Georgia State University
Digital Archive @ GSU

Computer Science Theses | Department of Computer Science

12-7-2006

Structure Learning of a Behavior Network for Context Dependent Adaptability

Ou Li

Follow this and additional works at: http://digitalarchive.gsu.edu/cs_theses

Recommended Citation
Li, Ou, "Structure Learning of a Behavior Network for Context Dependent Adaptability" (2006). *Computer Science Theses.* Paper 38.

This Thesis is brought to you for free and open access by the Department of Computer Science at Digital Archive @ GSU. It has been accepted for inclusion in Computer Science Theses by an authorized administrator of Digital Archive @ GSU. For more information, please contact digitalarchive@gsu.edu.

STRUCTURE LEARNING OF A BEHAVIOR NETWORK FOR

CONTEXT DEPENDENT ADAPTABILITY

by

OU LI

Under the Direction of Xiaolin Hu

ABSTRACT

One mechanism for an intelligent agent to adapt to substantial environmental changes is to change its decision making structure. Pervious work in this area has developed a context-dependent behavior selection architecture that uses structure change, i.e., changing the mutual inhibition structures of a behavior network, as the main mechanism to generate different behavior patterns according to different behavioral contexts. Given the important of network structure, this work investigates how the structure of a behavior network can be learned. We developed a structure learning method based on generic algorithm and applied it to a model crayfish that needs to survive in a simulated environment. The model crayfish is controlled by a mutual inhibition behavior network, whose structures are learned using the GA-based algorithm for different environment configurations. The results show that it is possible to learn robust and consistent network structures allowing intelligent agents to behave adaptively in a particular environment.

INDEX WORDS: Mutual Inhibition, Behavior Network, Structure learning, Behavior pattern, Generic algorithm

STRUCTURE LEARNING OF A BEHAVIOR NETWORK FOR CONTEXT DEPENDENT ADAPTABILITY

by

OU LI

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in the College of Arts and Sciences

Georgia State University

2006

Copyright by
OU LI
2006

STRUCTURE LEARNING OF A BEHAVIOR NETWORK FOR CONTEXT DEPENDENT ADAPTABILITY

by

OU LI

Major Professor: Xiaolin Hu
Committee: Raj Sunderraman
Ying Zhu

Electronic Version Approved:
[ ]
Office of Graduate Studies
College of Arts and Sciences
Georgia State University
December 2006

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Xiaolin Hu and other committee members Dr. Raj Sunderraman and Dr. Ying Zhu for their valuable suggestions and spending time reviewing this work.

TABLE OF CONTENT

ACKNOWLEDGEMENTS……..... iv

LIST OF TABLES……... vi

LIST OF FIGURES…….. vii

CHAPTER

1 INTRODUCTION…….. 1

2 RELATED WORK……. 4

3 REVIEW OF THE CONTEXT-DEPENDENT ARCHITECTURE…….. 6

4 SYSTEM DESIGN OF STRUCTURE LEARNING……. 9

The Simulation Environment……... 9

Definition of Behavior Network and Structure…… 11

GA-based Structure Learning…….. 13

5 EXPERIMENTAL RESULTS AND ANALYSIS……. 16

Convergence Analysis……. 16

Consistency Analysis……... 18

Robustness Analysis…….……..……. 22

6 DIVERSITY ANALYSIS…….……. 26

Increasing the Distance between Shelter and Fixed Food Source…….. 26

Generate Food Source at Random Position……. 32

Apply Same Structures in Different Environments……...……..... 38

Apply a Different Fitness Function……..……... 42

7 CONCLUSION AND FUTURE WORK……... 47

REFERENCES…… 49

LIST OF TABLES

Table 1. 3 of 10 Behavior Network Structures obtained by GA…..…… 19

Table 2. Mean and Standard Deviation of 10 structures…….…… 20

Table 3. Mean Fitness, Standard Deviation of Fitness, Maximum, and Minimum for the 3 structures…….…… 23

Table 4. Mean and Standard Deviation of 4 structures…….…….. 24

Table 5.1. Mean and Standard Deviation of 10 structures…….…… 27

Table 5.2. Mean and Standard Deviation of 10 structures…….…… 28

Table 6. Pair relationships in current environment and in original environment…….…… 29

Table 7. Mean Fitness, Standard Deviation of Fitness, Maximum, and Minimum for the 3 structures…….…… 30

Table 8. Mean and Standard Deviation of 7 structures…….…….. 33

Table 9. Mean and Standard Deviation of 9 structures…….…….. 35

Table 10. Numbers of behaviors selection…….……... 35

Table 11. Mean Fitness, Standard Deviation of Fitness, Maximum, and Minimum for the 3 structures…….…… 38

Table 12. Mean fitness and Survival steps of 3 structures in different environments…….……. 41

Table 13.1. Four Structures learned by GA with new fitness function…….……... 42

Table 13.2. Fitness (Steps) of 4 learned Structures in Table 13.1…….…….. 43

Table 14. Mean and Standard Deviation of 4 structures…….…….. 43

LIST OF FIGURES

Figure 1. The context-dependent behavior selection architecture…… 7

Figure 2. Simulation environment…… 10

Figure 3. Interaction and Pseudo-code of GA and Simulation……. 14

Figure 4.1. Population fitness changed over many generations…….. 17

Figure 4.2. Best fitness of each generation……. 17

Figure 5. Summed inhibition of all the seven behaviors…….. 20

Figure 6. Summed inhibition of all the seven behaviors in [20]…….. 21

Figure 7. Fitness of behavior networks for 3 structures……... 23

Figure 8. Mean Fitness and Standard Deviation for 10 structures……... 23

Figure 9. Summed inhibition of all the seven behaviors based on Table 4……. 25

Figure 10. New simulation environment…… 27

Figure 11. Summed inhibition of all seven behaviors in new environment……... 28

Figure 12. Robustness test for the 3 structures…….. 30

Figure 13. Summed inhibition of all seven behaviors……... 33

Figure 14. Summed inhibition of all seven behaviors……... 35

Figure 15. Robustness test for the 4 structures…….. 37

Figure 16. Robustness test for the 3 structures…….. 39

Figure 17. Robustness test for the 3 structures…….. 40

Figure 18. Summed inhibition of all seven behaviors based on Table 14……. 43

Figure 19. Robustness test for the 4 structures…….. 45

CHAPTER 1 INTRODUCTION

An important skill of intelligent systems is being able to adapt to external and internal environmental changes. Most these changes are limited in scope, significance, and occurrence speed, to which intelligent systems can adapt through continuous adjustment and learning. However, there are situations where changes are extensive, significant, and happen in an abrupt speed. These changes result in remarkably new circumstances that demand intelligent systems to significantly change their behavior patterns in order to survive in the new environments. For example, in animal behavior, the formation of social dominance hierarchy of a group of crayfish will dramatically change individual crayfish's behaviors: when two (or more) size-matched crayfish are initially put together, they fight aggressively to compete for the dominant position in the hierarchy. But after the dominance hierarchy is formed, an abrupt change of behavior is observed as the new subordinate retreats and escapes from the attacks and approaches of the dominant [1, 2]. This type of environmental change that dramatically affects a system's behavior is not unusual and calls for effective adaptation mechanisms that are able to quickly and efficiently address the substantial changes. One mechanism that has been employed by intelligent systems to respond to such changes is to modify the network structures of decision making. The change of network structures happens in coordinated and well-designed ways, as the new network structure leads to a new behavior pattern, or solution, that is suitable for the new environment. For example, in animal's social dominance

hierarchy formation, it has been suggested that the new dominant and subordinate quickly change their neural circuit connections thus exercising behavior patterns, e.g., aggressive and defensive, corresponding to their new social positions [1, 3, 4]. Using structure change as an adaptation mechanism has the advantage that it can significantly and quickly change a system's behavior. Meanwhile, from the decision theory point of view, since each network structure represents a durable behavior pattern, a system can reserve a discrete set of well-designed structures as candidate solutions, thus greatly reducing the search space of decision making in time-critical and resource-bounded situations. Given the importance of network structures, our work aims to develop architectures and methods that exploiting the structural property of decision networks for adaptability of intelligent agents.

In an earlier work [5], we have developed a context-dependent behavior selection architecture that uses structure change as the main mechanism to generate different behavior patterns according to different behavioral contexts, each of which represents a particular type of operating environment. In this architecture, the bottom layer is an asymmetry mutual inhibition behavior network where different behaviors inhibit each other for behavior selection. The top "behavioral context" layer sets the behavior selection context by changing the structure (inhibitory coefficients) of the behavior network, thus modulating an agent's overall behavior patterns according to different operating conditions. This architecture provides a simple yet effective mechanism for intelligent agents to dynamically and significantly change their behavior patterns in a dynamic environment. Based on this architecture, we developed a dynamic team formation multi-agent system, where mobile agents can switch from aggressive behavior

patterns before finding their partners to conservative behavior patterns after finding their partners by changing their behavior network structures.

The context-dependent architecture assumes that an intelligent agent has a set of pre-defined network structures corresponding to the different behavioral contexts. In order for the agent to exercise the “right” behavior pattern for a particular behavioral context (operating environment), it is important to provide it with the “right” network structure. Given a particular operating environment, a network structure can be manually defined by the designer. Alternatively and more interestingly, it is desirable to be able to find the “best” network structure using machine learning. In this work, we present the recent progress that we made on structure learning of a behavior network for context-dependent adaptability. Specifically, we designed a method based on Genetic Algorithm (GA) to evolve the behavior network structure of a crayfish model in a dynamic environment with a predator, a food source, and a shelter. The results of this work are encouraging and help us to explain the following important research questions related to structure learning: Is it possible to evolve a “best” behavior network structure for a particular operating environment? Will different runs of GA result in the “same” structure? Will different structures be learned for different operation environments (behavioral contexts)?

CHAPTER 2 RELATED WORK

Supporting adaptability is important for both artificial agents and robots. Researchers have developed techniques to support adaptability from different aspects such as decision architectures, action selection, and perception and motor control. Much research is conducted in the context of social behaviors [6, 7]. The rich diversity of social interactions (either with human or with other agents/robots) presents great challenges to adaptability of individual agents' behaviors. A widely employed technique to handle these challenges is to incorporate cognitive concepts such as emotions, motivations, and affections. In these works, the underlying mechanism that enables adaptive behavior is based on internal state variables. For example, emotions are modeled as state parameters to compute and arbitrate an agent's behaviors according to different situations [8, 9, 10]. We refer to this adaptation approach as state-based adaptation. The state-based approach, as argued by the work of "dual dynamics" [11], is insufficient to model the apparent stability and non-mixability of different behavioral patterns that are needed in extremely dynamic environments. Our work takes a different approach, the structure-based approach, to support adaptability for substantial environment changes.

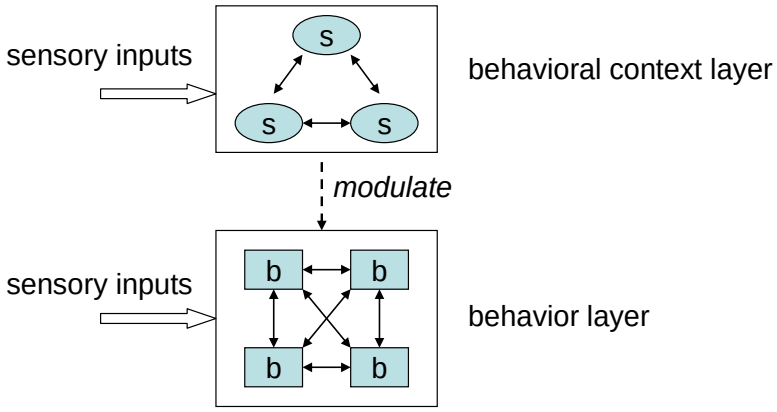
Learning and evolution has long been used to find the correct values for control parameters that lead to a desired behavior [12, 13]. In addition, techniques inspired by biological evolution have also been used in artificial agents and robots. For example, the work of [14] uses genetic programming to evolve behavior strategies for the predator

agents. [15, 16] use the method of co-evolution to evolve behaviors in multi-agent environments. However, none of them has focused on the structural aspect of decision networks. There are some works studied how to evolve the structure of neural network controller. For example, the work of [17] uses a genetic algorithm to evolve neural network controllers for simulated “prey” creatures that learn a herding behavior to help avoid predators. Not much work has been done to evolve network structures at the behavior level. Probably the most related work is [18], which employed GA to evolve the structure of Maes’ spreading activation network [19]. This work evolves the action selection network links among behaviors and between behaviors and motivations. The results show that it is possible to use GA to evolve networks that perform better in a given environment. However, little work was done to analyze the “similarity” of different evolved networks, and no analysis was provided to the problem that if different structures will be evolved for different environments.

