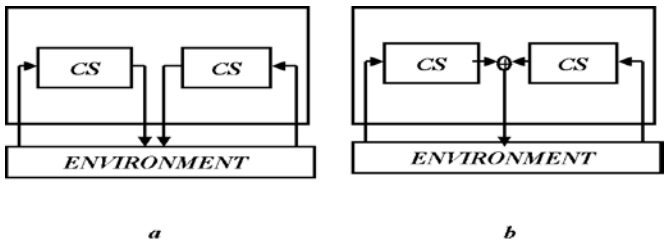

Fig. 2. Flat architecture

2.3. Hierarchical Architectures

In a flat architecture, all Classifier Systems (CS) receive input only from the sensors. In a hierarchical architecture, the set of all Classifier Systems can be partitioned into a number of levels. By definition, a Classifier System belongs to level N if it receives input from systems of level N–1 at most, where level 0 is defined as the level of sensors. An N-level hierarchical architecture is a hierarchy of Classifier Systems having level N as the highest one, Fig.3 shows two different 2-level hierarchical architectures. First level Classifier Systems implement basic behaviors, higher level Classifier Systems implement coordination behaviors [2].

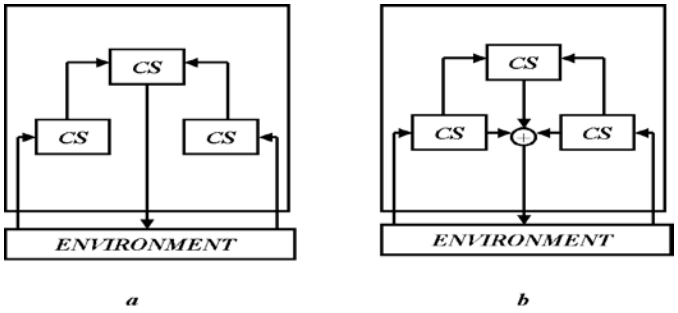

Fig. 3. Two level heretical architecture

3 Seek For Food And Avoid Tree System(FTS).A Case Study

The Seek for food and avoid Tree (FTS) system is built of three learning classifier systems. Organized in two level hierarchical architecture, interacting together to perform complex behavior, consist of three classifier systems (LCS-Switch), (LCS-Seek) and (LCS-Avoid).the FTS structure is shown in Fig.4.

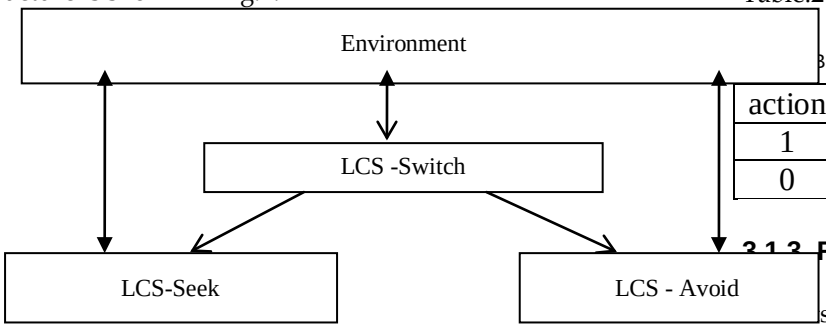

Fig.4. Hierarchal architecture for FTS system

3.1.The Controller (LCS-Switch) Development

The LCS-Switch is use as control system to switch between two classifiers after its analysis the environmental messages, which are receive from the environment. The Controller LCS-Switch is use to learn artificial Robot to choose one of the two classifiers, the basic classifier LCS-Seek, that is, moving towards food when the object has smell, test and color or stop otherwise. The Controller (LCS-Switch) should learn to suppress the LCS-Avoid whenever the Seek behavior proposes an action, which represents complex behavior.

