A Regulatory Theory of Cortical Organization and Its Applications to Robotics

by

Jekanthan Thangavelautham

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Institute for Aerospace Studies
University of Toronto

Abstract

A Regulatory Theory of Cortical Organization and Its Applications to Robotics

Jekanthan Thangavelautham
<jekan.thangavelautham@utoronto.ca>

Doctor of Philosophy
Graduate Department of Institute for Aerospace Studies
University of Toronto
2008

Fundamental aspects of biologically-inspired regulatory mechanisms are considered in a robotics context, using artificial neural-network control systems . Regulatory mechanisms are used to control expression of genes, adaptation of form and behavior in organisms. Traditional neural network control architectures assume networks of neurons are fixed and are interconnected by wires. However, these architectures tend to be specified by a designer and are faced with several limitations that reduce scalability and tractability for tasks with larger search spaces. Traditional methods used to overcome these limitations with fixed network topologies are to provide more supervision by a designer. More supervision as shown does not guarantee improvement during training particularly when making incorrect assumptions for little known task domains. Biological organisms often do not require such external intervention (more supervision) and have self-organized through adaptation. Artificial neural tissues (ANT) addresses limitations with current neural-network architectures by modeling both wired interactions between neurons and wireless interactions through use of chemical diffusion fields. An evolutionary (Darwinian) selection process is used to 'breed' ANT controllers for a task at hand and the framework facilitates emergence of creative solutions since only a system goal function and a generic set of basis behaviours need be defined.

Regulatory mechanisms are formed dynamically within ANT through superpositioning of chemical diffusion fields from multiple sources and are used to *select* neuronal groups. Regulation drives competition and cooperation among neuronal groups and results in areas of specialization forming within the tissue. These regulatory mechanisms are also shown to increase tractability without requiring more supervision using a new statistical theory developed to predict performance characteristics of fixed network topologies. Simulations also confirm the significance of regulatory mechanisms in solving certain tasks found intractable for fixed network topologies. The framework also shows general improvement in training performance against existing fixed-topology neural network controllers for several robotic and control tasks. ANT controllers evolved in a low-fidelity simulation environment have been demonstrated for a number of tasks on hardware using groups of mobile robots and have given insight into self-organizing system. Evidence of sparse activity and use of decentralized, distributed functionality within ANT controller solutions are found consistent with observations from neurobiology.

All stable processes we shall predict. All unstable processes we shall control.

—John von Neumann

Acknowledgements

I wanted to first thank Professor Gabriele M.T. D'Eleuterio for his constant optimism, encouragement, enthusiasm and confidence he had in me. His open, free-thinking style of supervision helped me explore new and fascinating areas of interest in this long PhD journey, something, I am very thankful for. What truly kept me motivated was his will to set higher expectations, put on challenges and never give up even in the thickest of times. Many thanks for everything. You are a true inspiration, Gabe!

I am also very thankful for the tremendous effort put in by several of my colleagues at the lab including Alexander Smith, Nader Abu El Samid, Alexander Ho and Kenneth Law. They shared in this journey from Sudbury to Montreal. Their tireless efforts on the resource gathering and excavation projects through all those ups and downs, storms and all nighters contributed towards Chapter 4.

I would also very much like to thank Dale Boucher, NORCAT and Jim Richards, EVC. Both had bold vision and foresight to help nurture the ANT concept. They helped me gain invaluable understanding of the practicalities of what we were after when we were merely dabbling in theory. Their gracious financial support for the excavation and resource gathering projects, patience, understanding and unmatched expertise kept us going. Their unforgettable hospitality, open and free-thinking style of operation melded very well with the crew from UTIAS.

Furthermore, I would also like to thank Ernest Earon and Tim Barfoot for their invaluable insight and helpful discussions during the early years of my grad work. I would like to acknowledge Mustafa Mirza, Chris Skonieczny, Paul Grouchy and Luke Ng for their countless (often humorous) discussions on various topics in evolution, robotics, philosophy and many other subjects.

There are many others whom I would like to thank including Stephen Chee, Yinan Wang, Bernd Schimdt, Ling Xiao, Terence Fu and Raymond Oung. They provided invaluable service in the assembly and setup of various single and multirobot experiments throughout Chapters 3, 4 and 6. I would also like to thank my parents, my aunts and my uncle in this long journey. Their patience, understanding, guidance and support throughout this process was truly critical. Thanks also to Prof. Chris Damaren, Prof. Peter Grant, Prof. James Mills and Prof. Dario Floreano (EPFL) for providing insightful reviews and helpful comments.

Preface

Many parts of this dissertation have appeared elsewhere in peer-reviewed publications and a book chapter. Parts of Chapter 2 initially appeared in Thangavelautham and D'Eleuterio (2004) and was expanded into Thangavelautham and D'Eleuterio (2004b). Parts of Chapter 3 appeared in Thangavelautham and D'Eleuterio (2005). Elements of Chapter 3 and Chapter 4 appeared in Thangavelautham *et al.* (2007) and was expanded in Thangavelautham *et al.* (2007b). Work on excavation, elaborated in Chapter 4 was published in Thangavelautham *et al.* (2008). Elements of Chapter 2, 3 and 4 appear in a book chapter Thangavelautham, Barfoot and D'Eleuterio (2008). Elements of Chapter 5 appeared in Thangavelautham and D'Eleuterio (2007). Note that a number of papers contain some overlapping content and were intended for different audiences.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

In the survival of favoured individuals and races, during the constantly-recurring struggle for existence, we see a powerful and ever-acting form of selection.
—Charles Darwin

Chapter 1

INTRODUCTION

The question of how a brain is organized, how it operates remains one of the great scientific enigmas of modern time. At its foundation are the regulatory systems evident within most forms of life. These are nature's equivalent of a dynamic control system.

One must truly marvel at the functions of the nervous systems that exhibit computational capabilities in vision, memory, decision making/control, learning and that remains unmatched by current computer systems. Moreover, *engineering* these advanced features inherent in brains onto robotic control systems is highly desirable. It is of immense practical value to have robots act creatively, augment human intelligence in solving some of the great questions in science, be our guides in exploring/navigating the outer reaches of space and handle difficult organizational tasks without human intervention. Part and parcel of engineering artificial neural structures from a reductionist viewpoint is to determine the underlying principles (if any) evident in these biological organisms. The challenge from a philosophical perspective is to determine how a brain gives rise to a mind.

The theory of evolution remains an important means of explaining how such regulatory systems have arisen [131]. Analyzing these colossal achievements of evolution, one wonders in the reductionist sense, what are the underlying principles that govern these neurobiological structures? Reductionism has been very successful in the field of physics, where the works of Einstein, Newton and others have made key discoveries by identifying the underlying principles.

Reductionist thinking would outwardly appear to face challenges in modeling evolution. In particular, how is it possible to extract the essence of form and function in these vast, parallel systems? As Dawkins notes, understanding evolution is to analyze the products of improbabilities [35]. Increased structure and modular complexity of an organism does not always serve as a disadvantage over simpler organisms. Both strategies can coexist, or either one dominates over the other due to exploitation of different niches.

Yet these interactions within biological systems are governed by the fundamental laws of physics and hence there has to be a reconciliation of the reductionist physical framework in the process. From a materialist viewpoint, it must be possible to model biological systems from a bottom-up approach and recreate (*reengineer*) the myriad possibilities from biology. Hence, the mind and brain are one and the same, where all thoughts and mental actions maybe explained by physical phenomena. The path taken due to stochasticity inherent in an evolutionary process is but only one of many other possibilities, shaped and molded by both internal and external environmental factors. Fields such as Artificial Life have the benefit of asking these "what if" questions in a biological context.

It is of immense interest to determine the role of regulatory systems within adaptive processes, including development and learning. Regulatory systems are evident within most known forms of life including both prokaryotes and eukaryotes. Translation of the contents of a gene into a phenotype is a cyclic process, governed by gene regulation. Gene regulation determines the activation and inhibition of specific genes, forming parallelized protein networks, that is used to translate the genetic contents into a phenotype. Regulation at the gene level consists of molecular interaction resulting in formation and interaction of protein molecules, the building blocks of bigger structures. The interaction of these protein molecules is a competitive-cooperative process, where various protein molecules compete for resources such as binding sites and amino acids. The process on the whole is massively parallel, stochastic and decentralized.

Several biologists have argued that natural selection occurs not only among whole organisms but also within cell populations resident in an organism [98, 131]. Darwin [34] had also put forth similar ideas in his 'hypothesis of pangenesis,' where gemmules and pangenes are influenced by the diverse conditions of life of different tissues within a single metazoan body [131]. This area of research has gained focus in molecular cell biology, where there is much debate as to how cell differentiation, specialization and morphogenesis are triggered. One of the key factors necessary for Darwinian-type selection mechanisms is population variability. Variability occurs even within multicellular organisms, where cell differentiation mechanisms introduce errors in the development program, where gene regulatory networks activate or inhibit genes once critical development milestones or environmental factors are reached. Such variability enables unique combination of features to be selected.

A biological system can act functionally invariant to individual component variability, damage and noise. This is a testament to the robustness of the architecture, where individual cells have a limited lifespan, can be subjected to damage such as mutation, yet the impact on the system as a whole is limited. These properties of biological systems make it well suited for applications in control particularly under hostile environments

such as outer space.

Pauling among others theorized that the immune system is an instructional system [43]. The instructionist view of that time assumed an external *instruction* in the form of a template helped mould the shape of the antibody molecule. The purpose of the immune system is to identify and eliminate foreign pathogens, while leaving internal cells within the organism intact, thus being able to recognize self from not-self [23]. Rather, within the antibody molecules exists enormous variability and hence an antigen *selects* a particular antibody from this available repertoire [43]. These systems are categorized as *exploratory* or *selective*, where gene contents represent an enormous set of hardwired options (initially) that are then pruned/activated based on sensory stimulus using trial and error encounters. Added to this is the role of evolution in the process. Evolution has helped promulgate a constant cycle of adaptation, where populations of viruses and other foreign pathogens evade detection through innovation. The immune system has coevolved by gaining plasticity, allowing for identification and learning encounters with new pathogens.

Edelman [42], Jerne [80], Young [169] have put forth the idea that the higher functions in the brain are also composed of selective systems. According to Edelman [42],

> Selection implies that after ontogeny and early development, the brain contains cellular configurations that already respond discriminatingly to outside signals because of their genetically determined structures or because of epigenetic alterations that have occurred independently of the structure of outside signals. These signals serve merely to select among preexisting configurations of cell or cell groups in order to create an appropriate response.

In neuroscience, physiological theories have dominated the field, with detailed anatomy of the brain showing designated areas of specialization based on neuronal activity during specified experiments. As Gibson [57] notes, if the brain is indeed exploratory, then the anatomy alone will have little relevance in determining how the system operates. He notes, the change of activity in the brain is dynamic, with "this anatomical pattern changing from moment to moment."

Mountcastle's landmark discovery of a uniform columnar organization of neurons in the neocortex led him to believe that since the different brain lobes share a lot in common in terms of cortical structure, that they perform the same underlying operations, regardless of sensory inputs and outputs [114]. This radical view of cortical organization has drawn a lot more interest with more recent experimental evidence. In particular, transplantation of retinal fibers into the auditory cortex in newborn ferrets results in visual orientation modules emerging in the auditory cortex [140].

It should also be noted that not all brain functions are expected to be exploratory in nature. Feature-detector functions forming facial imprinting behaviours are thought to be hardwired [71] and involve a newborn infant (particularly birds) gathering an imprint of its parent(s) and apply a different value system to encounters with others. Both selectionist and instructionist views account for hardwired brain functions. By the selectionist argument, there exists a *repertoire* of hardwired brain functions that is selected based on sensory stimulus and development, while instructionist models lack this variability feature.

Constructivism is another theory that has gained recent prominence and assumes the process of brain development is incremental with increased physiological functionality being added through maturation [128]. Learning through a Hebbian-type process is seen as the primary means of adaptation. The role of nurture from the constructivist viewpoint is not clear. It is assumed the cortical structure is generic, and environmental encounters help to develop the various faculties. With respect to synaptic connections, constructivism predicts a steady growth in the number of synapses with progression in development, while selectionism predicts a pruning of synaptic connections. While selectionism relies on a Darwinian-like framework to explain dynamic process of neuron interactions. It is by current accounts a regressive process, where selection implies pruning and tuning from a wider array of repertoires through environmental stimuli. These repertoires are a product of the evolutionary process.

Evidence from biology indicates that the brain has greater density of synaptic connections between 2 and 4 months after birth, with a steady pruning of connections till puberty which favours a selectionist viewpoint although this remains controversial [128]. Selectionism also assumes that new functionality is acquired through exploratory selection on a repertoire of hard wired functions. This is in contrast to instructionist modes of thinking that rely on *a priori* instruction set or knowledge base. Instructionist models lack the need for competing functional repertoires. Instead environmental stimulus triggers predefined noncompeting sets of commands.

Ultimately each biological system is engineered through a process of Darwinian selection to survive and work in an integrated fashion in its environment. The question is how should this research be pursued? Should some of the theories of neural regulation presented in thesis be elucidated in a 'wet' environment with live brain cells? Current limitations in technology, particularly in the detection and analysis of nitric oxide at the cellular level prevent any meaningful study in neuroregulatory function using live cells. Thus, simulation facilities are the only known means of testing some of these ideas.

The next best thing would be to demonstrate the integrated capabilities of a biological model through embodiment in a physical system such as a robot. Embodiment serves as a major change in philosophy within robotics and has spawned the "Nouvelle AI" movement. Embodiment doesn't merely imply a physical realization of a robotic system, but also tries to merge the individual into its ecological environment.

The work outlined here fits into the Nouvelle AI line of thinking. Nouvelle AI, in contrast to Old AI is interested in the *emergence* of functionality and creativity rather than the development of elaborate human knowledge-based models and heuristics of the world. Nouvelle AI has shunned the *system symbol hypothesis* [142], where it is assumed intelligence operates on a system of symbols.

In contrast, the *physical grounding hypothesis* within Nouvelle AI replaces the model of the world with the world itself [21]. This simplification sidesteps the *frame problem* [107] that has plagued Old AI. The frame problem involves determining how best to form an internal representation and synchronize it with the external world. The barriers have been technological and task dependent, where computation, knowledge and resource limits reduce the rate of update. The perceptibility limit of each agent or robot restricts how

much data representation of the real world is possible at each instant. Within the Nouvelle AI line of thinking, the ability to map internal representations of an external world needs to *emerge* from interaction with the environment.

The assumption with Old AI is that there exists vast general rules and repositories of predetermined programs that quantify intelligence [93]. The controller merely needs to search for the matching predetermined program based on its current sensory input or execute preprogrammed symbolic operations. All of these factors limit the scalability of the Old AI framework beyond what it was programmed for. The logic-based heuristic systems have little or no plasticity. The system cannot be adapted or retuned to handle new circumstances unless it is manually reprogrammed to include additional functional modules.

In contrast, while selectionist systems account for the benefits of having vast repertoires of hardwired function, the process that results from the emergence of these functional repertoires is entirely different. With selectionist systems, the hardwired functions are gene specified and are a product of self-organization through natural selection. Thus, the repertoire of hardwired functions is biased towards survival of the organism in its ecological environment. With Old AI, the general rules and programs are determined based on *ad hoc* choices of a designer.

Adaptation within Nouvelle AI is consistent with evolution, where certain lower-level basis functionalities (behaviours) are already existent due to synthesis of interaction between the creature (a robot) and its ecological environment [21]. Nouvelle AI sees the need for embodiment rather than the need to maintain an internal model of the environment. Nouvelle AI is also similar to connectionism. Both approaches are fundamentally bottom-up and seek to explain or develop higher level behaviours based on composition of low-level behaviours. In contrast, traditional top-down implementations in AI assumes high-level functionality is independent of low-level dependencies or programming nuances. In Nouvelle AI, the specifics of implementation are seen as important, particularly with suitability or interaction within the real-world environment. The notion of intelligence within Nouvelle AI is entirely variable and context dependent.

However, there also exist some inconsistencies between a Nouvelle AI perspective and natural selection. The need to do away with a model of the world and replace it with the world itself does pose challenges in a reductionist, Darwinian framework. For a Darwinian framework, having a bounded set of environments is sufficient. It doesn't explicitly require *access* to something as general as the world itself. Choice of environment will have differential impact on an evolving artificial system. Next it should also be argued that artificial systems as with biological ones will likely have limited interactions with the whole environment. It is highly unlikely that an individual senses nor interacts with *all* the details that are embedded in the 'physical world.' From a Darwinian framework, sensing or interacting with *every* such detail is not necessary unless if it provides a survival advantage. Nevertheless, environmental conditions will have an impact on the evolving artificial system.

If what is sensed by an artificial system is bounded within certain limits (i.e. constraint imposed on what sensory capabilities can be evolved), which is often the case with engineering applications then this makes

simulation feasible. Therefore a simulation of an environment (a *simulacrum* of reality) [12] could also be used in place of the physical 'world' and remain indistinguishable due to the bounded limits of perceptibility and interaction of its participants (evolving artificial systems). The ability to build a *simulacrum* of reality that is statistically representative of the physical world and that cannot be distinguished by its participants can address the practical challenges of evolution but at the price of reduced detail. By doing so one may build this simulated reality in a computer and then 'fast forward' the effects of evolution. It should be noted that the use of simulated reality does not fall into the trap of Cartesian dualism [38], where a mind is assumed to be detached and working on a different set of principles from its physical body. For the simulacrum of reality to be complete, the brain also needs to be 'physically grounded' in the *simulation*. This approach also bypasses the *frame problem* because there is no need to synchronize a physical world with the contents of the simulated reality. Furthermore, some form of validation is necessary whereby artificial systems evolved in this simulacrum be grounded in the 'physical world.' This involves verifying whether the contents and events of the simulacrum is statistically representative of the physical world and that cannot be distinguished by its participants.

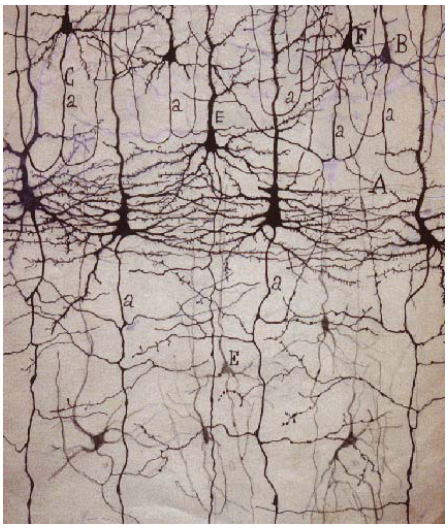


Figure 1.1: Drawings showing Golgi stained neurons of cat visual cortex circa 1921 (Textura de la corteza visual del gato. Archivos de Neurobiologa 2: 338-368). Obtained with permission thanks to Dr. Alberto Ferrús. Original deposited in the Historical Archive of the Cajal Institute. C.S.I.C. Madrid. Spain.

A bottom-up approach supported both in the Nouvelle AI line of thinking and connectionism emphasizes modularity of representation. The fact that a nervous system is modular is largely accepted and was born of the Neuron Doctrine. Santiago Ramón Cajal had initially proposed the Neuron Doctrine, in which neurons were considered as discrete cells that form the basic structural and functional units within the nervous system [83]. The more popular, competing theory of the day, the Reticular Theory described the brain as a single meshwork of fibers and was advocated by anatomist including Camillo Golgi. This is akin to the centralization vs. decentralization debate between Old and Nouvelle AI. Ultimately it was Golgi's staining

technique that conclusively confirmed a decentralized, cell theory of the nervous system (Figure 1.1) and both Golgi and Cajal shared the 1906 Nobel Prize in Physiology for their work.

However the question remains as to how such a modular neuronal framework is organized and performs its day to day functions? The original Neuron Doctrine that advocates fixed networks of synaptic connection tieing billions of cells is under severe strain [22]. Evidence of chemical (wireless) signaling between neurons and the importance of gene/protein regulatory functionality has further altered our understanding of the nervous system [22]. The work outlined in this thesis mirrors the difficulty with *traditional* connectionism. This work is particularly interested in the *evolvability* [87] of neural network-based architectures, the ability for genes to be heritable and selectable phenotypes, less susceptible to lethal mutations and produce novel traits with fewer mutations.

Motivated by the notion of *evolution*, this thesis considers the role of regulatory mechanisms in the organization of neural network control systems. Similar to the original Neuron Doctrine proposed by Cajal, traditional connectionism assumes network of neurons are fixed and interconnected solely by wires. These configurations are called fixed topology networks and reviewed in Chapter 2. However, fixed-network topologies are shown to face several limitations that reduce the scalability and tractability of these architectures for harder tasks presented in Chapter 3. In addition, these architectures tend to be specified by a designer, thus an incorrect *ad hoc* assumption may make training difficult. A common solution to overcoming this limitation is to provide more supervision, namely by explicitly encouraging particular behaviours to solve a task However, biological organisms do not always require such external intervention and have adapted through self-organization.

Neurons within biological nervous system have more recently been theorized to communicate wirelessly through diffusion of chemicals. However, diffusion of chemical stimuli from a single source forms a coarse, omnidirectional field. In contrast, coordinated chemical signaling by multiple sources can target a specific, fine volume. This enables neurons to selectively communicate with particular groups of neurons and ignore others through masking. Such functionality can be used to perform regulation by activating and inhibiting groups of neurons. Neural regulation as we show through analytical and empirical analysis can be used to overcome limited supervision in solving certain tasks. The biological plausibility of this theory is evaluated in Chapter 5.