CHAPTER 3 REVIEW OF THE CONTEXT-DEPENDENT ARCHITECTURE

The problem of behavior selection, often referred to as action selection in AI, is the problem of run-time choice between multiple parallel, competing, and overlapping behaviors to respond to a dynamic environment. Context-dependent behavior selection used in this paper refers to the capability for an intelligent agent to modulate its behavior selection process based on different explicitly modeled behavioral contexts, i.e., different operating environments. To support this, we developed a two-layer behavior-based architecture [5]. This architecture uses mutual inhibition as the major mechanism to achieve adaptive behavior selection among competitive behaviors. A behavior inhibits other behaviors through *inhibitory coefficients*. The coefficients for the different pairs of inhibiting and inhibited behaviors are different, and these differences help express the relative priorities of the behaviors. For example, if a behavior has relatively high priority, it will inhibit other behaviors with large values of inhibitory coefficients, and will be inhibited by other behaviors with small values of coefficients. The set of all inhibitory coefficients among behaviors defines the structure of the behavior network. In the context-dependent architecture, this behavior network structure can be changed by a top layer called “Behavioral Context”. This allows the relative priorities of the behaviors to be dynamically changed when the system switches to different operating conditions, which in turn makes the system exhibit different behavior patterns. Fig. 1(a) shows this two-layer behavior network architecture. Fig. 1(b) shows that the mutual inhibition

structure of three sample behaviors can be represented by a table of inhibitory coefficients.



(a) The two layer architecture

Table: behaviors' inhibitory coefficients

inhibiting / *inhibited*	b1	b2	b3
b1	x	c_{21}	c_{31}
b2	c_{12}	x	c_{32}
b3	c_{13}	c_{23}	x

(b) The mutual inhibition structure of the behavior network

Fig. 1: The context-dependent behavior selection architecture

The bottom layer is the behavior layer with a network of behaviors. Each behavior mediates a different fixed action pattern and is excited by a particular pattern of sensory inputs. Behaviors inhibit each other and the strengths of inhibitions among pairs of behaviors are specified by inhibitory coefficients. These coefficients are real numbers between 0 and 1, with 0 meaning no inhibition and 1 meaning full inhibition. Mutual inhibition among the behaviors enables the one with the strongest net activity (i.e., its excitation minus the sum of the inhibitions from other behaviors) to govern the agent's behavior. To give an example, assuming *behavior_i* and *behavior_j* are two behaviors

and c_{ij} is the inhibitory coefficient from *behavior_i* to *behavior_j*. Then the amount of inhibition $B_{ij}(t)$ from *behavior_i* to *behavior_j* at time step t is calculated according to Formula *(1)*, where $A_i(t-1)$ is the activation of *behavior_i* at the previous step. Formula *(2)* shows *behavior_j*'s activation $A_j(t)$ is calculated from its excitation $E_j(t)$ subtracting the total inhibitions from other behaviors.

$$B_{ij}(t) = c_{ij} * A_i(t-1) \qquad (1)$$

$$A_j(t) = E_j(t) - \sum_{i}^{i \neq j} B_{ij}(t) \qquad (2)$$

The top layer is the behavioral context layer, which consists of several discrete behavioral contexts. Each behavioral context defines its own set of inhibitory coefficients, i.e., the behavior network structure. Change from one behavioral context to another means the behavior network structure is switched accordingly. A rule-based system may be used to specify when to transit from one behavioral context to another. The transitions between behavioral contexts, by changing the behavior network structure, provide an effective way for intelligent agents to adapt to substantial environmental changes.

CHAPTER 4 SYSTEM DESIGN OF STRUCTURE LEARNING

In order to learn behavior network structures that lead to desired behavior patterns in a given environment, we implemented a genetic algorithm (GA) based evolution algorithm. We applied this algorithm to a model crayfish that uses a mutual inhibition behavior network to govern its behavior choice in a dynamic simulated world containing a predator, a shelter, and a food source. The model crayfish must avoid the predator, find the food source, and eat food to regain the energy depleted by its activity. We seek to determine if a "right" behavior network structure can be evolved to allow the crayfish to respond adaptively and make smooth transitions in this dynamic environment.

4.1. The Simulation Environment

The simulation environment, Fig. 2, is based on a crayfish behavior research that was originally implemented in Edwards' work [20]. The simulated dynamic world contains a food source, a shelter, a predator, and crayfish itself.

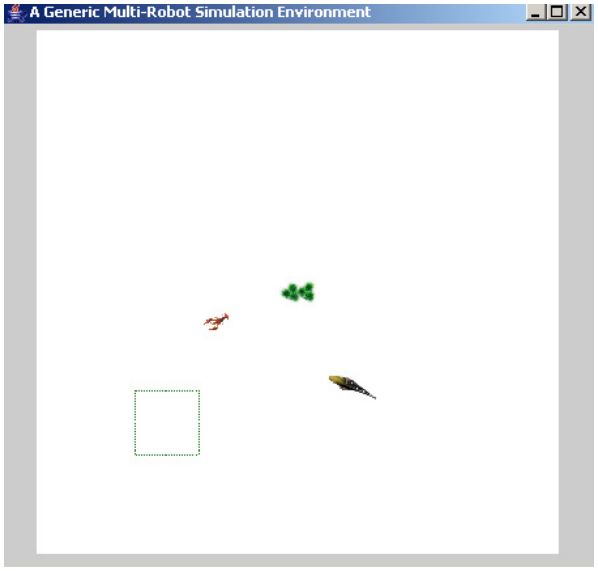


Fig. 2: simulation environment

Both crayfish and predator are able to move around in any directions within a 400×400 2-dimensional world map and will reappear on the opposite side of the map if they happen to move off of the map at one edge. The predator is controlled by a state-based machine with random numbers to make its movement look realistic. In particular, the predator appears with an initial direction and cruising movement. If, while cruising, it comes within a fixed distance of 100 of the crayfish, it will double its speed and start to chase the crayfish. If the crayfish escapes to outside that distance or into its shelter, the predator will give up the chase and resume its cruising speed. The crayfish is considered to have been caught and eaten if the predator comes into contact with it (within a distance of 6). In order to simulate the dynamic world, the initial positions and direction of both crayfish and predator are specified randomly for each simulation round.

The crayfish is governed by one of seven behaviors: ESCAPE, RETREAT, DEFENSE, HIDE, EAT, FORAGE, and SWIM. Each behavior is excited by some sensory inputs and carries out some actions. In general, behavior ESCAPE is to escape from the predator, RETREAT is to move away from the predator and to retreat to the

shelter, DEFENSE is to confront the predator, HIDE is to hide inside the shelter, EAT is to eat food and gain energy, FORAGE is to move towards the food, and SWIM is to swim away in a fast speed. Except the EAT behavior, all other behaviors consume energy (in different rate corresponding to their moving speeds). The crayfish is considered as dead if its energy drops below zero. Detailed descriptions for these behaviors, including how they are excited and what are the associated actions, can be found in [20]. The seven behaviors mutually inhibit each other and compete for behavior selection as described in [20]. A set of inhibitory coefficients are also provided in [20], which provides a reference for us to evaluate the structures evolved in this work.

4.2. Definition of Behavior Network and Structure

To establish the foundation for structure learning, it is important to provide a formal definition for the behavior network and its structure. We define a behavior network *BN* as a directed weighted graph. $BN = (B, L, W)$, where B is the set of behaviors, L is the set of inhibitory links, and W is the set of coefficients. Each inhibitory link is a pair (b, d), where $b, d \in B$, and has an associated coefficient w, where $w \in W$. A coefficient is a real number in [0, 1] and defines the inhibition strength from one behavior to another, with 0 meaning no inhibition and 1 meaning full inhibition. The range of [0, 1] can be extended to [-1, 1] to support not only inhibitory but also excitatory relationships between behaviors.

An important task for the structure-based adaptation approach is to determine what kind of changes will make a network generate different behavior patterns. For example, considering w_{ij} being the inhibitory coefficient from behavior b_i to behavior b_j in a

network, a change of w_{ij} from 0.1 to 0.15 is unlikely to make much difference to an agent's behavior pattern, because both cases represent weak inhibitions from b_i to b_j. However, a change from 0.1 to 0.9 probably will make a noticeable difference. More significantly, if the inhibitory relationship between b_i to b_j is reversed, *e.g.*, b_j starts to inhibit b_i (assuming b_j does not inhibit b_i in the first place), more difference will result. Based on this observation, we explicitly differentiate a network and its structure. A structure is the enduring, orderly, and patterned relationship among elements in a network. It defines a general pattern and can be mapped to multiple specific networks. Given a behavior network $BN = (B, L, W)$, we define its structure $BNS = (B, L, S)$, where S is the finite set of structural relationships and is related to W by a transformation $F: W \rightarrow S$. The kind of transformation that we use in this work is to map inhibitory coefficients to the domain of discrete structural relationships *{weak inhibition, medium inhibition, strong inhibition}* based on their numeric values. For example, coefficients in the range of [0, 0.3] are mapped as *weak inhibition*; coefficients in the range of [0.7, 1] are mapped as *strong inhibition*; the rest are mapped as *medium inhibition*. Oppositely, given a particular structure, a behavior network can be defined based on an inverse transformation $F': S \rightarrow W$. Since this is a one to many mapping, random numbers are needed to generate the specific network parameters.

Structure learning of the asymmetry mutual inhibition behavior network is to find the "right" set inhibitory coefficients that allow an intelligent agent to perform well in a particular dynamic environment. The explicit differentiation between a behavior network and its structure gives us two options to evolve the behavior network of the agent. We can evolve the specific values of the set of coefficients, or we can evolve the structural

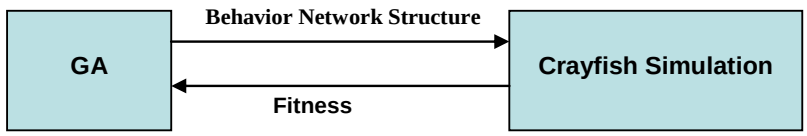
relationships such as *weak inhibition, medium inhibition,* and *strong inhibition.* In this paper, we focus on the second option to evolve the structural relationships between pairs of behaviors. The evolved structure are then mapped to specific randomly-generated behavior networks and be evaluated to see if they lead to the same desired and stable behavior pattern.

4.3. GA-based Structure Learning

A GA is a stochastic general search method which proceeds in an iterative manner by generating new populations of individuals from the old ones. We integrate a standard GA with the crayfish behavior simulation environment to evolve network structures for the model crayfish as described above. The interaction between GA and Crayfish simulation and the pseudo-code of GA and Crayfish simulation are presented in Fig. 3.

Definition of a chromosome As the crayfish's behavior is determined by its network, and the functioning of a network is controlled by its inhibitory coefficients, these coefficients are encoded in a fixed order to construct each individual of population. As mentioned before, the goal of this work is to extract the structure features of behavior network, which indicate several relative priorities of the behaviors, instead of aiming to obtain specific set of inhibitory coefficients. We expect to see these relative priorities can express a behavior pattern more appropriately than the specific set of inhibitory coefficients. So, instead of based on specific numeric values, the initial value of each inhibitory coefficient in the chromosome is randomly selected among L (weak

inhibition), M (medium inhibition), and H (strong inhibition), which represent the range of 0 - 0.2, 0.4 – 0.6, and 0.8 – 1.0 respectively.




```
begin GA
  g:=1  { generation counter }
  Initialize population P(g) with 100 structures
  for each structure in P(g)  { Evaluate P(g) }
     Invoke Crayfish simulation with structure  { compute fitness }
  end for
  while not done do
     g:=g+1
     Select P'(g) from P(g)
     Crossover P'(g)
     Mutate P'(g)
     for each structure in P'(g)    { Evaluate P'(g) }
     Invoke Crayfish simulation with structure  { compute fitness }
     end for
     P := survive P, P'(g) { select the survivors from actual fitness }
  end while
end GA

----------------------

For a given Behavior Network Structure provided by GA
begin Simulation
   for each simulation cycle up to 100 cycles
      Initialize positions of crayfish and predator
      Generate seed for predator's random move selection
      for each simulation step up to 2500 steps
         Update simulation dynamics
         Calculate step fitness
         If(DEAD) Break
      end for
      Calculate simulation fitness
   end for
   Calculate overall fitness for given structure
end Simulation
```

Fig. 3 Interaction and Pseudo-code of GA and Simulation

Design of Fitness function Basically the fitness of a crayfish is determined by how well it is able to survive in the environment in which it is placed. The main idea is that the crayfish should maintain a high level of energy and stay far away from the predator

all the time. Based on this idea, the fitness F is defined as

$$F = \sum_{i=1}^{N} E_i \Big/ N + \sum_{i=1}^{N} \left(1 - e^{\left(-1 \times D_I / 20\right)}\right) \Big/ N$$

Where Ei is crayfish's Energy and Di is the distance between crayfish and predator at each simulation step. N is the number of simulation steps, and F is the Fitness of each simulation. The final Fitness of each individual structure is calculated by averaging the summed Fitness of 100 simulations. This indicates that those structures which can keep maximizing the Fitness function over its active lifetime are considered to be "salient" and can depict the behavior pattern of crayfish appropriately.