3.1.1. Coding (LCS – Switch) Conditions

LCS-Switch receives 3-bit message from environment mapping it to 8 states from 0 to 7 of two bit only. The two bit represent as fallows:

- First bit represented if object has smell. (1 the object has smell, 0 the object has not smell).
- Second bit represented if object has test. (1 the object has test, 0 the object has not test).
- Third bit represented if object has color. (1 the object has color, 0 the object has not color).

LCS – Switch Conditions has the form and meaning as in Table.1.

TABLE.1. FORM AND MEANING OF LCS – SWITCH CONDITIONS

message	Its meaning
000	The object has no smell or test or color
001	The object has color with out smell or test
010	The object has test with out smell or color
011	The object has test &color with out smell
100	The object has smell with out test or color
101	The object has smell & color with out test
110	The object has smell & test & with out color
111	The object has smell & test &color

3.1.2.Coding (LCS – Switch) Actions

LCS – Switch has one action consisting of only one bit, LCS – Switch actions have the form and meaning as in Table.2

TABLE.2 FORM AND MEANING OF LCS – SWITCH ACTIONS

action	Its meaning
1	LCS – switch toward LCS – Seek for food
0	LCS – switch toward LCS – avoid tree

3.1.3. Representation of (LCS – Switch)

Performance system of the LCS-Switch s of a message list and classifier store. The classifier stores of LCS-Switch 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 (3 bit), and action part consists of (1 bit). The size of classifier store for LCS-Switch will be (8) Rules and all classifiers have the same strength value at the beginning. for Example:

The representation of the rule "If the object has smell &test &color then the robot seeks for food ": Smell test color / seek for food

1 1 1 / 1

3.2.The (LCS-Seek) development

LCS-Seek is use to learn robot seek for food i.e. move single step toward food when it has smell or test or color. The movement capability is completely symmetric a long the two axis .the direction of movement is illustrated in Fig.5

NW 111	N 000	NE 001
W 110		E 010
SW 101	S 100	SE 011

Fig.5.Direction of robot movement

3.2.1.Coding (LCS – Seek) Conditions

The length of message, which LCS – Seek is received, is always, 3 – bit environment message mapping it to eight states from 0 to 7 of three bit only. The meaning of three bit in input message of LCS – Seek determine relative position of food from robot. The form and meaning of three bit LCS – Seek is shown in Table. 3.

TABLE 3. FORM AND MEANING OF LCS – SEEK CONDITIONS

message	Its meaning
0 0 0	Relative position of food from robot is to north
0 0 1	Relative position of food from robot is to north - east
0 1 0	Relative position of food from robot is to east
0 1 1	Relative position of food from robot is to south - east
1 0 0	Relative position of food from robot is to south
1 0 1	Relative position of food from robot is to south - west
1 1 0	Relative position of food from robot is to west
1 1 1	Relative position of food from robot is to north - west

3.2.2. Coding (LCS – Seek) Actions

The desired action should be the same as system input message. Therefore we have eight actions. Action has the form and meaning as in Table.4.

TABLE .4: FORM AND MEANING OF LCS – SEEK ACTIONS

The action	Its meaning
0 0 0	Means robot move to the north
0 0 1	Means robot move to the north - east
0 1 0	Means robot move to the east
0 1 1	Means robot move to the south - east
1 0 0	Means robot move to the south
1 0 1	Means robot move to the south – west
1 1 0	Means robot move to the to west
1 1 1	Means robot move to the north – west

3.2.3. Representation of (LCS – Seek)

LCS – Seek consists of a condition part of(3bit) representing the position of food in the environment and form action part of (3 bit) representing action to be done in the environment the size of classifier store for LCS – Seek will be (8) rules. for Example:

The representation of the rule "if a relative position of a food from simulated robot is to the west then the action to be taken by simulated robot is moving to the west, and so on.

Position of food west of robot / robot move to the west
1 1 0 / 1 1 0

3.3.The (LCS – Avoid) development

LCS – Avoid is used to learn robot to still in its position if the object is tree i.e. has wood or solid or big . It receives its message from the controller (LCS – Switch).