The artificial neural tissue (ANT) architecture introduced in Chapter 3 models superpositioning of nitric oxide chemical concentration fields emitted by decision (nitrergic) neurons. The superpositioning of chemical concentration fields is modeled as coarse coding interactions. These coarse coding interactions allow for neural regulatory functionality through selection of groups of motor neurons based on sensory stimuli. The ANT architecture and its genotype to phenotype mapping system are shown to address several tractability concerns inherent within fixed topology neural network architectures.

The advantage of the ANT framework is posed in terms of the statistical theory introduced in Chapter 2 and through several evolutionary simulation experiments including on the double pole balancing, tile

formation, phototaxis and a sign-following task. ANT is shown to have improved evolutionary training performance over fixed topology networks for a wide range of tasks. In particular, the ANT architecture is shown to solve the sign-following task, which is incidently found to be intractable (empirically) for fixed network topologies that lack regulation. The ANT controller solutions are also validated on hardware for several of these tasks. It is theoretically shown that increased neutrality in a phenotype increases the probability of finding solutions. This is argued to be possible through use of neural regulatory systems. Further evidence for this theoretical result is obtained from ablation experiments performed on ANT functionality. Alternately, evolvability can be increased by introducing more supervision but this is argued to be an unsuitable *ad hoc* procedure particularly for little known task domains.

The ANT framework is extended to real-world multirobotic control tasks in Chapter 4. These include a resource gathering and excavation task with terrestrial and lunar applications. Controllers are evolved in simulation and validated on several hardware platforms. The controllers are analyzed in terms of scalability of system parameters and for robustness in case of component failures. In addition, the ANT framework is extended by evolving both system parameters like number of robots concurrently with the controllers.

In Chapter 6, the effects of increased structural variability in neuronal structure are compared using the ANT framework for two variants of the sign-following task. The neuron structure is described as a set of dynamic gene-protein interactions using a coarse coding framework. The coarse coding framework facilitates selection/arbitration of multiple internal representation within each neuron. This model is used to derive a developmental neuron that can concurrently represent multiple states and have memory functionality closer to its biological analogue. It is shown that this coarse coding gene-regulatory framework increases variability and this produce fitter, more robust ANT controller solutions.

While much of this work has been oriented towards robotics, it is hoped the work will also have direct implications in biology. These models can serve as means to test biological theories of cognition and control. The work presented here also has potential applications in software systems, particularly in the design of computer players for video games, context search, indexing and for computer security applications.

The actual science of logic is conversant at present only with things either certain, impossible, or entirely doubtful, none of which (fortunately) we have to reason on. Therefore the true logic for this world is the calculus of Probabilities, which takes account of the magnitude of the probability which is, or ought to be, in a reasonable man's mind.

—James Clerk Maxwell

Chapter 2

NEUROEVOLUTION AND TASK DECOMPOSITION

2.1 Introduction

A fundamental goal of machine learning is to develop learning algorithms that require limited supervision but can facilitate the discovery of novel solutions and better handle environmental uncertainties. A key objective for machine learning algorithms, therefore, is to perform task decomposition with limited supervision. Task decomposition involves partitioning/segmenting of a complex task into subtasks. Partitioning is expected to facilitate solving the simpler subtasks which in turn is combined to yield a solution to the overall task.

Consider social insects such as ants, bees and termites or more generally, multicellular organisms that form *synergistic* (one whose capabilities exceed the sum of its parts) multiagent systems without any centralized supervision. Insects societies consist of simple individuals but as a collective can solve complex tasks. For example, termites have evolved the ability to survive the harsh Australian outback by building towering nests with internal heating and cooling shafts [37]. Similarly, individual cells within a multicellular organism are imprecise, inefficient, unreliable and have limited lifespan, yet the organism as a whole is more robust to these limitations and can solve complex tasks.

Task decomposition in these systems is done through a process of self-organization. An external supervisor is not present to decompose a high-level task into a series of subtasks, rather insect societies have acquired the skills and low-level behaviours to accomplish these marvels of engineering through evolutionary adaptation. There lacks explicit top-down direction or guidance in performing task decomposition. Yet there is implicit guidance or biasing through Darwinian selection. A termite nest that can build a cathedral mound in the harsh Australian outback with internal heating and cooling shafts has a survival (fitness) advantage over a nest that lacks this capability.

In this chapter, we hope to visit this self-organization process in an artificial setting. Here instead of Darwinian selection, we shall be using rather simplified artificial processes, namely Evolutionary Algorithms (EAs), applied to robotic tasks. Only a global fitness function, sensory input scheme and generic allowable basis behaviours needs to be specified for the training techniques outlined in this chapter. The global fitness function is equivalent to the implicit biasing mechanism from biology and serves as the system goal.

Limited supervision makes it difficult for artificial evolutionary algorithms to solve increasingly complex robotics control tasks since it becomes harder to rank and select for incrementally better solutions for crossover and mutation resulting in premature search stagnation. This is known as the *bootstrap problem* [116] in evolutionary robotics. With limited supervision, the fitness function makes it difficult to distinguish between incrementally better solutions. Instead the system depends on 'bigger leaps' in fitness space (through sequence of mutations) for it to be distinguishable during selection. However increasingly bigger leaps are also more improbable hence resulting in search stagnation.

A common solution to the bootstrap problem is to decompose a complex task into a set of simpler tasks through supervised task decomposition. This often requires *a priori* information of the task and supervisor intervention in determining a suitable task decomposition strategy. The task becomes decomposed into a set of subtasks and a set of transitionary fitness (*shaping*) functions are assigned to select for individuals with traits to solve each subtask.

This form of *shaping* reduces diversity among potential solutions, since the transitionary fitness functions encourages particular known solution strategies. It should be noted that proper choice of transitionary fitness functions should reduce the number evolutionary evaluations needed to solve for a task. Yet in nature, biological systems can accomplish such a feat with no such external intervention.

For multirobot applications, the necessary local and global behaviours need to be known *a priori* to make task decomposition steps meaningful. We believe that for a multirobotic system, it is often easier to identify and quantify the system goal, but determining the necessary cooperative behaviours is often counterintuitive. Hence, the challenge is to come up with a scalable top-down controls approach that can perform self-organized task decomposition given a global goal.

A typical approach involves using fixed-topology monolithic neural network architectures. Although such architectures are easy to define and have gained popularity in the field, they face scalability issues using a global fitness function. Making the wrong choice in terms of selecting a network topology can inad-

vertantly increase the effects of *spatial crosstalk* where disruptive neurons interfere and drown out signals from feature-detecting neurons [81]. Crosstalk in combination with limited supervision is also expected to further accentuate the 'bootstrap problem' [116].

We introduce here a modular Emergent Task Decomposition Network (ETDN) architecture that requires limited supervision and can decompose a complex task into a set of simpler tasks through competition and self-organization. The ETDN is a connectionist (neural-network based) architecture trained using evolutionary algorithms, with only a prespecified global fitness function. The network is composed of *decision neurons* (mediates competition) and several *expert networks* (that compete for dominance). In this chapter, ETDNs are demonstrated on a multirobotic tiling pattern formation task (similar to a segment of the termite nest building task [37]). A monolithic Cellular Automata lookup table control architecture is shown unable (empirically) to solve more complex versions of the task due to the bootstrap problem [151].

This chapter also focuses on developing analytical methods to quantify the effects of spatial crosstalk (interference caused by parallel hidden neurons) within fixed topology neural networks. Strategies to limit the effects of spatial crosstalk have been one of the driving forces in developing scalable modular connectionist architecture that can perform task decomposition.

2.2 Background

Robot control problems as we have presented them in this thesis can be framed as a series of tasks. A machine learning algorithm is used to search for a suitable robot controller solution that solves for a particular user-defined task. For these tasks, we use Evolutionary Algorithms (EAs) to train artificial neural network controllers and this methodology is widely known as *Neuroevolution*. Artificial neural networks are simplified models of cell networks in the nervous system. They have three properties considered useful in machine learning, this includes ability to *generalize* (i.e., find patterns, classify data), have synaptic *plasticity* and are capable of *universal* approximations. Synaptic plasticity refers to the ability to alter synaptic connections permanently. Machine learning algorithms in turn are used to train neural networks by altering the synaptic weights. It is the ability to feed multiple signal sources via synapses to a single neuron that allows for generalization. Furthermore, it has been shown that a two-layer feed-forward neural network with an appropriate choice of activation function and with sufficient number of hidden neurons can be a universal approximator. This network can represent any continuous function within a desired accuracy [32].

The EA search process involves representing individual neural network controllers as a genome. A population of genomes undergoes an evaluation phase, in which all the individuals are evaluated for a task at hand according to a given fitness function. This is followed by a selection phase, where fitter individuals are chosen for crossover (reproduction) and mutation while unfit individuals are culled off. This process of evaluation and selection is repeated for a fixed number of generations. Mutation and crossover are used to alter the contents and functionality of a controller. Unlike backpropagation, where a learning function needs

to be smooth and differentiable in order to perform weight updates, EAs also allow for use of discrete and noncontinuous functions.

The key intention in this thesis is to develop a set of methodologies to design neural network control architectures that reduce the need for supervision and facilitate self organized task decomposition. Jordan, Jacobs and Barto [81] had shown the potential for using modular neural networks in performing task decomposition for vision tasks using backpropagation. Here we revisit the need for modularity with respect to task decomposition both in a biological and machine learning context followed by a detailed presentation of the Emergent Task Decomposition Network architecture and related work.

2.3 What is a Task?

Robot control problems as we have presented them in this thesis can be framed as a series of tasks. Before we define a task, we begin by introducing the concept of *actions*. The definition of an action presented here is based on the definitions developed by Fikes and Nilson [49], with further refinements from Allis and Koetsier [4]. According to Allis and Koetsier, "An action, when applied in a given world state results in a change of state", where it consists of pairing $< a^{del}, a^{add} >$ in which a^{del} is the delete set and a^{add} the add set. An action a is applicable in the current world state S if $a^{del} \subseteq S$. Thus application of action a_1 on S, $a_1(S)$, when applicable, results in the following state change $(S \setminus a_1^{del}) \cup a_1^{add}$. In otherwords, action a_1 in the delete set is removed from the current world state and a_1 in the add set is set within the current world state.

In addition, one special action known as the *null action* results in no change to the environment. There are number of competing definitions of a task. In one of these definitions, a task, T, is simply a sequence of actions $(a_1, a_2, \ldots, a_j)$ [66]. A further refinement of this definition has been proposed by Allis and Koetsier, where a task, T, is a finite sequence of non-null actions $a_1, \ldots, a_j$ and where an infinite number maybe null actions performed in finite time.

Crowley *et al.* [31] counters Allis and Koetsier's definitions by implying *states* as being the basic concept underpinning both actions and tasks. A *state* is a possible combination of predicates which are properties of the observed world. A graph of state transitions consisting of arcs represents actions. The definition of an action has common agreement between the works of Crowley *et al.* and Allis and Koetsier. Crowley *et al.* defines a task as the association of a current state to a goal state. The current state in this case is the state of the universe at present. To attain the goal state, one must perform a sequence of actions. A task according to Crowley *et al.* does not explicitly determine a sequence of user's action but is open ended and can be determined on the fly. Crowley *et al.* definition highlights the issue of whether a task is a specific sequence of actions or can be a generalized set.

We propose a further refinement of Allis and Koetsier's definition. In our definition, a task, T, consists of a finite set of non-null actions sequences in finite time that results in a change of state. $S^f(t*) =$

$T(a_1, \ldots, a_m, S^i(t))$, where n is a positive integer and $a_1, \ldots, a_m$ are a set of non-null actions, $S^i(t)$ is the initial state at time t, $S^f(t*)$, is the final state at time $t*$ and $\lim_{n\to\infty} \frac{t*-t}{n} = 0$. It should be noted that what is meant by finite time is a finite contigous time segement.

In addition, there exists *null Tasks*, $0T$, that consist of a finite set of non-null actions in finite time that results in no change in state. If $T(a_1, \ldots, a_m, S) = 0T$, then $T(a_1, \ldots, a_m, S(t)) = S(t*) = S(t)$, where $\lim_{n\to\infty} \frac{t*-t}{n} = 0$.

In these definitions, a task, T, in addition to the earlier definition may contain an infinite number of null actions. A finite set of action sequences accounts for a generalized set of actions being performed in serial, parallel or in combinations. Hence this definition does not imply one particular action sequence. Accounting for a generalized set allows for open ended, on the fly changes of user's action in completing a task and is consistent with the Crowley *et al.* definition.

We introduce a set of operators including a pair of brackets $()$ to indicate serial sequencing and $||$ to indicate parallel sequencing. Hence for example, a task, T, that is to be performed with actions a_1, a_2 and a_3 is denoted as $T(a_1, a_2, a_3, S)$ starting at state S. As an example, for this task, action a_1 is executed on state S followed by a_2 and a_3 in parallel thus $T(a_1, a_2, a_3, S) = a_2||a_3(a_1(S)) = a_3||a_2(a_1(S))$.

For parallel sequencing where actions a_1, a_2 are defined, this property always holds true: $a_1||a_2(S) = a_2||a_1(S)$. For serial sequencing, $(a_2(a_1(S)) = (a_1(a_2(S))$ only when actions a_1 and a_2 are of equal precedence or act independently on world state S.

Task Decomposition and Subtasks

Having defined a task, we shall define what we mean by task decomposition. Task decomposition involves partitioning/segmenting a complex task into a finite number of subtasks in order to simplify solving for the overall task [82]. Let $T = \hat{T}_1 = \left(\breve{T}_1, \breve{T}_2, \ldots, \breve{T}_n\right)$, where $\hat{T}_1$ is a non decomposed task equivalent to T and $\breve{T}_1$ is a subtask of T and $\breve{T}_1 \ldots \breve{T}_n$ is the ordered list of subtasks that in combination is equivalent to T.

With the 'divide and conquer' approach solving for the list of subtasks $\breve{T}_1, \ldots, \breve{T}_n$ individually in some order and combined to then be equivalent to T is expected to be easier to solve than solving $\hat{T}_1$ separately without segmentation. Thus task decomposition is expected to facilitate solving the simpler subtasks which in turn is combined to yield a solution to the overall task.

We impose the following condition on subtasks, namely each subtask needs to be completed in order to complete the overall task, thus if $T = \left(\breve{T}_1, \breve{T}_2, \ldots, \breve{T}_n\right) = \left(\breve{T}_1, \breve{T}_2, \ldots, \breve{T}_n, \breve{T}_{n+1}, \breve{T}_{n+2}\right)$ then $\breve{T}_{n+1}$ and $\breve{T}_{n+2}$ are not subtasks and $\breve{T}_{n+1}\breve{T}_{n+2} = 0T$. This added condition is imposed since individual subtasks result in a change of state, sequences of these subtasks could form null tasks. Hence for the task decomposition process to be meaningful, each subtask is necessary in completing the overall task. It should be clarified that task decomposition also does not necessarily imply decomposing a task into a specific sequence of subtasks or actions but could also imply a generalized set of subtask sequences.

2.3.1 Modularity in Biology

Having defined what we mean by a task and task decomposition, we are interested in looking at techniques used to facilitate task decomposition when evolving neural network controllers. It had been shown earlier that modularity offers functional advantages when training artificial neural networks. With biological neural systems, modularity has been identified at various taxonomic levels namely at the gene-protein, cellular and within multicellular ensembles of hypercolumns. One should ask, did modularity emerge because of physical constraints or is there explicit functional advantages to hierarchical modularity? In this chapter, most of the work presented imply modularity offers functional advantages consistent with Jordan, Jacobs and Barto [81].

The brain is known to be modular, based on its day-to-day pattern of activity and various physiological experiments. Ballard [7] argues that certain centers in the brain use coarse coding to represent multidimensional space and thus there are upper limits in the number of dimensions that can be represented by the cells within a given cortical area due limits on how many neurons can be packed into a given volume. Hence, different sets of dimensions need to be represented in different areas (partititioned), subscribing to a modular, decentralized organization of multidimensional space. This is analogous to the issue of spatial crosstalk, where the spatial organization of cells limit the representational capacity of the ensemble. With spatial crosstalk it is the wiring of many parallel neurons that results in disruptive interference and drown out signals from feature-detecting neurons, making training difficult.

Other ideas, particularly by Barlow [10], Ballard [7] and Cowey [28], that have gained popularity is the notion that the brain harbors competing or even multiple cooperative representation of signals collected from its environment. This implies that a modular interface needs to be in place to arbitrate between these representations. Geschwind and Galaburda [56] suggest competition between different networks of neurons in the brain drives certain networks to become specialized for a particular function.

Hawkins [70] based on Mountcastle's finding [114] notes the similarity in modular interface/structure of the visual, auditory and other cortex areas. He puts forward the idea that the brain consists of similar modular components that specialize based on their sensory inputs. Experiments on ferrets with visual connections rerouted to the auditory cortex at birth showed no noticeable visual/auditory impairment [140] further strengthing this hypothesis.

2.3.2 Network Scalability and Spatial Crosstalk

Having reviewed the role of modularity in a biological context, we wish to review how modularity offers functional advantages for artificial neural networks. A critical element in devising a modular approach to network organization is to address the issue of *spatial crosstalk* [81]. Spatial crosstalk involves parallel disruptive neurons interfering and drowning out signals from feature-detecting neurons making training more difficult. A means of helping reduce the effects of spatial crosstalk can also address how best to

scaleup the network topology for more difficult tasks.

Plaut and Hinton [126] argue that by limiting the number of hidden neuron connections to an output neuron reduces the effects of spatial crosstalk (Figure 2.1). Effects of spatial crosstalk was understood in the context of standard backpropogation type learning algorithms, where the output units of a network may provide conflicting error signals back to the hidden units.

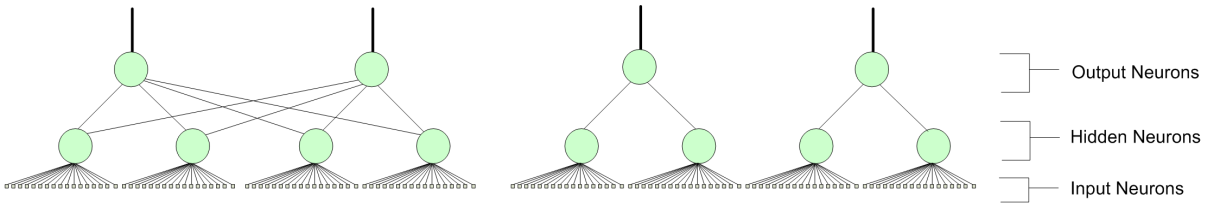


Figure 2.1: (Left) A typical neural network susceptible to spatial crosstalk using backpropagation. (Right) A modular network architecture not susceptible to spatial crosstalk using backpropagation.

Jordan, Jacobs and Barto [81] claimed the advantage of their network is that given a function can be decomposed into modular subcomponents, it is "generally easier to train a network into a set of simpler functions with an easy to switch mechanism between subcomponents." Jordan, Jacobs and Barto [81] argue that nonmodular networks that attempt to perform 'global generalization' are difficult to train, because functions would have to be found to match and balance features within the entire task space. It is argued that a means of arbitration among 'local specialist' functions requires less effort. Modular systems can take advantage of task decomposition, with each subtask being allocated to a different network.

However, this also raises the issue of over-decomposition. Over-decomposition of a task into subtasks may present a disadvantage, as this will counter one of the assumptions made in [81]. That original assumption was that the arbitration system would require less training than the expert networks particularly in the case of switching between two experts. But for an arbitrarily large number of expert networks this assumption may not hold true. In this situation there occurs a trade-off; what needs to be learned by the switching system may be more challenging than what needs to be learned by each expert network. In the work presented here, we consider this as a network scalability issue and compare a number of arbitration schemes.

In addition, given that the networks can be trained on one function, this may simplify the task of training on similar functions. This is widely known as the positive transfer of training. Alternatively if the system in trained on a dissimilar function, then there needs to be a 'locking mechanism' to prevent interference with existing learned traits (a case of spatial crosstalk). Compartmentalization facilitates specialization of subcomponents but can also be used to limit the interference effects of the compartments through a suitable switching mechanism.

2.4 Method

Some of the control system architectures presented in this section consist of artificial neural networks trained using evolutionary algorithms. We use a fixed network topology, with a one-to-one genotype-to-phenotype mapping (Figure 2.2). An integer encoding scheme is used to describe the weights, biases/thresholds and choice of activation function. With a fixed-network topology, the network structure (i.e., number of layers, position of neuron relative to other neurons in a graph) is not encoded in the genome. Some of the advantages of Neuroevolution, over gradient descent and other classical neural-network learning techniques is that neither the training function nor the neuron activation functions need be smooth nor differentiable and solutions tend to be more robust against local minima. In addition, unlike traditional machine learning algorithms, EAs could be used to select suitable activation functions, apart from perturbing the weight during training (as shown with our modular neuron architecture).

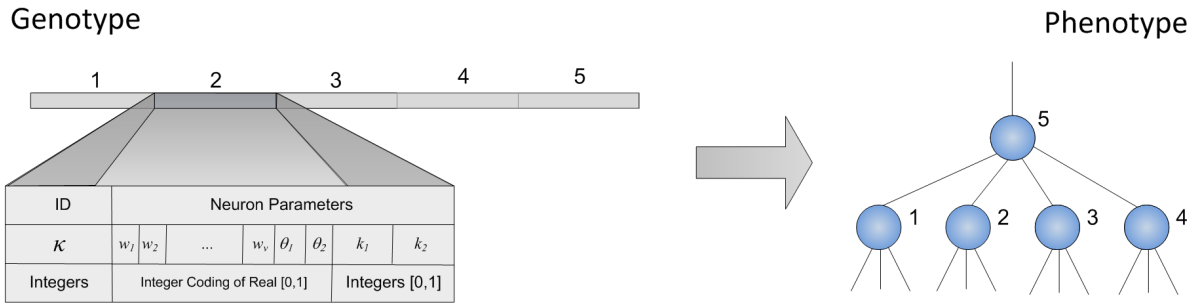


Figure 2.2: A one-to-one genotype-to-phenotype mapping used to construct the neural network controller. The network topology is prespecified by the experimenter.