Application of GA operators After evaluating of each structure in the population, we employ a standard "Roulette Wheel" selection process to generate the next generation of individuals. Next, a standard single point crossover operator and mutation operator are applied to the new population with probability p_crossover = 0.8 and probability p_mutation = 0.02 respectively.

CHAPTER 5 EXPERIMENTAL RESULTS AND ANALYSIS

This section presents some results obtained from our work of structure learning based on the crayfish behavior simulation. With these results, we conduct preliminary analysis and focus on several important aspects related to the structure learning: convergence, consistency, robustness, and diversity. The aspect of *convergence* tests how many generations it will need for GA to converge to a good structure; the aspect of *consistency* tests if different runs of GA will result in a "similar" structure; the aspect of *robustness* tests if a learned set of structural relationships, i.e., {*weak inhibition, medium inhibition, and strong inhibition*}, are robust, i.e., they are insensitive to the specific values of coefficients; the aspect of *diversity* tests if different behavior network structures will be learned in different environments.

5.1. Convergence Analysis

To conduct this analysis, we initiate a population consisting of 100 behavior network structures. As described before, the initial value of each inhibitory relationship in the chromosome is randomly generated among L (week inhibition), M (medium inhibition), and H (strong inhibition). In all simulations, 0.1, 0.5, and 0.9 are used to represent L, M, and H respectively in order to get a real number value of fitness. For each individual in the population we run 100 simulations (each simulation has 2500 time steps) and calculate the average fitness for that structure. Finally, we calculate an average fitness for

all 100 structures as the mean fitness for that GA generation. Our goal is to test if as the generation increases, the overall fitness will also increase and eventually stabilize, and when it stabilizes.

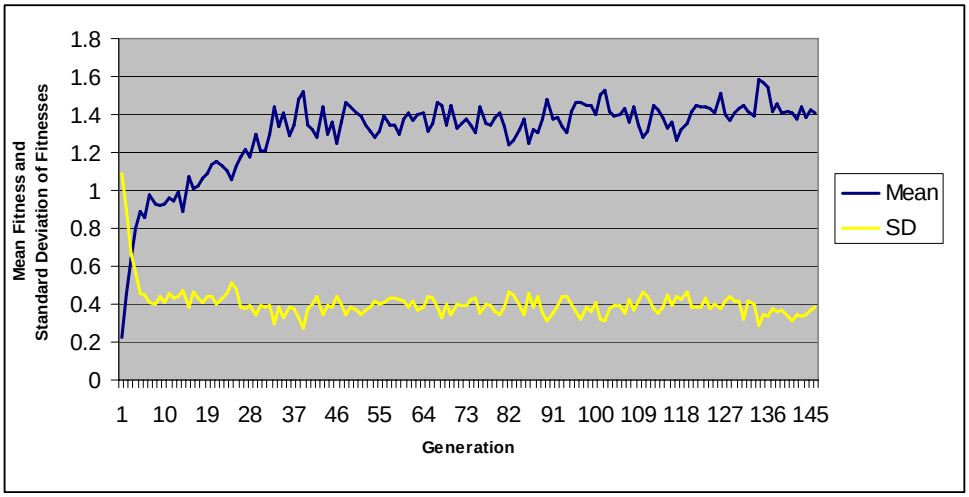


Fig. 4.1: Population fitness changed over many generations (averaged over several runs)

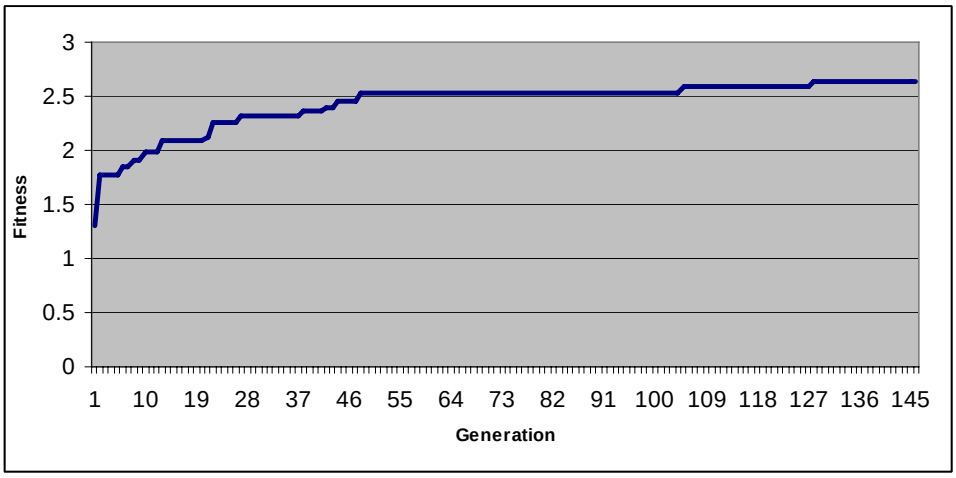


Fig. 4.2: Best fitness of each generation (averaged over several runs)

Fig 4 shows the mean fitness and normalized standard deviation of fitness in each generation in our experiment. As can be seen, the fitness of population improves rapidly in the first 40 or so generations. By about 60 generations, the population began to settle into its final level with standard deviation also leveling off to a minimum. Based on this analysis, in all the following experiments, we set the maximum generation for the GA

process to be 80. This analysis confirms that it is feasible to find good behavior network structures using GA in a reasonable number of generations.

5.2. Consistency Analysis

This analysis aims to answer a very important question: will the network structures learned by different runs of GA consistent (similar) to each other? To conduct this analysis, we carried out multiple runs of GA using the same environment setting (actually they are not exactly the same because random numbers were used to decide the initial positions, directions, and the predator's movement behavior). We expect that a consistent structure pattern (relative priorities of the behaviors), which can express the behavior pattern for that particular environment, can be discovered.

In total 10 structure generated after 10 runs of the learning process, 3 of them are shown in Table 1, in which integer number 1, 2, and 3 represent L (week inhibition), M (medium inhibition), and H (strong inhibition) respectively.

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	……	2	2	2	1	1	3
RETREAT	3	……	3	2	2	2	2
DEFENSE	1	1	……	3	2	1	2
HIDE	1	3	1	……	2	2	3
EAT	2	2	2	3	……	1	1
FORAGE	3	1	3	2	3	……	3
SWIM	1	2	2	2	2	1	……

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	……	2	2	2	1	1	3
RETREAT	3	……	2	3	2	3	1
DEFENSE	2	1	……	3	2	1	3
HIDE	3	3	2	……	2	1	2
EAT	2	1	1	2	……	1	2
FORAGE	3	2	3	1	2	……	3
SWIM	1	1	1	1	1	1	……

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	…….	1	2	2	1	1	2
RETREAT	3	……	2	2	3	2	3
DEFENSE	1	3	……	2	2	1	3
HIDE	2	2	3	……	2	2	2
EAT	2	3	2	1	……	1	3
FORAGE	3	1	2	1	3	……	3
SWIM	1	1	1	2	3	1	……

Table 1. 3 of 10 Behavior Network Structures obtained by GA

Based on these three structures, some interesting conclusions can be drawn. For example in Fig. 5, it is clear that SWIM has the highest relative priority. This is because compared to other behaviors, SWIM generally strongly inhibits other behaviors (as indicated by the column of SWIM in the tables), and is weakly inhibited by other behaviors (as indicated by the row of SWIM in the tables). Similarly, HIDE has the second highest relative priority since crayfish will spend the most time in Shelter due to the short distance between Food Source and Shelter. Also it is clear that FORAGE has the lowest relative priority as it is strongly inhibited by other behaviors and weakly inhibits other behaviors. For other behaviors, it can be seen that ESCAPE and EAT also have high relative priorities, although not as strong as SWIM. Furthermore, between pairs of behaviors, ESCAPE always weakly inhibit SWIM; EAT and FORAGE always weakly inhibit ESCAPE. What is interesting is that all these salient structural relationships are expected and makes sense from the biological point of view. For example, SWIM and ESCAPE have high priority because so the crayfish can move away from the predator to save itself. In fact, the structures learned in our work is comparable with the set of coefficients manually tuned by the author of [20]. Another interesting feature we observed is that the salient structural relationships described above are commonly shared

by all the 10 structures. To better show this, Table 2 shows the Mean and Standard Deviation of the 10 structures.

MEAN	Inhibiting behavior						
Inhibited behavior	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	…..	1.4	2.0	2.2	1.0	1.0	2.1
RETREAT	2.7	…..	2.0	2.2	2.4	2.6	2.2
DEFENSE	1.9	2.0	…..	2.0	2.1	1.5	2.3
HIDE	1.9	2.2	2.1	…..	1.8	1.9	2.3
EAT	2.0	2.4	1.5	2.3	…..	1.2	2.4
FORAGE	2.9	1.7	1.8	2.1	2.6	…..	3.0
SWIM	1.0	1.5	1.7	2.0	1.7	1.0	…..

SD	Inhibiting behavior						
Inhibited behavior	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	…..	0.49	0.63	0.60	0.00	0.00	0.70
RETREAT	0.64	…..	0.45	0.75	0.49	0.49	0.87
DEFENSE	0.70	0.77	…..	0.89	0.70	0.67	0.64
HIDE	0.83	0.75	0.83	…..	0.75	0.54	0.78
EAT	0.45	0.66	0.67	0.64	…..	0.40	0.66
FORAGE	0.30	0.64	0.75	0.83	0.49	…..	0.00
SWIM	0.00	0.67	0.78	0.63	0.64	0.00	…..

Table 2. Mean and Standard Deviation of 10 structures

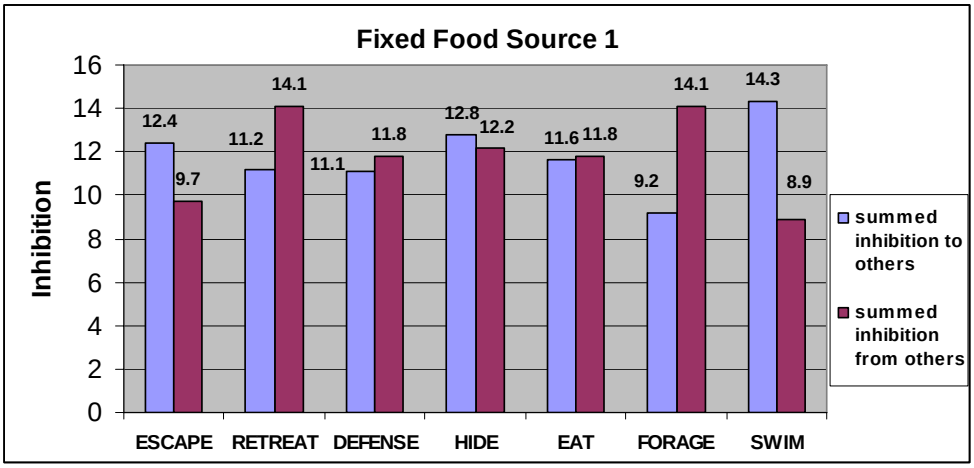


Fig. 5: Summed inhibition of all the seven behaviors

Based on Table 2, Figure 5 shows the summed inhibitions for all the seven behaviors to better support our analysis. Again, it can be seen that SWIM has the highest priority and FORAGE has the lowest priority. Figure 6 shows the “best” structure manually tuned by the author in [20]. Comparing Figure 5 and Figure 6, we can see the

structure features in current environment mostly conform to the original design in [20]. The differences between them might be caused by their different implementation. For example, ESCAPE has the highest priority in Figure 6 while in Figure 5 SWIM has the highest priority. This is because ESCAPE will cost double amount of energy than SWIM. Based on the Fitness Function in GA-based structure learning, a structure will be considered as "best" only if it can make crayfish maintain a high level of energy and stay far away from the predator all the time. Since in current environment the Food Source is close to Shelter, SWIM will be mostly selected to move the crayfish back to Shelter. So it makes sense that SWIM should have the highest priority.

