A COMPARISON OF PREDATOR TEAMS WITH DISTINCT GENETIC SIMILARITY LEVELS IN SINGLE PREY HUNTING PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÇAĞRI YALÇIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

AUGUST 2009

Approval of the thesis:

A COMPARISON OF PREDATOR TEAMS WITH DISTINCT GENETIC SIMILARITY LEVELS IN SINGLE PREY HUNTING PROBLEM

submitted by **ÇAĞRI YALÇIN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Müslim Bozyiğit
Head of Department, **Computer Engineering** ________________

Dr. Onur Tolga Şehitoğlu
Supervisor, **Computer Engineering Department** ________________

Examining Committee Members:

Prof. Dr. Göktürk Üçoluk
Computer Engineering, METU ________________

Dr. Onur Tolga Şehitoğlu
Computer Engineering, METU ________________

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering, METU ________________

Assist. Prof. Dr. Tolga Can
Computer Engineering, METU ________________

Dr. Sinan Kalkan
- ________________

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ÇAĞRI YALÇIN

Signature :

ABSTRACT

A COMPARISON OF PREDATOR TEAMS WITH DISTINCT GENETIC SIMILARITY LEVELS IN SINGLE PREY HUNTING PROBLEM

Yalçın, Çağrı
M.S., Department of Computer Engineering
Supervisor : Dr. Onur Tolga Şehitoğlu

August 2009, 45 pages

In the domain of the complex control problems for agents, neuroevolution, i.e. artificial evolution of neural networks, methods have been continuously shown to offer high performance solutions which may be unpredictable by external controller design. Recent studies have proved that these methods can also be successfully applied for cooperative multi-agent systems to evolve the desired team behavior. For a given task which may benefit from both cooperation and behavioral specialization, the genetic diversity of the team members may have important effects on the team performance. In this thesis, the single prey hunting problem is chosen as the case, where the performance of the evolved predator teams with distinct genetic similarity levels are systematically examined. For this purpose, three similarity levels, namely homogeneous, partially heterogeneous and heterogeneous, are adopted and analyzed in various problem-specific and algorithmic settings. Our similarity levels differ from each other in terms of the number of groups of identical agents in a single predator team, where identicalness of two agents refers to the fact that both have the same synaptic weight vector in their neural network controllers. On the other hand, the problem-specific conditions comprise three different fields of vision for predators, whereas algorithmic settings refer to varying number of individuals in the populations, as well as two different selection levels such as team and group levels. According to the experimental results within a simulated grid environment, we show that different genetic similarity level-field of vision-algorithmic setting combinations beget different performance results.

Keywords: multi-agent systems, predator-prey problem, evolutionary methods, genetic algorithms, artificial neural networks

ÖZ

TEK AV AVLAMA PROBLEMİNDE FARKLI GENETİK BENZERLİK DÜZEYLERİNE SAHİP AVCI TAKIMLARININ KARŞILAŞTIRILMASI

Yalçın, Çağrı
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Dr. Onur Tolga Şehitoğlu

Ağustos 2009, 45 sayfa

Yapay sinir ağlarının yapay evrimi yöntemlerinin, erkinler için karmaşık kontrol problemleri alanında yüksek başarımlı çözümler sunduğu bilinmektedir. Harici bir kontrol edici tasarımı, bu çözümleri tahmin edememektedir. Yakın zamandaki çalışmalar, bu evrimsel yöntemlerin, işbirliği yapan çok erkinli sistemlerde arzu edilen takım davranışı evrimleştirilmesi için de başarıyla uygulanabileceğini kanıtlamaktadır. İşbirliği ve davranışsal özelleşmeden yararlanabilen bir görev için, takım üyelerinin genetik çeşitliliği takım başarımında önemli etkilere sahip olabilir. Bu tezde, tek av avlama örnek problem olarak seçilmiş ve farklı genetik benzerlik düzeylerine sahip evrimleşmiş avcı takımlarının başarımı sistematik olarak sorgulanmıştır. Bu amaçla, türdeş, orta seviye ve türdeş olmayan olmak üzere üç benzerlik düzeyi kullanılmış ve probleme ve algoritmaya özgü çeşitli ayarlamalarda analiz edilmiştir. Benzerlik düzeyleri, bir avcı takımındaki özdeş erkin gruplarının sayısı bakımından birbirlerinden ayrılmaktadır. İki erkinin özdeşliği yapay sinir ağı kontrolörlerinde aynı sinaptik ağırlık dizilerine sahip olmalarına karşılık gelmektedir. Öte yandan, probleme özgü şartlar, avcılar için üç farklı görme alanına; algoritmaya özgü ayarlamalar ise populasyonlardaki birey sayılarının değişimine ve grup ve takım seçim düzeyleri olmak üzere iki farklı seçim seviyesinin kullanımına karşılık gelmektedir. Benzetim ortamındaki ızgara dünyasında gerçekleştirilen deney sonuçlarına göre, farklı genetik benzerlik düzeyi-görme alanı-algoritmaya özgü ayar birleşimleri farklı başarım sonuçları doğurmaktadır.

Anahtar Kelimeler: çok erkinli sistemler, avcı-av problemi, evrimsel yöntemler, genetik algoritmalar, yapay sinir ağları

To my family

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Onur Tolga Şehitoğlu for his open-mindedness, trust, creative guidance and support throughout the research.

I would like to thank Assist. Prof. Dr. Erol Şahin for giving me a chance to work under the framework of KOVAN Research Laboratory, which was indisputably the most important experience I have had during my first steps along the exciting and laborious way of the academic maturity.

I am grateful to The Scientific and Technological Research Council of Turkey (TÜBİTAK) for their National Scholarship Programme which has financially supported me during my master education.

I would like to thank my excellent housemate Volkan Uysal, a recently graduated industrial designer of METU, for creating a clean, tidy and elegant home atmosphere and also for carefully listening to my Kafkaesque think-aloud processes throughout the study.

Finally, I would like to thank my family for teaching me what the true love and dedication mean, as well as for their encouraging arguments stimulating my powers of thought.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

DEDICATION viii

ACKNOWLEDGMENTS ix

TABLE OF CONTENTS x

LIST OF TABLES xii

LIST OF FIGURES xiii

CHAPTERS

1 INTRODUCTION 1

1.1 Multi-Agent Systems 3

1.2 Homogeneity vs. Heterogeneity in Cooperative Agent Teams 3

1.3 Evolving Controllers for Agents 5

1.4 The Predator-Prey Problem 6

1.5 Systematic Comparison of Evolved Predator Teams 7

2 ARTIFICIAL EVOLUTION 8

2.1 Genetic Algorithms 10

2.2 Artificial Evolution in Predator-Prey Domain 12

2.3 Artificial Evolution in Other Multi-Agent Systems 15

3 SIMULATOR 16

3.1 Simulation Environment 16

3.2 Agent Architectures 17

3.2.1 Prey Architecture 17

3.2.2 Prey Controller 17

3.2.3 Predator Architecture 18
3.2.4 Predator Controller 18
3.2.5 Sensor Model 19
4 EXPERIMENTAL FRAMEWORK 21
4.1 Introduction 21
4.2 Levels of Genetic Similarity 21
4.3 Genetic Algorithm 23
4.3.1 Parameter Settings 24
4.3.2 Genetic Architecture of Predator Controller 25
4.3.3 Genetic Operations 25
4.3.4 Fitness Evaluation 26
4.3.5 Initial Set-ups 27
5 EXPERIMENTS AND RESULTS 29
6 CONCLUSION 38
REFERENCES 42

LIST OF TABLES

TABLES

Table 5.1 Experimental Set-ups. *FOV* denotes the height of the right triangle-shaped field of vision of each directional sensor on each predator of a team, i.e. the half of the edge length of the square-shaped total field of vision of a predator. num_{pop} and $size_{pop}$ refer to the number of populations and the number of individuals in each population, respectively. 29

LIST OF FIGURES

FIGURES

Figure 2.1 A capture of the prey. The filled squares correspond to predators, whereas the circle is the prey. On the right image, the predator located in the right of the prey is about to occupy the square of it. 14

Figure 3.1 A snapshot of the simulator. The filled squares are predator agents, whereas the single circle depicts the prey. 16

Figure 3.2 Predator architecture and its field of vision. 19

Figure 3.3 The architecture of the neural network controller. 9 input neurons (1-4: predator sensors, 5-8: prey sensors, 9: bias), 3 hidden neurons, and 5 output neurons (1:up, 2: down, 3:left, 4:right, 5:stay) are shown. First 8 input neurons are connected to hidden ones, whereas bias is connected to both hidden and output neurons. 20

Figure 4.1 Formation of teams with different genetic similarity levels during genetic algorithm. Pi refers to predator *i*. 22

Figure 4.2 A sample initial set-up. The same predator positions are used in all set-up sets at all generations of the genetic algorithm. The position of the prey is randomly generated. The numbers on the agent bodies reflects the order of turns. . 27

Figure 4.3 A representative example of levels of selection for partially heterogeneous teams. Gij denotes the group (sub-team) *j* of predator team *i*. Each population is designed to have 7 individuals. The elite individuals are supposed to be the first 3 ones of the unevaluated, previous populations. In group level of selection, new coupling possibilities can be observed, wheras in team level of selection, the selected teams come from previous team compositions. *C* and *M* signs denote crossover and mutation, respectively. 28

Figure 5.1 Average fitness values of 10 evolved teams for each experimental set-up with standard deviations. The same, randomly generated 100 member prey positions set is used for the evaluation of each team. *Hom* refers to a homogeneous team, whereas *PartHet* and *Het* are partially heterogeneous and heterogeneous teams, respectively. The numbers coming after them are population sizes. *TL* and *GL* are the abbreviations for team level and group level of selections, respectively. 34

Figure 5.2 Average fitness values and standard deviations of 10 best performing teams after 25, 50, 100, and 200 generations for each experimental set-up. 35

Figure 5.3 Linewise representation of Figure 5.2 without standard deviations. 36

Figure 5.4 The capture rate and the related average simulation steps for three best performing, evolved teams of each set-up with a population size of 100. A single simulation step refer to one global motion cycle, where each of 5 agents (including prey) completes a single sense-act operation. The same, randomly generated 100 member prey positions set is used. 37

CHAPTER 1

INTRODUCTION

According to the famous German philosopher Max Scheler, one of the most crucial characteristics of the human being is its consciousness that can objectify himself [1]. In this context, via the contributions of the fields like philosophy, mathematics, medicine, economics, psychology and engineering from Ancient Greek to recent date, the humankind's examinations on his knowledge, mind, reasoning, self-improvement and creativity and his attempts to build artifacts realizing these properties have given rise to the birth of a new science that was called "*artificial intelligence*" (AI) firstly by McCarthy in 1955 [2]. Although there exists a strong consensus about the name of the field, the debates regarding how it can be defined have continued. Russell and Norvig [2] give the following definitions of AI:

- "Systems that think like humans
- Systems that act like humans
- Systems that think rationally
- Systems that act rationally"

The concept of *intelligent agent* has been constructed in the light of these different AI approaches that have been in interaction with each other. Historically, the word *agent* comes from the Latin *agere*, to do [2]. Woolridge et al. [3] define agent as: "a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives." *Autonomy* is used to mean that the action of agents are totally independent from any interference of humans or other systems, i.e. they are on their own for their operations [4]. Surely, *intelligent agents* should have some additional properties than the simple agents do. In [3], they are suggested as the entities "that are capable of *flexible* autonomous action in order to meet their design objectives", where *flexibility* is composed of three particular points:

- "*reactivity*: intelligent agents are able to perceive their environment, and respond in timely fashion to changes that occur in it in order to satisfy their design objectives
- *pro-activeness*: intelligent agents are able to exhibit goal-directed behavior by taking the initiative in order to satisfy their design objectives
- *social ability*: intelligent agents are capable of interacting with other agents (and possibly humans) in order to satisfy their design objectives"

As can be seen clearly, the complexity of building intelligent agent architectures directly depends on the conditions of the agent environment. In [2], the following classifications of environment properties are presented:

- *Fully Observable vs. partially observable*: Fully observable environment is the one where agent can sense the complete and up-to-date state of it. Partially observable ones give only the partial information about its state.
- *Deterministic vs. stochastic*: Deterministic environment is the one where the next environmental state is totally determined by the current state and the agent's action in it. In stochastic environments, there exist an uncertainty about the state from the agent's point of view.
- *Episodic vs. sequential*: Episodic environment is the one in which a single sense-act sequence of the agent is independent from the previous ones. In sequential environments, the agent's performance is the result of interacting sense-act sequences.
- *Static vs. dynamic*: Static environment is the one where not any external change in the environmental state occur during the agent's operation. Dynamic ones are the opposites.
- *Discrete vs. continuous*: Discrete environment is the one which has finite number of distinct states and sense-act sequences. Continuous ones offer continuous value opportunities for them.