2.4.1 Emergent Task Decomposition Network Architecture

We propose an Emergent Task Decomposition Network (ETDN) architecture that consists of a set of decision neurons that mediate competition and a set of expert network modules that compete for dominance as shown in Figure 2.3 (right). This architecture exploits network modularity, evolutionary competition, specialization and *functional neutrality* to facilitate emergent (self-organized) task decomposition

Unlike traditional machine learning methods where handcrafted learning functions are used to train the decision and expert networks separately, our architecture requires only a *global* fitness function. The intent is for the architecture to evolve the ability to decompose a complex task into a set of simpler tasks with limited supervision. The decision neuron is connected to all the sensory input and is used to select an expert network based on the output state. In turn, the output from the selected expert network triggers a predefined basis behaviour (i.e. move forward or turn right).

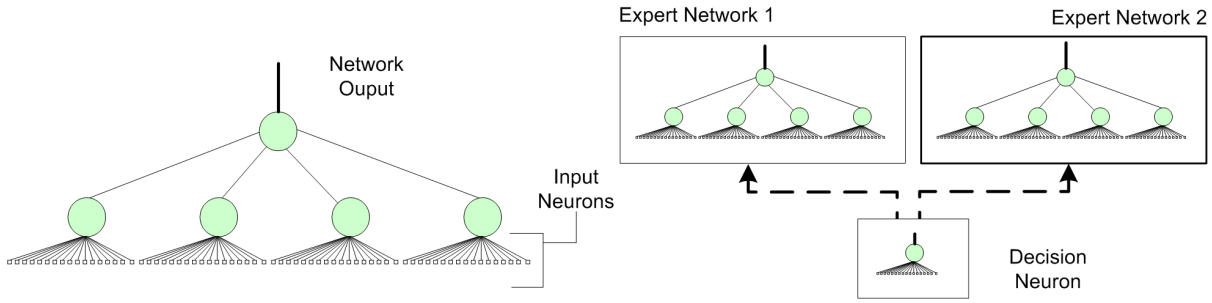


Figure 2.3: (Left) An example of a standard neural network used in our experiments. (Right) ETDN architecture consisting of a decision neuron that arbitrates between 2 expert networks.

2.4.2 Extending Emergent Task Decomposition

Our proposed ETDN architecture, unlike previous work in the field of evolutionary robotics, could be generalized to n_E number of expert networks. Here we discuss two proposed extensions to the ETDN architecture, namely the Binary Relative Lookup (BRL) architecture and Binary Decision Tree (BDT) architecture.

The Binary Relative Lookup (BRL) architecture consists of a set of n_d non-interconnected decision neurons that arbitrate between 2^{n_d} expert networks. Starting from left to right, each additional decision neuron determines the specific grouping of expert networks relative to the selected group. Since the decision neurons are not interconnected as shown Figure 2.4 (right), this architecture is well suited for parallel implementation.

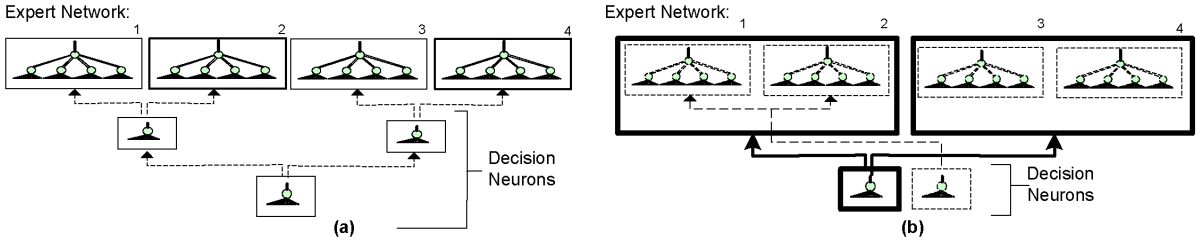


Figure 2.4: (Left) BDT Architecture with 4 expert networks and 3 decision neurons. (Right) BRL Architecture with 4 expert networks and 2 decision neurons.

The Binary Decision Tree (BDT) architecture as shown in Figure 2.4 (left) could be represented as a binary tree where the tree nodes consist of decision neurons and the leaves consist of expert networks. For this architecture, n_d decision neurons arbitrate between $n_d + 1$ expert networks. The tree is traversed by starting from the root and computing decisions neurons along each selected branch node until an expert network is selected. For both architectures the computational cost of the decision neurons, $C_d \propto \log_2 n_E$ since 2^{n_d} decision neurons need to be evaluated in both architectures in order to select an expert network.

2.4.3 Modular Neurons

Our intention is to develop a compact modular neuron architecture for implementation on hardware, where a single neuron could be used to simulate AND, OR, NOT and XOR functions. The common theme in the development of this neuron model is modularity. With the switch of a single genetic parameter, namely the entry that encodes for the choice of an activation function, a neuron can switch functionality with the reuse of the existing modular set of weights.

This assumption is confirmed later in this chapter, that a compact yet sufficiently modular (functional) neuron will speed up evolutionary training since this will reduce the need to tune more neurons in the hidden layers. Increased number of parallel hidden neurons are a source of spatial crosstalk as discussed earlier. At the same time, over-reduction of the number of neurons in the hidden layer is also expected to reduce performance since there may not be enough neurons to perform the necessary feature recognition. Hence there exists a trade-off between the two extreme scenarios when seeking to maximize network performance (see Section 2.8 for further details).

For the modular activation function, the EAs are used to train the weights, threshold parameters and choice of activation function for each neuron. The choice of activation function is determined by two binary parameters, $k_1, k_2 \in \{0, 1\}$ contained within each neuron gene. The inputs and outputs from the modular threshold neurons are binary states. The modular neuron could assume one of four different activation functions listed below:

$$
\begin{aligned}
\phi_1: \quad & s_{out} = \begin{cases} 0, & \text{if } p(x) \le \theta_1 \\ 1, & \text{if } p(x) > \theta_1 \end{cases} \qquad
\phi_3: \quad s_{out} = \begin{cases} 0, & \text{if } \theta_2 < p(x) < \theta_1 \\ 1, & \text{otherwise} \end{cases} \\
\phi_2: \quad & s_{out} = \begin{cases} 0, & \text{if } p(x) \ge \theta_2 \\ 1, & \text{if } p(x) < \theta_2 \end{cases} \qquad
\phi_4: \quad s_{out} = \begin{cases} 0, & \text{if } p(x) > 0.5 \\ \text{rand(0,1)}, & \text{if } p(x) = 0.5 \\ 1, & \text{if } p(x) < 0.5 \end{cases}
\end{aligned} \tag{2.1}
$$

The threshold functions shown in (2.1) can summarized in a single analytical expression as:

$$\phi = (1 - k_1)[(1 - k_2)\phi_1 + k_2\phi_2] + k_1[(1 - k_2)\phi_3 + k_2\phi_4] \tag{2.2}$$

Each neuron outputs one of two states $s \in \{0, 1\}$, where $\theta_1, \theta_2 \in [0, 1]$ and are the threshold parameters and

$$p(x) = \frac{\sum\limits_i w_i x_i}{\sum\limits_i x_i} \tag{2.3}$$

$w_i \in [0, 1]$ is a neuron weight and $x_i \in \{0, 1\}$ is an element of the active input state vector.

2.5 Related Work

Dorigo and Colombetti [39] introduced the idea of *incremental learning* or *shaping* in classifier systems where a robot learns a simplified version of the task and the task complexity is progressively increased by modifying the learning function. Stone and Veloso [150] developed a task decomposition scheme known as *layered learning*, where the learning function for a complex task (playing soccer) is partitioned into a set of simpler learning functions (corresponding to subtasks) and learned sequentially. These traditional task decomposition techniques expect a supervisor to have domain knowledge of a task at hand to determine how best to incrementally shape or layer the learning process. This is particularly difficult to implement in the multirobotics domain as the necessary local and the resulting global (collective) behaviours need to be known *a priori* for the task decomposition process to be meaningful.

Jordan, Jacobs and Barto [81] introduced an automated task decomposition technique now widely known as the Mixture of Experts, using a modular neural network architecture for the 'what' and 'where' vision task. The architecture consists of a *decision network* (mediates competition) and two *expert networks* (explicitly specialized for predefined subtasks). Mediation involves weighing the outputs from both expert networks as shown in Figure 2.5.

The Mixture of Experts architecture was designed to addresses the issue of *spatial crosstalk*. Spatial crosstalk occurs when parallel neurons provide conflicting signals to neurons in the next layer resulting in longer training times and poor network performance [81]. To address this issue, networks are subdivided into modules of experts. These modules include fewer neurons at the output layer thus reducing the source of neuron-to-neuron interference during backpropagation training. Figure 2.1 (left) is a network suspectible to spatial crosstalk during backpropagation training, while the one on the right, is not suspectible to spatial crosstalk since the output neurons are separated and cannot provide conflicting signals when fed back to the hidden layer. The ETDN architectures imposes a feed-forward topology with the use of Evolutionary Algorithms (EAs) for training and hence, the source of conflicting signals is from the neurons in the hidden layer.

The expert networks and the decision networks are also trained separately according to a set of learning algorithms. Supervisor designed objective functions are devised for training each network separately. The process of devising these objective functions relies on user knowledge of the task at hand and can implicitly bias for a particular task decomposition strategy. This is in contrast to the ETDN architectures presented here, in which both the expert and decision networks are evolved together with functional roles being a product of self-organization.

Hierarchical Mixture of Experts Model [82], is a further expansion of the early model, with a binary tree consisting of gating networks at each node. The output from the architecture is a mixture of outputs from the experts at each node and experts at the leaves.

The Mixture of Experts architecture involves use of broadly tuned feature representation distributed

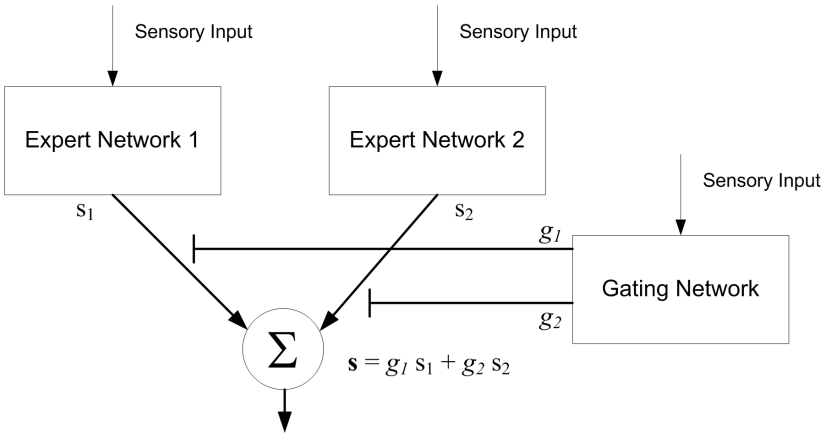


Figure 2.5: The Mixture of Experts architecture.

among the expert networks. The gating networks merely arbitrates the weighing parameters between the experts. In contrast, the Product of Experts model [73] uses finer tuned expert, which provides improved efficiency for higher-dimensional data such as facial recognition task, involving 35 dimensions and can simultaneously satisfy many low dimensional constraints. Each expert model specializes itself and can satisfy just one constraint. In addition each model can be constrained to different dimension, with the product spanning all of the dimensions in the data.

The Mixture of Experts is less efficient for higher dimensional search spaces as each modular component attempts to correlate features from a broad number of dimensions [73]. In contrast, experts within the Product of Experts model specialize and span only a few dimensions. The limitation within the mixture and product models is that both have preconceived data representation schemes in which experts are predetermined as to how they interact (either a gated linear model or a product model). It is not always apparent which preconceived scheme is appropriate nor even necessary for a particular task. A more generic system, where the interactive factors between the experts is not prespecified as in the ETDN model has advantages, since it's not task specific. Less knowledge is required in picking an appropriate scheme and this facilitates emergence of creative solutions for a problem at hand.

Darwen and Yao [33], Liu, Yao and Higuchi [100] developed a hybrid automated approach to task decomposition. Localized training is done using backpropagation or negative correlation learning and in turn EAs are used to modify training algorithm parameters. Rather than using a gating neuron, a prespecified gating algorithm such as voting is used. Diversity amongst the expert modules is protected through fitness sharing based speciation techniques.

Algorithms such as ESP (Enforced Subpopulations) [59] and SANE (Symbiotic Adaptive Neuro-Evolution) [113] select for individual neurons, within a fixed neural network topology. In contrast to training the whole network architecture as with ETDN, ESP uses separate subpopulations to select neurons for each placeholder within the neural network control system [164, 59, 61] while SANE relies on a common population

[113]. Unfortunately, this approach is susceptible to premature convergence in some cases, due to use of isolated neuron population and a 'localist' approach to neuron selection [59]. Task decomposition using ESP has consisted of devising a binary tree by hand (with branches representing subtasks) and setting up networks with a predetermined topology for the leaves. The networks are trained individually through a layered learning processes [150] until the controller can accomplish the overall task.

Nolfi [115] has introduced an *emergent modular architecture*, where decision neurons arbitrate individual motor control neurons for the garbage collection task [116]. This architecture requires fewer genetic evaluations and produces better solutions than other comparable networks including multilayered feedforward and recurrent architectures.

There are some key difference between our ETDN architecture and Nolfi's *emergent modular* architecture. In his experiments, selection was among two competing motor neurons rather than whole networks. In addition, there is a one-to-one mapping between his decision neurons and control neurons, where each decision neuron votes for a single motor neuron. This implementation is applicable for simpler problems, where individual neuron responses are sufficient for a solution. Using our ETDN architecture (Binary Relative Lookup variant), arbitration could be extended to $2n_E$ expert networks.

2.6 Application: Tiling Pattern Formation Task

The tiling pattern formation task [151] involves redistributing objects (blocks) piled up in a 2-D world into a desired tiling structure (see Figure 2.6). The robots need to come to a consensus and form one 'perfect' tiling pattern. This task is similar to a segment of the termite nest construction task that involves redistributing pheromone filled pellets on the nest floor [37]. Once the pheromone pellets are uniformly distributed, the termites use the pellets as markers for constructing load balancing pillars to support the nest structure.

More important, the tiling pattern formation task could be decomposed into a number of potential subtasks consisting of emergent behaviours. These may include foraging for objects (blocks), redistributing block piles, arranging blocks in the desired tiling structure locally, merging local lattice structures, reaching a collective consensus and finding/correcting mistakes in the lattice structure. Instead with this controls approach we are interested in seeing solutions to these subtasks emerge through self organization. Inspired by nature, we are interested in evolving homogeneous decentralized controllers (similar to a nest of termites) for the task [90]. Decentralized control offers some inherent advantages including the ability to scale up to a larger problem size. Task complexity is dependent on the intended tile spacing as more sensor data would be required to construct a 'wider' tiling pattern.

Shannon's entropy function has been shown to be a suitable fitness function for the tiling formation task [151]. The test area spans $M \times M$ squares and is divided into J $l \times l$ cells, A_j (Figure 2.7), where the fitness value, $f_{i,x,y}$, for one set of initial condition i, after T discrete time steps, with cells shifted x squares in the x-direction and y squares in the y-direction is given as follows:

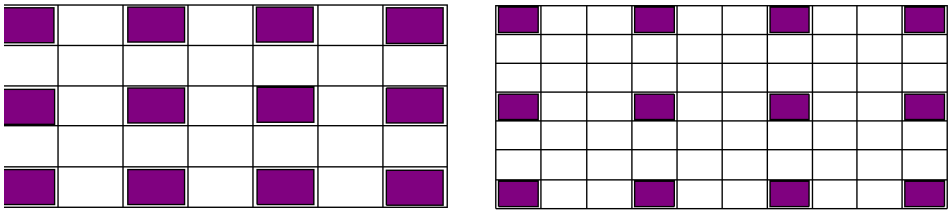


Figure 2.6: The 2×2 tiling pattern (left) and 3×3 tiling pattern (right).

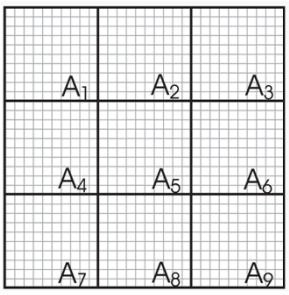


Figure 2.7: Partition of the grid world into bins, A_j for the fitness calculations.

$$f_{i,x,y} = \frac{\sum_{j=1}^{J} p_j \ln p_j}{\ln J} \tag{2.4}$$

where i is an index over a set of initial conditions :

$$p_j = \frac{n(A_j)}{\sum_{j=1}^{J} n(A_j)} \tag{2.5}$$

and where $n(A_j)$ is the number of blocks in cell A_j. To encourage the desired tiling pattern, the fitness function is applied by shifting (and wrapping around) the cells a maximum of $l-1$ squares and the total fitness is calculated as follows:

$$f_i = \frac{\sum_{y=0}^{l-1}(\sum_{x=0}^{l-1} f_{i,x,y})}{l^2}. \tag{2.6}$$

When the blocks are uniformly distributed over J cells, according to the desired tiling pattern, we have $f_i = 1$ (a successful epoch).

2.6.1 Simulated Robot Model

The robots are modelled as modified Khepera^{TM} robots equipped with a gripper turret. We have developed a fast 2-D grid world simulator for our evolutionary experiments and we have verified this simulator using Webots (Figure 2.11). For the 2×2 tiling pattern formation task, each robot can detect blocks and other robots in the 5 squares as shown in Figure 2.8 (left). For the 3×3 tiling pattern formation task, the robots can detect objects in 7 surrounding squares as shown. The output state from the robot controller activates one of two predefined basis behaviours, namely **Move** and **Manipulate Object** [151].

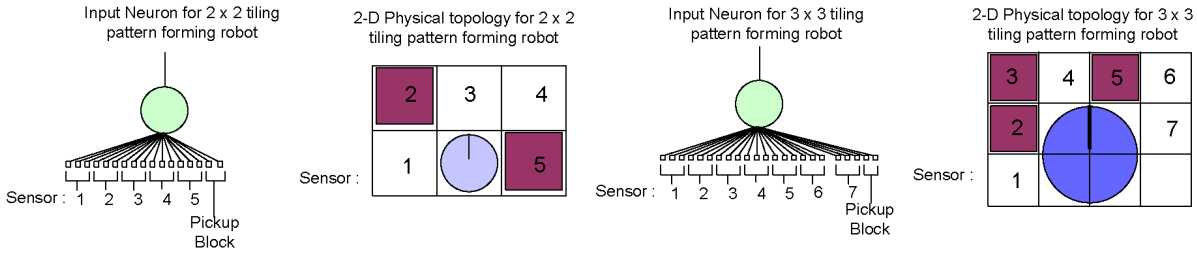


Figure 2.8: Input layer neuron and physical topology for the 2×2 (left) and 3×3 (right) tiling pattern forming robots. The input layer is fully connected to all the sensory input states as shown.

2.7 Experiments

The evolutionary performance of various control systems architectures are compared for the tiling pattern formation task (see Figure 2.9). Through experimentation, we found the best standard neural network (without using ETDNs) for the tiling pattern formation task as shown in Figure 2.3 (left) using a modular activation function (2.2). We use this network as our expert network module for the ETDN architectures. The fitness for each initial condition is evaluated after $T = 3000$ timesteps for a 11×11 world (2×2 task) or 16×16 world (3×3 task) with 11 robots and 36 blocks and is averaged over 15 different initial conditions per individual controller.

In our simulations, the EA population size is $P = 100$, number of generations $G = 200$, crossover probability $p_c = 0.7$, mutation probability $p_m = 0.025$ and tournament size of $0.06p_c$ (for tournament selection). Tournament selection involves randomly selecting individuals from the population into a tournament group, with the number of individuals determined by the tournament size. The fittest individual from a pair of tournament groups is selected for mating. The probability of parent genomes being interchanged using crossover to form a child genome is 70% otherwise the genome from a single parent is copied over to form the child genome. Mutation is applied on the child genome according to the mutation probability per gene site.