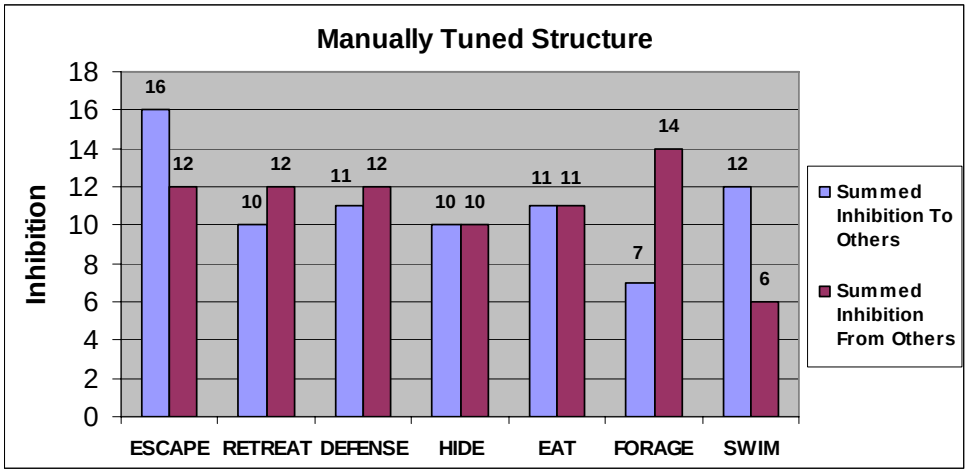
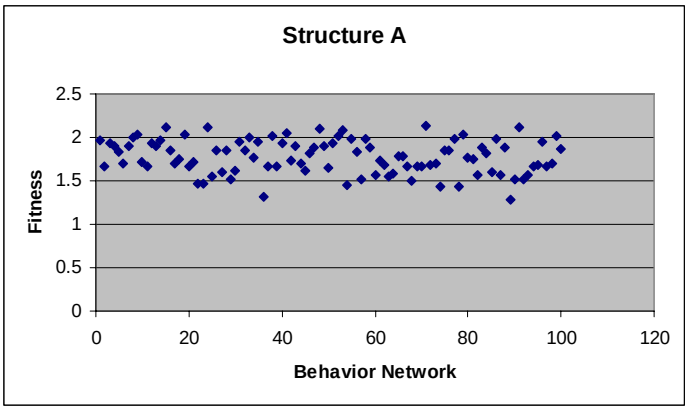


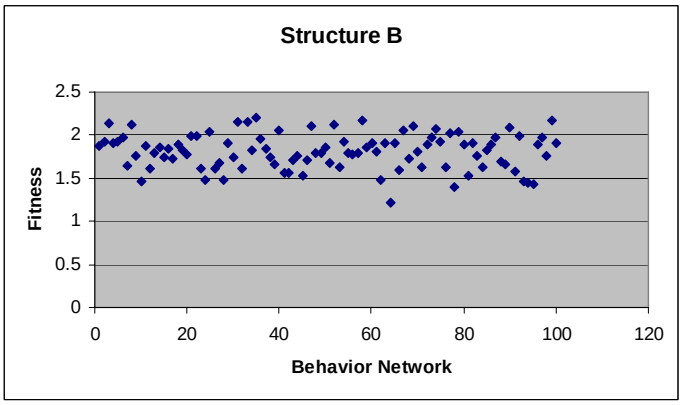
Fig. 6: Summed inhibition of all the seven behaviors in [20]

All these results show that all the learned network structures in current environment have a high degree of consistency. The reason behind this is likely because a "best" structure is highly dependent on the environment. Thus given a particular environment, the learned structures will always share a strong similarity.

5.3. Robustness Analysis

The robustness analysis aims to prove our hypothesis that it is the structure (i.e., the salient structural relationships between pairs of behaviors), not the specific values of coefficients, that defines the behavior pattern of a behavior network. To conduct robustness analysis, each learned structure is used as the prototype to randomly generate 100 behavior networks (with inhibitory coefficients) of that kind. Specifically, for each structural relationship that is depicted with L (week inhibition), M (medium inhibition), and H (strong inhibition), its coefficient is generated randomly from the range of 0 - 0.2, 0.4 – 0.6, and 0.8 – 1.0 respectively. Then each behavior network is evaluated in the same simulation environment used in the learning process. The fitness is calculated after simulation as the measure indicating how well a specific behavior network performs.





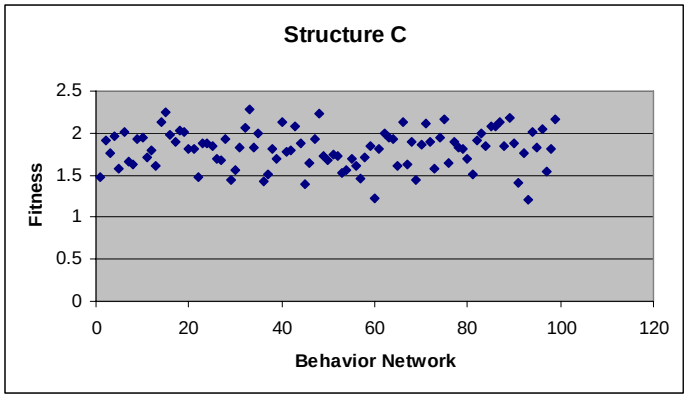


Fig.7. Fitness of behavior networks for 3 structures (averaged over 100 simulations)

Structure	Mean Fitness	SD	MAX	MIN
A	1.776974885	0.194122328	2.131124097	1.276399159
B	1.811156524	0.203804919	2.19823398	1.212239444
C	1.810032587	0.227045309	2.274207994	1.200681876

Table 3. Mean Fitness, Standard Deviation of Fitness, Maximum, and Minimum for the 3 structures

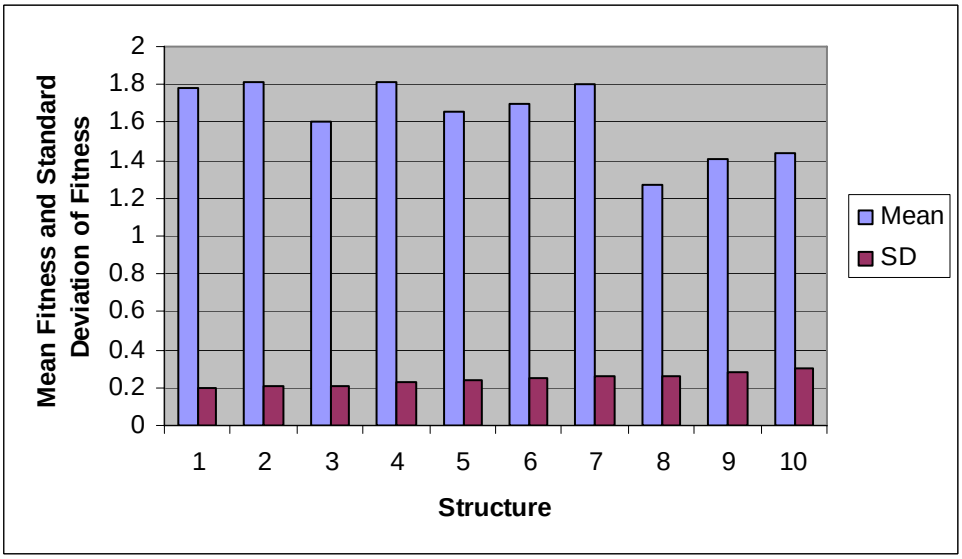


Fig. 8: Mean Fitness and Standard Deviation for 10 structures

Fig. 7 plots the Fitness of the randomly generated behavior networks for the three structures given in Table 1. Table 3 shows the Mean Fitness, Standard Deviation of Fitness, Max Fitness, and Min Fitness of the behavior networks for these three structures. We can see from Figure 7 that all the fitness for a particular structure appear within a

range and there are no outliers as can be seen from Table 3. The stable distribution of fitness indicates that the structures learned by GA are not sensitive to specific value of coefficients and thus can be used to define the crayfish's behavior pattern. To better support this analysis, Figure 8 shows the Mean Fitness and Standard Deviation of Fitness of behavior networks for all the 10 structures that have been learned. The results show that most of the structures have higher Mean Fitness and Lower SD, and thus indicate the learned structures are robust.

However, not all the structures have relatively high fitness and low SD in Figure 8. These structures with relatively low fitness and high SD, such as structure 8, 9, and 10, cannot represent the behavior pattern of crayfish as well as those with relatively high fitness and low SD since the fitness can show how well crayfish is capable of surviving in a given environment. We expect those with relatively high fitness and low SD can bring stronger consistency on describing the behavior pattern.

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	……	1.5	1.8	1.8	1	1	2.3
RETREAT	2.8	……	2.3	2.5	2.3	2.5	2.3
DEFENSE	1.5	1.5	……	2.5	2.3	1.3	2.3
HIDE	1.8	2.3	2.3	……	1.8	2	2.5
EAT	2	2.3	1.5	2	……	1	2.3
FORAGE	3	1.5	2.3	1.3	2.5	……	3
SWIM	1	1.5	1.3	2	2	1	……

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	……	0.5	0.43	0.43	0	0	0.83
RETREAT	0.43	……	0.43	0.5	0.43	0.5	0.83
DEFENSE	0.5	0.87	……	0.5	0.43	0.43	0.83
HIDE	0.83	0.83	0.83	……	0.43	0.71	0.5
EAT	0	0.83	0.5	0.71	……	0	0.83
FORAGE	0	0.5	0.83	0.43	0.5	……	0
SWIM	0	0.5	0.43	0.71	0.71	0	……

Table 4. Mean and Standard Deviation of 4 structures

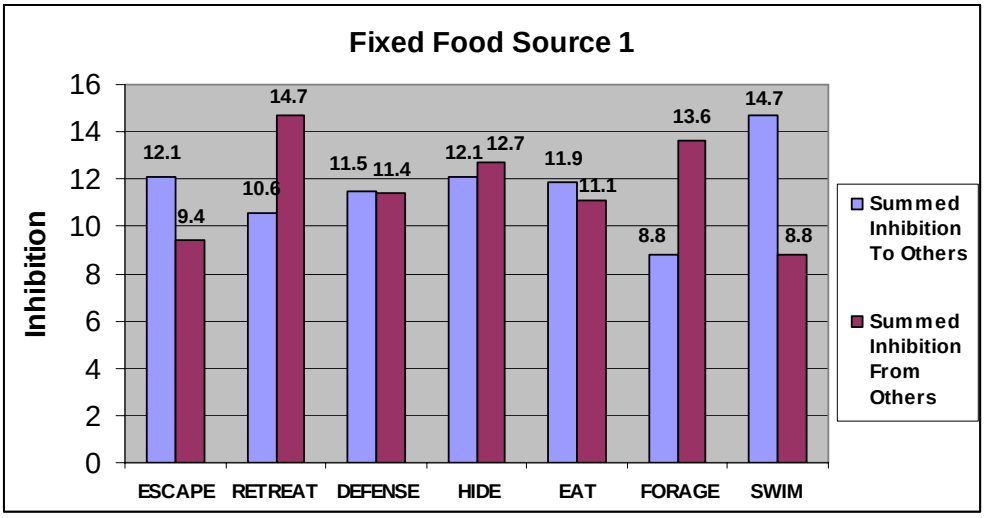


Fig. 9: Summed inhibition of all the seven behaviors based on Table 4

Table 4 shows the Mean and Standard Deviation of 4 structures, which are structure 1, 2, 4, and 7 in Figure 8. Figure 9 shows the Summed inhibition of all the seven behaviors based on Table 4. Compared with Table 2, It is clear that Table 4 shows a higher degree of similarity on structure features. Especially two more pairs of behavior have the same inhibition relationship, which follows the conclusions of original design in [20]. ESCAPE always strongly inhibits FORAGE while FORAGE always weakly inhibits ESCAPE; ESCAPE always has middle level inhibition to EAT while EAT always weakly inhibits ESCAPE. Also Figure 9 shows that it mostly conforms to Figure 5 and Figure 6 on relative priority of behaviors and makes sense from the biological point of view. These results indicate that robustness test can be used to filter the "non-appropriate" structures and improve the consistency of structures learned by GA, and as a result the behavior pattern of crayfish can be better described. However, it is not clear enough that to what extent would a structure learned by GA be considered as "appropriate", which indicates the method used here to select the structures need to be further elaborated by more detailed analyses.

CHPTER 6 DIVERSITY ANALYSIS

Previous analysis have shown that consistent and robust structures of Mutual Inhibition Behavior Network can be learned by GA in a given environment context. So an intuitive question followed is that whether the structures learned in different environment can still show consistency and robustness. To answer this question is the first purpose of this analysis. Changes will be made to the original simulation environment to define two different environments, which are “Increasing the distance between Shelter and fixed Food Source” and “Generate Food Source at random position”. In each situation consistency analysis and robustness analysis will be performed to see whether the learned structures will share a strong similarity and will be sensitive to specific value of coefficients. Also structures learned from different context will be compared to find the structure features related to the changes in different environment since the changes on context will affect how the Crayfish behaves.

6.1. Increasing the distance between Shelter and fixed Food Source

The Food Source in original simulation environment (Fig.2) is fixed at the center of the map and very close to the Shelter. It is very easy for Crayfish to reach the Food Source for energy supply and to escape from Predator by entering the Shelter. So the distance between Shelter and Food Source is extended and the Food Source is fixed at the upright of the map (Fig.8). We expect to see the increase on relative priority of FORAGE since

Crayfish has to have a higher tendency of moving towards to the Food Source than in the previous environment in order to keep its energy level and not to become starved.