- *Single agent vs. multi-agent*: Single agent environment is the one where only one agent operates. In multi-agent ones, there are more than one agent.

In the light of these classifications, in some hard environmental conditions, e.g. partially observable, sequential and stochastic, it is very difficult to build control architectures that can successfully fulfill the aforementioned characteristics of intelligent agents.

1.1 Multi-Agent Systems

The research in the multi-agent systems (MAS) started in the mid 1970s and they are generally defined as: "systems in which several interacting, intelligent agents pursue some set of goals or perform some set of tasks."[4] These systems cover various research fields, from robotic agents to computational ones, such as software and artificial life agents; and are shown to be useful in various industrial application domains, such as vehicle control or network communication [5], [6]. The complexity of building control strategies for MAS stems not only from the given environmental conditions, but also from its very nature where the action of each agent depends also on its interaction with other agents. Task-oriented *coordination* is the crucial part of interaction in MAS for both *cooperative* and *competitive* scenarios, where in the former "several agents try to combine their efforts to accomplish as a group what the individuals cannot" and in the latter "several agents try to get what only some of them can have" [4].

1.2 Homogeneity vs. Heterogeneity in Cooperative Agent Teams

Floreano and Keller [16] emphasize that "*Cooperation* applies the situations where two or more individuals obtain a net benefit by working together". This directly reflects the fact that for a given goal, the agent architectures should be designed so that the interactions of agents with each other and the environment provide them a reasonable task performance that cannot be guaranteed to be managed by a single agent. For this purpose, various agent cooperation and communication protocols have been proposed in MAS [4]. From the perspective of the artificial evolution of cooperative teams, a wide variety of successful studies has been continously submitted [6], [9], [10], [11], [15], etc..

For a particular task assigned to an agent team, deciding the control rule of each member is a challenging issue, since various types of team compositions directly affect the resulting goodness of the team. In the multi-robot team research, sundry studies have been proposed where each robot in the team is assigned a predetermined role via different sensory morphologies and behavioral controller design [7], [8]. Intelligent agent teams built with machine learning techniques, e.g. artificial evolution, have also harnessed the controller diversity of members, for which the proposed works have generally adopted two distinct approaches: The first approach tries to differentiate the agent controllers with the help of the suboperations of the related learning algorithm using a single potential solution set [6]. In the second one, the decision about the difference of agent controllers is given previously by creating subteams (groups) each of which acts with a different controller obtained as a result of the operations in a discrete potential solution set [9]. These *heterogeneous* systems are taking the advantage of *specialization* where each specialized agent or agent group is expected to focus on particular subtasks to display a high performance behavior in team level. However, as Quinn *et al.* [10] point out, *homogeneous* teams, whose members share the same identical control structures and sensory morphologies, do not necessarily offer behavioral homogeneity during their operations, since the behavior of each identical agent in a team can vary due to its current and history of sensory inputs. Baldassare *et al.* have succeeded to observe some behavioral specializations in evolved homogeneous teams for the task of aggregation and moving together toward a light target. On the other hand, in homogenenous teams, agent's ability of specialization can be improved due to the type of the selected controller architecture. In the multiple predator, single prey problem, Nitschke [11] showed that homogeneous teams with controllers that use both current and historical sensory information display better specialization and performance than homogenenous teams that have controllers that use only current sensor input for action.

In the artificial evolution of cooperative heterogeneous teams, an important issue that should be coped with is *credit assignment problem*, i.e. ambiguity about the individual contributions to the team performance. In fact, the contribution of each agent to the team performance may differentiate from each other. This problem can be solved in cooperative scenarios that easily allow the direct calculation of individual effect to the team goodness [6]. However, in some cases, this calculation may not be so easy. Bull and Holland [12] proposed the approach of *fitness sharing*, where each individual is assumed to have equal role on the team performance. This simple approach has been effectively adopted in various studies related to MAS and other fields [11], [13]. In some other studies [14], [15], for the individual contribution, various indirect evaluation methods have been proposed, which are said to give more robust measurements than *fitness sharing*.

1.3 Evolving Controllers for Agents

In his famous study, Turing [17] draws the attention to the difficulty and inefficiency of programming his intelligent machines manually and offers a learning method where so-called "child" programs, which are initially ineffective, are improved through the evaluations of an external experimenter. He explains this process by bridging an analogy on the *natural evolution* as follows:

- "Structure of the child machine = hereditary material"
- "Changes of the child machine = mutation"
- "Natural selection = judgment of the experimenter"

He also mentions the importance of the randomness and discusses the learning process "as a search for a form of behaviour which will satisfy the teacher (or some other criterion)". Actually, Turing's ideas created crucial fundamentals for the development of intelligent agent controllers which are expected to show high performance results in different initial environmental conditions.

By taking inspirations from the rules of *natural evolution* such as competition, selection reproduction and random variation, the approach of *evolving controllers for agents* takes an initial controller set and applies repetitive evaluations and modifications on the set members to obtain better and better performing hypotheses through iterations, where *goodness* of a behavioral controller is determined by the human's point of view [18]. Hence, it can be seen as an intelligent search technique in the hypothesis space to obtain an optimal or sub-optimal agent controller design.

Up to now, a huge number of studies harnessing this approach has been presented. From the robotics field, one of the first examples is the study of Beer and Gallagher [19], where controllers are evolved for homing and locomotion of simulated legged robots. Another early study is performed by Koza [20], where wall-following behavior is evolved for robots in simulation environment. Nolfi *et al.* [21] use a single physical robot during evolution and manage to evolve a controller which makes the robot display straight motion and object avoidance. In all of these works, a-priori design of a proper fitness function which reflects the desired behavior and reasonably differentiates high performing controllers from low performing ones is crucial.

1.4 The Predator-Prey Problem

The Predator-Prey problem is a well-known problem in MAS and various definitions as well as methods have been proposed for it. In its original form, four predator agents try to catch a single prey by occupying its four orthogonally adjacent squares in a simulated, infinite grid world [22]. All agents are allowed to move to vertical or horizontal squares and the behavioral strategy of prey is totally random. Korf [23] extended the problem by adding diagonal motion capability and suggested a greedy approach for predator agents. In his grid world, initially, the predators are allowed to see the complete world once and start to move in turn. Without any communication, in each turn, a predator is designed to move to the best possible square via utilizing an evaluation function which calculates the fitness of each movable square by substracting its distance to the nearest predator from its distance to prey. Hence, each predator is exposed to an attractive force towards the prey and a repulsive one from the nearest predator. Korf's prey adopts a probabilistic evading strategy. In [49], Stephens and Merx focused on various different control strategies, such as local, distributed and centralized. In the local strategy, a predator broadcasts its global positional information, only when it is close to the prey. In the distributed one, communication of global positional information occurs for all steps. In the centralized strategy, a single predator rules other predators to occupy each quadrant of the prey's neighborhood by a different predator [26]. They found that while distributed and centralizad strategies resulted in high level capture rates, the local strategy was insufficient for a reasonable success. Using reinforcement learning, Tan [24] proposed that sharing of sensor readings and experiences gives important advantages to predator agents.

Above studies can be seen as non-evolutionary approaches to the predator-prey domain. In Section 2.2, we will mention various studies applying evolutionary methods to the domain and clarify the definition of our problem.

1.5 Systematic Comparison of Evolved Predator Teams

The methods of artificial evolution induce impressive results in the development of intelligent agent behaviors that cannot be offered by manual designs. They are also shown to be effective for the derivation of controllers in cooperative agent teams, where agents should operate together to achieve a task via their local interactions with each other and environment. According to the task characteristics, the choice of various team compositions, perceptual and algorithmic parameter settings may have important effects on the success of evolved behaviors.

Our work mainly focuses on the evolution cooperative predator agent teams whose similarity levels are previously defined as *homogeneous*, *partially heterogenous* and *heterogenous*. In the first composition, each of the four predators in a team has an identical controller. In the next one, we define two groups each of which consists of a different controller for the use of a distinct two predator sub-team. In the last one, each agent adopts a different controller. Throughout our experiments, we evolve them with different fields of vision, population sizes and level of selections such as team and group level. The results of the experiments are compared with respect to various performance metrics.

The forthcoming organization of the thesis is as follows: Chapter 2 will give descriptions about artificial evolution methods and the details on *Genetic Algorithms*. Besides, it will present a literature survey on the use of these methods for predator-prey domain and other multi-agent systems. Chapter 3 will clarify the details of simulation environment and the adopted architectures of prey and predators. In Chapter 4, we will present the details of our team compositions, population sizes, level of selections, genetic algorithm and initial environmental set-ups. Chapter 5 will show the performance results of our evolutions. Chapter 6 will put a general review of the results and final comments, as well as some future works waiting to be done.

CHAPTER 2

ARTIFICIAL EVOLUTION

According to neo-Darwinian paradigm, populations and species of living organisms have been exposed to sundry physical processes such as *competition*, *selection*, *mutation* and *reproduction*. *Reproduction* is a fundamental aspect of species, where the genetic information of individuals are transferred to next generations by creating new ones. *Mutation* is the result of the inevitable errors during these transfers. *Competition* occurs within the share of the limited resources among the members of growing populations. As a direct result of this, *selection* is performed where individuals that are better suited to their environment survive and the weaker ones are eliminated. Hence, natural evolution can be seen as the total product of these processes [31].

With the increasing availability and computing power of digital computer systems, the field of evolutionary computation has started to emerge in the early 1960s by taking inspirations from the aforementioned characteristic processes of natural evolution. The proposed methods have been shown to be effective in various areas such as optimization, the design of controllers for intelligent agents and even the natural evolution itself to get better insights on it.

The three main types of evolutionary algorithms are as follows: *Genetic algorithms*, *Evolution strategies*, *Evolutionary programming*. Besides, there are also other techniques such as *genetic programming* and *classifier systems* which are seen as the derivatives of *genetic algorithms*. All of these methods share some common aspects [31]:

- Population of individuals (candidate solutions)
- Randomized generation of the descendants of individuals
- A fitness value assigned to individuals with respect to their evaluations

- Selection of better individuals

Genetic algorithms were originally proposed by Holland [32], [33] and mainly favors *crossover* operator. It also adopts *mutation*, however, with a less probability. Besides, a probabilistic selection method is applied. *Genetic programming* was suggested by Koza [34] and adopts the general characteristics of *genetic algorithms* to evolve executable computer programs with tree-formed representations. On the other hand, *classifier systems* were described by Holland [35] and Goldberg [36] and utilizes *genetic algorithms* to search the solution space of condition-action pair rules.

Evolution strategies were described by Rechenberg [37], [38] and Schwefel [39], [40] are exposed to some modifications through the history. In its modern state, each individual contains a solution offering in search space, as well as some parameters to control individual mutation distribution. Crossover is also used and selection is performed in a deterministic fashion by taking only the best performing ones [31].

Evolutionary programming was firstly developed by Fogel [41] to evolve finite state machines. It focuses on mutation and does not adopt recombination (crossover). It harnesses a probabilistic selection operator. In its modern state, it is also applicable for real-valued vector solution spaces.