2.8 Results and Discussion

The experiments with the tiling formation tasks shows both the advantages and limitations with using neural network architectures. For the 2×2 tiling pattern formation task, the lookup table approach evolves desired solutions with fewer genetic evaluation than the neural network architectures (see Figure 2.9, left). This suggests that ETDNs and neural networks may not be the most efficient strategy for smaller search spaces. The advantage of a neural network approach is that the entries within a lookup table could be generalized however this also requires dealing with neuron-to-neuron interactions, namely spatial crosstalk. Through generalization of sensory inputs, the effective sensory search space is shrunk. For example, the search space maybe divided into situations where another agent is present within sensory range or not. A feature detector

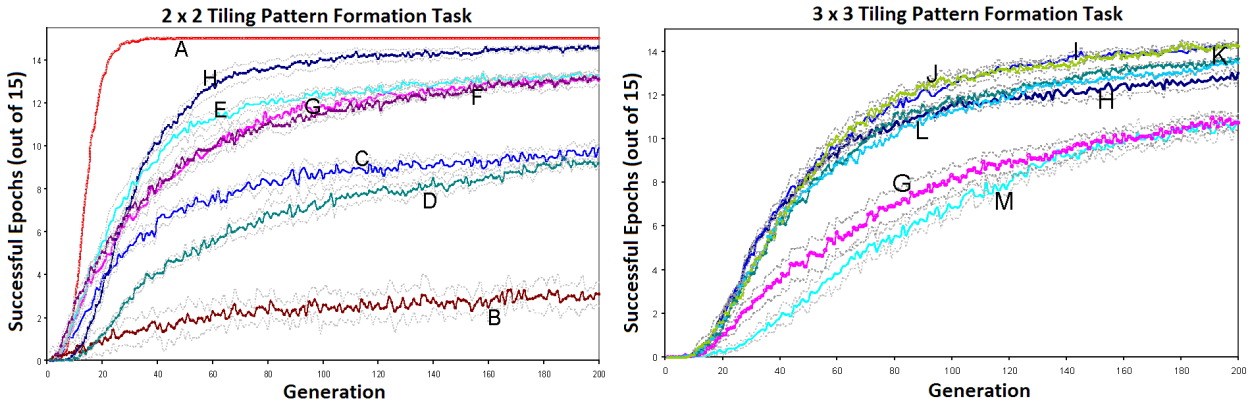
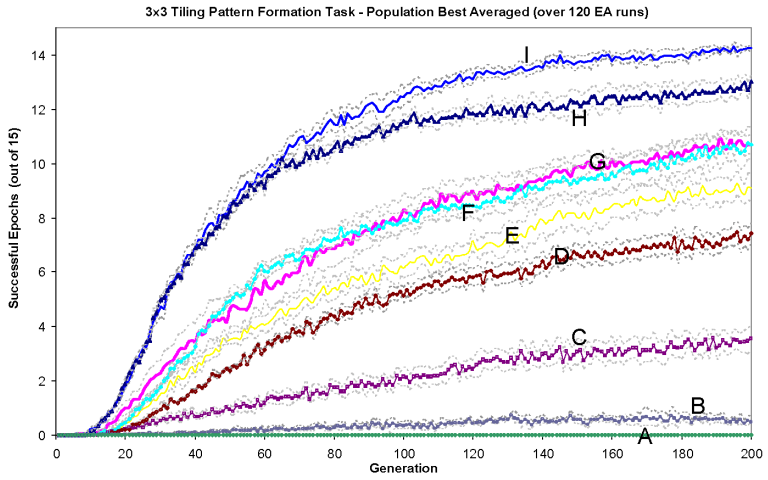


Figure 2.9: Evolutionary performance comparison, 2×2 (left), 3×3 (right, bottom) tiling pattern formation task, population best averaged over 120 EA runs. **(A)** Look-up Table, **(B)** ESP (using Standard Neural Net), **(C)** Standard Neural Net (Sigmoid), **(D)** ETDN (Exp. Net 2, Sigmoid), **(E)** Standard Neural Net. (Threshold), **(F)** ETDN (2 Exp. Nets, Threshold), **(G)** Standard Neural Net (Modular), **(H)** ETDN (2 Exp. Nets, Modular), **(I)** BRL (16 Exp. Nets, Modular), **(J)** BRL (32 Exp. Nets, Modular), **(K)** BRL (8 Exp. Nets, Modular), **(L)** BRL (4 Exp. Nets, Modular), **(M)** BDT (4 Exp. Nets, Modular). Standard deviation is shown in gray.



neuron tuned to obstacle detection can group situations where another agent is present.

In the case of the 2×2 tiling formation task, a neuron needs to detect another agent in 5 different squares. Let us assume the ability detect an obstacle is encoded in a single neuron. If this 'obstacle detection neuron' can trigger in response to any of these five situations, then a combination of these scenarios is also expected to the trigger the 'obstacle detection neuron', provided the signal stays the same or is further amplified. However for the obstacle detection feature neuron to correctly trigger an *obstacle avoidance behaviour* at the output layer, spurious (disruptive) signals from other neurons need to be ignored.

In contrast, with a lookup table approach, the obstacle avoidance behaviour needs to be correctly specified for each unique combination of sensory inputs in the lookup table, where for the 2×2 case, its 2^5 entries. However each entry is independent, hence an incorrect response to one set of sensory input does not carry

over to another instance. Hence what is apparent is a trade-off, where for smaller search spaces, the effect of neuron-to-neuron interference (spatial crosstalk) slows down convergence to a solution. Spatial crosstalk (see Figure 2.1) in the feed-forward case occurs amongst parallel neurons competing to signal neurons in the next layer. With the 2×2 lookup table controller, this issue is not present since each reaction rule (entry within table) is independent. Our conventional ETDN architecture consisting of a single threshold activation function evolves slower than the nonemergent architectures.

The ETDN architectures in contrast to regular neural networks include an additional 'overhead,' since the evolutionary training performance is dependent on the evolution of both expert networks and decision neurons resulting in slower performance for simpler tasks. However the BRL architecture that combines our modular neuron architecture outperforms all other network architectures tested in Chapter 2 (Figure 2.9). The performance of the modular neurons appears to offset the 'overhead' of the bigger ETDN architectures. A 'richer' choice of activation function is hypothesized to improve the ability of the decision neurons to switch between suitable expert networks.

For the more difficult 3×3 tiling formation task, a lookup table architecture is unable to find a desired solution [151] (see Figure 2.9 right). The resulting search space is significantly larger (owing to increased number of sensors required), 2^{4374} potential solutions for the 3×3 tiling formation task compared to 2^{486} for the 2×2 version. With a larger search space, this architecture falls victim to the *bootstrap problem*, since EAs are unable to find an incrementally better solution during the early phase of evolution. Increased mutation doesn't appear to help either, since higher rates of mutation can damage newly acquired traits.

The ESP algorithm show lower evolutionary training performance than the other neural network architectures for both tiling formation tasks. It is suspected that evolving individual neurons using isolated populations may not be the most effective strategy in facilitating task decomposition for this task. When each population of neurons are evolved separately, there is no mechanism in place to arbitrate between the population of neurons to remain diverse. Hence the same feature detector neuron maybe evolved within multiple isolated populations reducing the search efficiency or worse, resulting in premature search stagnation.

The ETDNs outperform standard neural network architectures for the more complex 3×3 tiling formation task (regardless of the activation function used). Analysis of a typical solution (for ETDN with 2 expert nets) suggests the expert networks have specialized since the fitness performance of the networks evaluated in isolation is quite low as shown in Figure 2.10 (left). It appears the decision neuron arbitrate among the expert networks not according to 'recognizable' distal behaviours but as a set of emergent proximal behaviours (organized according to proximity in sensor space) [115].

Although conventional network architectures can perform task decomposition, (evident from solutions to the 3×3 tiling formation task), it is more prone to *spatial crosstalk* [81] resulting in slower training performance. For the ETDNs, it appears, the task is distributed over several finely tuned expert networks resulting in fewer hidden neurons being used as feature detectors within each network, thus reducing the overall effect of spatial crosstalk. The decision neurons within the BRL architecture each act 'independently'

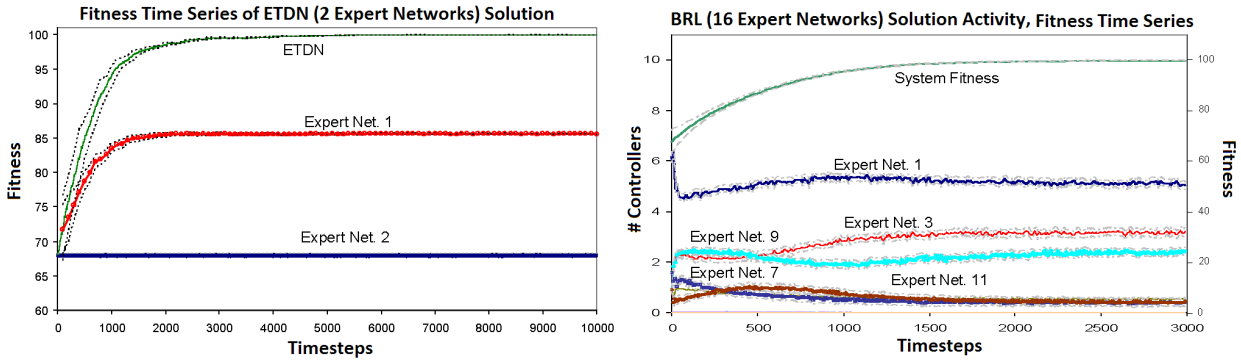


Figure 2.10: (Left) Fitness of individual expert networks and the entire control system (ETDN, 2 expert nets.) averaged over 1000 simulations. (Right) Expert Network activity and overall system fitness for a BRL (with 16 expert networks).

Table 2.1: Simulation results of success rate among population best (generation 200) scaled up to a 100×100 world (76 robots, 1156 blocks)

Architecture	% Successful Epochs	% Batch Deviation
Standard Neural Net.(Sigmoid)	0.0	-
Standard Neural Net.(Modular)	27.0	0.45
ETDN (2 Expert Nets., Modular)	44.5	1.5
BRL (16 Expert Nets., Modular)	70.5	0.63

when selecting an expert network through a superpositional selection process. In contrast, with the BDT architecture, the root node is used to select the next branch in the tree until a leaf is reached. We suspect this characteristic, in addition to the fewer number of decision neurons for the BRL network improves the ability to select suitable expert networks. This is evident from Figure 2.9 (right), where the BDT with 4 expert networks show reduced evolutionary training performance to a comparable BRL network.

Much like biological nervous systems, the larger BRL architecture outperformed (or performed as fast as) the smaller ones after about 80 generations as shown in Figure 2.9 (bottom). By increasing the number of expert networks, competition among candidate expert networks is further increased thus improving the chance of finding a desired solution. However, as the number of expert networks is increased (beyond 16), the relative improvement in training performance is minimal for this particular task.

2.8.1 Evidence for Task Decomposition

Figure 2.10 and 2.9 show evidence of task decomposition. From the incremental evolution of additional functionality in solving the multirobot tiling formation, it is evident that the controllers are performing task decomposition. As had been defined earlier, a task $T = \hat{T}_1 = \left(\breve{T}_1, \breve{T}_2, \ldots, \breve{T}_n\right)$ can represented as one non decomposed task, $\hat{T}_1$ or an ordered set of subtasks $\left(\breve{T}_1, \breve{T}_2, \ldots, \breve{T}_n\right)$. For the tiling formation task, the

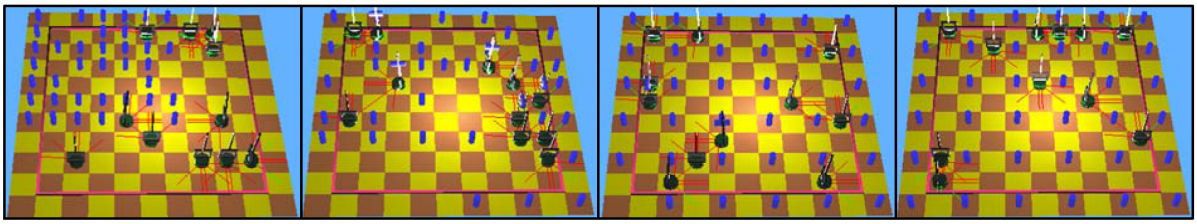


Figure 2.11: Simulation snapshots taken after 0, 100, 400 and 410 timesteps using a evolved solution for the 2×2 tiling formation task on a 11×11 world (11 robots, 36 blocks). The robots reach a consensus and form a 'perfect' tiling pattern after 410 timesteps.

controllers are evaluated for different initial conditions. The system reaches a fitness, $f_i = 1$ once it has solved the task for a particular set of initial conditions. It is argued that each set of initial conditions unevenly bias for certain set of subtasks over the general ordered set of subtasks required to complete the *overall* task. Hence with the appropriate choice of a fitness function, the system has the opportunity to tune individual subtasks at a time instead of the *overall* task all at once. When a controller is successful in solving the task for another initial condition, it gains a fitness advantage over those that do not have this capability. Since the controllers make incremental improvement toward solving the task instead of doing so in one generation, this shows evidence of the controller using task decomposition.

Furthermore Figure 2.10 (left) show that the controller functionality is distributed among more than one expert network. This confirms that the decision neuron is helping to arbitrate between the expert networks and therefore has decomposed a task into substasks among the expert networks. It should also be noted that these subtasks are not distal as mentioned earlier.

Figure 2.10 (right) shows that the individuals play different roles in completing the task. Evidence from Figure 2.11 show some robots are involved in carrying/distributing the blocks, while others are involved in searching/correcting existing blocks placements. This shows multiple roles being assumed by the collective, more commonly known as 'role assignment.' However since the controllers are homogenous and decentralized, the individuals are not constrained to one role, instead they may transition from one role to another based on sensory input.

2.8.2 Scalability

The larger BRL network solutions evolve the ability to limit the number of expert networks used (see Figure 3.19 (right) for a typical solution). The other expert networks remain dormant throughout the simulations. The ability to limit the number of expert networks used for task decomposition is particularly advantageous, as 'over segmenting' the task into subtasks over more 'expert networks' (and associated decision neurons) involves tuning more parameters than is necessary thus resulting in slower evolutionary training performance.

One of the main advantages of using a decentralized controller approach, as has been well noted in Barfoot and D'Eleuterio [9], for a task involving multiple decentralized agents or robots is that it can be

scaled up to larger problem sizes without requiring retraining. Minor differences in fitness performance and 'success rates' for controllers trained on a 19×19 world, with 12 agents is 'magnified' once we scaleup to a larger problem size (100×100 world, 76 agents). This is where the BRL architectures shows utility over conventional network architectures (see Table 2.1).

2.8.3 Sensory Input Noise

We measure the robustness of the evolved controller solutions by introducing sensor error, something not introduced during evolution (Figure 2.12). The BRL architectures tend to out-perform the standard neural network architectures for a sensor error of below 10%. Beyond 10% sensor error, the net effect is drastic independent of the controller architecture. This could be explained by the fact that beyond a critical threshold of sensory error, the simulated robots are unable to reach a consensus since the net actions are not constructive. It appears the ability of the BRL architecture to limit the number of expert networks used for task decomposition limits overfitting amongst the networks and reduces the effect of spurious sensor input. The general expectation is that with more expert networks in place, the systems may tend towards oversegmentation of the task into subtasks. Larger ETDNs, in particular the BRL (with 16 expert networks) have a tendency of evolving more robust, scalable solutions that can better handle unique situations not encountered during training.

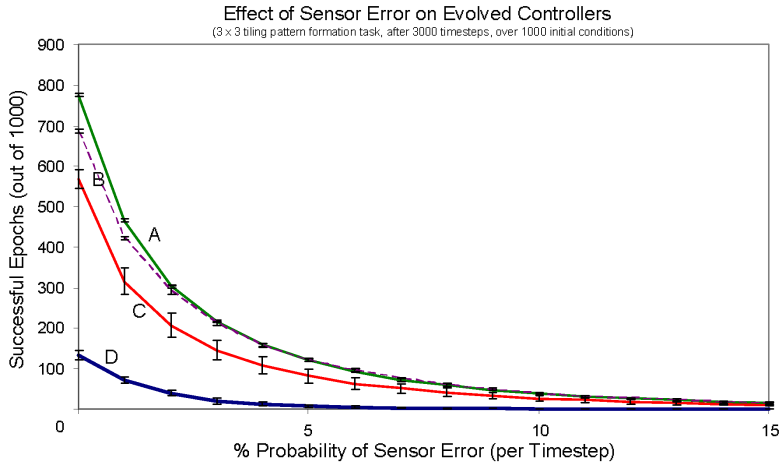


Figure 2.12: Effect of sensor error on evolved controllers population best at Generation 200) for the 3×3 tiling pattern formation task, 16×16 world, 11 robots, 36 blocks, after 3000 timesteps) averaged over 1000 simulations. (A) BRL (16 Expert Nets, Modular), (B) ETDN (2 Expert Nets, Modular), (C) Standard Neural Net. (Modular), (D) Standard Neural Net (Sigmoid). Error bars indicate standard deviation.

2.9 Spatial Crosstalk and Network Performance

Apart from analyzing the performance of ETDN networks, we are also interested in better quantifying and validating some of the observations on spatial crosstalk made by Jordan, Jacobs and Barto [81]. Within

this section, we consider both empirical simulations results and analytical techniques for predicting the performance of two layer fixed-topology neural networks for the tiling formation task. Figure 2.16 shows the number of successful (normalized to 1) of a two-layer fixed topology network, after 200 generations, with one output neuron and number of hidden neurons varied. The fitness performance peaks for around 7-8 hidden neurons present and shows significant drop in fitness when increasing the number of hidden neurons from 11 to 12. Beyond 15 hidden neurons, we see the fitness performance dwindle, making it unlikely to find a desired solution to the task. Corroborating the hypothesis made earlier, we can confirm that number of parallel hidden neurons does impact the training performance of the networks. Making a wrong choice of network topology can make training more difficult. To better understand these observations, we seek to build an analytical model that can be used to predict network performance.

2.9.1 Theoretical Framework

We start by considering simplified three layer networks, consisting of an input neuron layer, n hidden neurons and a single output neuron (that serves as a thresholded summing junction for the hidden neurons). Only the weights associated with the hidden neurons are trained. We assume the output neuron sums the signal from the hidden neurons and thresholds them to produce a network output.

We define a *solution neural network* to be a network that consist of a required number of *feature detector* hidden neurons and no disruptive neurons to complete a particular computation task successfully. The term *feature detector* is borrowed from biological vision research, where neurons are theorized to react to particular (finely tuned) set of inputs and remain dormant for most others. For an object character recognition (OCR) task, feature detectors can be a set of pre-positioned windows used to identify defining (unique) elements of the characters. Figure 2.13 shows four such pre-positioned windows that can be used to distinguish between the uppercase characters shown. In general, we refer to feature detectors as a set of sensory inputs that are used to identify unique characteristics of the current surrounding state.

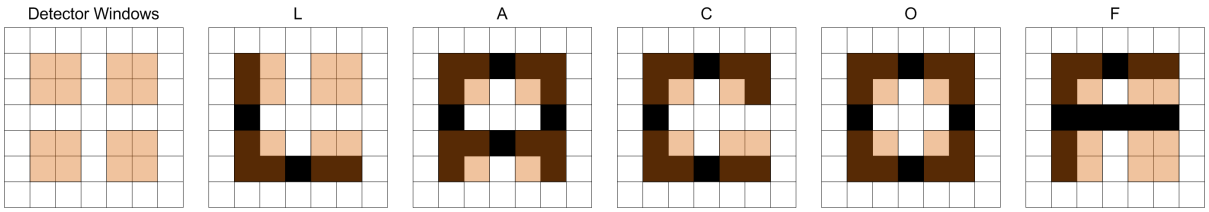


Figure 2.13: For an example OCR task, four pre-positioned windows (shaded orange) are used to distinguish features (unique characteristics) among 5×5 pixel monochrome characters.

There can exists many 'classes' of solution neural networks for a task at hand with not all required feature detectors being shared between these 'classes' of solutions. With a particular solution network, we theorize that the network consist of a number of feature detectors (subcomponents of a network required to complete a task). For a feature neuron a, we define p_a to be the probability of finding this feature neuron assuming random initializations of the neuronal parameters.

In addition, we assume there are no disruptive neurons present in the network. We define p_{neg} to be the probability of finding disruptive neurons assuming once again random initializations of the neuronal parameters. The effect of disruptive neurons located in parallel to the feature detector neurons results in disruption or noise being added to the signal that is being summed by the output neuron. The net effect being that the network produces errors that prevent it from successfully completing a particular task.

We also introduce a third category, namely, *neutral neurons*. Neutral neurons can be present in a solution network where, unlike the disruptive neurons, will remain dormant and will not affect the network output. We term these neurons neutral, borrowing from the concept of neutrality in genetics. We define p_{neut} to be the probability of finding neutral neurons assuming once again random initializations of the neuronal parameters.

If a solution network were to consist of more than one feature detector neuron, then we may term the solution as being distributed among the feature detectors. The distribution of feature detectors may occur in multiple ways. One way maybe that the same functionality within a feature detector exists in a redundant fashion within many multiple neurons. Removal of one of these neurons will not alter the overall functionality of the network based on the described setup. The other possibility is that the functionality is subdivided among several feature detectors, where if one feature neuron is damaged or removed it will result in some loss of functionality.