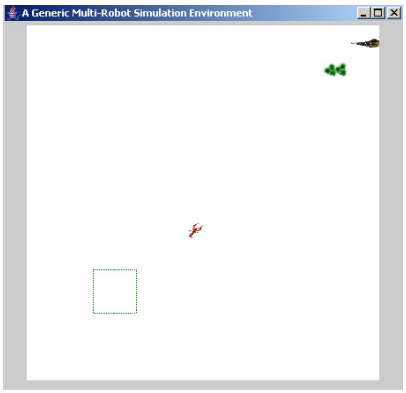


Fig. 10: new simulation environment

Structure A	Inhibiting behavior						
Inhibited behavior	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	0	1	2	2	1	1	2
RETREAT	1	0	2	2	1	3	2
DEFENSE	1	1	0	1	3	1	3
HIDE	3	3	1	0	1	3	1
EAT	2	3	2	2	0	1	1
FORAGE	3	1	1	2	3	0	3
SWIM	1	1	1	2	2	1	0
Structure B	**Inhibiting behavior**						
Inhibited behavior	**ESCAPE**	**RETREAT**	**DEFENSE**	**HIDE**	**EAT**	**FORAGE**	**SWIM**
ESCAPE	0	2	2	3	1	1	1
RETREAT	2	0	3	3	2	3	2
DEFENSE	2	1	0	1	3	2	1
HIDE	1	1	3	0	1	2	2
EAT	2	3	3	1	0	1	2
FORAGE	3	1	1	1	3	0	2
SWIM	2	1	1	1	1	1	0
Structure C	**Inhibiting behavior**						
Inhibited behavior	**ESCAPE**	**RETREAT**	**DEFENSE**	**HIDE**	**EAT**	**FORAGE**	**SWIM**
ESCAPE	0	2	2	2	1	1	3
RETREAT	2	0	1	3	1	3	2
DEFENSE	3	3	0	1	1	3	1
HIDE	3	1	2	0	2	2	1
EAT	1	3	3	3	0	1	1
FORAGE	3	1	1	1	3	0	3
SWIM	1	2	2	2	1	1	0

Table 5.1. Mean and Standard Deviation of 10 structures

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	…..	1.6	1.8	2.6	1.0	1.0	2.2
RETREAT	1.8	…..	1.8	2.6	1.6	3.0	2.4
DEFENSE	1.8	1.8	…..	1.0	2.4	2.4	2.2
HIDE	2.2	1.8	2.0	…..	1.6	2.6	1.4
EAT	1.8	2.8	2.6	2.0	…..	1.0	2.0
FORAGE	3.0	1.0	1.4	1.2	3.0	…..	2.8
SWIM	1.2	1.2	1.2	1.8	1.6	1.2	…..
Inhibited behavior	**Inhibiting behavior**						
	ESCAPE	**RETREAT**	**DEFENSE**	**HIDE**	**EAT**	**FORAGE**	**SWIM**
ESCAPE	…..	0.49	0.40	0.49	0.00	0.00	0.75
RETREAT	0.75	…..	0.75	0.49	0.49	0.00	0.49
DEFENSE	0.75	0.75	…..	0.00	0.80	0.80	0.98
HIDE	0.75	0.75	0.89	…..	0.49	0.49	0.49
EAT	0.40	0.40	0.49	0.63	…..	0.00	0.89
FORAGE	0.00	0.00	0.49	0.40	0.00	…..	0.40
SWIM	0.40	0.40	0.40	0.40	0.49	0.40	…..

Table 5.2 . Mean and Standard Deviation of 10 structures

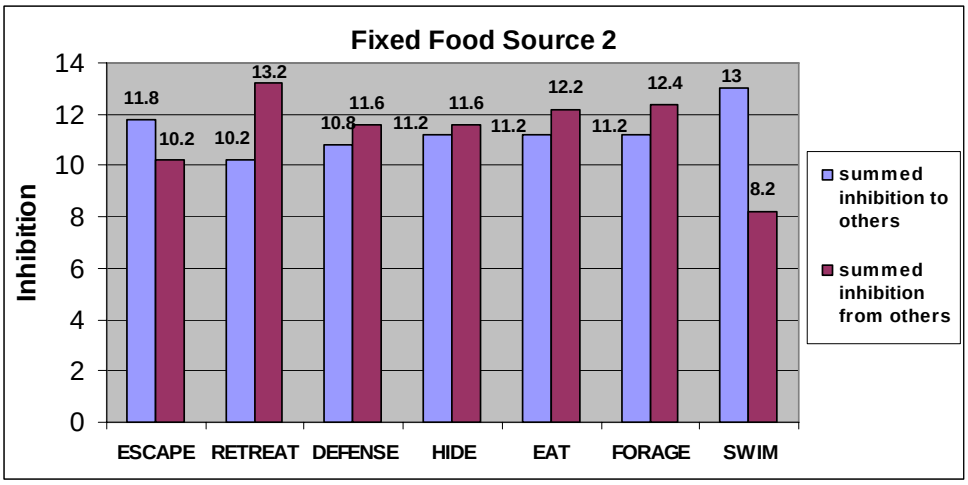


Fig. 11: Summed inhibition of all seven behaviors in new environment

Table 5 shows the Mean and Standard Deviation of the 8 structures and Fig. 11 shows the summed inhibitions for behavior ESCAPE, RETREAT, DEFENSE, HIDE, EAT, FORAGE, and SWIM based on the results in Table 5. The pair relationships of structures learned by GA in two different environments are compared in Table 5.

Behavior Pair			Fixed Food Source 1		Fixed Food Source 2	
	A	B	A→B	A←B	A→B	A←B
1	ESCAPE	RETREAT	2.7	1.4	1.8	1.6
2	ESCAPE	DEFENSE	1.9	2.0	1.8	1.8
3	ESCAPE	HIDE	1.9	2.2	2.2	2.6
4	ESCAPE	EAT	2.0	1.0	1.8	1.0
5	ESCAPE	FORAGE	2.9	1.0	3.0	1.0
6	ESCAPE	SWIM	1.0	2.1	1.2	2.2
7	RETREAT	DEFENSE	2.0	2.0	1.8	1.8
8	RETREAT	HIDE	2.2	2.2	1.8	2.6
9	RETREAT	EAT	2.4	2.4	2.8	1.6
10	RETREAT	FORAGE	1.7	2.6	1.0	3.0
11	RETREAT	SWIM	1.5	2.2	1.2	2.4
12	DEFENSE	HIDE	2.1	2.0	2.0	1.0
13	DEFENSE	EAT	1.5	2.1	2.6	2.4
14	DEFENSE	FORAGE	1.8	1.5	1.4	2.4
15	DEFENSE	SWIM	1.7	2.3	1.2	2.2
16	HIDE	EAT	2.3	1.8	2.0	1.6
17	HIDE	FORAGE	2.1	1.9	1.2	2.6
18	HIDE	SWIM	2.0	2.3	1.8	1.4
19	EAT	FORAGE	2.6	1.2	3.0	1.0
20	EAT	SWIM	1.7	2.4	1.6	2.0
21	FORAGE	SWIM	1.0	3.0	1.2	2.8

Table 6: Pair relationships in current environment and in original environment

.

We can see from Fig. 11 that SWIM has highest relative priority and weakly inhibited by other behaviors; ESCAPE has the second highest relative priority and second lowest inhibition from other behaviors. This is because they are performed most often in order to get away from Predator. Also we can see, as expected, FORAGE has a higher relative priority. In Table 6, EAT always weakly inhibits ESCAPE; ESCAPE always has strong inhibition to FORAGE and only weakly inhibited by FORAGE; FORAG always strongly inhibits RETREAT and RETREAT always inhibits FORAGE with low inhibition level; HIDE always weakly inhibits DEFENSE; EAT always strongly inhibits FORAGE and is inhibited by FORAGE in a low inhibition level. Since the results in Table 5 have shown that all the 8 structures learned by GA in current simulation environment share the same structure relationships, we can consider them consistent.

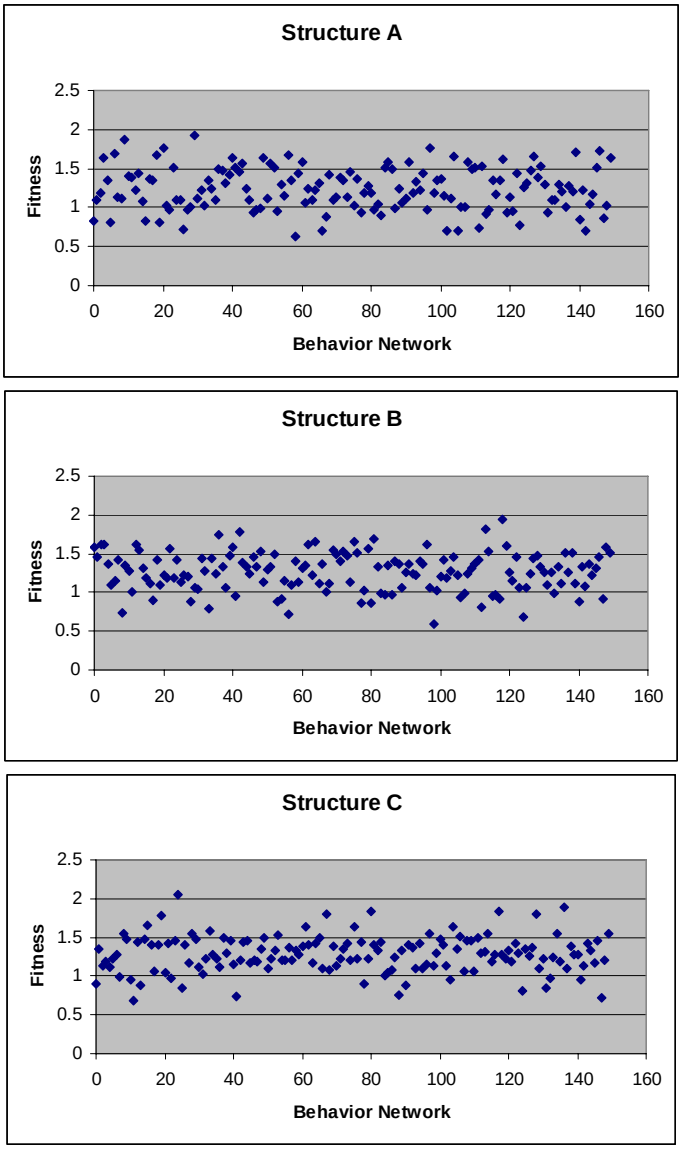


Fig.12. Robustness test for the 3 structures (averaged over 100 simulations)

Structure	Mean Fitness	SD	MAX	MIN
A	1.236526875	0.274809342	1.919545041	0.629055047
B	1.266434783	0.248548845	1.94017511	0.591297946
C	1.283501023	0.239092536	2.042778542	0.689778919

Table 7. Mean Fitness, Standard Deviation of Fitness, Maximum, and Minimum for the 3 structures

Known the structures learned by GA in current environment are consistent, we should perform Robustness test to see if these structure are sensitive to the specific value of coefficients. The results of Robustness test for 3 structures learned in current

environment are obtained and presented in Fig. 12 and Table 7. Following the analysis method described in Chapter 5, these results indicate that the learned structures are robust and can be used to describe the behavior pattern of Crayfish in current environment.

So far the first goal of this test has been achieved. Next we will try to figure out the structure features in the salient structures. Based on the results in Table 6, we can find six pair behaviors with “Significant” inhibition relationship, which are Pair9(RETREAT and EAT), Pair10(RETREAT and FORAGE), Pair12(DEFENSE and HIDE), Pair13(DEFENSE and EAT), Pair14(DEFENSE and FORAGE), and Pair17(HIDE and FORAGE).

1) For Pair9(RETREAT and EAT) and Pair13(DEFENSE and EAT).

RETREAT and EAT in original environment (Env1) inhibits to each other with equal level while in Current environment (Env2) RETREAT strongly inhibits EAT. This is because the Food Source in Env1 is very close to Shelter. Crayfish do not have to start to RETREAT back to the Shelter very earlier while it is eating although the predator is approaching. But in Env2 Food Source is far from Shelter. Crayfish might be caught if it did not plan for RETREAT earlier while it is eating; Pair13 shares the same reason as Pair9.

2) For Pair10(RETREAT and FORAGE) and Pair14(DEFENSE and FORAGE).

In Env1 RETREAT inhibits FORAGE in a middle level and FORAGE inhibits RETREAT in a strong level. But In Env2, FORAGE has strong inhibition to RETREAT and RETREAT only weakly inhibits FORAGE. This change indicates that the crayfish has to take risk to move towards to the Food Source due to the low energy

level even though it is dangerous and most likely to get caught by predator; Pair14 shares the same reason as Pair10.

3) For Pair17(HIDE and FORAGE) and Pair12(DEFENSE and HIDE).

HIDE and FORAGE in Env1 inhibits to each other with almost equal level while in Env2 FORAGE strongly inhibits HIDE and only is weakly inhibited by HIDE. This is because HIDE is not excited outside the shelter. Since Food Source is far from shelter, Crayfish will move to Food Source more often and spend more time outside the Shelter than in Env1; Pair12 shares the same reason as Pair17.