A strict claim which scatters the advantages of evolutionary methods to all problem domains is questionable. In [31], Schwefel gives his insights on the point as follows:

"Since, according to the no-free-lunch (NFL) theorem (Wolpert and Macready 1996), there cannot exist any algorithm for solving all (e.g. optimization) problems that is generally (on average) superior to any competitor, the question of whether evolutionary algorithms (EAs) are inferior/superior to any alternative approach is senseless. What could be claimed solely is that EAs behave better than other methods with respect to solving a specific class of problems-with the consequence that they behave worse for other problem classes."

He suggests that classical optimization methods are more efficient than evolutionary ones "in solving linear, quadratic, strongly convex, unimodal, separable, and many other special problems". However, evolutionary methods should be taken into account "when discontinuous, nondifferentiable, multimodal, noisy, and otherwise unconventional response surfaces are involved". Afterwards, he concludes his ideas in the following simple implications:

- Evolutionary algorithms should not be adopted, if a known traditional method solves the problem in an exact fashion.
- Evolutionary algorithms are preferable, when a manual attempt for a new solution method is costly.

A clarifying addition to Schwefel's former suggestion can be the fact that evolutionary methods can be used if a known method is not easily calculable. Furthermore, they can also be adopted in the domains where a known method gives only one sub-optimal result, since they provide the opportunity of obtaining a wide variety of optimal and sub-optimal solutions due to their use of randomness and their maintenance of a solution candidate set. On the other hand, the use of artificial evolution for agent controllers takes its motivation mainly from the latter suggestion, since, in most of the cases, the environmental and problem-specific conditions make manual designs costly and even unpredictable.

2.1 Genetic Algorithms

As a population-based method, a genetic algorithm starts to operate on a set of usually randomly generated candidate solutions, which are also called *individuals* or *chromosomes*. Then, each individual of the population is evaluated and assigned a fitness value. Afterwards, some of these individuals are selected and copied to a mating buffer for recombination (crossover) and mutation. This selection process is performed probabilistically, where individuals with higher fitness values have higher chance to be selected. After the recombination and mutation are completed, depending on a combining rule, the original population and the mating buffer are merged to build a new population, i.e. next generation. This whole process continues iteratively until a satisfactory condition is reached.

Algorithm 1 depicts the general scheme of a genetic algorithm. In most of the cases, a genetic algorithm maintains a fixed-sized population through iterations. Therefore, different methods can be proposed for the combination of mating buffer and original population. A simple approach is using all individuals in the buffer and randomly selected ones from the original set. Another one may be making the size of the buffer equal to the original population and replacing all population members with the newly generated ones in the buffer. Both of these approaches have a main drawback: Relying on mating set and random additions (former case)

Algorithm 1 The Genetic Algorithm

1: Initialize population P
2: **while** Termination condition is not satisfied **do**
3: Evaluate individuals in P
4: Select mating buffer M from P
5: Apply crossover and mutation on M
6: Combine M and P to build a new P
7: **end while**

or relying only on mating set (latter one) may cause some deteriorations of individual performances through generations. A fine responsive method is combining the whole buffer with a subset of best performing individuals from the original population, which is called *elitist strategy* and generally leads to faster and better convergence results due to its preservation of top ranking ones in each iteration [31].

On other hand, the probabilistic selection techniques offered to build mating buffer have also shown some varieties. A simple technique is *fitness proportional selection*, also known as *roulette-wheel selection*, which selects the buffer members with respect to probabilities calculated for each individual via dividing its fitness by the total fitness value in the original population. In some situations where fitness differences between individuals are very high, this technique leads to a high level *selection pressure*, which refers to the danger of continuous selection of only a very small set of top ranking members. This situation is unwanted, since it hinders the genetic diversity, and therefore, the possibility of building diverse solutions in the buffer. To overcome this, another technique called *ranked selection* is suggested, where the individuals of the original population are ranked and probabilistically selected according to these ranks rather than their fitnesses. Another simple, but effective technique is *tournament selection* which performs a number of tournaments each of which consists of equal number of randomly selected individuals. The output of a single tournament is the highest ranking chromosome, and outputs coming from all tournaments are combined to form the mating buffer. The selection pressure of the tournament selection can be adjusted with the choice of size for a tournament. A very small size causes a low level pressure and is undesirable, since the selection probability of high performing individuals become less. In the same way, a very large number begets a high level pressure and should also be avoided, since it increases the selection probability of only the same best ranking individual, and by this way, does not allow a desired genetic diversity in the mating buffer [31], [42].

The main rationale behind *crossover* is the fact that we may obtain higher performing individuals by combining sundry random parts of two high ranked ones. Although it is not always the case, crossover has shown to be an effective operator to obtain fitter individuals. Mostly, it is applied with a high probability in the mating buffer. There are different recombination approaches in the literature. One is *one-point crossover*, where a random point is firstly determined and the whole segments after that point are swapped between two individuals. Another one is *two-point crossover*, where two different random points are selected at first and the swapping operation occurs between two individuals' corresponding parts that are located in between these points. A third approach is *uniform crossover*, which swaps randomly selected single points between two individuals.

Mutation is performed after crossover with a less probability by changing random and very small parts of chromosomes. By this way, new solution candidates are obtained. In binary individual representations, it is achieved by flipping randomly selected bits. In representations of floating point vectors, it is applied by adding random floating point numbers to randomly selected points of chromosomes.

2.2 Artificial Evolution in Predator-Prey Domain

In the original definition of predator-prey problem, where the prey capture is performed via occupying orthogonally adjacent grid cells to the prey, Haynes and Sen [25] adopted genetic programming to evolve cooperative predators, as well as a competitive co-evolution setup where both predator and prey controllers are evolved. They conclude that a prey that linearly moves in an infinite world is impossible to be surrounded. In [26], they modeled the problem so that all agents act simultaneously and diagonal moves are not allowed. Regardless of any field of vision, their predators can see the prey, however, cannot see each other. Their prey is able to see all four predators and follows a strategy that makes it escape from the nearest predator. Their predators can travel faster than the prey, since the prey is designed so that it does not move in a particular percent of total simulations steps allowed. In such configuration, they again harnessed genetic programming where after each move their evaluation function increments the fitness of each predator controller according to the inverse of the Manhattan distance of that predator to the prey. Besides, the evaluation method adds extra rewards to the predator controllers that make that predators locate adjacent to prey in the end of a simulation trial. They concluded that heterogenous teams perform slightly better than homogeneous ones, since their behavioral diversity can eliminate the deadlock situations that are faced in homogeneous teams. Jim and Giles [28] designed simultaneously moving agents that have the same speed and showed that for predator agents a communication language can be evolved via a genetic algorithm and can increase their capture performance significantly. In a recent study, Reverte *et al.* [30] extended Korf's model by modifying the evaluation function and adding transmission of relative location information between predators. Besides, they adopt artificial evolution to evolve controllers which take the location information and the results of evaluation functions of the other predators that are in the field of vision of that predator as input. Their fitness evaluation method takes prey capture time and number of collisions occured between predators into consideration. They use two different prey strategies such that random and evading prey.

Yong and Miikkulainen [29] altered the problem so that occupying the square of the prey by only one predator is said to be enough to capture the prey. They used three predators and one prey where all agents are allowed to move only in orthogonal directions and to have the same speed. Their prey is aware of its own global position, as well as the ones of all predators and moves directly away from the nearest predator. They proposed two different predator teams such as non-communicative and communicative ones. In the former, each predator can get its own global position and that of the prey. In the latter, in addition to its own global position, each predator can have both the global positions of the prey and other predators. Throughout the experiments with communicative predators, they evolved homogeneous predator teams by using a single population and heterogeneous predator teams by using a separate population for each predator controller using cooperative coevolution. They showed that evolved heterogeneous teams are much more successful than the homogeneous ones, since their problem demands role-based specializations and heterogenous agents are more responsive to these requirements. An interesting result of their study is the fact that non-communicative and heterogenous teams perform better than communicative and heterogenous ones, although the former offers a less variety of role assignments. Due to this result, they suggested that for their problem the information about other predator positions, i.e. communication, is not required, even expands the solution space redundantly and creates some sort of noise that diverts the predators from best possible solutions. In all of their experiments, the predators adopt neural network controllers and their evolutionary method is "The Enforced Subpopulations Method" (ESP) where evolution occurs at the neuron level instead of the complete neural network and each neuron is evolved in its own neuron subpopulation. Since three predators are used, three ESP's are run in a parallel fashion for heterogenous teams. The fitness of a team is calculated with respect to the difference between the average final and initial distances of the predators to the prey. This fitness is assigned to each of three predator controllers equally. Then, each shared fitness value is again equally shared among the neurons that forms that controller.

In the robotics domain, Nolfi and Floreano [27] applied a competitive co-evolution method to evolve a single predator and a single prey robot. A binary fitness value of 1 is assigned to a predator controller and a value of 0 to a prey controller, if the predator robot touches to prey during a trial, and vice versa if not. Nitschke [11] evolved controllers of both homogeneous and heterogeneous predator robot teams, where the desired task is to immobilize a prey robot adopting a static obstacle avoidance behavior. His fitness calculation for a predator team is proportional to how much it is able to slow down the prey. He concluded that with respect to several performance metrics, evolved heterogeneous teams show better performance results with a greater variety of behavioral specializations than homogeneous ones.

In this study, we take the definition of Yong and Miikkulainen [29], except the fact that we use four predators. Our simulated grid world is obstacle-free and toroidal as in the aforementioned studies, where if an agent runs off a side of the arena, it will reappear on the opposite side. Besides, our agents move sequentially and have limited range of perception. An example

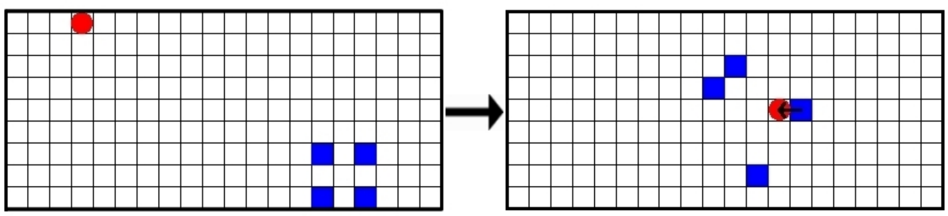


Figure 2.1: A capture of the prey. The filled squares correspond to predators, whereas the circle is the prey. On the right image, the predator located in the right of the prey is about to occupy the square of it.

of our prey capture is depicted in Figure 2.1. The details about the simulator and the agent architectures will be given in Chapter 3.

2.3 Artificial Evolution in Other Multi-Agent Systems

Among the other domains, an interesting study was presented by Cardon *et al.* [43], where they evolved heterogenous agent teams with a single population, multi-objective genetic algorithm for a job scheduling problem whose objectives are delay minimization and completion of predefined jobs. In [44], Bull and Fogarty evolved cooperative heterogeneous agents for a simulated trail following task via classifier systems. In [45], they evolved agent teams for the gait of a wall-climbing robot, where each leg is seen as a different agent and controlled by a different classifier system. In [46], Nelson *et al.* evolved robot controllers for a competitive searching task in a maze environment and showed that competition among robots leads to better performance results than the evolution of search behavior for a single robot offers. Trianni *et al.* [47] evolved aggregation behavior for swarm robotic systems and observed different behaviors emerged as a result of evolution. Bahceci [18] studied the effect of sundry paramater settings in the genetic algorithm by selecting the evolution of aggregation behavior in robot swarms as the case. In [9], Luke *et al.* evolved coordinated robot soccer teams via genetic programming. They adopted both homogeneous and partially heterogeneous teams, where the latter had various subteams, each of which operated with a different controller evolved in a different population, and were shown to outperform the former. In [6], in addition to evolving altruistic cooperation, Waibel *et al.* studied evolution of traditional cooperation for a foraging task which do not benefit from the specialization of team members. They adopted a single population genetic algorithm for the evolution of both homogeneous and heterogeneous teams. Besides, they proposed two different selection methods for the formation of mating buffer. The first one is team level selection and selects high performing teams via roulette wheel selection. The second one is individual level selection and selects high performing individuals again with roulette wheel selection. They concluded that homogeneous teams outperforms the heterogeneous teams regardless of these selection methods.