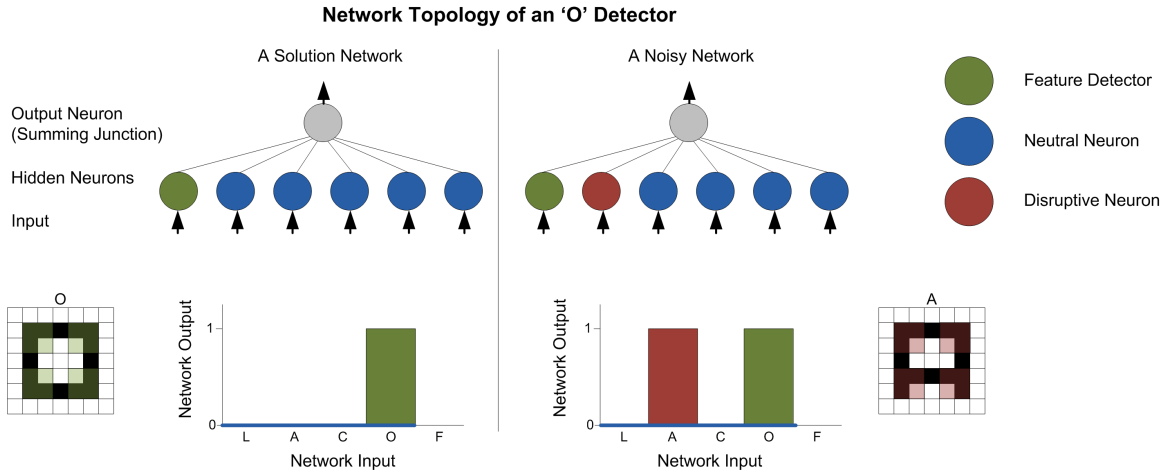


Figure 2.14: (Left) An 'O' detector solution network used to identify an uppercase 'O' using pre-positioned feature detector shaded windows as shown. The network outputs a 1 when an uppercase 'O' is encountered and 0 otherwise. The solution network consists of a single feature detector neuron and multiple neutral neurons. (Right) A noisy network consisting of the required feature detector neuron and a disruptive neuron. The resultant network outputs a 1 when an uppercase 'O' or 'A' is encountered.

Returning to an OCR task, consider a situation where a neural network is to correctly identify an uppercase 'O' by outputting a 1 and 0 otherwise. Figure 2.14 (left) shows a solution network consisting of a single feature detector (combining four feature windows) that can correctly identify an uppercase 'O'. Each neuron produces a binary output, $\{0,1\}$. The neutral neurons as mentioned earlier remain dormant (out-

putting a 0 for the allowable range of sensory input) and doesn't impact network performance. The output neuron is a summing junction and produces 1 when the summed signal is greater than or equal to 1. The feature detectors as shown, correctly identifies the four corners of the uppercase 'O' and the resultant signal is summed to produce the correct network output. The probability p_a of obtaining the 'O' feature detector neuron through random initialization of neuron parameters is shown in Appendix 8.

Figure 2.14 (right) shows a noisy network. This noisy network not only contains the correct feature detector (used to identify the uppercase 'O') but also contains a disruptive neuron shown in red. The disruptive neuron is activated when an uppercase 'A' is encountered and hence the network incorrectly signals having detected an 'O' when encountering an 'A'.

2.9.2 Solution Networks for One Feature

We start with a simple scenario, a solution network that only consists of a single type of feature detector neuron (with probability of finding this neuron from random as p_a). For the network to be a solution network, the number of feature detector neurons present, $x \geq 1$ and remaining number of hidden neurons can be neutral. Therefore, the probability P_1, of a finding a suitable network, consisting a single of type of feature detector neuron meeting the afore mentioned criteria is the following:

$$P_1(n) = \sum_{x=1}^{n} \begin{pmatrix} n \\ x \end{pmatrix} p_a^x p_{\text{neut}}^{n-x} \tag{2.7}$$

for $n > 1$, where n is the total number of hidden neurons present. Now if we were to assume a network configuration without any constraints on the number of feature detector neurons a present for a total of n hidden neurons, we get P_{Bin}, the probability of finding this network configuration in the form of a binomial distribution:

$$P_{\text{Bin}}(n) = \sum_{x=0}^{n} \begin{pmatrix} n \\ x \end{pmatrix} p_a^x p_{\text{neut}}^{n-x} \tag{2.8}$$

The probability of finding the one-feature solution network as a function of the binomial distribution is therefore:

$$P_{Bin}(n) = \sum_{x=1}^{n} \begin{pmatrix} n \\ x \end{pmatrix} p_a^x p_{\text{neut}}^{n-x} + \begin{pmatrix} n \\ 0 \end{pmatrix} p_a^0 p_{\text{neut}}^{n-0} \tag{2.9}$$

Using the binomial theorem, we can further simplify the probability $P_1(n)$ to the following:

$$P_1(n) = (p_a + p_{\text{neut}})^n - p_{\text{neut}}^n \tag{2.10}$$

2.9.3 Solution Networks for Two-Features, Three-Features and m-Features

We further extend the approach outlined in the previous section to include two or more feature neurons. The probability of finding a two feature network, $P_2(n)$ is the following:

$$P_2(n) = \sum_{x_1=1,x_2=1,x_{\text{neut}}=0}^{x_1+x_2+x_{\text{neut}}=n} \begin{pmatrix} n \\ x_1\ x_2\ x_{\text{neut}} \end{pmatrix} p_a^{x_1} p_b^{x_2} p_{\text{neut}}^{x_{\text{neut}}} \tag{2.11}$$

for $n \geq 2$, where n is the total number of hidden neurons in the network, x_1 is the number of feature neurons a, x_2 is the number of feature neurons b and x_{neut} is the number of neutral neurons and $p_\square$ is the probability associated with each type of neuron. To further simplify (2.11), we can express $P_2(n)$ in terms of the trinomial distribution $P_{\text{tri}}(n)$ for three types of neurons:

$$P_{\text{tri}}(n) = \sum_{x_1=0,x_2=0,x_{\text{neut}}=0}^{x_1+x_2+x_{\text{neut}}=n} \begin{pmatrix} n \\ x_1\ x_2\ x_{\text{neut}} \end{pmatrix} p_a^{x_1} p_b^{x_2} p_{\text{neut}}^{x_{\text{neut}}} \tag{2.12}$$

We may then rewrite, $P_2(n)$ as follows:

$$P_2(n) = P_{tri}(n) - \sum_{x_1=0,x_{\text{neut}}=0}^{x_1+x_{\text{neut}}=n} \begin{pmatrix} n \\ x_1 \end{pmatrix} p_a^{x_1} p_{\text{neut}}^{x_{\text{neut}}} - \sum_{x_2=0,x_{\text{neut}}=0}^{x_2+x_{\text{neut}}=n} \begin{pmatrix} n \\ x_2 \end{pmatrix} p_b^{x_2} p_{\text{neut}}^{x_{\text{neut}}} + p_{\text{neut}}^n \tag{2.13}$$

Using the multinomial theorem, we obtain a simplified expression for (2.13):

$$P_2(n) = (p_a + p_b + p_{\text{neut}})^n - (p_a + p_{\text{neut}})^n - (p_b + p_{\text{neut}})^n + p_{\text{neut}}^n \tag{2.14}$$

Following from this, the probability of finding a solution network for the three feature network, $P_3(n)$, is the following:

$$\begin{aligned} P_3(n) = (p_a + p_b + p_c + p_{\text{neut}})^n - (p_a + p_c + p_{\text{neut}})^n - (p_b + p_c + p_{\text{neut}})^n - \\ (p_a + p_b + p_{\text{neut}})^n + (p_a + p_{\text{neut}})^n + (p_b + p_{\text{neut}})^n + (p_c + p_{\text{neut}})^n - p_{\text{neut}}^n \end{aligned} \tag{2.15}$$

for $n \geq 3$ we also introduce feature neuron c, with corresponding probability p_c. Similar to the one-feature network, it would be expected that if $p_a = 0$ or $p_b = 0$, then it follows $P_2(n) = 0$. Also $P_3(n) = 0$ if either $p_a = 0$, $p_b = 0$ or $p_c = 0$. Thus we generalize this observation to say that if the probability of finding a constituent feature neuron with a solution network is zero, then it follows the probability of finding a solution network is also zero.

For the m-feature scenario:

$$P_m(n) = \sum_{x_{\text{neut}}=0,x_i=1 for\ i=1...m,}^{n=x_{\text{neut}}+\sum_{i=1}^m x_i} \begin{pmatrix} n \\ x_1 \ldots x_m\ x_{\text{neut}} \end{pmatrix} p_{\text{neut}}^{x_{\text{neut}}} \prod_{j=1}^{m} p_j^{x_j} \tag{2.16}$$

for $n \geq m$, where x_i is the number of feature neuron i and p_i is the corresponding probability. Using the multinomial theorem as done previously, we may express $P_m(n)$ as follows:

$$P_m(n) = \sum_{k_1=0}^{k_{1=1}} \cdots \sum_{k_m=0}^{k_{m=1}} C_{k_1,\ldots,k_m} \left(\sum_{i=1}^{m} k_i p_i + p_{\text{neut}} \right)^n \tag{2.17}$$

for $n \geq m$. For m features, there exists 2^m entries for C, where $C_{k_1,\ldots,k_m} = 1$ for $k_1 = k_2 = \ldots = k_m = 1$ and that satisfy the condition where if $p_j = 0$ for $1 \leq j \leq m$, then $P_m(n) = 0$.

2.9.4 Distributed Features: An Analysis

Based on the equations derived we can now proceed to compare the relative probability of finding solutions for one, two and three feature networks by varying the number of hidden neurons. These are conceptual studies from where we seek to understand general trends. We consider one scenario in which $p_{\text{neut}} = 0.7$ and we assume the probability of a single feature detector is $p_a = 0.28$. For the two and three feature cases, we assume the probability of finding each feature is $0.28/m$, where m is the number of features.

Plots of the probability distribution for these scenarios are shown in Figure 2.15. What is clearly evident from the plot is that increased number of features (more distributive), solutions are less probable assuming p_{neut} is held constant. For a two-solution scenario to match the probability distribution of a single feature solution, the probability of finding each of the feature neurons needs to be higher. Therefore what we see is a trade off, where distributed solutions are more likely if the probability of finding the component features is greater than is for a single feature.

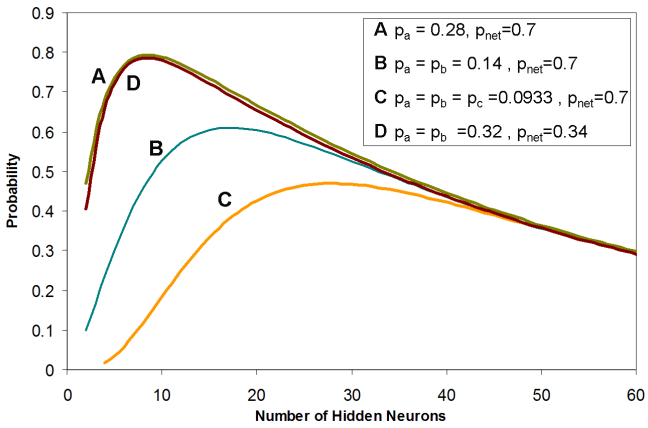


Figure 2.15: Four different solution networks considered (where p_{neg} = 0.02) including (A) single feature network (B) Two-feature (C) Three-feature (D) Two feature network with solution probability distribution similar to (A).

It should also be noted that distributed solutions in this scenario reach a peak probability with a greater number of hidden neurons present. Furthermore, with increased number of hidden neurons, each of these scenarios trend toward matching probabilities. Thus, in a typical network training scenario, with increased number of hidden neurons present, each type of solution will have matching occurrence probabilities. What we expect is that with an increased number of hidden neurons being prespecified, more distributed solutions should also be present (i.e., solutions with greater number of distinct component feature neurons).

2.9.5 Model Comparison for the Tiling Formation Task

The theoretical models presented up to this point give the probability distribution of network solutions with respect to number of hidden neurons present. However this probability distribution is not directly available from the simulation experiments. Instead what is available is the success rate (out of 15 initial conditions) among the population best. The theoretical model does not account for any aspect of searching for the solution, such as search efficiency nor any details regarding the fitness and task domain. However, the observational data (that measures the success rate) depends on these conditions. Therefore, as a first step, what we are after is to apply a transformation function to observational data (success rate among the population best) to get a probability distribution function (to obtain a solution) that is similar to what have seen in the previous analysis. To summarize, we apply a transformation function $\chi[\cdot]$ to the probability distribution $P(n)$ to get a model $T(n)$ of the observed data $Q(n)$, where n is the number hidden neurons:

$$T(n) = \chi\left[P(n)\right] \tag{2.18}$$

$P(n)$ is given below and is a mixture model of probability distribution $P_m(n)$ defined in (2.18), that is the probability distribution for a m-feature solution network

$$P(n) = \sum_{m=1}^{\infty} v_m P_m(n) \tag{2.19}$$

where, $\sum_{m=1}^{\infty} v_m = 1$ and v_m is the weighing constant associated with a m feature solution network. Since we don't have sufficient knowledge to calculate χ based on task-specific or search-specific parameters, χ is assumed to be a sigmoid function of the following form:

$$\chi(x) = \frac{k}{1 + e^{-(x+\varphi)\beta}} \tag{2.20}$$

and where k, φ and β are task/search specific parameters. For this experiment, we found $k = 0.69$, $\varphi = -0.32$ and $\beta = -36$ from manually attempting to fit $P(n)$ to $Q(n)$. The effect of the sigmoid function transformation could be thought of as an efficiency parameter, where number of solutions needs to exceed a certain threshold before it could be readily found by the evolutionary search process within a given finite time. With too few solutions present, the evolutionary search process will take longer to find a suitable solution. It should be noted however, that the relative amplitude (gain) of this sigmoid function is assumed and an exact value cannot be determined without determining the feature probability values.

Some simplifications are made to (2.19) such that, $P(n) = P_m(n)$, where $m = 2, 3, \ldots$. From analyzing the observation data, it is evident solutions emerge for $n \geq 2$. Figure 2.16 (left) shows a two-feature network probability distribution, $P_2(n)$, fitted to the observation data, with $p_a = 0.26$, $p_b = 0.26$ and $p_{\text{neut}} = 0.4$.

Further, we also attempt to fit a mixture model that combines two different solution probabilities, $P(n) = v_2(n) \cdot P_2(n) + [1 - v_2(n)] \cdot P_3(n)$ and follow the same procedure. For $P_3(n)$, we set $p_a =$

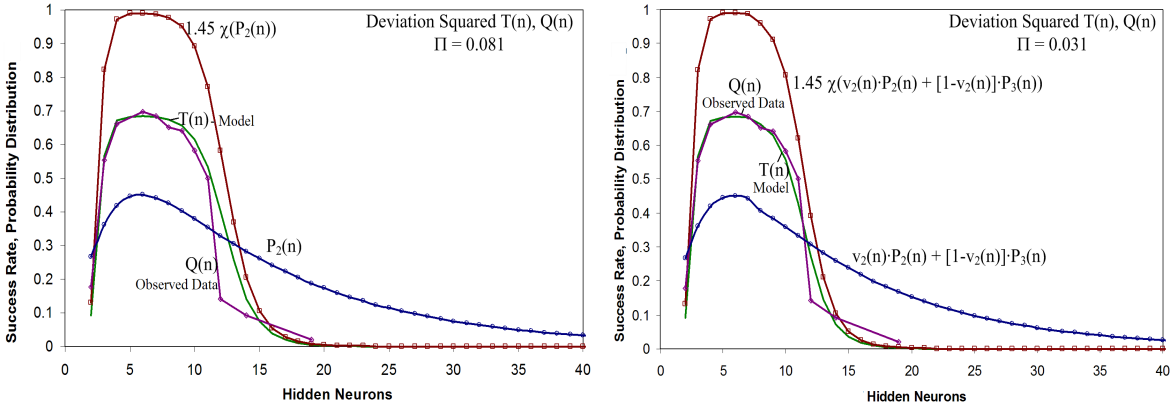


Figure 2.16: (Left) Plot of Q(n) showing success rate, while varying number of hidden neurons. T(n) assumes a two-feature solution network model. (Right) Plot of T(n) assuming a mixture model consisting of a two-feature and three-feature solution, using weighing parameters $v_2(n)$

$p_b = p_c = 0.3$ and $p_{\text{neut}} = 0$. Determination of feature probability values for the two-feature and three-feature case shows viability of a two-feature solution transforming into a three-feature solution, since both solution networks depend on three types of neurons. It is possible with the right sequence of mutations for a neutral neuron from a two-feature solution to transform into feature neuron c for the three-feature solution and for corresponding mutations among the remaining two feature neurons.

The results of this fit are shown in Figure 2.16 (right). Plots of $v_2(n)$ are shown in Figure 2.17 (right). Naturally a mixture model with a greater number of component functions can be expected to yield a better fit, but for us to show plausibility of this model, we also extract additional data for verification. Since the two-feature and three-feature networks solutions show a better fit to the observational data, we then seek to verify from the simulation experiments whether the observed solutions match the theoretical results.

It should be noted that the values guessed for the feature and neutral probability parameters are fitting terms which enable us to extract the shape of the underlying probability distribution but not the exact probability values of the features. For us to be able to extract the feature probability values, we must establish the relative amplitude (gain) of χ with respect to the $P(n)$. Interestingly, this is not necessary for us to determine peaks in the probability distribution.

Figure 2.17 (left) shows results of the ablation test indicating average of number feature neurons among the population best measured after 200 generations. From the ablation tests, we obtain $u_2(n)$, the observed weight associated with a two-feature solution network and used to determine $P(n) = u_2(n) \cdot P_2(n) + [1 - u_2(n)] \cdot P_3(n)$. Observed values for $u_2(n)$ are shown in Figure 2.17 (right). For the ablation tests, each hidden neuron is removed independently and the performance of the controller determined. When there is no effect on the network performance, the neuron is considered neutral. Similarly, if removing multiple neurons shows similar performance contributions to the controller network individually, then the neurons are categorized as copies. This above assumption is in contrast to an assumption made for the theoretical setup. In the theoretical setup, removal of a redundant copy is not expected to alter the network performance, since

the output neuron is simply a thresholded summing junction feeding in signals from the hidden neurons. However, in the experimental setup, the output neuron cannot be simplified (constrained) as being just a thresholded summing junction and can attain other functionality. This implies that removal of redundant feature neurons is expected to alter the performance of the network since the output neurons cannot be guaranteed to be this summing junction.

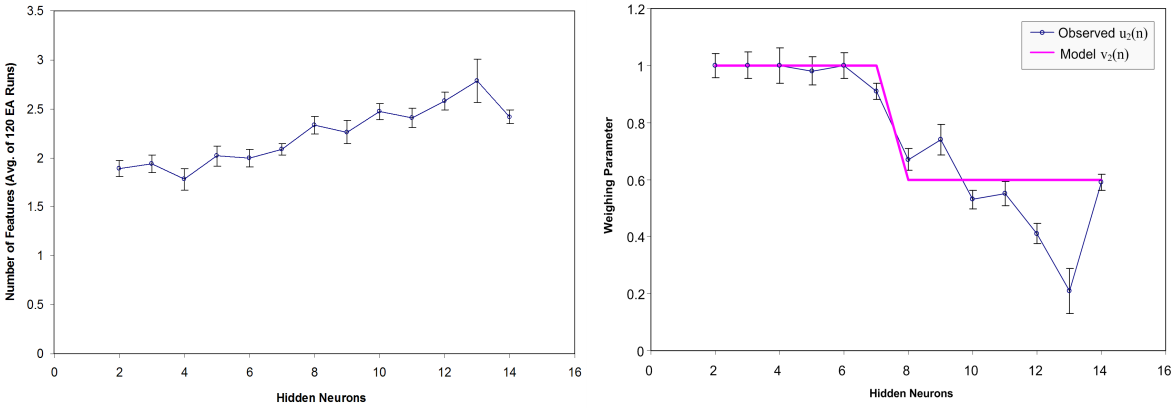


Figure 2.17: (Left) Average number of feature neurons (population best) and (right) Weight model $v_2(n)$ and observed weights $u_2(n)$ obtained from ablation tests.

Similarity is measured by comparing the average number of successful epochs contribution by each neuron and determining whether matching performance were obtained for matching initial conditions. The average number of non-redundant feature neurons solutions for the range of data (from 2 hidden neurons to 14) is between 2-3 neurons as shown. For 14 and more hidden neurons the number of instances of successful epochs are markedly less and thus not enough solutions exist to get a better measure of number of non-redundant feature neurons with it being more prone to noise.

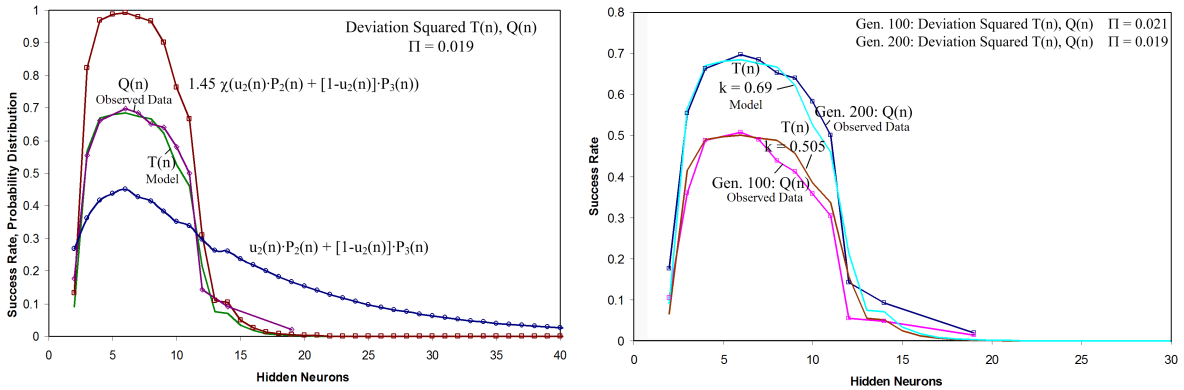


Figure 2.18: (Left) Plot of $T(n)$ and $Q(n)$ after 200 generations using weighing parameters $u_2(n)$ obtained from the ablation tests. (Right) Plot of $T(n)$ and $Q(n)$ after 100 and 200 generations, using weighing parameters $u_2(n)$ from ablation tests, while varying k as shown.