Since above analyses can explain both the changes on Pair relationship and the changes on relative priority of behavior in Fig. 11, these six “Significant” pair inhibition relationships can be considered as the structure features related to the environment changes which affect the behavior pattern of Crayfish in current environment.

6.2. Generate Food Source at random position

The Food Source in previous two simulation environments was placed at a fixed position and the amount of Food Source never changed. In order to make the simulation environment more realistic and expect to see various changes on structure, the Food Source will be generated at random position in the map with certain amount when it is consumed up by Crayfish. Except changing the simulation environment, we will also change the configuration of behavior network by increasing the excitation and decreasing the energy cost of behavior FORAGE. We expect FORAGE to have a higher relative priority since Crayfish has to spend the most of time to search for Food Source in order to keep its energy level and expect to see the consistent and robust structures can still be

learned. Also we will compare the structures learned in both environments (Fixed Food Source with new behavior network configuration and randomly generated Food Source with new behavior network configuration) in order to find the changes on behavior pattern.

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	……	1.0	1.6	1.6	1.0	1.0	1.9
RETREAT	2.3	……	1.7	2.3	1.9	1.4	1.4
DEFENSE	2.3	2.3	……	2.1	2.1	2.1	1.9
HIDE	1.9	1.6	2.0	……	2.1	2.3	2.0
EAT	2.0	1.7	1.9	2.0	……	1.0	2.0
FORAGE	3.0	2.0	2.7	1.7	3.0	……	2.1
SWIM	2.0	1.6	1.6	2.1	1.6	1.6	……

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	……	0.00	0.49	0.73	0.00	0.00	0.64
RETREAT	0.45	……	0.70	0.70	0.83	0.73	0.73
DEFENSE	0.70	0.70	……	0.83	0.83	0.83	0.83
HIDE	0.83	0.49	0.93	……	0.99	0.45	0.93
EAT	0.53	0.70	0.83	0.93	……	0.00	0.76
FORAGE	0.00	0.53	0.70	0.45	0.00	……	0.83
SWIM	0.93	0.73	0.49	0.83	0.49	0.73	……

Table 8. Mean and Standard Deviation of 7 structures
(Fixed Food Source at center with new behavior network configuration)

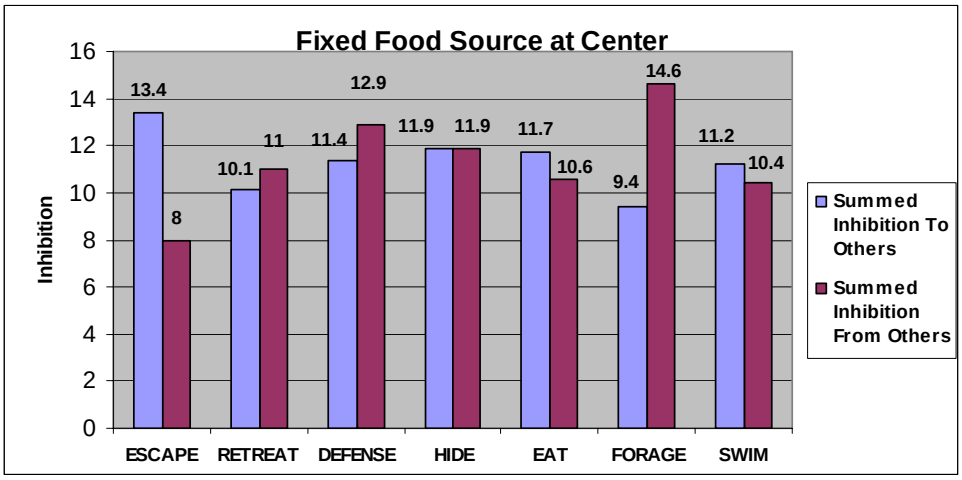


Fig.13. Summed inhibition of all seven behaviors
(Fixed Food Source at center with new behavior network configuration)

Table 8 shows that Mean and Standard Deviation of 7 structures learned in the environment of “Fixed Food Source at center with new behavior network configuration”. Figure 13 presents the summed inhibition of seven behaviors based on Table 8. Comparing Table 8 with Table 2, the results show that they have the same inhibition relationship on some behavior pairs. For example, ESCAPE always has middle level inhibition to EAT while EAT always weakly inhibits ESCAPE; ESCAPE always strongly inhibits FORAGE while FORAGE always weakly inhibits ESCAPE. There are also some differences between these two patterns which caused by the changes of behavior network configuration. For example, in Table 8 FORAGE has higher inhibition to SWIM while SWIM has lower inhibition to FORAGE; in Table 8 ESCAPE always has middle inhibition level to SWIM while in Table 2 ESCAPE always has low inhibition level to SWIM. This is because the excitation of FORAGE is increased and the energy cost of FORAGE is decreased in new environment so that Crayfish can keep a higher level of energy. With enough energy, Crayfish do not need to worry about the high energy cost of behavior ESCAPE. So ESCAPE will be mostly selected instead of SWIM since ESCAPE can escape from Predator and return to Shelter in a very fast speed. To better support this, Figure 13 shows the Summed inhibition of all seven behaviors. We can see from Figure 13 that the relative priority of SWIM (summed inhibition to other behaviors) significantly decreased and the relative priority of ESCAPE, FORAGE, and EAT do get improved. These results do demonstrate that the changes between this two behavior patterns are caused by the changes of behavior network configuration (increase on excitation and decrease on energy cost of FORAGE).

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	…..	1.4	1.8	1.8	1.0	1.0	1.9
RETREAT	2.1	…..	2.0	2.0	2.6	3.0	2.4
DEFENSE	2.1	1.7	…..	1.8	2.1	1.9	2.0
HIDE	2.4	1.3	2.0	…..	2.2	2.3	1.7
EAT	2.6	2.2	2.0	2.2	…..	1.1	1.9
FORAGE	2.7	1.0	1.7	1.2	2.8	…..	2.7
SWIM	1.0	1.0	1.9	1.8	1.9	1.0	…..

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	…..	0.50	0.79	0.79	0.00	0.00	0.87
RETREAT	0.57	…..	0.94	0.82	0.50	0.00	0.50
DEFENSE	0.87	0.82	…..	0.42	0.87	0.87	0.67
HIDE	0.83	0.67	0.82	…..	0.79	0.67	0.82
EAT	0.50	0.63	0.67	0.92	…..	0.31	0.87
FORAGE	0.47	0.00	0.67	0.42	0.42	…..	0.47
SWIM	0.00	0.00	0.87	0.79	0.87	0.00	…..

Table 9. Mean and Standard Deviation of 9 structures
(Randomly generated Food Source with new behavior network configuration)

	ESCAPE	RETREAT	HIDE	FORAGE	SWIM
Rondom Food Source	12	377	221	1214	46
Fixed Food Source	38	491	403	1361	0

Table 10: Numbers of behaviors selection (average over 100 simulations)

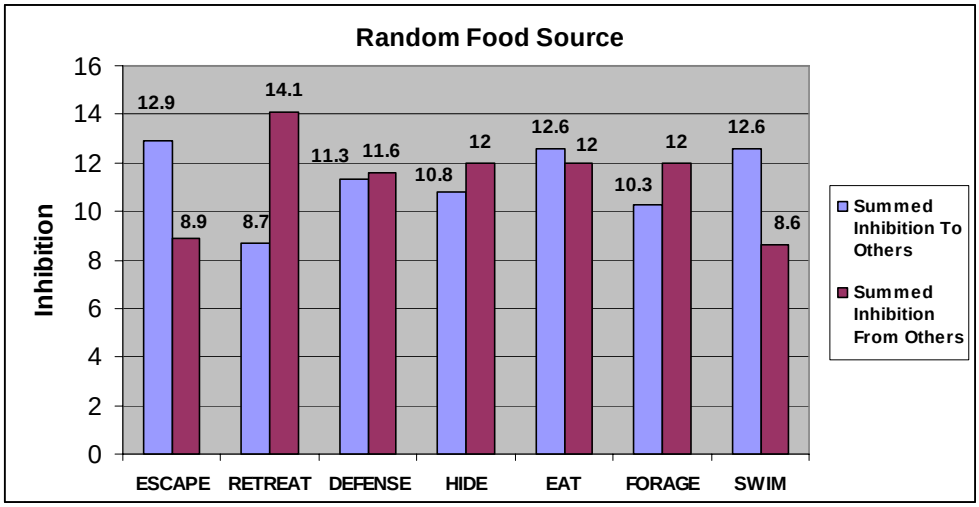


Fig.14. Summed inhibition of all seven behaviors
(Randomly generated Food Source with new behavior network configuration)

Table 9 shows the Mean and Standard Deviation of 9 structures learned in the environment of "Randomly generated Food Source with new behavior network

configuration". Figure 14 presents the summed inhibition of seven behaviors based on Table 9. Based on the results, we can see that the relative priority (summed inhibition to other behaviors) of ESCAPE and RETREAT get lower while the relative priority of FORAGE and EAT do get improved as we expected. This is because Crayfish has to spend most of time to keep searching for Food Source in order to keep its energy level even though it might be caught by Predator. So FORAGE must have stronger inhibition to and weaker inhibition from other behaviors in order to be selected to perform; It also shows that SWIM has higher relative priority. This is because SWIM will cost less energy than ESCAPE since Crayfish cannot always keep a high level of energy as in the previous environment; Another observation is that the relative priority of HIDE is decreased. This is because HIDE will be excited only in Shelter. Since Crayfish has to spend most of time outside the Shelter, HIDE will less likely be selected. To better support this, Table 10 shows the averaged numbers of behaviors selection for ESCAPE, RETREAT, HIDE, FORAGE, and SWIM. We can find the same results in Table 11 as those described above.

The above analyses show that the changes of environment (randomly generated Food Source) do cause the changes on the behavior pattern of Crayfish as we expected. Also since the structure features described above are commonly shared by all the structures as shown in Table 9, they can be considered to be consistent and can describe the behavior pattern of Crayfish in current environment.

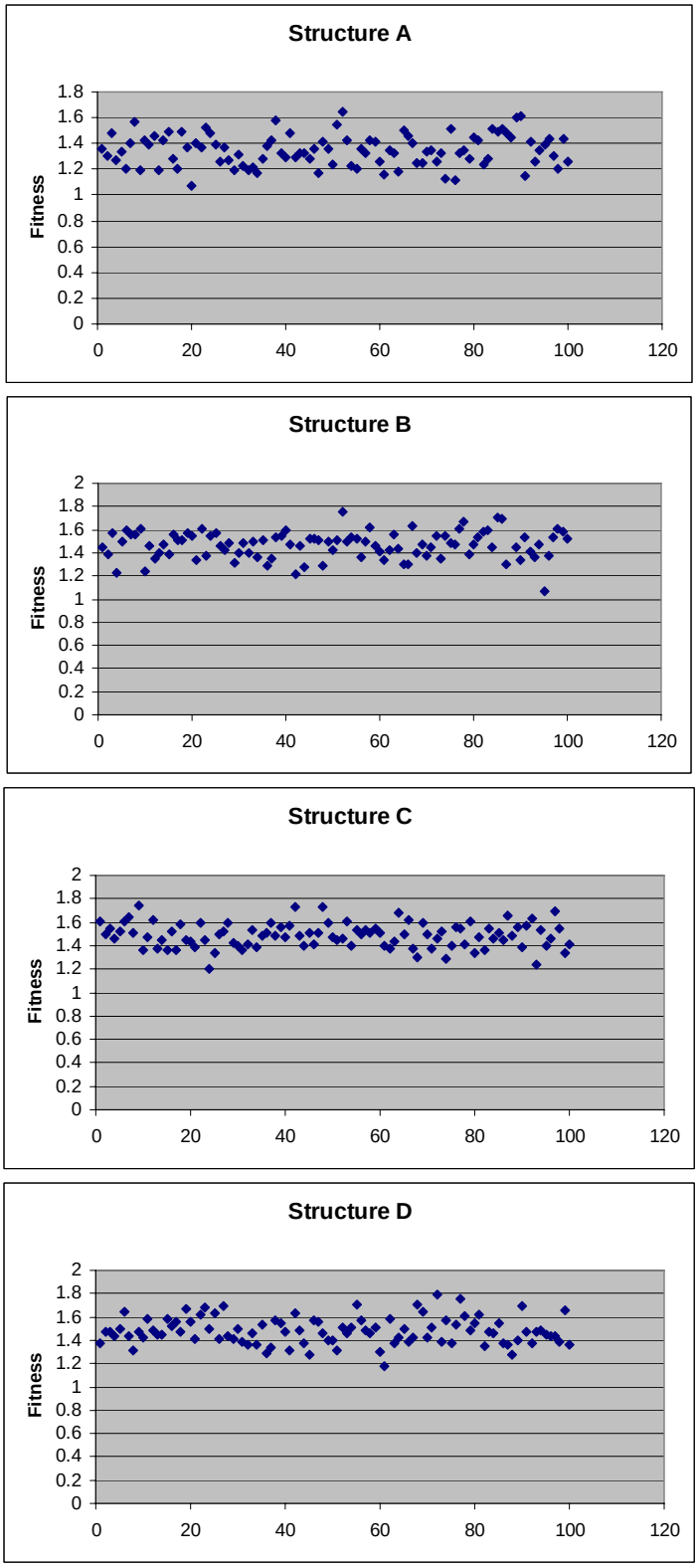


Fig.15. Robustness test for the 4 structures (averaged over 100 simulations)
(Randomly generated Food Source with new behavior network configuration)

Structure	Mean Fitness	SD	MAX	MIN
A	1.34790	0.12033	1.64230	1.06621
B	1.46671	0.11545	1.76014	1.07232
C	1.48765	0.10471	1.74480	1.19777
D	1.47982	0.11449	1.78552	1.17752

Table 11. Mean Fitness, Standard Deviation of Fitness, Maximum, and Minimum for the 3 structures (Randomly generated Food Source with new behavior network configuration)

The results of Robustness test for 4 structures learned in current environment are obtained and presented in Fig. 15 and Table 11. We can see that all these structures display a stable distribution of fitness. Following the analysis method described in Chapter 5, these results demonstrate that the learned structures are robust and can be used to describe the behavior pattern of Crayfish in current environment.