4 (FTS) SYSTEM USING FLAT ARCHITECTURE

The Seek for food and avoid Tree (FTS) system is repeated again using flat architecture by built of two learning classifier systems only (instead of three classifier systems) Organized in one level architecture(instead of two level architecture), (LCS-Seek) and (LCS-Avoid).the FTS structure is shown in fig.6,the two classifiers receive message from the environment and thee is no controller to coordinate between them.

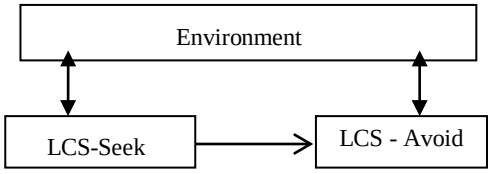

Fig.6.flat architecture for FTS system

5 SEEK FOR FOOD AND AVOID TREE SYSTEM CODE (FTS)

In this work we compare between the flat architecture and the hierarchal architecture ,difference in performance between these two architectures is examined ,each experiment was run for 100 test cycle.

5.1 .Learning Classifier System using hierarchal architecture.

Executing the FTS code, the system responds by presenting the initial report display in Appendix A. for LCS - Switch and LCS-Seek. The classifier system run for 100 iterations, termination with the snapshot report display in Appendix A for LCS - Switch and LCS – Seek respectively.

5.2.Learning Classifier System using flat architecture.

The results using flat architecture are presented in Appendix B for LCS-Seek, in the long run the classifier system is able to sustain this relatively high level of performance even without controller coordinate between two classifiers.

The best performing architecture was the hierarchal architecture, it was able to achieve the same performance of flat architecture using much lest computing time. these results can be explained by the fact that in the switch architecture each single CS, having shorter classifiers ,has a smaller search space, therefore the over all learning task is easier.

6 Conclusion

- Experimentally shows the time need to perform the task when using hierarchal architecture is less than the time is need when using flat architecture.
- Experimentally we obtain more accuracy of the signal when use hierarchal architecture.
- Hierarchal architecture used to implement the task with interesting properties and shorting the number of cycles required to learn.
- The parallel implementation of the algorithm would speed up the training process.
- Design a hierarchal architecture reduce learning complexity.
- Results using hierarchal architecture show improvement over the flat architecture in time need to perform task, in speed, in signal accuracy, in reducing the learning complexity.

7 Refrences

1. Amir Kharmandar, Alireza Naeimi, Alireza Molla Alizadeh, Shaghayegh Jafari, Samira Chavoshi,(2011)," Soccer Simulation 2DTeam Description Proposal for Robocup , ,Payame Noor University, Iran.
2. Brownlee Jason,(2007) "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, Australiajbrownlee@ict.swin.edu.au.
3. Ryan J. Urbanowicz and Jason H.Moore,(2009) "Learning Classifier Systems:A Complete Introduction, Review, and Roadmap"Department of Genetics, Dartmouth College, Hanover, NH 03755, USACorrespondence should be addressed to Jason H. Moore, jason.h.moore@dartmouth.edu.
4. Bull.Larry,(2004), "Learning Classifier Systems: A Brief Introduction", Faculty of Computing, Engineering & Mathematical Sciences University of the West of England, Bristol BS16 1QY, U.K. Larry.
5. Zhou. Qing Qing and Purvis. Martin,(2004) "A Market-Based Rule Learning System" aGuangDong Data Communication Bureau China Telecom 1 Dongyuanheng Rd., Yuexiunan, Guangzhou 510110, China, Department of Information Science, University of Otago, PO Box 56, Dunedin, New Zealand and/or improving the comprehensibility of the rules.
6. Jakobsen. Troels,(2004),"Classifier System.
7. Hartley Adrian R, (1999)" Accuracy-based fitness allows similar performance to humans in static and dynamic classification environments". The University of Birmingham School of Computer Science Edgbaston ,Birmingham, B15 2TT, United Kingdom Email arh@cs.bham.ac.uk Telephone Abstracts "Aarhus school of business ,Denmark. (+44) (0)121 414 3711.
8. Togelius. Julian,(2003) "Evolution of The Layers In a Subsumption Architecture Robot Controller" Dissertation for the Master of Science in Evolutionary and adaptive systems University of Sussex at Brighton.
9. Crook. Stamati.(2003) "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.
10. Eriksson. Anders,(2002) "Evolution of Meta-parameters in Reinforcement Learning" Master's Thesis in Computer Science, at the School of Computer Science and Engineering, Royal Institute of Technology, Stockholm, Sweden.
11. Dorigo. Marco. and Colombetti. Marco (1994). "Training Agents to PerforSequentialBehavior"International Computer Science Institute, to appear in adaptive behavior, MIT press, 2 (3),.
12. Dorigo. Marco and Colombetti. Marco(1993)."Robot Shaping: Developing Autonomous Agents through Learning" international computer science institute.