Applying the $u_2(n)$ values obtained using ablation tests back into the model yields a better fit against the observational data compared with a single two-feature solution network and the mixture model presented

earlier (Figure 2.18). This verification further strengthens our underlying assumptions made earlier. Use of the $u_2(n)$ taken after generation 200 also shows a good fit of the performance measured at generation 100, by just setting $k = 0.505$. This indicates the model generalizes well and strenghtens the hypothesis that k parameter used within χ is a time dependent search-method specific parameter hypothesized earlier and shows that that probability distribution appears to be largely constant between 100 and 200 generations. This increase in k value indicates that the system is fine tuning the networks configurations found. χ can be thought of as a search effectiveness parameter that tells us the expected (population best) fitness performance after a specified number of generations for a fixed number of hidden neurons. When the solution probability distribution exceeds a critical threshold, the effectiveness of the population (in finding solutions) is shown to dramatically increase and this is facilitated by setting the number of hidden neurons between 4 and 11.

Based on the techniques outlined in Appendix 8, it is also possible to determine the expected number of hidden neurons present, once the probability of obtaining the feature detector and neutral neurons are known. As we see for the tiling formation task, there appears to be two dominant solutions present. Recalling that the probability of obtaining feature neurons for the two feature and three feature solutions are $p_a = 0.26$, $p_b = 0.26$ and $p_{\text{neut}} = 0.4$ and $p_a = p_b = p_c = 0.3$ and $p_{\text{neut}} = 0$ then the expected number of hidden neurons for the two feature solution is 16.2 and for the three feature case its 14.06 based on the method outlined in Appendix 8. Since the evidence point to both solutions co-existing and the net probability of obtaining a solution is a mixture model of the two different solutions, then the expected value of solutions should be between 14.06 and 16.2 hidden neurons.

To verify whether the expected value of hidden neurons from the evolutionary simulations matches that from theory, we run another experiment, where we modify the architecture discussed previously and include genes for 25 hidden neurons (within genotype). We also add an additional genome parameter, n_{hidden}, that determines how many hidden neurons are expressed in the phenotype. n_{hidden} is limited to a value of between 1 and 25. The first n_{hidden} neuron genes are expressed and the remainder of the hidden neuron genes remain repressed. Figure 2.19 shows a histogram of solutions for the tiling formation task found according to number of hidden neurons expressed. The expected value of hidden neurons from the histogram is 15.43, out of 30 solutions found and falls within the range of predicted expected values mentioned earlier. It is also important to note that the system doesn't tend towards an n value that maximizes the solution probability (see Appendix 8). For the two-feature and three-feature network solutions, $n_{\text{max}} = 7$. This is inpart because the search process cannot determine whether the solution is at the solution probability peak or not. Instead the solutions tend towards a distribution *weighted* by the solution probabilities, $P_m(n)$.

Both the mixture model and observed values of $u_2(n)$ also confirms the viability of both two-feature and three-feature network solution for networks with seven or more hidden neurons (for this task). The theoretical models presented show good fits of a fixed network topology performance characteristics through an artificial evolutionary search process. Based on this analytical model, it is possible as we have shown to determine number of feature neurons present within a solution network and obtain estimates of the ratio of

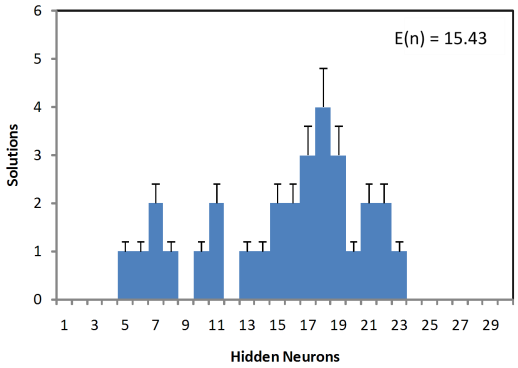


Figure 2.19: Histogram of solutions for the tiling formation task according to number of hidden neurons expressed. For the experiment, the first n_{hidden} neurons out of 25 hidden neurons genes are expressed. The parameter n_{hidden} is an additional entry in the genome that is evolved.

solutions with different number of unique feature neurons present. This is done macroscopically by measuring the performance of a fixed network topology while varying the number of hidden neurons. In contrast, the ablation experiment is a microscopic procedure that involves systematically isolating the performance contribution of each neuron in the network. From the analytical model present, it is also possible to predict for the n value of the optimal network topology knowing the probability values of the necessary feature neurons and expected ratios of different m-feature solution networks. Search and task specific parameters such as k, φ and β could be determined through inference from preforming limited sampling runs.

2.10 Summary

The analytical approach presented in the previous section further confirms that a wrong choice of network topology can adversely impact finding a suitable solution using a neural network approach. To determine an ideal network topology suited for a particular task, using this analytical approach, one would require knowing the probability of finding the necessary feature neurons. This would imply decomposing a task at hand and determining the necessary subtasks. However, as emphasized earlier, this remains a challenging endeavor within the collective robotics domain. It is often not clear what set of local behaviours and subtasks are necessary to accomplish a global goal within multirobotic systems. Instead, it is often easier to define a global goal and expect a self-organized top-down approach that will find the necessary local behaviors.

ETDN overcomes the issue of separate training of network components, namely decision networks and expert networks that existed with previous architectures. The approach allows for a global goal to be defined and expect emergence of a suitable multirobotic controller as in for the tiling pattern formation task. However, this approach still requires experimenter knowledge in determining number of expert networks needed and network topology of both expert and decision networks. These limitations are also usually task dependent factors. Although for the tiling pattern formation task, increased number of expert networks

shows no degradation in training performance (number of genetic evaluations), this may not hold for other tasks. To automate finding suitable network topologies (and number of experts), one may naively use a brute force approach that systematically varies the topological structure and number of experts. The problem with this brute force approach is that it is not very scalable, primarily due to the combinatorial explosion of possibilities. In addition, there is a need for a search process that can home in on and retain suitable parameters. We are also intent on reducing the number of evaluations needed since the bulk of the training time is taken up by operation of robotic hardware or software simulation and not by the genetic operators (when using an EA approach).

A more efficient approach is to tune the topology and network contents in parallel. In the next chapter Artificial Neural Tissues (ANT) addresses the limitations with fixed topology ETDNs and extends emergent task decomposition to variable length topologies.

We're inquiring into the deepest nature of our constitutions: How we inherit from each other. How we can change. How our minds think. How our will is related to our thoughts. How our thoughts are related to our molecules.
—Gerald M. Edelman

Chapter 3

ARTIFICIAL NEURAL TISSUES

3.1 Introduction

We are motivated by multicellular organisms where complex organizational behaviors have emerged through evolution and collective interaction. The answer to the training of controllers for complex problems has often been to introduce more supervision *ad hoc*, where the experimenter decomposes a complex task into a set of simpler tasks based on domain knowledge of the task at hand [116]. In biological systems, such intervention (more supervision) rarely exists, yet these systems can adapt and thrive with relative ease.

Evolving open-ended variable-length neural systems with large phenotypes remains a significant challenge in the field of artificial life [134, 65]. One of the problems encountered with larger phenotypes is the *bootstrap problem* [116]. The bootstrap problem occurs when evolutionary algorithms (EAs) are unable to pick out incrementally better solutions for crossover and mutation resulting in premature stagnation of an evolutionary run. This is evident in monolithic neural-network topologies with many hidden neurons, where the effects of spatial crosstalk [81] can further accentuate the bootstrap problem.

Current solution techniques involve starting with a single cell or a small phenotype and allowing for the system to grow incrementally in size and complexity until the system can find a satisfactory solution to a given task [146]. In biology, there often exist nervous systems that may already have the neural capacity to adapt easily to a new task/scenario. In such circumstances, starting over with a minimalist system may be a much slower process owing to the reliance of topological growth directed by evolution.

Consider a binary gene encoding scheme with a genome consisting of n bits. Evolutionary techniques

are expected to find solutions from a set of 2^n candidate solutions. Growth in the length of the genome implies an exponential growth in the candidate solution space. This can be both an advantage or a disadvantage depending on the context. While there may be more candidate solutions, the search space has also significantly increased. The increased number of candidate solutions may also present a challenge in machine learning, namely the *competing conventions problem* (permutation problem) [129]. The competing conventions problem occurs when competing representations of a controller, with similar fitness are mutually incompatible from crossover interactions resulting in damage to the genotypes. The end result is that the search process is computationally inefficient and results in slower convergence to a solution.

In nature, competing conventions in the form of *homology* (where multiple alleles represent the same trait) is widely known to exist in the genome yet this problem has been overcome through use of *synapsis*. In synapsis, a protein known as RecA aligns homologous genes between two genomes in preparation for crossover [130, 141] thus preventing crossover-induced damage. The trick has been to determine how a protein such as RecA determines whether two genes are homologous without having to execute the programmatic code buried in the genes. Previous artificial techniques attempt to mimic biology in this respect but they either introduce historical origin markers [144] considered to be biologically implausible owing to its centralized setup or introduce restrictions on subnetwork combinations [127].

Artificial nervous systems can benefit from adaptive mechanisms whose performance is invariant to negative changes to the topological structure and a genetic structure that prevent or reduces the effects of the competing-conventions problem. Use of variable-length neural systems offer significant advantages over a fixed-topology structure as such systems often require less information *a priori* and topological growth is directed by evolutionary pressure.

Often a fixed-network topology is based on task-specific information or *ad hoc* assumptions. This methodology is problematic when very little is known about a task at hand and wrong assumptions may inadvertently lead to poor performance. A fixed-network topology may also heavily influence the EA search process and thus limit the emergence of potentially innovative solutions.

Emergent Task Decomposition Networks (ETDNs) presented in Chapter 2 can perform task decomposition by assigning expert networks to the subtasks through self-organization. However a limitation with this approach it that it requires human input in determining the topology and number of expert networks needed. While evaluation of such parameters can be automated and exhaustively searched, such brute-force techniques become infeasible owing to the curse of dimensionality when scaling to more involved tasks that require extended evaluation time.

A biologically plausible answer to many of these limitations has been the use of regulatory systems that can limit the negative effects of changes to the topological structure, mediate growth and death of cells depending on evolutionary pressure, reduce the effects of the competing conventions problem and facilitate self-organized task decomposition. Taking inspiration from the mammalian visual cortex, we have developed an evolvable Artificial Neural Tissue (ANT) model presented in this chapter. The genotype for the

evolvable ANT defines a developmental program that constructs a neural tissue (phenotype) and associated neural-regulatory functionality.

The variable-length tissue architecture can be characterized as a lattice of neurons mapped into a three-dimensional structure. Gene regulatory networks control cell growth and cell death in the tissue. Neural regulatory functionality activates/inhibits portions of the tissue based on external sensory input using a coarse-coding framework, while gene regulatory functionality is used during development. In addition the ANT genotype to phenotype mapping system also addresses the competing conventions problem.

The Artificial Neural Tissue (ANT) model is compared to existing neuroevolutionary approaches on several control tasks, including the benchmark double-pole balancing task, a multirobot tiling pattern formation task, a phototaxis task with obstacles and a sign-following task (a more difficult variant of the T-maze task [160]).

3.2 Background

In nature, regulatory systems within the nervous system are found at the gene/molecular level and at intercellular levels [83]. Regulation refers to mechanisms used to control adaptation of form or behavior of an organism to changing conditions[1]. More specifically, gene regulation is a mechanism used to control gene expression. All biological functions within an organism can be traced back to expression and inhibition of its genetic content. Regulation at intercellular levels within an organism's nervous system involves interaction among cells that leads to control of sensory processing and behavior [131]. It had been hypothesized that arrangements with regulatory systems provide certain computational and tractability advantages [8, 85]. The ANT model is built on this assumption.

At the gene level, the ability to activate and inhibit parts of a genome allow for a biological control system to either explore (through trial and error) or ignore/shut off genes that may be malformed or problematic. The process by which the proteins shut off or activate genes is thought to be a competitive process [87]. For proteins to attach to an intended binding site, enough concentration of proteins needs to be produced to make the situation probable. In addition, the protein maybe deluged with another inhibitory protein that may alter its functionality. The proteins are perhaps also competing against other proteins for limited resources, such as amino acids (the building blocks to produce the proteins) and available binding sites.

From analyzing neural systems, it has been suggested that many functions including neural growth, synaptic pruning/organization and functional organization and learning are regulated [43, 131]. For neuronal growth, functional organization and learning, regulatory functions are distributed among neurons. Each neuron is noisy, inefficient, unreliable and slower than present-day computers yet on the whole, neural systems can perform reliably and fast enough for many survival activities. It is apparent from these observations that neuronal organization plays a critical part in achieving robustness. But how is this done?

[1] Based on definition by American Heritage Stedman's Medical Dictionary.

It has been argued that *selectionist* mechanisms underpin regulation and help to offset and 'counter-balance' the robustness and reliability limitations of the individual subcomponents in biological systems [87]. The artificial neural tissue (ANT) architecture superimposes on a typical neural-network structure a coarse-coding neural regulatory mechanism inspired by the work of Albus [3] and Hinton [72]. The neural regulatory mechanism functions through the diffusion of chemicals that trigger into operation or inhibit the functionality of other neuronal groups, a form of selection. The mechanism has a biological reference as neurons can communicate not only electrically by exchanging signals along axons but also through longer-range diffusion of chemicals. Analysis into the biological plausibility of this mechanism and its implications are covered in Chapter 5. The following section presents the ANT model followed by analysis of key features, related work and experiments on several robotic and control tasks.

3.3 Artificial Neural Tissue Model

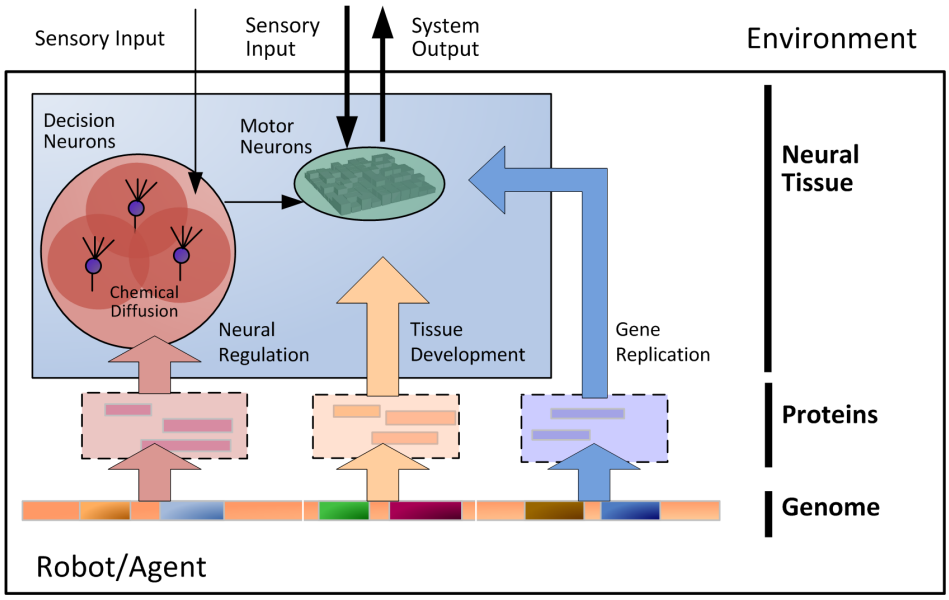


Figure 3.1: Schematic of the Artificial Neural Tissue (ANT) Architecture.

The ANT architecture (Figure 3.1) presented in this chapter consists of a developmental program, encoded in the 'genome,' that constructs a three-dimensional neural tissue and associated regulatory functionality. Regulatory functionality exists at two levels, gene/molecular level and at the cellular level. The tissue consists of two types of neural units, *decision neurons* and *motor-control neurons*, or simply motor neurons. Neural Regulation is performed by the decision neurons that dynamically excite or inhibit motor-control neurons within the tissue based on a coarse coding framework. Let us discuss the computational mechanism of the tissue first and then outline the process by which the tissue is created.

3.3.1 Motor Neurons

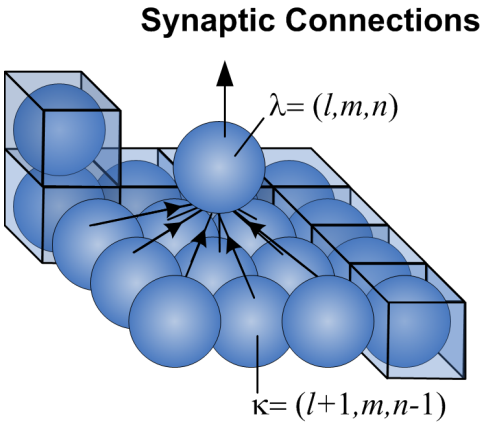


Figure 3.2: Synaptic connections between ANT motor neurons from layer $l+1$ to l.

We imagine the motor neurons within the tissue to be arranged in a regular rectangular lattice in which the neuron $N_{\boldsymbol{\lambda}}$ occupies the position $\boldsymbol{\lambda} = (l, m, n) \in \mathbb{I}^3$ (Figure 3.2). Depending on the activation functions used (see Section 3.3.3 for details), the state $s_{\boldsymbol{\lambda}} \in \mathbb{S}$ of the neuron is either binary, i.e., $\mathbb{S}_b = \{0, 1\}$ or can be real, $\mathbb{S}_p = [0, 1]$ or $\mathbb{S}_r = [-1, 1]$.

Each neuron $N_{\boldsymbol{\lambda}}$ nominally receives inputs from neurons $N_{\boldsymbol{\kappa}}$ where $\boldsymbol{\kappa} \in \Uparrow(\boldsymbol{\lambda})$, the nominal input set. Here we shall assume that these nominal inputs are the 3×3 neurons centered one layer below $N_{\boldsymbol{\lambda}}$; in other terms, $\Uparrow(\boldsymbol{\lambda}) = \{(i, j, k) \,|\, i = l-1, l, l+1;\ j = m-1, m, m+1;\ k = n-1\}$. (As will be explained presently, however, we shall not assume that all the motor neurons are active all the time.)

The sensor data are represented by the activation of the sensor input neurons $N_{\boldsymbol{\alpha}_i}, i = 1 \ldots m$, summarized as $A = \{s_{\boldsymbol{\alpha}_1}, s_{\boldsymbol{\alpha}_2} \ldots s_{\boldsymbol{\alpha}_m}\}$. Similarly, the output of the network is represented by the activation of the output neurons $N_{\boldsymbol{\omega}_j}, j = 1 \ldots n$, summarized as $\Omega = \{s_{\boldsymbol{\omega}_1^1}, s_{\boldsymbol{\omega}_2^1}, s_{\boldsymbol{\omega}_3^2} \ldots s_{\boldsymbol{\omega}_n^b}\}$, where $k = 1 \ldots b$ specifies the output behavior. Each output neuron commands one behavior of the agent. (In the case of a robot, a typical behavior may be to move forward a given distance. This may involve the coordinated action of several actuators. Alternatively, the behavior may be more primitive such as augmenting the current of a given actuator.) If $s_{\boldsymbol{\omega}_j^k} = 1$, output neuron $\boldsymbol{\omega}_j$ votes to activate behavior k; if $s_{\boldsymbol{\omega}_j^k} = 0$, it does not. Since multiple neurons can have access to a behavior pathway, an arbitration scheme is imposed to ensure the controller is deterministic where $p(k) = \sum_{j=1}^{n} \gamma(s_{\boldsymbol{\omega}_j^i}, k) s_{\boldsymbol{\omega}_j^i} / n_k$ and $n_k = \sum_{j=1}^{n} \gamma(s_{\boldsymbol{\omega}_j^i}, k)$ is the number of output neurons connected to output behavior k where $\gamma(s_{\boldsymbol{\omega}_j^i}, k)$ is evaluated as follows:

$$\gamma(s_{\boldsymbol{\omega}_j^i}, k) = \begin{cases} 1, & \text{if } i = k \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

and resulting in behavior k being activated if $p(k) \geq 0.5$.

As implied by the set notation of Ω, the outputs are not ordered. In this embodiment, the order of activation is selected randomly. We are primarily interested here in the statistical characteristics of relatively

large populations but such an approach would likely not be desirable in a practical robotic application. However, this can be remedied by simply assigning a sequence *a priori* to the activations (as shown for example in Table 3.4 for the phototaxis task).

We moreover note that the output neurons can be redundant; that is, more than one neuron can command the same behavior, in which case for a given time step one behavior may be 'emphasized' by being voted multiple times. Neurons may also cancel out each other such as when one output commands a forward step while another a backward step. Finally, not all behaviors need be encoded in the neural tissue. This is left to the evolutionary process.

3.3.2 The Decision Neuron

The coarse coding nature of the artificial neural tissue is provided by the decision neurons. Decision neurons occupy nodes in the lattice as established by the evolutionary process (Figure 3.3). The effect of these neurons is to excite into operation or inhibit the motor control neurons (shown as spheres). Once a motor control neuron is excited into operation, the computation outlined in (3.6) is performed. Motivated as we are to seek biological support for ANT, we may look to the phenomenon of chemical communication among neurons. In addition to communicating electrically along axons, some neurons release chemicals that are read by other neurons, in essence serving as a 'wireless' communication system to complement the 'wired' one. Further details on the biological plausibility of this process is presented in Section 5.3.