Despite these studies, for the problem domains which harness both the genetic diversity and cooperation of team members, the performances of artificial evolution applied to teams with different genetic similarity levels have not been fully studied. Within the existence of various problem-specific and algorithmic settings, this point deserves even much more attention. Hence, this study should be viewed as an attempt to build some answers on it.

CHAPTER 3

SIMULATOR

3.1 Simulation Environment

The simulator developed for this study is an obtacle-free, toroidal 31x31 grid environment, where the agents on it move sequentially, i.e. at each simulation step only a single agent is allowed to perform its sense-act operation.

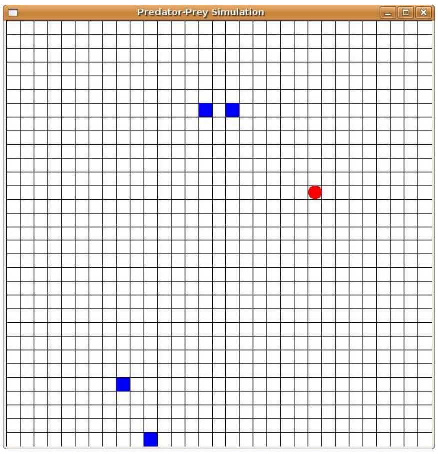


Figure 3.1: A snapshot of the simulator. The filled squares are predator agents, whereas the single circle depicts the prey.

A snapshot of our simulator is shown in Figure 3.1. It contains four predators and one prey. Each agent has a limited range of sense and occupies one different grid square during each of its moves. Besides, each one maintains the same constant orientation in the north direction without any rotational capability. From a single agent's point of view, the environment is a *partially observable*, *stochastic*, *sequential*, *dynamic*, *discrete* and *multi-agent* one.

3.2 Agent Architectures

3.2.1 Prey Architecture

The prey agent used in this study has a hypothetical architecture so that it knows the global x and y coordinates of its own square and of the predators that are located in its square-shaped field of vision.

3.2.2 Prey Controller

The prey employs a manually designed control algorithm, which generally forms an evading strategy. Algorithm 2 explains the controller of the prey. In each of its turns, the prey firstly

Algorithm 2 The Controller of the Prey

```
Initialize x offset, x_off, and y offset, y_off, with zero
Find the closest predator p in the field of vision
Calculate the relative x and y offsets of p by subtracting the global x and y positions of p
    from the ones of itself and assign them to x_off and y_off, respectively
if Both x_off and y_off are zero then
    Do nothing
else if One of x_off and y_off is zero and the other is not then
    Select an orthogonal square in the opposite direction of non-zero offset
else
    Find the offset that has smaller absolute value
    Select an orthogonal square in the opposite direction of that offset
end if
if The selected square is already filled by another predator then
    Do nothing
else
    Move to that square
end if
```

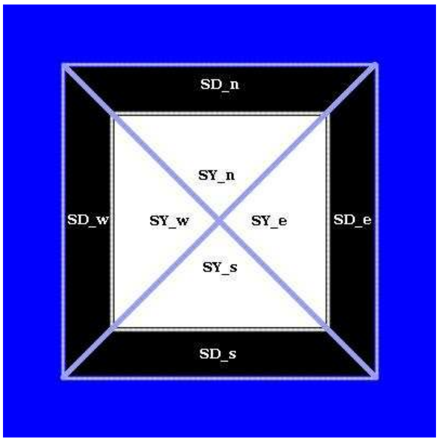
determines the closest predator in its field of vision in terms of Euclidean distance. Then, it compares the absolute values of relative x and y offsets of that predator and selects an orthogonally adjacent grid square in the opposite direction of the relative offset that has a smaller absolute value. If one of the offsets is zero, it is not taken into account in the comparison. If the prey cannot see any predator in its field of vision or the selected square is already occupied by a predator, it does not move and preserves its current position.

3.2.3 Predator Architecture

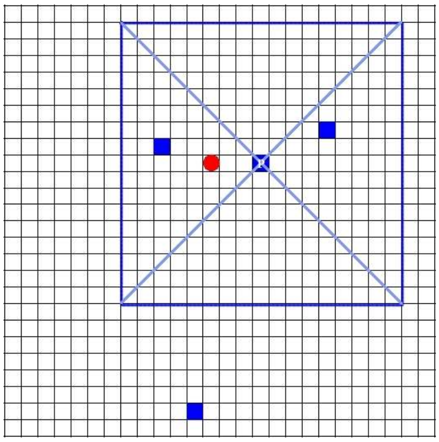
Each predator is designed to have 4 directional predator and 4 directional prey sensors located at the center of its circular body. With such structure, the architecture of our predators can be said to be inspired from the real robot models. Since the predators used in this study are designed to have limited range of sense, i.e. not all agents are sensible to each other always, we find the use of directional sensors convenient. By this way, our predators are more informed about the advantageous directions to move and are tried to be compensated for their lack of global position information of other agents in the environment. To the best of our knowledge, up to now, not such a robotic-inspired architecture has been harnessed for the predator-prey problem studied in the simulated grid world. Figure 3.2(a) shows a view of the predator architecture.

3.2.4 Predator Controller

Each predator adopts a fixed topology multi-layer feed forward neural network controller. It directly maps the sensory input taken at a simulation step to the 4 orthogonally adjacent square and 1 staying options in a reactive manner, i.e. without any memory or plan. It has 9 input neurons, where 4 of them are connected to predator sensors, the other 4 are connected to prey sensors and the last one is the bias, which has a constant value of 0.1 during all time steps. Besides, it has 3 hidden neurons and 5 output neurons for the 4 possible orthogonal directions such as up, down, left and right and an additional option for staying on the current position. In this way, the predators and the prey have the same speed, i.e. both can move only to orthogonally adjacent squares, and for a single predator it is impossible to catch the prey via simply chasing it. Therefore, predators must cooperate somehow to block the possible escape routes of the prey. After the calculation of the values for output neurons, the predator selects the square with the highest value to move. If the selected square is already occupied



(a) A view of the predator architecture. The big square represents the body of a predator. The triangles refer to 4 directional predator sensors and 4 directional prey sensors. For each direction, each prey and predator sensor pair is designed to be overlapped. SY denotes a prey sensor, whereas SD is a predator sensor. *n*, *s*, *e*, and *w* refer to north, south, east, and west directions, respectively.



(b) A predator with its square-shaped field of vision. *P* denotes the said predator. The diagonal lines are used to differentiate the sensing fields of directional sensors.

Figure 3.2: Predator architecture and its field of vision.

by another predator, not any move occurs. If the prey is on the selected square, the predator is said to catch the prey. Figure 3.3 shows the architecture of the neural network controller of predators.

3.2.5 Sensor Model

Each predator has the same square-shaped field of vision which is divided to 4 right triangles to sense in 4 directions. Each directional sensor senses the related objects located in its own 90 degree field of vision, i.e. in the area of its right triangle. This area is taken equal for both predator and prey sensors. Figure 3.2(b) represents a predator with its field of vision. The north directed predator and prey sensors of the mentioned predator *P* are not activated, since not any object is situated in the area of their right triangle-shaped fields of vision. On the other hand, the west directed predator and prey sensors of *P* are activated by another predator and the prey, respectively. Besides, the east directed predator sensor of *P* also senses another predator located at the upper right. However, the predator at the bottom cannot be seen by *P*, since it is out of the field of vision of *P*.

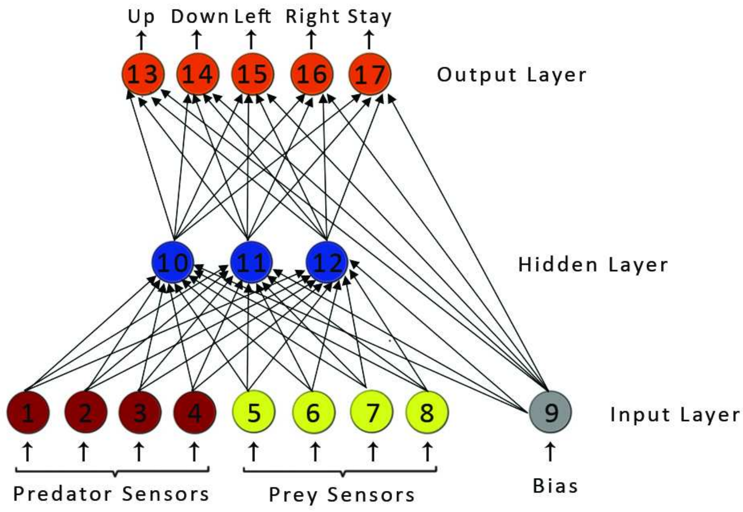


Figure 3.3: The architecture of the neural network controller. 9 input neurons (1-4: predator sensors, 5-8: prey sensors, 9: bias), 3 hidden neurons, and 5 output neurons (1:up, 2: down, 3:left, 4:right, 5:stay) are shown. First 8 input neurons are connected to hidden ones, whereas bias is connected to both hidden and output neurons.

In a single time step, the reading of a predator sensor is calculated by the inverse of the Euclidean distance between the sensing predator and the closest sensible predator, i.e. other sensible predators are ignored. Since only one prey exists in the environment, the reading of a prey sensor is directly equal to the inverse of the Euclidean distance between the sensing predator and the prey located in 90 degree field of vision of that predator.

CHAPTER 4

EXPERIMENTAL FRAMEWORK

4.1 Introduction

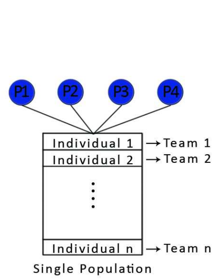
In our predator-prey problem, where the benefit of controller heterogeneity in a team has been already shown [29], studying the effect of evolved *partially heterogeneous* teams seems to form an interesting step, since it focuses on a relatively limited solution space than totally heterogeneous ones, whereas it preserves the controller diversity in the team. Within this motivation, we adopt three genetic similarity levels for predator teams, such as *homogeneous*, *partially heterogeneous* and *heterogeneous*. Besides, since the directional sensors of the predators are designed to have a particular sensing range, to obtain more comprehensive results, we alter the fields of vision of directional sensors for each similarity level throughout the experiments. To the best of our knowledge, the effect of varying the field of vision of predators has not been studied in the predator-prey problem, where the capture occurs by occupying the cell of the prey. On the other hand, for the *partially heterogeneous* and *heterogeneous* levels, we vary sizes of populations and also use two different selection levels, such as team and group level.

4.2 Levels of Genetic Similarity

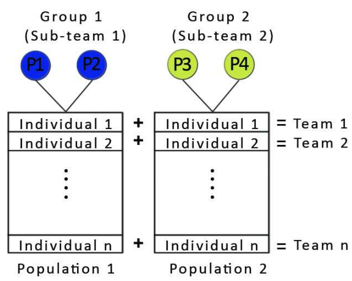
The first similarity level in this study is *homogeneous* one, where a single population is used and each individual in the population is evaluated via assigning it to all four predator controllers. Therefore, the team members operate with the same controller during a trial with that individual.

In the next level, a *partially heterogeneous* team is formed by two chromosomes which come from two distinct populations and have the same index numbers in their population arrays. By this way, in a single trial, two predators have the same controller from the first population, while the other two predators act with another controller from the second population. Therefore, in this configuration, we have two groups (sub-teams) in a single team, where a group has two agents that use the same controller.