Appendix. A

[Seek for food and avoid tree (FTS) code in hierarchal architecture for LCS-switch]

```
population parameters
number of classifiers     =      20
number of positions        =       3
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 parameter
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 to select/gen =  0.8000
Number to select         =       8
Mutation probability     =  0.0200
Crossover probability    =  1.0000
Crowding factor          =       3
Crowding subpopulation   =       3

snapshot report
[block: iteration]  -  [0:0]
current Statius
signal                  =     000
desired     output      =       0
classifier output       =       0
environmental message:      000
```

no.	strength	bid	ebid	M	classifier
1	10.00	0.00	0.00		111:[1]
2	10.00	0.00	0.00		11#:[1]
3	10.00	0.00	0.00		#11:[1]
4	10.00	0.00	0.00		1##:[1]
5	10.00	0.00	0.00		#1#:[1]
6	10.00	0.00	0.00		##1:[1]
7	10.00	0.00	0.00		1#1:[1]
8	10.00	0.00	0.00		000:[0]
9	10.00	0.00	0.00		0##:[0]
10	10.00	0.00	0.00		0##:[1]
11	10.00	0.00	0.00		#0#:[1]
12	10.00	0.00	0.00		#0#:[0]
13	10.00	0.00	0.00		0##:[1]
14	10.00	0.00	0.00		0##:[0]
15	10.00	0.00	0.00		00#:[1]
16	10.00	0.00	0.00		00#:[0]
17	10.00	0.00	0.00		#00:[1]
18	10.00	0.00	0.00		#00:[0]
19	10.00	0.00	0.00		0#0:[0]
20	10.00	0.00	0.00		0#0:[1]

```
new winner[1] : old winner[1]

snapshot report
[block: iteration]  -  [0:20]
current Statius
signal                  =     001
desired     output      =       1
classifier output       =       1
environmental message:      001
```

no.	strength	bid	ebid	M	classifier
1	6.68	0.00	0.00		111:[1]
2	6.68	0.00	0.00		11#:[1]
3	6.68	0.00	0.00		#11:[1]
4	6.68	0.00	0.00		1##:[1]
5	6.68	0.00	0.00		#1#:[1]
6	5.44	0.21	0.10	x	##1:[1]
7	6.68	0.00	0.00		1#1:[1]
8	6.68	0.00	0.00		000:[0]
9	5.44	0.21	0.11	x	0##:[0]
10	5.44	0.21	0.16	x	0##:[1]
11	5.44	0.21	0.19	x	#0#:[1]
12	5.44	0.21	0.42	x	#0#:[0]
13	5.44	0.21	0.14	x	0##:[1]
14	5.44	0.21	0.15	x	0##:[0]
15	103.30	5.07	5.02	x	001:[1]
16	5.44	0.28	0.09	x	00#:[0]
17	6.68	0.00	0.00		#00:[1]
18	6.68	0.00	0.00		#00:[0]
19	6.68	0.00	0.00		0#0:[0]
20	6.68	0.00	0.00		0#0:[1]

```
new winner[15] : old winner[15]
```