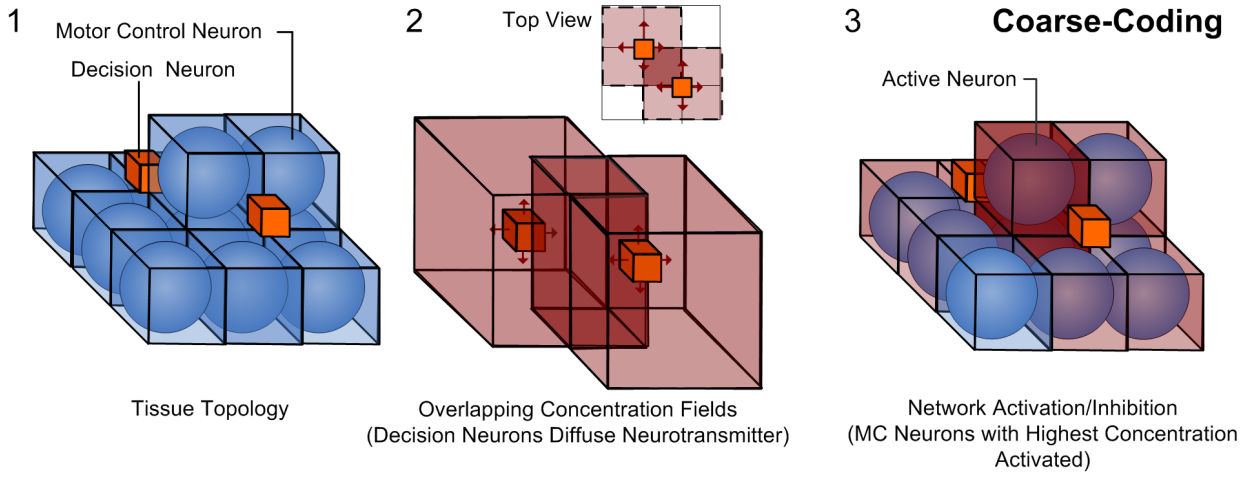


Figure 3.3: Coarse coding regulation being performed by two decision neurons (shown as squares) that diffuse a chemical, in turn activating a motor neuron column located at the center (right).

Each decision neuron can be in one of two states, one in which it diffuses a neurotransmitter chemical or remains dormant. The state of a decision neuron $T_{\boldsymbol{\mu}}$, $\boldsymbol{\mu}$ is binary and determined by one of the activation functions (see Section 3.3.3). Assuming the decision neurons use the modular activation function described in Section 3.3.3, the inputs to $T_{\boldsymbol{\mu}}$ are all the input sensor neurons $N_{\boldsymbol{\alpha}}$; *i.e.*, $s_{\boldsymbol{\mu}} = \psi_{\boldsymbol{\mu}}(s_{\boldsymbol{\alpha}_1} \ldots s_{\boldsymbol{\alpha}_m})$ where

$\sigma_{\boldsymbol{\mu}} = \sum_{\boldsymbol{\alpha}} v_{\boldsymbol{\alpha}}^{\boldsymbol{\mu}} s_{\boldsymbol{\alpha}} / \sum_{\boldsymbol{\alpha}} s_{\boldsymbol{\alpha}}$ and $v_{\boldsymbol{\alpha}}^{\boldsymbol{\mu}}$ are the weights. The decision neuron is dormant if $s_{\boldsymbol{\mu}} = 0$ and releases a virtual neurotransmitter chemical of uniform concentration $c_{\boldsymbol{\mu}}$ over a prescribed field of influence if $s_{\boldsymbol{\mu}} = 1$.

Motor control neurons within the highest chemical concentration field are excited into operation. Only those neurons that are so activated will establish the functioning network for the given set of input sensor data. Owing to the coarse coding effect, the sums used in the weighted input of (3.5) are over only the set $\overline{\Uparrow}(\boldsymbol{\lambda}) \subseteq \Uparrow(\boldsymbol{\lambda})$ of active inputs to $N_{\boldsymbol{\lambda}}$. Likewise the output of ANT is in general $\overline{\Omega} \subseteq \Omega$. The decision neuron's field of influence is taken to be a rectangular box extending $\pm d_{\boldsymbol{\mu}}^{r}$, where $r = 1, 2, 3$, from $\boldsymbol{\mu}$ in the three perpendicular directions. These three dimensions along with $\boldsymbol{\mu}$ and $c_{\boldsymbol{\mu}}$, the concentration level of the virtual chemical emitted by $T_{\boldsymbol{\mu}}$, are encoded in the genome.

Decision neurons emit chemicals that are used to selectively activate and inhibit motor control neurons. We label this component of ANT as the *neural regulatory system*. This is akin to genes being able to activate or inhibit other genes in a gene regulatory system.

3.3.3 Activation Functions

Three different activations functions are considered for the experiments presented in this chapter, including a sigmoid, hyperbolic tangent and a modular activation function. For the sigmoid activation function:

$$\psi = \frac{1}{1 + e^{-\sigma}} \tag{3.2}$$

where the weighted input $\sigma_{\boldsymbol{\lambda}}$ for neuron $N_{\boldsymbol{\lambda}}$ is nominally taken as follows:

$$\sigma_{\boldsymbol{\lambda}} = \sum_{\boldsymbol{\kappa} \in \Uparrow(\boldsymbol{\lambda})} w_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}} s_{\boldsymbol{\kappa}} + \phi \tag{3.3}$$

$w_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}} \in \mathbb{R}$ is the weight connecting neurons, $N_{\boldsymbol{\kappa}}$ to $N_{\boldsymbol{\lambda}}$, $s_{\boldsymbol{\kappa}} \in \mathbb{S}$ is the output from neuron $N_{\boldsymbol{\lambda}}$ and $\phi \in \mathbb{R}$ is the bias.

For the hyperbolic tangent activation function,

$$\psi = \frac{e^{\sigma} - e^{-\sigma}}{e^{\sigma} + e^{-\sigma}} \tag{3.4}$$

and σ is calculated from (3.3).

The modular activation function allows selection among four possible threshold functions of the weighted input σ. As emphasized in Chapter 2, the use of two threshold parameters, allows for a single neuron to compute the XOR function, in addition to the AND and OR functions. For this version of the modular activation function,

$$
\begin{aligned}
\psi_{\text{down}}(\sigma) &= \begin{cases} 0, & \text{if } \sigma \geq \theta_1 \\ 1, & \text{otherwise} \end{cases} \\
\psi_{\text{up}}(\sigma) &= \begin{cases} 0, & \text{if } \sigma \leq \theta_2 \\ 1, & \text{otherwise} \end{cases} \\
\psi_{\text{ditch}}(\sigma) &= \begin{cases} 0, & \min(\theta_1, \theta_2) \leq \sigma < \max(\theta_1, \theta_2) \\ 1, & \text{otherwise} \end{cases} \\
\psi_{\text{mound}}(\sigma) &= \begin{cases} 0, & \sigma \leq \min(\theta_1, \theta_2) \text{ or } \sigma > \max(\theta_1, \theta_2) \\ 1, & \text{otherwise} \end{cases}
\end{aligned}
\tag{3.5}
$$

and the weighted input $\sigma_{\boldsymbol{\lambda}}$ for neuron $N_{\boldsymbol{\lambda}}$ is nominally taken as

$$
\sigma_{\boldsymbol{\lambda}} = \frac{\sum_{\boldsymbol{\kappa} \in \Uparrow(\boldsymbol{\lambda})} w_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}} s_{\boldsymbol{\kappa}}}{\sum_{\boldsymbol{\kappa} \in \Uparrow(\boldsymbol{\lambda})} s_{\boldsymbol{\kappa}}} \tag{3.6}
$$

with the proviso that $\sigma = 0$ if the numerator and denominator are zero. Also, $w_{\boldsymbol{\lambda}}^{\boldsymbol{\kappa}} \in \mathbb{R}$ is the weight connecting $N_{\boldsymbol{\kappa}}$ to $N_{\boldsymbol{\lambda}}$. We may summarize these threshold functions in a single analytical expression as

$$
\psi = (1 - k_1)[(1 - k_2)\psi_{\text{down}} + k_2\psi_{\text{up}}] + k_1[(1 - k_2)\psi_{\text{ditch}} + k_2\psi_{\text{mound}}] \tag{3.7}
$$

where k_1 and k_2 can take on the value 0 or 1. The activation function is thus encoded in the genome by k_1, k_2 and the threshold parameters $\theta_1, \theta_2 \in \mathbb{R}$.

It may appear that ψ_{down} and ψ_{up} are mutually redundant as one type can be obtained from the other by reversing the signs on all the weights. However, retaining both increases diversity in the evolution because a single 2-bit 'gene' is required to encode the threshold function and only one mutation suffices to convert ψ_{down} into ψ_{up} or *vice versa* as opposed to changing the sign of every weight.

3.3.4 Evolution and Development

A population of ANT controllers is evolved in an artificial Darwinian manner [75]. The 'genome' for a controller contains a 'gene' for each cell with a specifier D used to distinguish the functionality (between motor control, decision and tissue). A constructor protein (an autonomous program) interprets the information encoded in the gene and translates this into a cell descriptor protein (Figure 3.4). The gene 'activation' parameter is a binary flag resident in all the cell genes and is used to either express or repress the contents of the gene. When repressed, a descriptor protein of the gene content is not created. Otherwise, the constructor protein 'grows' a cell. Each cell position is specified in reference to a seed-parent address. A cell-death flag determines whether the cell commits suicide after being grown. Once again, this feature in the genome helps in the evolutionary process with a cell committing suicide still occupying a volume in the lattice although it is dormant. In otherwise retaining its characteristics, evolution can decide to reinstate the cell by merely toggling a bit through mutation.

Motor Control Neuron Gene

Specifier	Reference Address	Position			Activation Function Parameters								Gene Activate	Cell Death	Replication Prob.	Output Behaviour	Reference Pointer
D	A	x	y	z	w_1	w_2	...	w_v	θ_1	θ_2	k_1	k_2	G	C	R	k	P
Integer [0,2]	Integer	Integers			Integer Coding / Real [0,1]						Binary		Binary	Binary	Real [0,1]	Integer [0,b]	Integer

Decision Neuron Gene

Specifier	Reference Address	Position			Diffusion Param.			Diffusion Concentration	Activation Function Parameters							Gene Activate
D	A	x	y	z	d_x	d_y	d_z	c	w_1	w_2	...	w_v	θ_1	θ_2	k_1	k_2	G
Integer [0,2]	Integer	Integers			Integers [1,3]			Integers [0,1]	Integer Coding / Real [0,1]						Binary		Binary

Tissue Gene

Specifier	Neuron Replication Prob.	Neuron Replication Ratios		Seed Address
D	T_r	n_d	n_m	T_s
Integer [0,2]	Integer Coding [0.001,0.1]	Integers [1,10]		Integer

Figure 3.4: ANT gene map showing tissue, motor neuron and decision neuron genes.

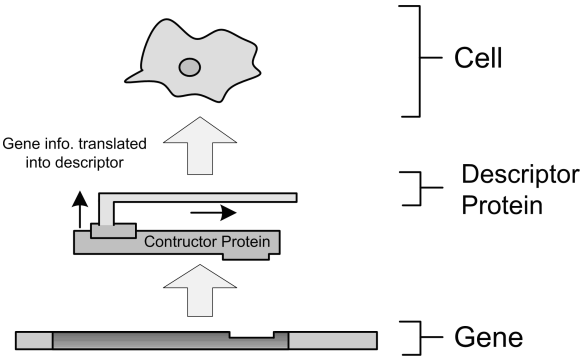


Figure 3.5: Genes are 'read' by constructor proteins that transcribe the information into a descriptor protein which is used to construct a neuron. When a gene is repressed, the constructor protein is prevented from reading the gene contents.

In turn mutation (manipulation of gene parameters with a uniform random distribution) to the growth program results in new cells being formed through cell division. The rate at which mutation occurs to a growth program is also specified for each tissue and is dependent on the cell replication probability parameter T_r. This probability parameter is used to determine whether a new cell is inserted. Cell division requires a parent cell (selected with highest replication probability relative to the rest of the cells within the tissue) and involves copying $m\%$ of the original cell contents to a daughter cell (where m is determined based on uniform random distribution), with the remaining cell contents initialized from a uniform random distribution. This process models a gene duplication process with the first $m\%$ being a redundant copy of an existing gene and the remaining contents being malformed thus resulting in a subfunctionalization-type process [172].

The cell type of each new cell is determined based on the ratio of motor control neurons to decision neurons, a parameter specified in the tissue gene. The new neuron can be located in one of six neighboring locations (top, bottom, north, south, east, west) chosen at random and sharing a common side with the parent

and not occupied by another neuron.

3.3.5 Crossover and Mutation

Within ANT, the genome is compartmentalized, through a direct encoding scheme, with each gene specifying the characteristics of each neuron within the tissue. Crossover involves interchange of a set of genes between the two parents. Before crossover, a parent affinity parameter, $\varpi \in \{0, 1\}$ is chosen at random for each child genome. The affinity parameter is used to establish if each child genome has closer 'affinity' to one of its parents (either parent A or parent B). Thus if $\varpi = 0$ then the genome has closer 'affinity' to parent A and $\varpi = 1$ if it has affinity with parent B. Once the affinity parameter is established for a child genome, a crossover plane is drawn along the same position shown for the two parents in Figure .

It should be noted that each gene remains intact and the crossover operation does not result in an arbitrary segmentation and interchange of gene contents. Each neuron has a unique position parameter $\boldsymbol{\lambda} = (l, m, n)$ relative to rest of the neurons within the tissue, a crossover is performed by drawing a plane (with a normal vector parallel to the *x* or *y*-axis) separating the tissue. Cell genes within the parent genome located on the side of the plane closer to the origin is copied directly onto the child genome with the associated 'affinity' parameter and the remaining genes are interchanged between the parents based on a 'compatibility criterion' (Figure 3.6).

The 'compatibility criterion' imposes the following condition, that gene for neuron $N_{\boldsymbol{\lambda}_1}$ from parent A and $N_{\boldsymbol{\lambda}_2}$ from parent B could be interchanged if $\boldsymbol{\lambda}_1 = \boldsymbol{\lambda}_2$, i.e., have the same position after development and only when *both* genes are expressed or repressed during development. Thus child 1 with $\varpi = 0$ (affinity to parent A) assumes the gene for ${}^{B}N_{\boldsymbol{\lambda}_2}$ and child 2 with $\varpi = 1$ (affinity to parent B) assumes ${}^{A}N_{\boldsymbol{\lambda}_1}$. If the compatibility criterion is not met, then no interchange occurs, and thus ${}^{A}N_{\boldsymbol{\lambda}_1}$ is passed onto child 1 and ${}^{B}N_{\boldsymbol{\lambda}_2}$ is passed onto child 2. Thus if ${}^{A}N_{\boldsymbol{\lambda}_1}$ is not expressed in parent A and ${}^{B}N_{\boldsymbol{\lambda}_1}$ is expressed in parent B, then this pair of genes fail the 'compatibility criterion.'

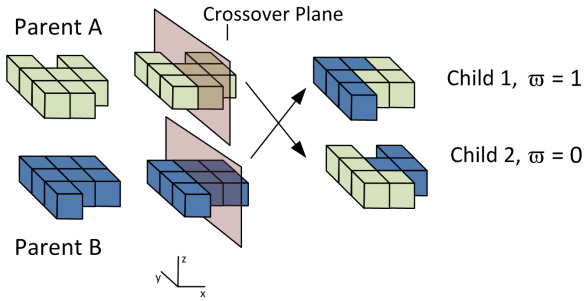


Figure 3.6: A crossover operation between two ANT parents. 'Compatible' neuron genes are interchanged as shown resulting in two offspring.

3.3.6 Phenotypic Neutrality

We argue that one of the advantages of a neural regulatory system is that it is able to facilitate phenotypic neutrality. We define phenotypic neutrality as a component within the phenotype when altered does not impose any fitness change to the rest of the system. In this section, we show that an increase in phenotypic neutrality increases the probability of finding solution networks as defined in Section 2.9.1. The probability of finding an m-feature solution network for a given task is presented in (2.16). Taking the derivative of $P_m(n)$ with respect to p_{neut} we get the following:

$$\frac{dP_m(n)}{dp_{\text{neut}}} = \sum_{k_1=0}^{k_{1=1}} \cdots \sum_{k_m=0}^{k_{m=1}} nC_{k_1,\cdots,k_m} \left(\sum_{i=1}^{m} k_i p_i + p_{\text{neut}} \right)^{n-1} = nP_m(n-1) \tag{3.8}$$

It follows that when $P_m(n) > 0$, i.e. $p_1, \ldots, p_m > 0$ then we get $\frac{dP_m(n)}{dp_{\text{neut}}} = nP_m(n-1) > 0$, for $n > m$ and because P_m and $\frac{dP_m}{dp_{\text{neut}}}$ are polynomials with $\sum_{i=1}^{m} p_i + p_{\text{neut}} \geq 0$. With the derivative, $\frac{dP_m}{dp_{\text{neut}}} > 0$ and $P_m > 0$, the function is thus a positive increasing function with respect to p_{neut}. Increasing the probability of finding neutral neurons, p_{neut}, is expected to increase the probability, $P_m(n)$, of finding a m-feature solution network, with n hidden neurons and with $p_1, \ldots, p_m$ given as constants. Taking into account the following:

$$1 - p_{\text{neg}} - \sum_{i=1}^{m} p_i = p_{\text{neut}} \tag{3.9}$$

and by increasing p_{neut}, with $p_1, \ldots, p_m$ held constant, translates into decreasing p_{neg}, the probability of finding disruptive neurons that causes spatial crosstalk.

Therefore, for a given task, where the probability of finding each feature neuron from a uniform random distribution is constant, then to improve the probability of finding a solution network, one must increase the probability of finding neutral neurons. We argue that a regulatory approach allows for a network to increase the probability of setting a neuron to neutral, particularly when one single regulatory neuron can signal many others to shut-off. This one-to-many messaging could be accomplished through diffusion of chemicals from a single neuron to its neighbors. Coordinated interactions of multiple regulatory neurons (based on a coarse coding selection process) improves the resolution of the selection field thus helping better pinpoint and activate localized volumes.

More Supervision

An alternative method of increasing the probability of finding a solution, as had been discussed earlier, is to introduce more supervision. The effect of introducing more supervision would be to change the probability of finding feature neurons. For simplicity sake, let us assume that p_{neut} remains constant under this scenario. Then the change in probability of finding a solution network is given as follows:

$$dP_m(n) = \sum_{j=1}^{m} \frac{\partial P_m(n)}{\partial p_j} dp_j \tag{3.10}$$

where,

$$\frac{\partial P_m(n)}{\partial p_j} = \sum_{k_1=0}^{k_1=1} \cdots \sum_{k_{j-1}=0}^{k_{j-1}=1} \cdots \sum_{k_{j+1}=0}^{k_{j+1}=1} \cdots \sum_{k_m=0}^{k_m=1} nC_{k_1,\ldots,k_{j-1},1,k_{j+1},\ldots,k_m} \left(\sum_{i=1}^{m} k_i p_i + p_{\text{neut}} \right)^{n-1} \tag{3.11}$$

for $n \geq m$, where $\frac{\partial P_m(n)}{\partial p_j} \geq 0$, since $C_{1,\ldots,1} \left(\sum_{i=1}^{m} p_i + p_{\text{neut}} \right)^n$ is greater than the sum of the remaining elements in right side of (2.16); otherwise $P_m(n) < 0$. Then as $\sum_{i=1}^{m} k_i p_i + p_{\text{neut}} \geq 0$, the derivative of the element with coefficient $C_{1,\ldots,1}$ in (3.11) is also greater than zero and is greater than the remaining elements on the right side of (3.11). With $\frac{\partial P_m(n)}{\partial p_j} \geq 0$, then assuming $dp_j \geq 0$ for $j = 1 \ldots m$ then it follows that $dP_m(n) \geq 0$. Hence by increasing supervision one needs to ensure $dp_j \geq 0$ for $j = 1 \ldots m$ in order to guarantee $dP_m(n) \geq 0$. However if $dp_j < 0$ for some values of j and $dp_j > 0$ of other values of j, then it is not certain that $dP_m(n) \geq 0$. This result intuitively matches what would be expected with more supervision namely, increased supervision can reduce the probability of finding a solution when introducing incorrect assumptions related to a task domain. In contrast, increasing phenotypic neutrality implies coming up with a mechanism or form of organization to facilitate the process and hence is not task specific.

As we have shown from this analytical analysis, there are at least two ways of increasing the probability of finding a solution network. One would be to increase the phenotypic neutrality, p_{neut}. This, as we argue, can be done through neuroregulation. The alternative is to introduce more supervision that runs the risk of decreasing the probability of finding a solution network. When there is not enough domain knowledge of a task at hand or where it is intended for the controller to solve the task through minimal supervision, then increased phenotypic neutrality can help improve the chance of finding a solution network.

Homologous Alignment and Gene Neutrality

The competing conventions problem occurs when competing representations of a controller with similar fitness are mutually incompatible from crossover interactions, resulting in damage to the genotypes. The end result is slower convergence to a solution. ANT's genotype to phenotype mapping also allows for implicit protection of innovative features due entirely to the genotype-to-phenotype mapping and without the need for 'innovation markers'. What is meant by innovative features are unique traits in the genome. In the case of ANT, this can be a group of neurons located well outside the original tissue 'boundaries'. Figure 3.7 shows an example of a unique feature evident with Parent A, not evident in Parent B nor rest of the population.