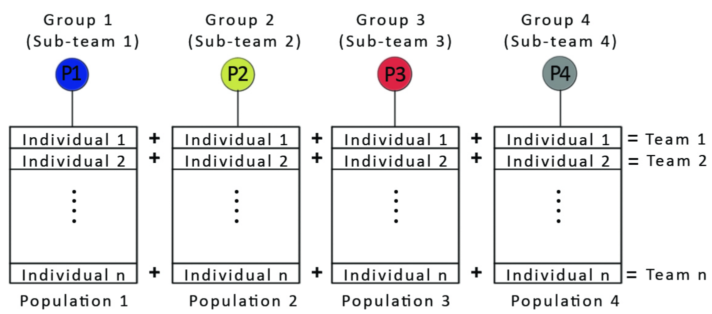
In the third level, four individuals coming from four distinct populations are assigned to four predator controllers, therefore, a totally *heterogeneous* team composition is obtained. Consequently, we have four groups in a single team, where each group has only one agent. Again, in a single trial, each selected chromosome has the same sequence number in its population as the other selected ones have in their populations. The formations of three levels of genetic similarity are depicted in Figure 4.1.



(a) Homogeneous Teams



(b) Partially Heterogeneous Teams



(c) Heterogeneous Teams

Figure 4.1: Formation of teams with different genetic similarity levels during genetic algorithm. Pi refers to predator *i*.

4.3 Genetic Algorithm

The general scheme of the genetic algorithm used for homogeneous teams is given in Algorithm 3. This algorithm starts with randomly initializing the individuals of the population with a size of p_{size}. For each of its iterations, i.e. generations, a random initial set-up set is built. Afterwards, each single individual in the population is assigned to all controllers of a predator team and is evaluated with the same initial set-up. Then, the individuals are sorted with respect to their fitness values and e best performing (elite) individuals is selected and

Algorithm 3 The genetic algorithm used for homogeneous teams

Initialize p_{size} individual population P randomly
Initialize current generation number, $g_{current} \leftarrow 0$
while $g_{current}$ is less than a predefined threshold **do**
    Create a random initial set-up set S
    for $i = 0$ to p_{size} **do**
        Evaluate individual i in each member of S and calculate its final fitness
    end for
    Sort individuals according to their fitness values
    Select e elite (best performing) individuals
    Select $p_{size} - e$ individual mating buffer M via tournament selection
    Apply crossover and mutation on M probabilistically
    Combine M and e elite individuals and assign them back to P
    $g_{current} \leftarrow g_{current} + 1$
end while
Return the best performing individual in P as the evolved controller

copied to a different location. Mating buffer is built via selecting $p_{size}-e$ number of individuals from the original population with the size of p_{size}. The building method of mating buffer is called *team level of selection*, since all predators share the same controller during a single trial. After applying probabilistic crossover and mutation operators, the new population is formed via combining e elite individuals and the mating buffer. This process continues for a predefined number of iterations and the best performing individual at the last generation is taken to be the evolved controller of this genetic algorithm run.

In partially heterogeneous and heterogeneous teams with *group level of selection*, a multiple number of these algorithms is run in parallel except the fact that in each generation a single randomly created initial set-up set is adopted for the operation of those teams whose controllers are built with corresponding individuals of separate populations. Hence, after sharing of team fitness among individuals that form the team, each seperate population performs its own elitist strategy, selection, as well crossover and mutation operators. On the other hand, in partially heterogeneous and heterogeneous teams with team level of selection, although separate populations are again adopted, the elitism strategy and formation of mating buffer are not performed in each population independently. Without any fitness sharing, they are realized at the team level with respect to team fitnesses by preserving evaluated team compositions. However, after team level of selection is completed, crossover and mutation are applied at each population separately. The details about selection levels and fitness evaluation methods will be explained in detail in Section 4.3.3 and Section 4.3.4, respectively.

4.3.1 Parameter Settings

Throughout the experiments with homogeneous teams, a population of 100 chromosomes is used. This forms evaluation of totally 100 teams in each generation of the genetic algorithm. To equalize the number of evaluated teams, the experiments with partially heterogeneous teams adopt two populations each of which has 100 individuals. With the same purpose, experiments with heterogeneous teams use four populations each of which consists of again 100 individuals. Apart from equalization of number of evaluated teams, we also equalize the total number of individuals among experiments with different similarity levels. Therefore, additional experiments are performed, where experiments with partially heterogeneous teams adopt two populations each of which has 50 individuals, which makes totally $2 \times 50 = 100$ individuals, and experiments with heterogeneous teams are carried out with four populations each of which has 25 individuals, which makes in total $4 \times 25 = 100$ individuals. In all experiments, the genetic algorithm lasts for 200 generations and the highest ranking individual at the last generation is accepted as the evolved controller. Surely, in the similarity levels other than homogeneous one, we obtain multiple evolved controllers coming from separate, co-evolved populations.

Since the performance of a single genetic algorithm run is highly dependent on the randomly generated initial population and probabilistic genetic operations throughout iterations, to reasonably compare the experimental set-ups, we perform 10 independent evolutionary runs for each experiment and take the average of the performances of evolved controllers built as a result of these runs.

4.3.2 Genetic Architecture of Predator Controller

The synaptic weights of the neural network in Figure 3.3 are encoded as $9 \times 3 + 3 \times 5 + 5 = 47$ floating point numbers on the genome of an individual in the population. In the initialization of the population, a random floating point number between [-1, 1] is assigned to each gene of each individual.

4.3.3 Genetic Operations

Elitist strategy and tournament selection are preferred for all evolutionary experiments. For 100 member populations, we adopt an elitist strategy which takes 10 best performing individuals unchanged and perform tournaments with a size of 5. For 50 and 25 member populations, elitist strategy takes 5 highest ranking individuals for both of them and tournament selection arranges tournaments with a size of 5 and 3, respectively.

For homogeneous teams, only team level of tournament selection is adopted, since each individual in the population is assigned to each of 4 predator controllers in a team during fitness evaluation. On the other hand, for partially heterogeneous and heterogeneous team compositions, the selection is applied from two distinct perspectives, such as group (sub-team) level and team level. Figure 4.3 depicts a schematic view of a representative example for both group and team levels of selection in partially heterogeneous teams. As can be seen, each population is supposed to have 7 individuals and corresponding individuals in two populations create controllers for a predator team as explained in Figure 4.1(b). In the group level of selection, after evaluation of teams, each team fitness value is equally shared among groups that create that team. Then, each population performs its own independent elitist strategy and tournament selection. If each team fitness has a different value, then the resulting new teams selected with elitism is guaranteed to preserve their previous team compositions. However, independent tournament selections cannot guarantee this, since independent probabilistic selections in two separate populations create previously unseen couplings, i.e. team compositions. Conversely, in the team level of selection, after evaluation of teams, selection directly occurs at the team level, i.e. for both of populations, a single elitist strategy and tournament selection are performed according to the team fitness values. With such selection level, separate populations can be thought to be merged and to behave like a single population, where a single individual is composed of the individuals that are responsible for the controllers of the defined sub-teams in a team. Therefore, the selected teams are guaranteed to be some of the ones in previous team compositions.

After selection is completed and mating buffer is built, as shown in Figure 4.3, a two-point crossover is applied between adjacent individuals, where the size of the swapped parts is $\lceil 47/3 \rceil = 16$. We give a probability of 0.8 to perform a crossover within each pair. Mutation is performed with a probability of 0.5 on each individual in the mating buffer by adding a random floating point number between [-1, 1] to a randomly selected gene of that individual. In this way, each gene, i.e. synaptic weight, on an individual in the buffer has a probability of $0.5 \div 47$ to be mutated. It is important to note that in both of group and team levels of selection, crossover and mutation are performed in each population in an independent fashion, therefore, any gene migration between populations is not allowed.

4.3.4 Fitness Evaluation

As explained in Algorithm 3, at each generation of genetic algorithm, an initial set-up set is randomly generated and each team is evaluated with this set. For homogeneous teams, the average of the fitness values coming from the members of this set is directly assigned to that individual. For other similarity levels, this average is equally shared among the individuals that form the controllers of sub-teams. In a single initial set-up, the fitness of a team is calculated with the following formula:

$$F_t = \begin{cases} (60 - \ d_{capture_t})/10 & \text{if prey captured} \\ (d_{init_t} - \ d_{final_t})/10 & \text{otherwise} \end{cases} \tag{4.1}$$

, where F_t denotes the fitness of team t, $d_{capture_t}$ is the average Euclidean distance of predators in t to the prey at the time of capture, and d_{init_t} and d_{final_t} are the average Euclidean distances of predators in t to the prey at the first and last simulation steps, respectively. Each simulation trial is limited to 100 moves for each agent. If the capture occurs during these moves, the current trial is stopped and the next one starts.

The above fitness function is similar to the one proposed in [29]. If the prey is captured, it takes capture positions of the predators into account. If the prey is not captured, the amount of approaching of the predator team to the prey is taken into consideration. The time passed up to the capture moment has no role in the fitness calculation. Hence, predator teams that catch the prey in a shorter time interval get not any extra reward.

4.3.5 Initial Set-ups

In each generation, each team is evaluated with the same 5 randomly generated initial-setups. In each set-up, the starting locations of predators are fixed on the bottom middle and only the initial position of the prey is random.

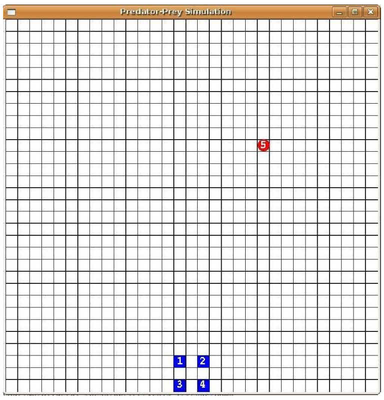


Figure 4.2: A sample initial set-up. The same predator positions are used in all set-up sets at all generations of the genetic algorithm. The position of the prey is randomly generated. The numbers on the agent bodies reflects the order of turns.

Figure 4.2 shows a sample initial set-up. The numbers on agent bodies determine the order of turns of agents. In partially heterogeneous agents, predators with numbers 1 and 2 adopt same controller, while the ones with numbers 3 and 4 operate with another same controller.

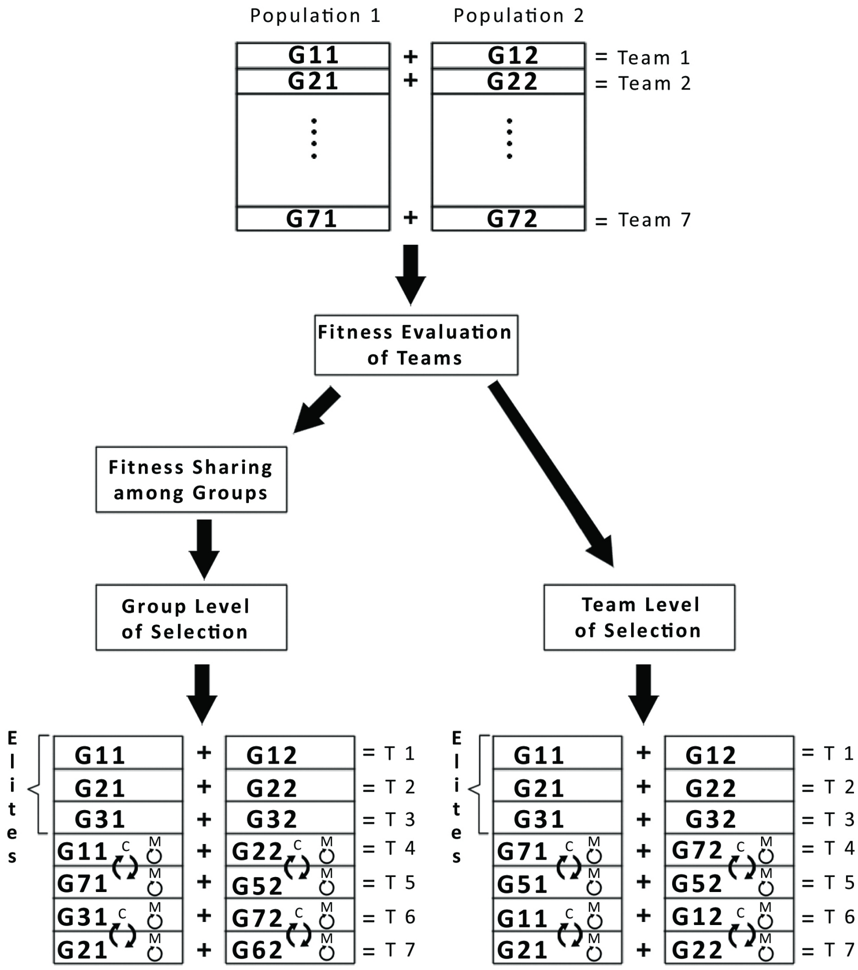


Figure 4.3: A representative example of levels of selection for partially heterogeneous teams. Gij denotes the group (sub-team) *j* of predator team *i*. Each population is designed to have 7 individuals. The elite individuals are supposed to be the first 3 ones of the unevaluated, previous populations. In group level of selection, new coupling possibilities can be observed, wheras in team level of selection, the selected teams come from previous team compositions. *C* and *M* signs denote crossover and mutation, respectively.