Seek for food and avoid tree (FTS) code in hierarchal architecture for LCS-seek

```
population parameters
number of classifiers     =      16
number of positions       =       3
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              =    000
desired output     =000
classifier output  =000
environmental message:       000
no.      strength       bid      ebid   M
classifier
-------------------------------------
    1     10.00      0.00      0.00        000:[000]
    2     10.00      0.00      0.00        001:[001]
    3     10.00      0.00      0.00        010:[010]
    4     10.00      0.00      0.00        011:[011]
    5     10.00      0.00      0.00        100:[100]
    6     10.00      0.00      0.00        101:[101]
    7     10.00      0.00      0.00        110:[110]
    8     10.00      0.00      0.00        111:[111]
    9     10.00      0.00      0.00        ###:[000]
   10     10.00      0.00      0.00        ###:[001]
   11     10.00      0.00      0.00        ###:[010]
   12     10.00      0.00      0.00        ###:[011]
   13     10.00      0.00      0.00        ###:[100]
   14     10.00      0.00      0.00        ###:[101]
   15     10.00      0.00      0.00        ###:[110]
   16     10.00     0.00      0.00        ###:[111]
new winner[1] : old winner[1]

snapshot report
block: iteration]  -  [0:20]
current Statius
signal              =    111
desired output     =111
classifier output  =111
environmental message:       111
no.      strength       bid      ebid   M
classifier
-------------------------------------
    1      6.68      0.00      0.00        000:[000]
    2      6.68      0.00      0.00        001:[001]
    3      6.68      0.00      0.00        010:[010]
    4      6.68      0.00      0.00        011:[011]
    5      6.68      0.00      0.00        100:[100]
    6      6.68      0.00      0.00        101:[101]
    7      6.68      0.00      0.00        110:[110]
    8     94.03      5.79      5.70   x    111:[111]
    9      5.44      0.14      0.06   x    ###:[000]
   10      5.44      0.14      0.15   x    ###:[001]
   11      5.44      0.14      0.09   x
###:[010]
   12      5.44      0.14      0.10   x    ###:[011]
   13      5.44      0.14      0.17   x    ###:[100]
   14      5.44      0.14      0.10   x    ###:[101]
   15      5.44      0.14      0.15   x    ###:[110]
   16      5.44      0.14      0.13   x    ###:[111]
new winner[8] : old winner[8]
```

APPENDIX B

Seek for food and avoid tree (FTS) code in flat architecture for LCS-seek

```
snapshot report
---------------
[block: iteration]  -  [0:40]
current Statius
----------------------------
signal              =    101
desired output     =101
classifier output  =101
environmental message:       101
no.      strength       bid      ebid   M
classifier
-------------------------------------
    1      6.68      0.00      0.00        000:[000]
    2      6.68      0.00      0.00        001:[001]
    3      6.68      0.00      0.00        010:[010]
    4      6.68      0.00      0.00        011:[011]
    5      6.68      0.00      0.00        100:[100]
    6     94.03      5.79      5.69   x    101:[101]
    7      6.68      0.00      0.00        110:[110]
    8      6.68      0.00      0.00        111:[111]
    9      5.44      0.14     -0.00   x    ###:[000]
   10      5.44      0.14      0.07   x    ###:[001]
   11      5.44      0.14      0.12   x    ###:[010]
   12      5.44      0.14      0.29   x    ###:[011]
   13      5.44      0.14      0.17   x    ###:[100]
   14      5.44      0.14      0.04   x    ###:[101]
   15      5.44      0.14      0.19   x    ###:[110]
   16      5.44      0.14      0.05   x    ###:[111]
new winner[6] : old winner[6]
```