6.3. Apply same structures in different environments

In this section we will apply robustness test to the structures learned from the first environment of "*Fixed Food Source at center with original behavior network configuration*" in previous two environments, which are "*Fixed Food Source at corner with original behavior network configuration*" and "*Randomly generated Food Source with changed behavior network configuration*". The 3 structures learned in the first environment as shown in Table 1 will be used in this test. The results will be compared with Figure 7 and Table 3. Since these structures have shown the consistent and robust behavior pattern in the first environment, we expect that they can still deliver stable performance in different environment. More important, we expect them to show different performance in terms of survival ability in different environment since the learned structures highly depend on the given environment in which they are learned.

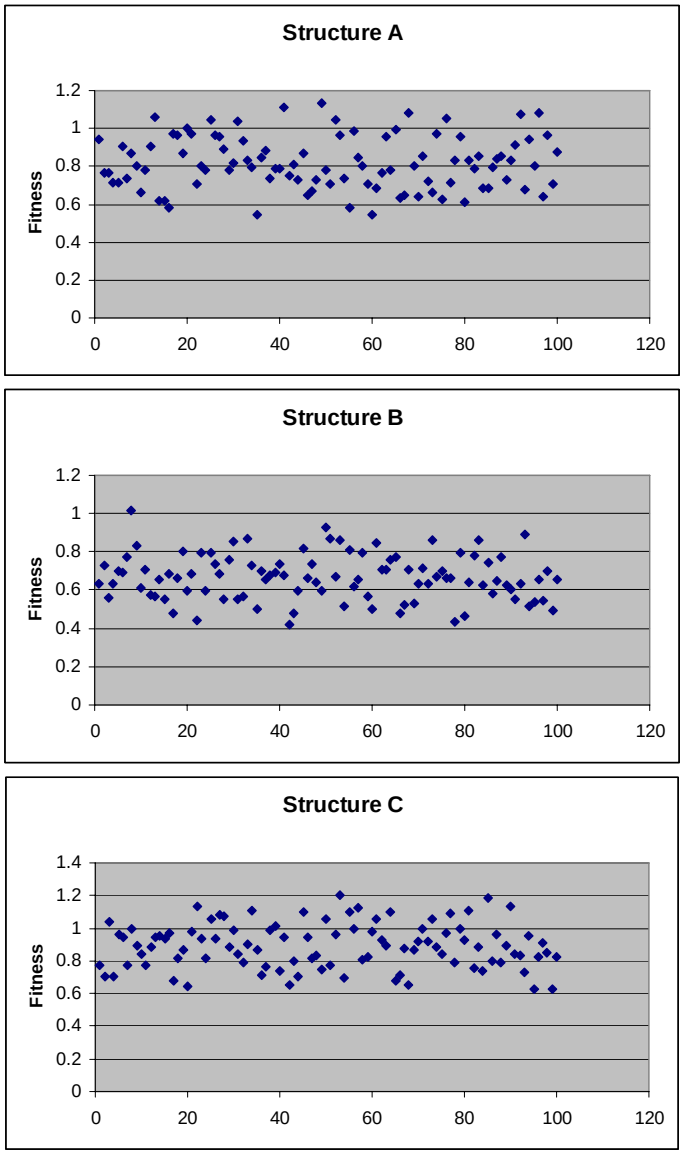


Fig.16. Robustness test for the 3 structures (averaged over 100 simulations)
(Fixed Food Source at corner with original behavior network configuration)

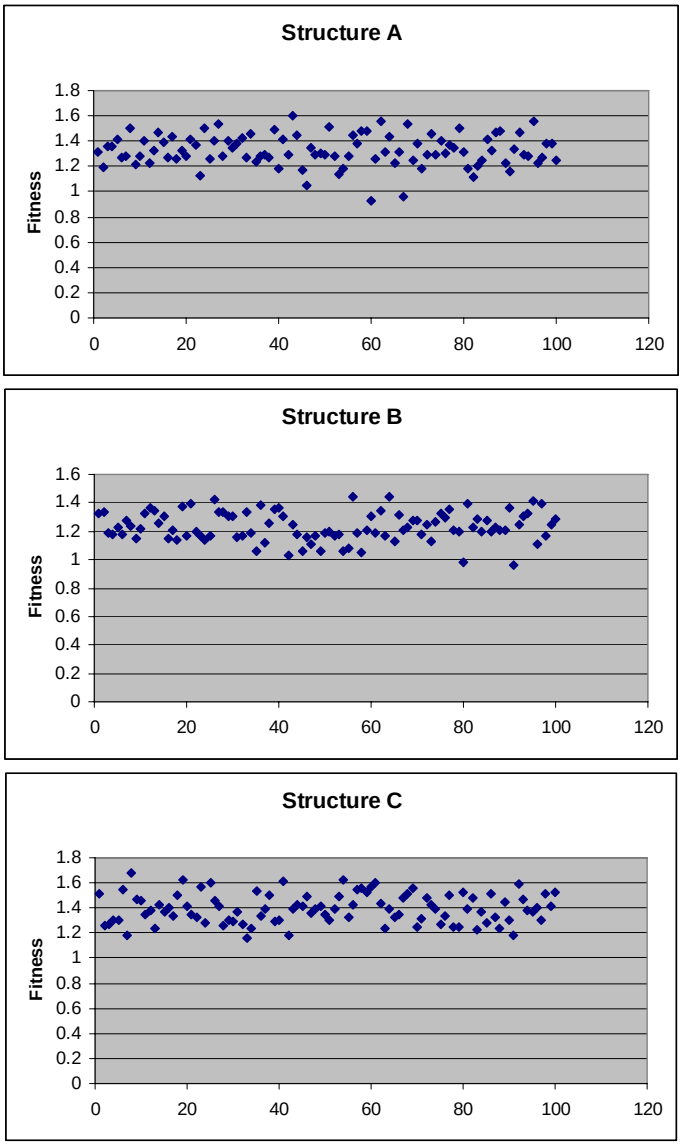


Fig.17. Robustness test for the 3 structures(averaged over 100 simulations)
(Randomly generated Food Source with changed behavior network configuration)

Structure	Environment	Mean Fitness	Survival Steps
A	Fixed Food Source at center	1.777	1886
	Fixed Food Source at corner	0.819	926
	Randomly generated Food Source	1.328	1717
B	Fixed Food Source at center	1.811	1910
	Fixed Food Source at corner	0.667	777
	Randomly generated Food Source	1.233	1626
C	Fixed Food Source at center	1.810	1937
	Fixed Food Source at corner	0.892	993
	Randomly generated Food Source	1.396	1778

Table 12. Mean fitness and Survival steps of 3 structures in different environments (averaged over 100 simulations)

The results in Figure 16, Figure 17, and Table 12 show that all these 3 structures can deliver stable performance in both testing environments in terms of Fitness distribution, Mean fitness, and Survival steps. But the results obtained in environment of "*Randomly generated Food Source with changed behavior network configuration*" are better than those in environment of "*Fixed Food Source at corner with original behavior network configuration*". This is because the behavior FORAGE in the former one has a higher excitation and a lower energy cost, so that Crayfish in this environment can keep a high level of energy which help it maintain the more stable performance in different environments. More important, Table 12 shows that these 3 structures have the highest Mean fitness and Survival steps in the environment in which they are learned by GA. These results clarify that the "best" structure learned by GA is highly dependent on the environment. Thus given a particular environment, the learned structures will mostly share a strong similarity and can describe the behavior pattern in the current context.

6.4. Apply a different fitness function

In order to test how the fitness function of the Genetic algorithm affects the learned structures, we will apply a different fitness function to the first simulation environment (*Fixed Food Source at center with original behavior network configuration*). The current fitness function is based on energy level of Crayfish and distance to Predator. We plan to change to a different fitness function that only depends on the "Survival steps" in the simulation, to see if a similar (or different) structure will be learned.

Structure 1	**Inhibiting behavior**						
Inhibited behavior	**ESCAPE**	**RETREAT**	**DEFENSE**	**HIDE**	**EAT**	**FORAGE**	**SWIM**
ESCAPE	……	2	2	1	1	1	2
RETREAT	2	……	2	3	3	2	2
DEFENSE	1	2	……	3	1	3	3
HIDE	1	1	3	……	1	2	3
EAT	2	2	3	2	……	1	2
FORAGE	3	3	2	1	2	……	3
SWIM	1	1	1	3	1	1	……
Structure 2	**Inhibiting behavior**						
Inhibited behavior	**ESCAPE**	**RETREAT**	**DEFENSE**	**HIDE**	**EAT**	**FORAGE**	**SWIM**
ESCAPE	……	1	1	3	1	1	3
RETREAT	1	……	3	2	2	3	1
DEFENSE	1	1	……	2	2	3	3
HIDE	3	2	3	……	3	2	2
EAT	2	1	2	2	……	1	2
FORAGE	3	3	3	1	2	……	2
SWIM	1	1	2	2	2	1	……
Structure 3	**Inhibiting behavior**						
Inhibited behavior	**ESCAPE**	**RETREAT**	**DEFENSE**	**HIDE**	**EAT**	**FORAGE**	**SWIM**
ESCAPE	……	1	2	3	1	1	3
RETREAT	3	……	3	3	3	3	2
DEFENSE	1	2	……	3	2	3	1
HIDE	2	2	2	……	3	3	1
EAT	2	3	3	3	……	1	2
FORAGE	3	2	1	2	3	……	3
SWIM	1	1	1	1	1	1	……
Structure 4	**Inhibiting behavior**						
Inhibited behavior	**ESCAPE**	**RETREAT**	**DEFENSE**	**HIDE**	**EAT**	**FORAGE**	**SWIM**
ESCAPE	……	2	1	3	1	1	2
RETREAT	2	……	3	3	2	2	1
DEFENSE	3	2	……	1	3	3	2
HIDE	2	1	3	……	2	2	2
EAT	3	1	3	1	……	1	1
FORAGE	3	3	3	2	2	……	3
SWIM	1	1	1	3	3	1	……

Table 13.1 4 Behavior Network Structures learned by GA with new fitness function

Structure	Survival steps
1	5044
2	4772
3	4441
4	4782

Table 13.2 Fitness (Steps) of 4 learned Structures in Table 13.1

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	……	1.5	1.5	2.5	1	1	2.5
RETREAT	2	……	2.8	2.8	2.5	2.5	1.5
DEFENSE	1.5	1.8	……	2.3	2	3	2.3
HIDE	2	1.5	2.8	……	2.3	2.3	2
EAT	2.3	1.8	2.8	2	……	1	1.8
FORAGE	3	2.8	2.3	1.5	2.3	……	2.8
SWIM	1	1	1.3	2.3	1.8	1	……

Inhibited behavior	Inhibiting behavior						
	ESCAPE	RETREAT	DEFENSE	HIDE	EAT	FORAGE	SWIM
ESCAPE	……	0.5	0.5	0.87	0	0	0.5
RETREAT	0.71	……	0.43	0.43	0.5	0.5	0.5
DEFENSE	0.87	0.43	……	0.83	0.71	0	0.83
HIDE	0.71	0.5	0.43	……	0.83	0.43	0.71
EAT	0.43	0.83	0.43	0.71	……	0	0.43
FORAGE	0	0.43	0.83	0.5	0.43	……	0.43
SWIM	0	0	0.43	0.83	0.83	0	……