CHAPTER 5

EXPERIMENTS AND RESULTS

In this study, we take the single prey capture problem as the case and aim to compare the performances of evolved predator teams with three different levels of genetic similarity within the existence of sundry variations in sensing ranges of predator agents, as well as in the level of selection and also in the size of populations in the evolutionary method. Table 5.1 shows these design choices. Therefore, a total of $3 + 3 \times 2 + 3 \times 2 + 2 + 2 = 19$ experimental set-ups is built and $19 \times 10 = 190$ evolutionary runs are performed, where each set-up is repeated 10 times independently. For all experiments, the prey employs the same evading algorithm, where the half of the edge length of its square-shaped field of vision is fixed to 6.

Table 5.1: Experimental Set-ups. FOV denotes the height of the right triangle-shaped field of vision of each directional sensor on each predator of a team, i.e. the half of the edge length of the square-shaped total field of vision of a predator. num_{pop} and $size_{pop}$ refer to the number of populations and the number of individuals in each population, respectively.

Level of Genetic Similarity	FOV	Level of Selection	num_{pop}	$size_{pop}$
Homogeneous (Hom)	4, 8, 12	Team (TL)	1	100
Partially Heterogeneous (PartHet)	4, 8, 12	Group (GL)	2	100, 50
Heterogeneous (Het)	4, 8, 12	Group (GL)	4	100, 25
Partially Heterogeneous (PartHet)	8	Team (TL)	2	100, 50
Heterogeneous (Het)	8	Team (TL)	4	100, 25

To measure the performances of evolved teams and to compare them in a more accurate fashion, we build a single, randomly generated 100 initial prey positions set and adopt it for the evaluation of each evolved team. Figure 5.1 shows the average of the fitness values of 10 evolved predator teams for each experimental set-up, where each fitness value is an average performance of that team on the same test set that consists of 100 randomly generated initial prey positions. Through the increase of field of vision of directional sensors, the general tendency for each team composition is the increase of performances. However, the fitness increase observed in partially heterogeneous teams is greater than the ones observed in other team compositions. On the other hand, as can be expected, the decrease in population sizes begets lower average performances. In all field of vision settings, evolved homogeneous teams are the worst performing ones. This is also an expected result which reflects an important characteristic of our problem, i.e. harnessing the genetic diversity among team members, since even heterogeneous teams evolved within 25 member populations can outperform them. An interesting observation is the fact that while heterogeneous teams evolved within 100 member populations take the lead in $FOV = 4$, partially heterogeneous teams evolved within 100 member populations become the best performing ones in other fields of vision. Besides, in $FOV = 12$, partially heterogeneous teams evolved within 50 member populations shows better average performance than heterogeneous ones evolved within 100 member populations.

At first glance, not any significant performance difference is observed among the teams evolved with different levels of selection. In both group and team levels, corresponding setups show similar average fitness values, except that heterogeneous teams evolved within 25 member population and team level of selection show better performance than their corresponding ones evolved with group level of selection. Besides, in both levels, the partially heterogeneous teams evolved within 100 member populations are the best performing ones, the heterogeneous teams evolved within 100 member populations become the second, the partially heterogeneous teams evolved within 50 member populations are the third, and the evolved homogeneous teams take the last position. Despite these observations, an important difference between team and group levels of selection is shown in Figure 5.2(b). At first generation samples, teams evolved with team level of selection generally outperform the ones evolved with group level of selection. However, as approaching to the 200th generation, teams evolved with group level of selection catch them with greater increases in their average fitnesses. Therefore, our team level of selection is more preferable in genetic algorithms with smaller number of generations. This result may be due to the fact that the preservation of team structures allows team level of selection better optimization opportunity in a short time interval; however, limits it in terms of finding better solutions in the next generations. Conversely, previously unseen coupling chances make group level of selection be able to have a fitness rise with an increasing slope, although they seem to form a detrimental effect on the team performance at the first periods of evolution.

In Figure 5.2, for all field of vision choices, the homogeneous teams are outperformed by other evolved teams even from the first generation sample 25. Generally, as the generation number increases, the average fitness also rises except some rare cases where sundry fitness decreases are observed at some intermediate generation samples. However, these decreases are eliminated by sharp fitness increases at the next generation samples. In Figure 5.3, which is the linewise representation of Figure 5.2 without standard deviations, as the field of vision increases, both the average performances of evolved homogenenous teams and heterogeneous teams evolved within 25 member populations are outperformed by other set-ups via increasing differences for each generation sample except the partially heterogeneous and heterogeneous teams evolved with the team level of selection where the average fitnesses are more compact. In Figure 5.2(a), partially heterogeneous teams and heterogeneous teams evolved within 100 member populations show similar performances at 25th generation. However, in the next generation sample, the average fitness of the partially heterogeneous teams decreases, whereas heterogeneous ones maintain their fitness increase and take the lead. In the remaining generation samples, although partially heterogeneous teams eliminate the previous decrease by strong fitness rises, they could not reach the level of heterogeneous ones. On the other hand, in the 25th and 50th generations, partially heterogeneous teams evolved within 50 member populations show greater average fitness than heterogeneous teams evolved within 25 member populations. In the next generation samples, heterogeneous ones start to approach partially heterogeneous teams such that at 200th generation, only a small difference remains between them.

In Figure 5.2(c), for the first generation sample, evolved partially heterogeneous teams beat the evolved heterogeneous ones regardless of the population size. In the next sample, heterogeneous teams evolved within 100 member populations outperform the partially heterogeneous teams evolved within 50 member populations; however, their average performance could not beat the one of the partially heterogeneous teams evolved within 100 member populations. In the long run, heterogeneous ones lose their rank and locate at the third position which is under the performances of evolved partially heterogeneous teams. For all generation samples, heterogeneous teams evolved within 25 member populations take the fourth best position which has only homogeneous teams below itself.

In addition to the above average fitness comparisons, we try to figure out which genetic similarity level is the most efficient one by comparing their prey capture rates and the average simulation steps required for them up to the capture moment. For this purpose, set-ups with 100 member population(s) are only considered and three best performing, evolved teams are selected for each of them. Figure 5.4 shows the capture rates and the related average simulation steps for best three teams of each set-up, where the test set is the previously adopted, 100 member initial prey position set. For each trial, a single simulation step is defined to be a single motion cycle at the global level, where each of 5 agents in the environment completes its single sense-act sequence. The teams located in the bottom right of the coordinate system in the figures are said to be the most efficient ones, because they are effectively able to fulfill our two objectives: they catch the greatest number of preys in the smallest time interval.

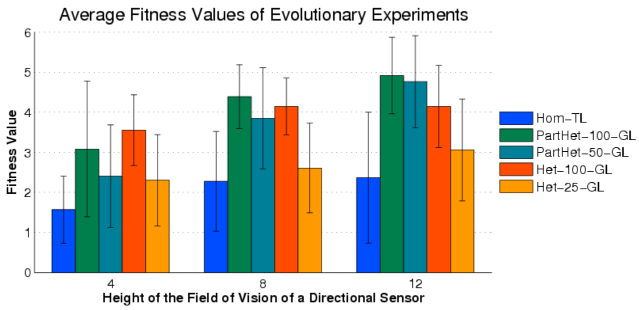
In $FOV = 4$, evolved homogeneous teams have the lowest capture rates with the lowest average times. Other similarity levels manage to catch greater number of preys with similar average times as in homogeneous level. Specifically, partially heterogeneous teams show the best capture rates. Compared to the results in Figure 5.1(a), an important implication for $FOV = 4$ can be the fact that although the evolved heterogeneous teams show better average fitness values, when only best three out of 10 evolved teams are considered for each, partially heterogeneous teams outperform all. A possible explanation for this lies again in Figure 5.1(a), where standard deviation of the performances of evolved partially heterogeneous teams is relatively high.

In $FOV = 8$, we observe a clear classification between similarity levels: evolved homogeneous teams have the lowest capture rates and low average times. Heterogeneous teams show greater capture rates with the highest average times, whereas partially heterogeneous teams show the greatest capture rates, as well as the lowest average times. Focusing on levels of selection, the effect of team level of selection in heterogeneous similarity level is a slight decrease in capture rates and a rather greater decrease in average times. On the other hand, the effect of team level of selection in partially heterogeneous similarity level is a slight decrease in capture rates and a rather greater increase in average times.

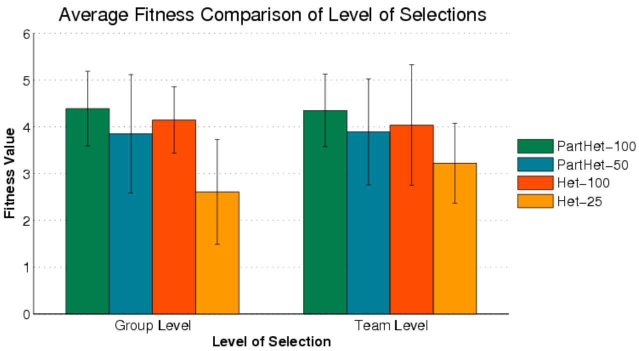
In $FOV = 12$, the performances of similarity levels scatters in a more compact fashion. The performance of homogeneous teams approach to the ones of other similarity levels. However, the ranking of levels in terms of the capture rates and the average times is the same as in

$FOV = 8$.

Focusing on the evolved behaviors of predator teams, a general result is the fact that although similar dynamic role specializations are observed throughout different fields of vision, the frequency of them increases, as the field of vision of predators rises. This fact is especially clear for evolved homogeneous teams: In $FOV = 4$, only a small percent of them has predators that can show dynamic roles, such as chasing the prey and waiting to block the prey, during a single trial. In some initial set-ups, the predators do not take specific roles and show the similar chasing behaviors that lead to unsuccessful trials. However, in $FOV = 12$, the number of the evolved homogeneous teams including predators that can show these dynamic roles increases significantly. Moreover, the members of these successful teams act in a compact fashion such that generally they start with a global flocking behavior and divide into two heterogeneously behaving sub-teams as getting closer to the prey. On the other hand, although a huge number of the evolved heterogeneous teams generally includes predators that can show the aforementioned dynamic role specializations even in $FOV = 4$, their predator agents mostly act in a heavily dispersed fashion which creates a waste of time to sandwich and capture the prey and even unsuccessful trials in some rare cases. Conversely, the partially heterogeneous teams can operate in a more compact way like some successful homogeneous teams again by building heterogeneously behaving sub-teams. Different from homogeneous teams, they can show this behavioral heterogeneity even at the start of a simulation trial. Besides, most of the time they have dynamically role taking predators even in $FOV = 4$ like heterogeneous teams. Therefore, it can be concluded that the partially heterogeneous teams combine the advantages of two worlds: the compact team level behavior of homogeneous teams and frequently observed behavioral specializations of heterogeneous teams.

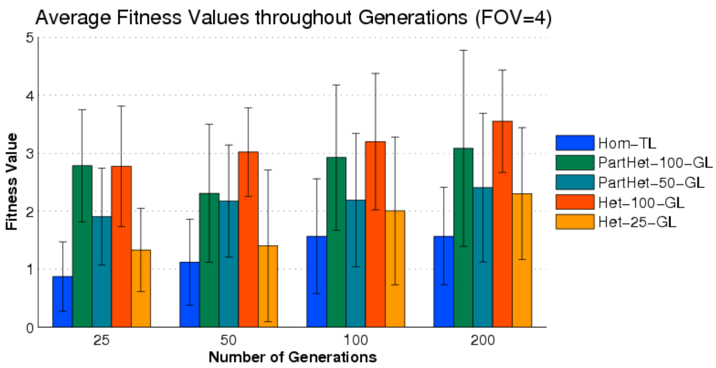


(a) Average fitness values and standard deviations of 10 evolved teams for each experimental set-up except the non-homogeneous ones evolved with team level of selection.

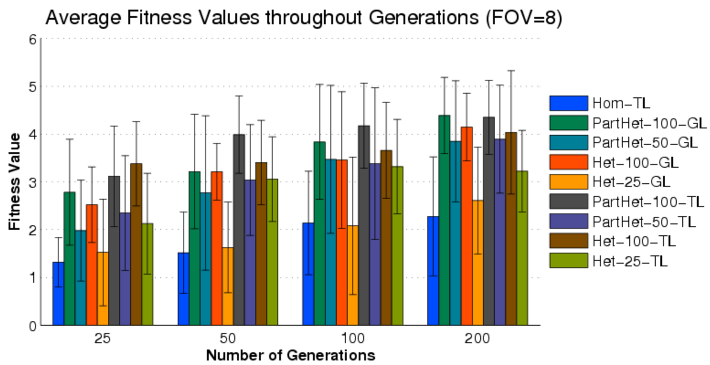


(b) Average fitness values and standard deviations of 10 evolved teams for each of partially heterogeneous and heterogeneous cases. The performances of team level and group level of selections are compared only for $FOV = 8$.

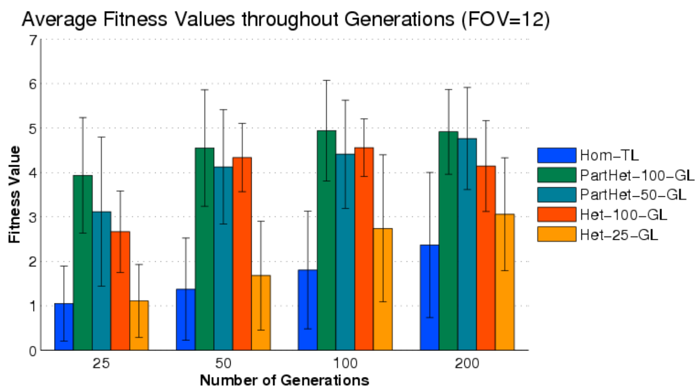
Figure 5.1: Average fitness values of 10 evolved teams for each experimental set-up with standard deviations. The same, randomly generated 100 member prey positions set is used for the evaluation of each team. *Hom* refers to a homogeneous team, whereas *PartHet* and *Het* are partially heterogeneous and heterogeneous teams, respectively. The numbers coming after them are population sizes. TL and GL are the abbreviations for team level and group level of selections, respectively.



(a) Average fitness values and standard deviations of 10 best performing teams after 25, 50, 100, and 200 generations for each experimental set-up in $FOV = 4$.

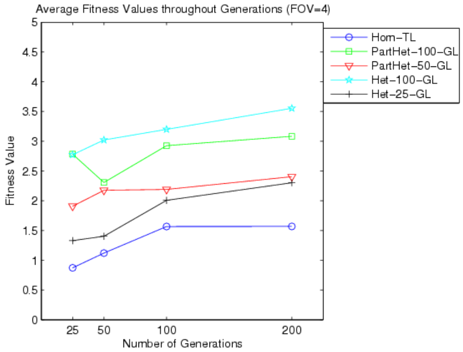


(b) Average fitness values and standard deviations of 10 best performing teams after 25, 50, 100, and 200 generations for each experimental set-up in $FOV = 8$.

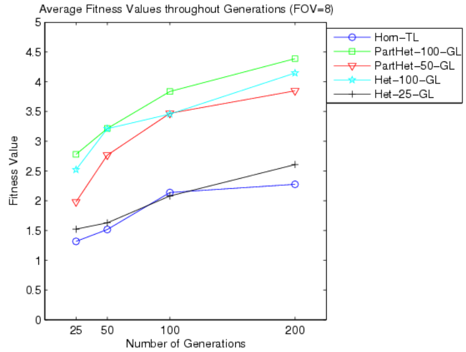


(c) Average fitness values and standard deviations of 10 best performing teams after 25, 50, 100, and 200 generations for each experimental set-up in $FOV = 12$.

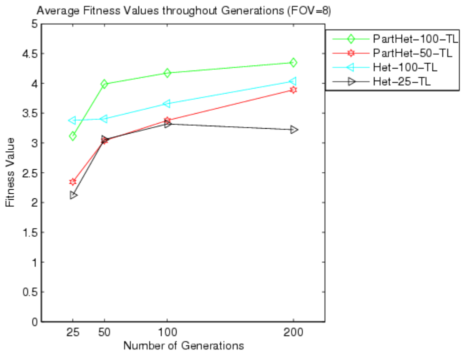
Figure 5.2: Average fitness values and standard deviations of 10 best performing teams after 25, 50, 100, and 200 generations for each experimental set-up.



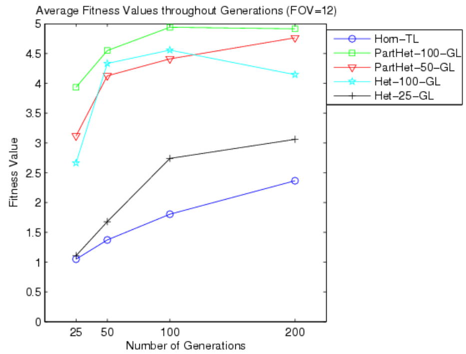
(a) Average fitness values of 10 best performing teams after 25, 50, 100, and 200 generations for each experimental set-up in $FOV = 4$.



(b) Average fitness values of 10 best performing teams after 25, 50, 100, and 200 generations for each experimental set-up in $FOV = 8$ except non-homogenenous teams evolved with team level of selection.

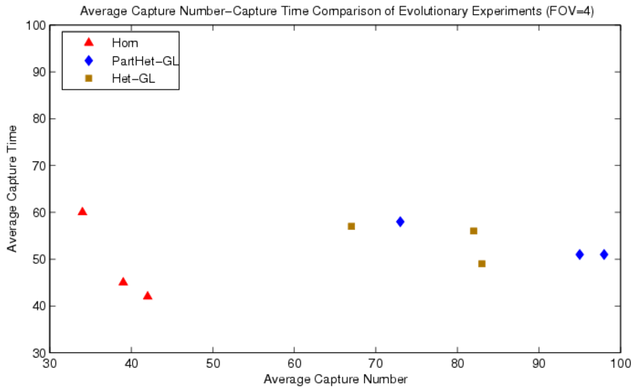


(c) Average fitness values of 10 best performing teams after 25, 50, 100, and 200 generations for each non-homogenenous team composition evolved with team level of selection in $FOV = 8$.

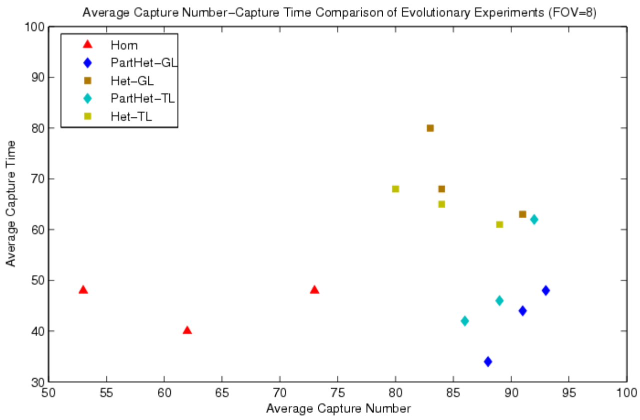


(d) Average fitness values of 10 best performing teams after 25, 50, 100, and 200 generations for each experimental set-up in $FOV = 12$.

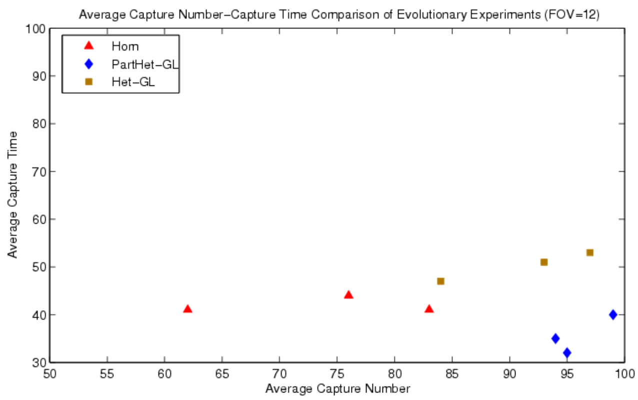
Figure 5.3: Linewise representation of Figure 5.2 without standard deviations.



(a) FOV=4



(b) FOV=8



(c) FOV=12

Figure 5.4: The capture rate and the related average simulation steps for three best performing, evolved teams of each set-up with a population size of 100. A single simulation step refer to one global motion cycle, where each of 5 agents (including prey) completes a single sense-act operation. The same, randomly generated 100 member prey positions set is used.

CHAPTER 6

CONCLUSION

In this study, we examined the performance comparison of evolved predator teams with distinct genetic similarity levels in a single prey capture problem, where the prey is said to be hunted, when its grid cell is occupied by one of the four predator agents in a simulated grid world. These levels were predefined as homogeneous, partially heterogeneous and heterogeneous. In the homogeneous level, all four predators in a team adopted the same controller that corresponded to an individual of a single population. In the partially heterogeneous level, two sub-teams each of which had two predators and operated with a different controller that comes from a separate population were built. In the heterogeneous level, the controller of each predator agent was different from each other. Hence, four controllers coming from four separate, co-evolved populations were used. For each similarity level, the effects of the use of three distinct fields of vision for the sensors of predators on the team performance were investigated. Besides, for each of partially heterogeneous and heterogeneous teams, two distinct population sizes were used with the aim of equalizing the number of teams and individuals adopted in the genetic algorithm with the ones in the evolution of the homogeneous team composition. In addition to this, two different levels of selection, such as group (sub-team) level and team level, were also analyzed and compared for the teams other than homogeneous ones.

The simulated environment used in this study was designed to be an obstacle-free, toroidal grid world. The agents were supposed to move via taking turns. For all experiments, the predator agents were assumed to have the same initial positions with the same order of turns; however, the prey was allowed to have random initial positions. Again in all experiments, the prey was supposed to employ the same manually designed control algorithm that made it move directly away from the nearest predator that was located in the constant field of vision of itself. The prey and the predators were allowed to move only in orthogonal directions and to have the same speed. Each predator was assumed to have directional sensors with a particular field of vision. The readings of these directional sensors were taken as input to the multi-layer feed-forward neural network controller of it.

Each individual in the population(s) encoded the connection weights of the neural network controller(s) of predator(s). The elitist strategy and tournament selection were used for all evolutionary experiments. While homogeneous teams were evolved within a single 100 member population, partially heterogeneous and heterogeneous teams were evolved within two 100 and 50 member populations and four 100 and 25 member populations, respectively. For the 100 member population(s), the elitist strategy was designed to take 10 best performing individuals unchanged and tournaments with a size of 5 were performed. For the 50 member populations, the elitist strategy took 5 best performing individuals unchanged and tournaments were performed again with a size of 5. For the 50 member populations, elitism took 5 best performing chromosomes and the tournaments had a size of 3. In the group level of selection, after the evaluation of each team, the fitness was equally shared among the chromosomes that form the controllers in the team; and independent elitism and tournament selection was applied for each population. On the other hand, in the team level of selection, separate populations acted like a single big population; and elitist strategy and tournament selection occurred at the team level without changing some of the previous team compositions. To obtain more reliable results, each evolutionary experiment was repeated 10 times with different seeds. Besides, each of 10 evolved teams of each experiment was tested in the same randomly generated 100 initial prey position set.