Table 14. Mean and Standard Deviation of 4 structures

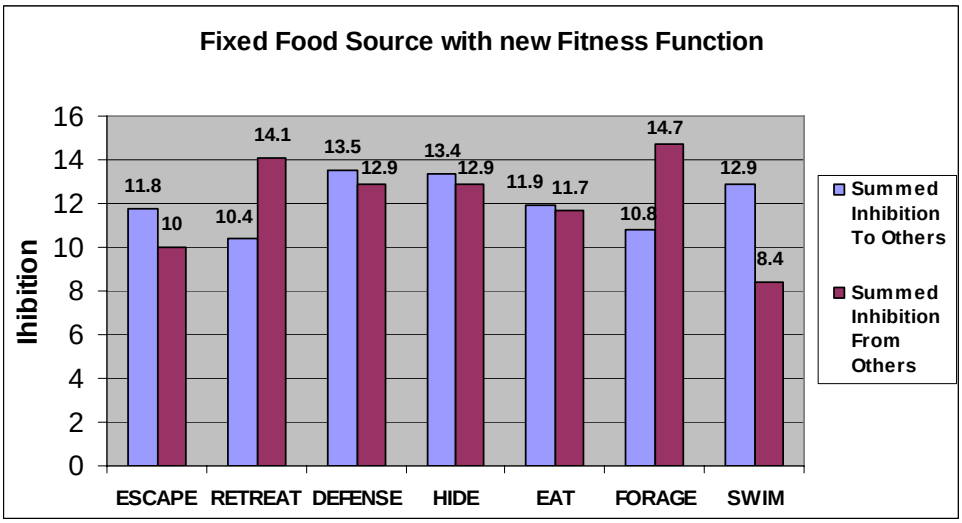
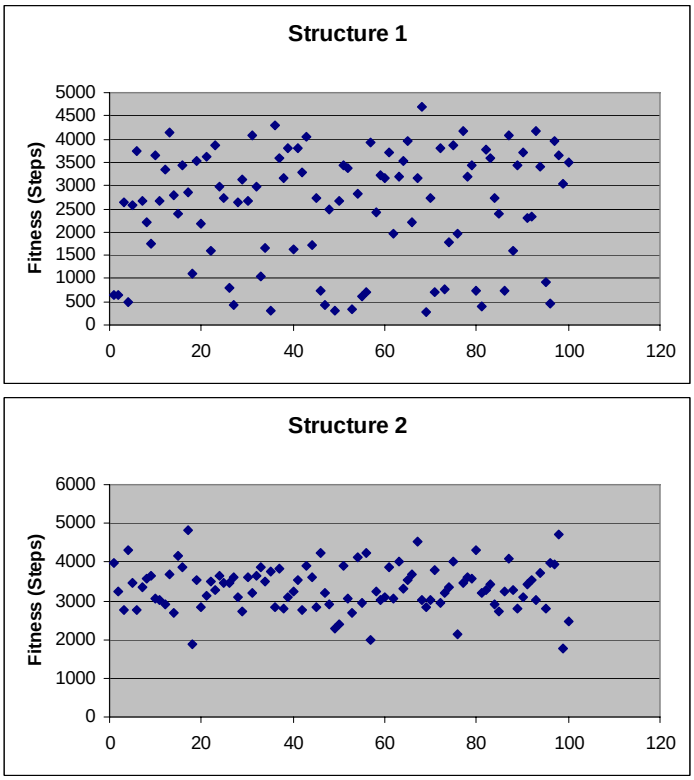


Fig.18. Summed inhibition of all seven behaviors based on Table 14

The structure learning with new fitness function follows the same process as described in Figure 3 except that the simulation step is increased up to 6000 (Steps) in order to prevent the fitness (Steps) from exceeding the original range of 2500 (Steps). Table 13.1 shows the 4 structures learned by GA with new fitness function (Survival steps). Table 13.2 presents the fitness for each of them and shows no one is higher than 6000 (Steps).

Table 14 shows the Mean and Standard Deviation of 4 structures. Figure 18 presents the summed inhibition of all seven behaviors based on Table 14. The results of robustness test for the 4 structures learned in current environment are obtained and presented in Figure 19.



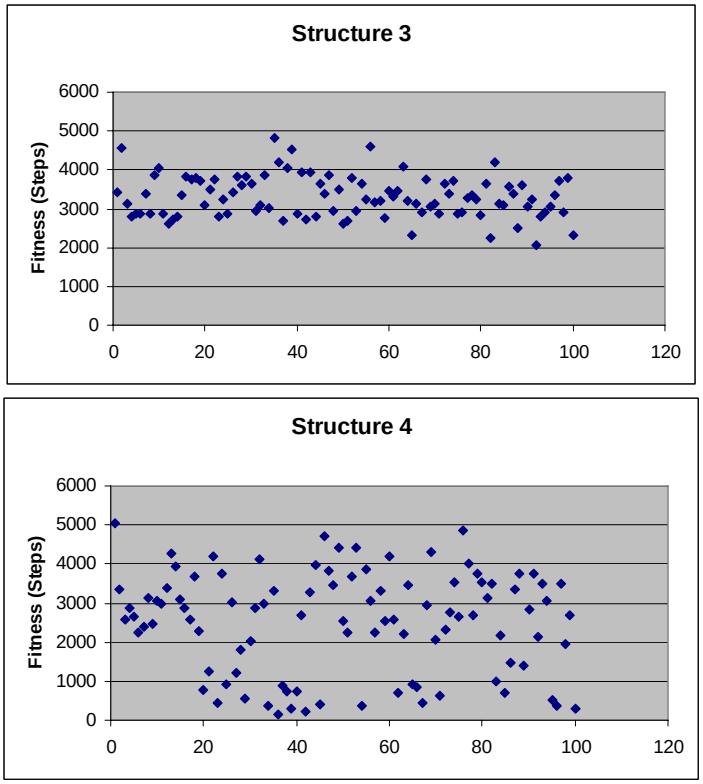


Fig.19. Robustness test for the 4 structures (averaged over 100 simulations)

Compared with Table 2, we can see from Table 14 that they share many inhibition relationships. For example, FORAGE always weakly inhibits EAT and ESCAPE always weakly inhibit SWIM. More important, they have the same or similar pair inhibition relationships. For example, ESCAPE always has middle level inhibition to EAT while EAT always weakly inhibits ESCAPE; ESCAPE always strongly inhibits FORAGE while FORAGE always weakly inhibits ESCAPE; SWIM always strongly inhibits FORAGE while FORAGE always weakly inhibits SWIM.

Compared with Figure 5, we can also see from Figure 18 that SWIM has a higher relative priority and is always weakly inhibited by other behaviors (lowest inhibition from other behaviors). Similarly, HIDE has the second highest relative priority since crayfish will spend the most time in Shelter due to the short distance between Food

Source and Shelter. Also it is clear that FORAGE has the second lowest relative priority as it is strongly inhibited by other behaviors and weakly inhibits other behaviors.

These results indicate that the structures learned by GA with two different fitness functions mostly conform to each other in terms of structure features. Therefore, they can be considered similar. However, the results of robustness test for these 4 structures in Figure 19 are not as stable as those shown in Figure 7. The primary reason might be we only have 4 structures. It makes sense that survival ability (steps) can be employed to evaluate the performance of Crayfish in a given environment. So we believe the better robustness can be shown if we have learned enough structures with survival ability as fitness function.

CHAPTER 7 CONCLUSION AND FUTURE WORK

In this work, we first presented a GA-based structure learning method and then conducted the experiments in different environment against a DEVS-based model Crayfish which employ mutual inhibition behavior network as its behavior selection mechanism. The analysis results indicated that consistent and robust structures can be learned by GA so that we concluded it is the structure, not the specific value of a set of coefficients, that defines the behavior pattern of a behavior network. In order to find the structure features corresponding to the environment changes we compared the relative priority (summed inhibition to other behaviors) of behaviors and inhibition relationship of behavior pairs in different environment. The results show that different environment does lead to different structures and the structures learned by GA in different environment are still consistent and robust. Also we applied robustness test to the same structures in different environments. The results clarify that the “best” structure learned by GA is highly dependent on the environment. Thus given a particular environment, the learned structures will mostly share a strong similarity and can describe the behavior pattern in the current context. Moreover, we applied a different fitness function (Survival steps) to the first simulation environment in order to see if the similar structure will be learned. These results indicate that the structures learned by GA with two different fitness functions mostly conform to each other in terms of structure features and therefore can be considered similar.

However, the differences between the structures learned in different environments are not as “distinct and simple” as we initially expect. The method used in finding the structure features and the method to filter the “non-appropriate” structures should be further elaborated. This suggests more complex analyses are needed in the future work to characterize the structures learned in different environments.

REFERENCES

[1] Herberholz J., F.A. Issa and D.H. Edwards (2001) Patterns of neural circuit activation and behavior during dominance hierarchy formation in freely behaving crayfish. The Journal of Neuroscience 21 (8): 2759-2767

[2] Issa FA, Adamson DJ, Edwards DH (1999) Dominance hierarchy formation in juvenile crayfish, Procambarus clarkii. J Exp Biol 202: 3497-3506.

[3] Yevgenia Kozorovitskiy, Elizabeth Gould, “Dominance Hierarchy Influences Adult Neurogenesis in the Dentate Gyrus”, The Journal of Neuroscience, July 28, 2004, 24(30):6755-6759

[4] Cha-Kyong Song, Jens Herberholz, and Donald H. Edwards, “The effects of social experience on the behavioral response to unexpected touch in crayfish”, Journal of Experimental Biology 209, 1355-1363, 2006

[5] X. Hu, D. Edwards, “Context-Dependent Control of Adaptive Behavior Selection”, proceedings of the Workshop on Bio-inspired Cooperative and Adaptive Behaviours in Robots, in co-operation with The Ninth International Conference on the SIMULATION OF ADAPTIVE BEHAVIOR (SAB'06)

[6] Dautenhahn, K.; Bond, A.H.; Canamero, L.; Edmonds, B. (Eds.), Socially Intelligent Agents, Creating Relationships with Computers and Robots, Springer, 2002

[7] Fong, T., Nourbakhsh, I., and Dautenhahn, K. A survey of socially interactive robots. Robotics and Autonomous Systems, 2003.

[8] Arkin, R.C., "Moving Up the Food Chain: Motivation and Emotion in Behavior-based Robots", in Who Needs Emotions: The Brain Meets the Robot, Eds. J. Fellous and M.~Arbib, Oxford University Press, 2005

[9] Delgado,C & Aylett,R.S (2004) Emotion and Action Selection: Regulating the Collective Behaviour of Agents in Virtual Environments The Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS04) pp1304-5

[10] Avila-Garcia, O., & Cañamero, L. (2004). Using hormonal feedback to modulate action selection in a competitive scenario. From Animals to Animats: Proceedings of the 8th International Conference of Adaptive Behavior (SAB'04) (pp. 243–252). Cambridge, MA: MIT Press.

[11] H. Jaeger and T. Christaller. Dual dynamics: Designing behavior systems for autonomous robots. Artificial Life and Robotics, 1998

[12] Peter Stone and Manuela Veloso. Multiagent Systems: A survey from a machine learning perspective. Autonomous Robots, 8(3):345–383, July 2000

[13]. Erfu Yang, Dongbing Gu, Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey (2004), University of Essex Technical Report CSM-404.

[14] Thomas Haynes and Sandip Sen. Evolving behavioral strategies in predators and prey. In Gerhard Weißand Sandip Sen, editors, Adaptation and Learning in Multiagent Systems, pages 113--126. Springer Verlag, Berlin, 1996

[15] Michail Maniadakis, Panos Trahanias, Hierarchical Cooperative CoEvolution Facilitates the Redesign of Agent-Based Systems, in Nolfi S., Baldassarre G., Calabretta R., Hallam J., Marocco D., Meyer J-A, Parisi D. (2006). From animals to animats 9: Proceedings of the Ninth International Conference on Simulation of Adaptive Behaviour, LNAI. Volume 4095. Berlin, Germany: Springer Verlag.

[16] Esben H. Ostergaard and Henrik H. Lund, "Co-Evolving Complex Robot Behavior", in Proceedings of ICES'03, The 5th International Conference on Evolvable Systems: From Biology to Hardware, Trondheim, Norway, Pages 308-319, March 17-20, 2003

[17]. G. Werner and M. Dyer. Evolution of herding behavior in artificial animals. In Proc. Simulation of Adaptive Behavior, 1992.

[18] Singleton, D. (2002). An Evolvable Approach to the Maes Action Selection Mechanism. Master Thesis, University of Sussex. Available at http://www.informatics.susx.ac.uk/easy/Publications.

[19] Maes, P. A Bottom-Up Mechanism for Behavior Selection in an Artificial Creature. In *From Animals to Animats*, MIT Press, 1991.

[20] Edwards D.H. (1991), Mutual inhibition among neural command systems as a possible mechanism for behavioral choice in crayfish. *J Neurosci* 11: 1210-1223.