In the light of the above configurations, we obtained the following contributions:

- In the evolutionary approaches to the single prey-multiple predators problem, where prey capture is done via occupying the grid cell of the prey, adopting directional sensors with limited range of sense instead of global positional information of agents is also an appropriate way to obtain successful results.
- For all fields of vision and generation samples, evolved homogeneous teams are generally outperformed even by partially heterogeneous and heterogeneous teams evolved within populations that have much smaller number of individuals than homogeneous

ones. Besides, as the field of vision increases, both the average performances of evolved homogeneous teams and heterogeneous teams evolved within 25 member populations are surpassed by other set-ups via increasing amounts for each generation sample except the partially heterogeneous and heterogeneous teams evolved with the team level of selection where the average fitnesses scatter in a more compact fashion.

- In the smallest field of vision, the heterogeneous teams evolved within 100 member populations give the highest average performance. As the vision area increases, partially heterogeneous teams evolved within 100 member populations take the lead. In the highest field of vision, even partially heterogeneous teams evolved within 50 member populations can outperform the heterogeneous teams evolved within 100 member populations.

- In the intermediate field of vision, the team level of selection and the group level of selection give similar final average performances for partially heterogeneous and heterogeneous team compositions. However, in the first generation samples, the teams evolved with the team level of selection give higher performance results and this difference is eliminated throughout the next samples by greater fitness increase of the teams evolved with the group level of selection.

- Partially heterogeneous teams are generally more preferable than homogeneous and heterogeneous teams, since they can capture more preys in smaller time interval via behaving globally compact and being able to show frequent dynamic roles, where the former observation comes from the homogeneous teams and the latter one is from heterogeneous ones.

- Although the types of role specializations in evolved predator teams do not change through the increase of the field of vision, their frequencies increase and this fact creates generally more successful predator teams.

This study can be expanded via adopting varying numbers of predators and preys with more diverse genetic similarity levels, as well as with varying order of turns and randomly generated initial predator positions. Besides, a different fitness function that takes the capture time into the consideration can also be adopted. An interesting addition would be the use of a more reliable fitness sharing method that assigns more fitness to the individuals that contribute to the team performance more than others. Furthermore, the effect of the crossover between the individuals of separate populations, i.e. gene migration, is also worth analyzing. Another noteworthy investigation would be the use of recurrent neural networks or neuromodulated plasticity for homogeneous teams and comparing the role specializations and performances of them with the ones of the partially heterogeneous and heterogeneous teams that have only feed forward neural network controllers. Finally, the benefit of partially heterogeneous team compositions should also be examined in the domains other than predator-prey.

REFERENCES

[1] M. Scheler, İnsanın Kosmostaki Yeri, Trans.: Harun Tepe, Ayraç Yayınevi, Ankara, Turkey, 1998.

[2] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Second Edition, Prentice Hall, Upper Saddle River, New Jersey, USA, 2003.

[3] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice", The Knowledge Engineering Review, 10(2), pp. 115-152, 1995.

[4] G. Weiss, Multiagent Systems: A Modern Approach to Distributed Aritifial Intelligence, MIT Press, Cambridge, MA, USA: MIT Press, 1999.

[5] S. Franklin and A. Graesser, "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.

[6] M. Waibel, L. Keller, D. Floreano, "Genetic Team Composition and Level of Selection in the Evolution of Cooperation", IEEE Transactions on Evolutionary Computation, vol. 13, issue 3, pp. 648-660, 2009.

[7] T. Balch and R. Arkin, "Behavior-based formation control for multiagent robot teams", IEEE Transactions on Robotics and Automation, vol. 14, issue 6, pp. 926-939, 1998.

[8] V. Gervasi and G. Prencipe, "Flocking by a set of autonomous mobile robots", Technical Report 515, Dipartimento di Informatica, University of Pisa, Italy, 2001.

[9] S. Luke, C. Hohn, J. Farris, G. Jackson, and J. Hendler, "Co-evolving soccer softbot team coordination with genetic programming", in Proceedings of The First International Workshop On RoboCup, IJCAI-97, Springer-Verlag, pp. 214-222, 1997.

[10] G. Baldassarre, S. Nolfi, and D. Parisi, "Evolving mobile robots able to display collective behaviors.", Artificial Life, vol. 9, pp. 255-268, 2003.

[11] G. Nitschke, "Emergence of Cooperation in a Multiple Predator, Single Prey Game", Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference, St. Augustine, Florida, USA, 2003.

[12] L. Bull and O. Holland, "Evolutionary Computing in Multi-Agent Environments: Eusociality", In Proceedings of the Second Annual Conference on Genetic Programming, Morgan Kaufmann, pp. 347-352, San Francisco, USA, 1997.

[13] F. Gomez, J. Schmidhuber, R. Miikkulainen, "Accelerated Neural Evolution through Cooperatively Coevolved Synapses", Journal of Machine Learning Research, vol. 9, pp. 937-965, 2008.

[14] T. Miconi, "A Collective Genetic Algorithm", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 876-883, San Francisco, CA, USA, 2001.

[15] T. Arita and Y. Suzuki, "A Comparative Evaluation of Methods for Evolving a Cooperative Team", in Frontiers in Evolutionary Robotics (H. Iba ed.), pp. 1-8, Crotia, 2008.

[16] D. Floreano and L. Keller, "Methods for Artificial Evolution of Truly Cooperative Robots", International Work Conference on Artificial Neural Networks 2009, Salamanca, Spain, 2009.

[17] A. Turing, "Computing Machinery and Intelligence", Mind, 59, pp. 433-460, 1950.

[18] E. Bahçeci, "Evolving Aggregation Behaviors for Swarm Robotic Systems: A Systematic Case Study", MS thesis, Middle East Technical University, Ankara, Turkey, 2005.

[19] R.D. Beer and J.C. Gallagher, "Evolving dynamical neural networks for adaptive behavior", Adaptive Behavior, vol. 1, no. 1, pp. 91-122, 1992.

[20] J.R. Koza, "Evolution of subsumption using genetic programming", in Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life (F.J. Varela, P. Bourgine eds.), pp. 110-119, The MIT Press, Cambridge, MA, 1992.

[21] S. Nolfi, D. Floreano, O. Miglino, F. Mondada, "How to Evolve Autonomous Robots: Different Approaches in Evolutionary Robotics", in Proceedings of the IV International Workshop on Artificial Life (R.A. Brooks, P. Maes eds.), Cambridge, MA, MIT Press, 1994.

[22] M. Benda, V. Jagannathan, R. Dodhiawalla, "On Optimal Cooperation of Knowledge Sources", Tech. Rep. BCS-G2010-28, Boeing AI Center, Boeing Computer Services, Bellevue, WA, USA, 1985.

[23] Richard E. Korf, "A simple solution to pursuit games", In Proceedings of the 11th International Workshop on Distributed Artificial Intelligence, pp. 183-194, Glen Arbor, MI., USA, 1992.

[24] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents", in Proceedings of the Tenth International Conference on Machine Learning (ICML), pp. 330-337, 1993.

[25] T. Haynes and S. Sen, "Evolving behavioral strategies in predators and prey", IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems, (S. Sen ed.), Morgan Kaufmann, pp. 32-37, Montreal, Quebec, Canada, 1995.

[26] T. Haynes and S. Sen, "Co-adaptation in a team", International Journal of Computational Intelligence and Organizations, vol. 1, pp. 1-4, 1997.

[27] S. Nolfi, D. Floreano, "Coevolving predator and prey robots: Do "arms races" arise in articial evolution" Articial Life, 4(4), 337-357, 1998.

[28] K. C. Jim, C. L. Giles, "Talking helps: Evolving communicating agents for the predator-prey pursuit problem", Artificial Life 6(3), pp. 237-254, 2000.

[29] C. H. Yong, R. Miikkulainen, "Cooperative Co-evolution of Multi-Agent Systems", Technical Report AI01-287, Department of Computer Science, University of Texas, USA, 2001.

[30] J. Reverte, F. Gallego, R. Satorre, F. Llorens, "Cooperation Strategies for Pursuit Games: From a Greedy to an Evolutive Approach", in MICAI 2008: Advances in Artificial Intelligence, Lecture Notes in Computer Science, (Berlin Heidelberg), vol. 5317, pp. 806-815, Springer-Verlag, 2008.

[31] T. Baeck, D. B. Fogel, Z. Michalewicz, "Handbook of Evolutionary Computation", IOP Publishing Ltd., Bristol, UK, 1997.

[32] J. H. Holland, "Outline for a logical theory of adaptive systems", J. ACM 3, pp. 297-314, 1962.

[33] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan, Ann Arbor, University of Michigan Press, MI, USA, 1975.

[34] J. R. Koza, Genetic programming: on the programming of computers by means of natural selection. Cambridge, MA, MIT Press, USA, 1992.

[35] J. H. Holland, K. J. Holyoak, R. E. Nisbett, P. R. Thagard, Induction: Processes of Inference, Learning, and Discovery, MIT Press, Cambridge, MA, USA, 1986.

[36] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, USA, 1989.

[37] I. Rechenberg, "Cybernetic solution path of an experimental problem", Library Translation No: 1122, Royal Aircraft Establishment, Farnborough, UK, 1965.

[38] I. Rechenberg, "Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution", Frommann-Holzboog, Stuttgart, Germany, 1973.

[39] H.-P. Schwefel, "Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik", Diplomarbeit, Technische Universitaet, Berlin, Germany, 1965.

[40] H.-P. Schwefel, "Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie", Interdisciplinary Systems Research, vol. 26, Basel, Birkhaeuser, Basel, Switzerland, 1977.

[41] L. J. Fogel, A. J. Owens, and M. J. Walsh, Artificial Intelligence through Simulated Evolution, John Wiley and Sons, New York, USA, 1966.

[42] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms", Foundations of Genetic Algorithms, (G. J. E. Rawlins eds.), pp. 69-93, Morgan Kaufmann, San Mateo, CA, USA, 1991.

[43] A. Cardon, T. Galinho, J. P. Vacher, "Genetic algorithms using multi-objectives in a multi-agent system", Robotics and Autonomous Systems, vol. 33, no. 2-3, pp. 179-190, 2000.

[44] L. Bull and T. C. Fogarty, "Evolutionary Computing in Cooperative Multi-Agent Environments", AAAI Technical Report SS-96-01, 1996.

[45] L. Bull and T. C. Fogarty, "Evolutionary computing in multi-agent environments: Speciation and symbiogenesis", Parallel Problem Solving from Nature - PPSN IV, Lecture Notes in Computer Science, vol. 1141, Springer Berlin / Heidelberg, 1996.

[46] A.L. Nelson, E. Grant, G.J. Barlow, M. White, "Evolution of autonomous robot behaviors using relative competitive fitness", in Proceedings of the 2003 IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems (KIMAS 03) Modeling, Exploration, and Engineering Systems, Boston MA, Oct. 1-3, 2003, pp. 145-150.

[47] V. Trianni, R. Gross, T. Labella, E. Sahin, and M. Dorigo, "Evolving Aggregation Behaviors in a Swarm of Robots", in Advances in Artificial Life - Proceedings of the 7th European Conference on Artificial Life (ECAL) (W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, and J. Ziegler, eds.), vol. 2801 of Lecture Notes in Artificial Intelligence, pp. 865-874, Springer Verlag, Heidelberg, Germany, 2003.

[48] C. Yalcin, "Evolving aggregation behavior for robot swarms: A cost analysis for distinct fitness functions", in 23rd International Symposium on Computer and Information Sciences, ISCIS '08, pp. 1-4, Istanbul, Turkey, 2008.

[49] M. L. Stephens and M. B. Merx, "The Effect of agent control strategy on the performance of a DAI pursuit problem", in Proceedings of the 10th International Workshop on Distributed Artificial Intelligence, 1990.