Damage to such unique features may occur through unrestricted intermixing of genes as in the standard Genetic Algorithms (GAs) crossover operator. In other words, due to the stochastic nature of the selection

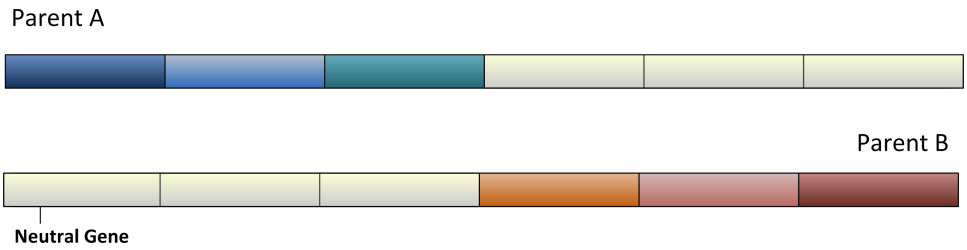


Figure 3.7: Genome of two parents with feature genes (shaded) and neutral gene (unshaded).

process within GAs, there is less of a chance (in comparison to ANT) that such features may remain as one 'intact' unit over a generation even if parent A is the fittest individual and descendants are subject to sexual reproduction. Although it's unlikely for such unique features to simply arise from a population, it will be even more improbable for these gene specified features to remain unaltered, as is, from inception. This result would make it improbable within GAs for new, highly fragile (with respect to crossover and mutation), nonincremental innovative traits to be retained after recombination, even if an individual is the fittest of the population.

Within ANT, unique features (i.e., unique grouping of neurons) do not get mixed among the children as readily and thus are not affected by the intermixing of genes due to crossover. This is maintained according to the compatibility criterion, where a similar (compatible) neuron gene must exist in both parents and both must be expressed or repressed for there to be a possibility of exchange. Where there is none, genes from a parent remains intact and gets passed onto a child genome with the affinity parameter ϖ matching the parent. Within the ANT genotype-to-phenotype mapping, it is still possible for competing network representations to be damaged due to crossover, provided the contents of the genes are incompatible and the networks occupy the same (phenotype) positions in the genome. To get an idea of the probability of such an occurrence, consider the gene representations shown in Figure 3.7. For Parent A and B, we may categorize the genes into $f+1$ categories, consisting of $f=3$ feature genes (coding feature neurons) and neutral genes (unshaded) with a genome consisting of $n=6$ genes and where the number of neutral gene sites $n_{\text{neut}} = n - f$. Based on this categorization, the neutral genes are indistinguishable. For this scenario, there exists $\frac{n!}{(n-f)!}$ distinguishable arrangement (where position matters). For two individuals, with f feature genes and n number of genes each, then $\left(\frac{n!}{(n-f)!}\right)^2$ possible arrangements exists. It follows that there exists, $\frac{n!}{(n-f)!} \cdot \frac{(n-f)!}{(n-2f)!}$ ways for two competing sets of gene representations with components occupying different spots on the genome thus ensuring no potential interference due to crossover. The probability of crossover interference P_{gi} among the two described individuals is as follows:

$$P_{gi}(n,f) = 1 - \frac{(n-f)!^2}{n!\,(n-2f)!} \tag{3.12}$$

and where $n \geq 2f$ genes. As can be seen, with an increase in the total number of genes (with extra neutral genes to spare) results in an inversely proportional decrease in probability of crossover interference. Setting $m! \approx \sqrt{2\pi} m^{m+1/2} e^{-m}$ according to the Stirling approximation, we get the following:

$$P_{gi}(n,f) \approx 1 - \frac{(n-f)^{2n-2f+1}}{n^{n+1/2}\,(n-2f)^{n-2f+1/2}} \tag{3.13}$$

where $n > 2f$. For us to minimize P_{gi} under these constraints, $\frac{n}{f} \to 0$ it follows that $n_{\text{neut}} \gg f$. Based on this analysis, the effects of the competing conventions problem can be reduced within ANT by having genomes with greater ratio of neutral genes to feature genes. However, this also means reduction in interchange between expressed genes from two different parents.

ANT allows for neutrality through use of gene and neural regulation. Furthermore, mutation to the development program results in addition of new neuron genes. Both of these features within ANT as we see from this analysis can help to reduce the effects of the competing conventions problem.

Genome Growth and Introns

The accumulation of *introns* or nonsense genes has been described to be a major problem in the field of Genetic Programming (GP) [30]. However, introns allow for genetic neutrality, an important facet of evolution [86]. With current GP approaches, intron accumulation results in increased computational inefficiency over time. Common techniques in controlling intron accumulation has been to prune the genotype regularly or explicitly impose a size limit on the genotype using the fitness function. Both strategies either lack biological plausibility or translate into more supervision. In addition, it is not clear if such approaches either help or hinder the evolutionary process, where a neutral gene maybe the source of future innovation.

Within ANT, two regulatory systems are at work. One is within the genome, namely a Gene Regulatory Network (GRN) that determines whether a gene gets expressed or not; another operates at the cellular level, where, neural regulation determines which neurons are excited into action. Inactive genes and neurons can be ignored from the evaluation process, which allows for improved computational efficiency over fully expressed genomes. To reduce computational inefficiency within a fully expressed genome, a parser is used to evaluate and identify nonsense elements to be ignored. The use of a parser will include an additional $O(n)$ computational penalty, where n is the number of gene modules. In contrast, a genome with gene regulatory functionality does not require this additional computational step and involves merely expressing genes with the 'active' flag turned on.

Neurons that will never be excited into action during an individual's lifetime could also be identified and ignored, if for example a motor control neuron is not within range of any active decision neuron chemical concentration fields or if the decision neuron has the 'cell death flag' turned on. Such interneuron dependencies can be easily identified during the development (mapping) process without the need to evaluate the neurons or identify nonsense elements.

It is noted that with traditional GP approaches, program size increases according to the square of the number of generations [92]. During crossover, a node within a tree is arbitrarily chosen from each parent and genetic code is exchanged about the node. This can result in significant genome growth in the children due

to mating of unbalanced trees during crossover and is biologically implausible. Owing to the homologous mapping scheme, genome growth within ANT is entirely dependent on the GRN growth model and the rate at which new genes are added to the genome (per generation) is determined by the parameter T_r in the ANT genome (Figure 3.4).

3.4 Related Work

Traditional machine-learning methods for task decomposition involve the supervisor decomposing the task and training separate 'expert networks' on the subtasks [81]. Arbitration among the expert networks is performed using a Product of Experts model [73] or cooperative Mixture of Experts model [81, 100]. With these automated task decomposition schemes, the gating function and the expert networks using different error functions (using backpropagation). In addition, the network topology is predetermined by the experimenter. This is in contrast to the ANT architecture, where the decomposition scheme is also a product of training process (artificial evolution).

For the ETDN architectures outlined in Chapter 2, decision networks (acting like gating functions) and expert networks were evolved together using a global fitness function [152]. We found larger BRL architectures (with more expert networks) tend to show better evolutionary training performance for the tiling pattern formation task. The decision neurons learned to limit the number of expert networks used thus preventing problems in over segmentation (over-fitting) to many expert networks. However, the BRL architecture is a fixed-network topology that requires user input in determining the network topology and the number of expert networks.

NEAT (NeuroEvolution of Augmenting Topologies) shows the potential advantage of evolving both the neuron weights and topology concurrently [144]. It is argued that growing the network incrementally through ('complexification') helps minimize the dimensional search space and thus improves evolutionary performance particularly for the benchmark double pole balancing task [144]. ANT is even more flexible and can be initialized with a large number of neurons without the need for incremental 'complexification.' This is accomplished using regulatory networks that can effectively suppress unnecessary/disruptive neurons (that NEAT lacks) while activating neurons specialized for specific input signals.

Inherent within NEAT are innovation markers embedded into the genome that are used to align corresponding homologous genes for crossover. The need for such a technique arises from having to deal with the competing conventions problem [129], in which similar network solutions are mutually incompatible during crossover. The innovation parameter scheme lacks biological plausibility as it is an external mechanism keeping track of new genes throughout the entire population and is also centralized. The modular nature of the ANT genome and its genotype-to-phenotype developmental process allows for homologous alignment during crossover, but without the need for centralized innovation parameter tracking.

Other techniques by Pujol and Poli [127] involve modular organization of the networks into subnet-

works. Here restrictions are imposed, such that different topologies may not be based on the same subnetwork, which limits the number of combinations possible. However, this technique also has advantages, namely, a whole subnetwork of neurons remains intact even after crossover. Homologous alignment of genes invariably reduces the combination of genes used to form a solution. Within ANT, instead of explicitly imposing restrictions on how subnetworks are mated, neuron genes between individuals are aligned for exchange based on the phenotype to genotype mapping scheme. Building from this procedure, it follows that networks of neurons remain intact, when there is no corresponding gene to perform homologous exchange within a partner individual (see Section 3.3.5).

AGE (Analog Genetic Encoding) [50, 105, 104] uses a biologically plausible technique, known as implicit encoding [18] to encode a genome consisting of sequence of characters in a finite alphabet into analog values that represent interconnection between gene elements. *Implicit encoding* techniques involve synthesis of coding regions that use distinct starting and ending characters (base-pairs) to encapsulate a coding region [50, 18]. As with other gene-regulation-based techniques, genes interact and are influenced by other genes in parallel networks.

Within AGE, genes are not necessarily segregated to one particular site but could rather include multiple parallel representations that need to be decoded, using an interaction map to form a phenotypic value. The system is gene-regulated through use of interaction maps to arbitrate/coordinate representation of multiple gene sites and allow for activation/inhibition. This is in essence a decentralized means of encoding that offers variable levels of redundancy and could result in high neutrality in the search space, where both facilitate evolvability [86].

AGE also provides additional levels of flexibility, enabling for variable-length genomes that can permit addition, insertion, deletion and substitution of single characters. AGE has also shown comparable performance with NEAT for the benchmark double-pole balancing task without velocity information [41]. Within AGE, parallel or alternate representation of neuron genes could exist but occupying different phenotypic positions, while ANT with a coarse-coding gene regulatory model for each neuron (see Chapter 5) allows for parallel representation of neuronal subelements.

The ANT genome described in this chapter only permits particular groupings of gene sites (forming a gene) limiting the flexibility of addition/deletion of individual characters. The AGE transcription process and genome in this regard more biologically plausible. While such added flexibility in AGE exists in biology, there are also exists autonomous genes that permit error-correcting gene sites or means of ignoring more obvious transcription errors, thus enabling a high degree of robust copying of gene segments. Nevertheless as Dawkins points out, evolution is about the improbable [35].

Another approach to evolving solutions to complex tasks involves use of encoding schemes that effectively partition the search space. This includes a multicellular developmental system through use of GRNs to control cell differentiation, division and death by Eggenberger [45]. Morphogenetic System (MS) originally used on POEtic [133] uses chemical diffusion. Eggenberger's earlier model demonstrates how cell

differentiation, cell division and cell death can be controlled by gene regulatory functionality and constructs a three-dimensional organism. In these two systems, the GRNs merely act on the developmental process in constructing the phenotype. In contrast, within ANT, regulatory networks are active throughout an individual's lifetime.

A more refined model by Gomez and Eggenberger [46] uses 'ligand-receptor interactions' allowing for one neuron to recognize/attach to a partner neuron and allow for emergence of Hebbian-type learning without specification of learning rules for a foveating artificial retina system. Astor and Adami's [1] NORGEV tissue architecture consists of cells arranged into a two-dimensional structure that perform logical functions. Cell replication and connections are formed through a gene-regulated development process using a Genetic Programming type command set.

Planar architectures such as NORGEV lack the laminar (columnar layering) of neurons found in cortical structures. It is well known that the cerebral cortex on average includes six layers [70]. Layering enables computational processing in stages, when one neuron alone cannot perform the required computation. While layering can be achieved by spatial segregation across a planar configuration, a comparably larger distance needs to be covered in comparison to a volumetric approach (Figure 3.8 for a comparison). This would be of concern if the tissue structure were to use diffusion of chemicals like nitric oxide for communication (see Section 5 on physical plausibility).

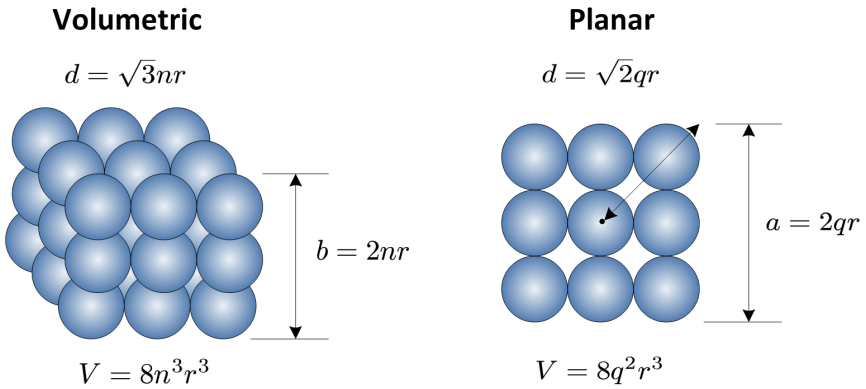


Figure 3.8: Volumetric and planar layout of neurons for chemical signalling.

From a chemical signaling viewpoint, if we were to equate the volume covered by a planar configuration to the volumetric configuration, then we obtain, $n = q^{2/3}$ where n and q are number of neurons stacked on each side of the configurations and r is the radius of the central neuron body. Then the ratio of the diagonal distances d (the furthest distance a chemical signal has to travel) between the volumetric and planar configuration is $\sqrt{3}q^{2/3} : \sqrt{2}q$ for $q \geq 1$. Thus for a planar configuration ($q > 1$), each chemical signal source will require a greater range, implying greater chemical production that in turn translates into more energy consumption. A fixed volumetric signaling scheme can also form natural barriers that can distinguish columnar bundles of neurons within range of the chemical signals from the surrounding neurons. Incidently bundles of neurons form what neurophysiologists call *hypercolumns* throughout the cortex [83].

From this brief analysis, the cortical structure modeled within ANT better matches existing observations from neurophysiology than NORGEV.

In the ANT architecture, regulation occurs during the developmental process in addition to when the tissue interacts with the environment. The decision neurons act much like gating neurons while helping to reduce the effects of *spatial crosstalk* [81] and perform sensory preprocessing enabling selection of *specialized* networks of neurons depending on the sensory input.

Another class of indirect developmental encoding schemes such as Artificial Embrogeny (AE) systems [146] produces phenotypes through recursive rewriting of the genotype code. These systems use an artificial chemistry as a grammar or model cellular metabolism and replication. Other recursive rewriting schemes include Cellular Encoding [64], edge-encoding [101], matrix-rewriting [88], L-Systems (see [143, 77]) and fractal rules [14]. L-systems have been mainly applied on locomotion tasks, where the body and brain (controller) are evolved concurrently. Some Artificial Embrogeny systems also combine multicellular development such as that by Dellaert and Beer [36] that can sense neighboring cells and use of Morphers (fixed topology modular feed-forward neural network without hidden layers) to signal cell replication and development with embryonal stages (DES) [47].

It has been argued that indirect developmental encoding schemes may effectively decrease the search space (by exploiting regularities and allowing for *pleiotropy*) but at the price of introducing a *deceptive fitness landscape* [134]. It has also been found that the overhead required for indirect encoding schemes appear to result in poor performance for smaller search spaces [134]. This presents a problem for task decomposition methodologies, where the objective is for solving subtasks that consist of smaller search spaces.

Although ANT's developmental program does not include recursive rewriting of gene contents during development, the model enables for gene duplication [118] and subfunctionalization [172] which results in production of daughter cells. This feature of genetic change, known as *neutral complexification* is also modeled within a multicellular AE model termed development with embryonal stages (DES) [47]. DES unlike other AE systems reduces the effects of pleiotropy and involves partitioning of the development process into stages. Mutation multipliers are evolved for each stage, with a scheme to reduce deleterious mutations using smaller multipliers for older embryonal stages.

3.5 Example Tasks

Evolutionary performance of the ANT architecture is compared with fixed network topologies (direct-encoding schemes) for control and robotic tasks. The pole-balancing control task has been considered a benchmark problem in control theory research [166, 145]. Extensive work has been done using various control paradigms to solve this task [166].

In addition, three other robotic tasks were chosen because it could be argued that self-organized task

decomposition strategies maybe necessary to accomplish the tasks given a 'global' fitness function. These tasks are also inspired by some remarkable behaviors evident in biology. By evaluating ANT on a wide set of robotic tasks, we hope to get an understanding of the applicability of the ANT architecture and its limitations. Tasks such as the phototaxis task with obstacles and the sign-following task are chosen to evaluate whether ANT can exploit short term memory (working memory) when evolved with limited supervision. The tile-formation task is used to perform comparison with the fixed network topologies introduced in Chapter 2. The intent of these comparisons have been to determine suitability of the controllers for various tasks, while keeping the evolutionary search parameters constant

The population size for phototaxis, tile formation and sign following tasks is $P = 100$, crossover probability $p_{\text{crossover}} = 0.7$, mutation probability $p_{\text{mutation}} = 0.025$ and tournament size of $0.06P$ (for tournament selection). A transcription error rate, $p_{\text{transcript}} = 0.005$, is used for all the experiments and mutations to the development program (T_r) results in growth of the genome at a rate of between 0 and 1 gene per generation.

3.5.1 Double-Pole Balancing

The pole balancing task has long been considered a benchmark task in control theory research [166, 145]. It involves controlling a system that is inherently unstable and has been acknowledged to be representative of a wider class of problems including walking. Extensive work has been done using various control paradigms to solve the single pole balancing task including use of reinforcement learning techniques [109, 11], cellular automata with gradient descent learning [96] and ADALINE networks using Widrow-Hoff Least Mean Square algorithm [165] to name a few. This type of control problem is also considered as a standard task for neural network controllers [166].

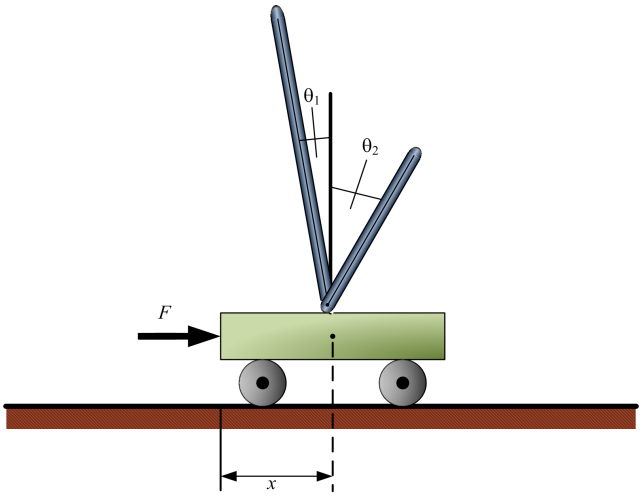


Figure 3.9: System schematic for the double pole balancing task.

Wieland [166] introduced the double pole balancing task as a more difficult extension to the single pole balancing. For the double pole balancing task (Figure 3.9), the objective is to balance a pair of poles, each hinged (1 degree of freedom) to a cart, within a finite track. Since the poles are of different lengths, the poles

react differently to the force applied and therefore the poles can be balanced concurrently. The equations of motion for the system are the following:

$$\ddot{x} = \frac{F - \mu_c \text{sgn}(\dot{x}) + \sum_{i=1}^{N} \tilde{F}_i}{M + \sum_{i=1}^{N} \tilde{m}_i} \tag{3.14}$$

and

$$\ddot{\theta}_i = -\frac{3}{4l_i}(\ddot{x}\cos\theta_i + g\sin\theta_i) + \frac{\mu_{pi}\dot{\theta}_i}{m_i l_i} \tag{3.15}$$

where $N = 2$ for the double pole balancing task. The system is defined by the state of six variables, the angle of the ith pole from vertical θ_i, the angular velocity of each pole $\dot{\theta}_i$ for $i \in \{1, 2\}$, position x of the cart on the track and its velocity $\dot{x}$. g is the acceleration due to gravity, m_i and l_i is the mass and half-length of the ith pole respectively, M is the mass of the cart, μ_c is the linear coefficient of friction of the cart on the tracks and μ_{pi} is the coefficient of friction for the ith hinge. Only a force, F, maybe exerted on the cart (directed either left or right) at regular time intervals to ensure the pole is balanced for a preset time, T, and the cart remains within the track boundaries. Two variations of the task are considered, namely one with velocity information and one without.

Pole Balancing Comparison

For this task, the controller outputs a force within the range of $[-10, 10]N$ once per timestep (every 0.02 seconds). The fitness measure is the number of timesteps a controller can keep the poles within $\pm 36^o$ from the vertical and ensure the cart remains within the 4.8 m long track. The poles are 1.0 m and 0.1 m long. Initially the long pole is at 1^o tilt and the short pole kept upright. The task is considered solved if the controller can balance the poles for $T = 100,000$ time steps (over 30 minutes).

For the double-balancing task with access to velocity information, ANT's motor neurons are arranged using feed-forward synaptic connections, with a real value encoding scheme to represent weights. The motor neurons use a standard sigmoid activation function (3.2) for the hidden layers and hyperbolic tangent function (3.4) for the output layers to match the other neuroevolutionary approaches tested [145]. The decision neurons use the modular activation function described by (3.5). While the ANT framework allows for greater flexibility in switching between activation functions for the motor control neurons, we are interested in specifically comparing ANT's regulatory functionality with the other evolutionary control approaches.

A second variant of the double pole balancing task considered more difficult [145] involves balancing the poles under similar conditions described earlier but without access to velocity information. In this scenario, ANT's motor neurons consist of fully recurrent synaptic connections, with a real-value encoding scheme to represent weights

The intention is for the controller to determine the velocities internally through feedback/recurrent connections. Hence, although less sensory input is provided more internal synaptic connections (owing to the recurrent connection) will mean a larger network and thus an expanded search space. In addition, a fitness

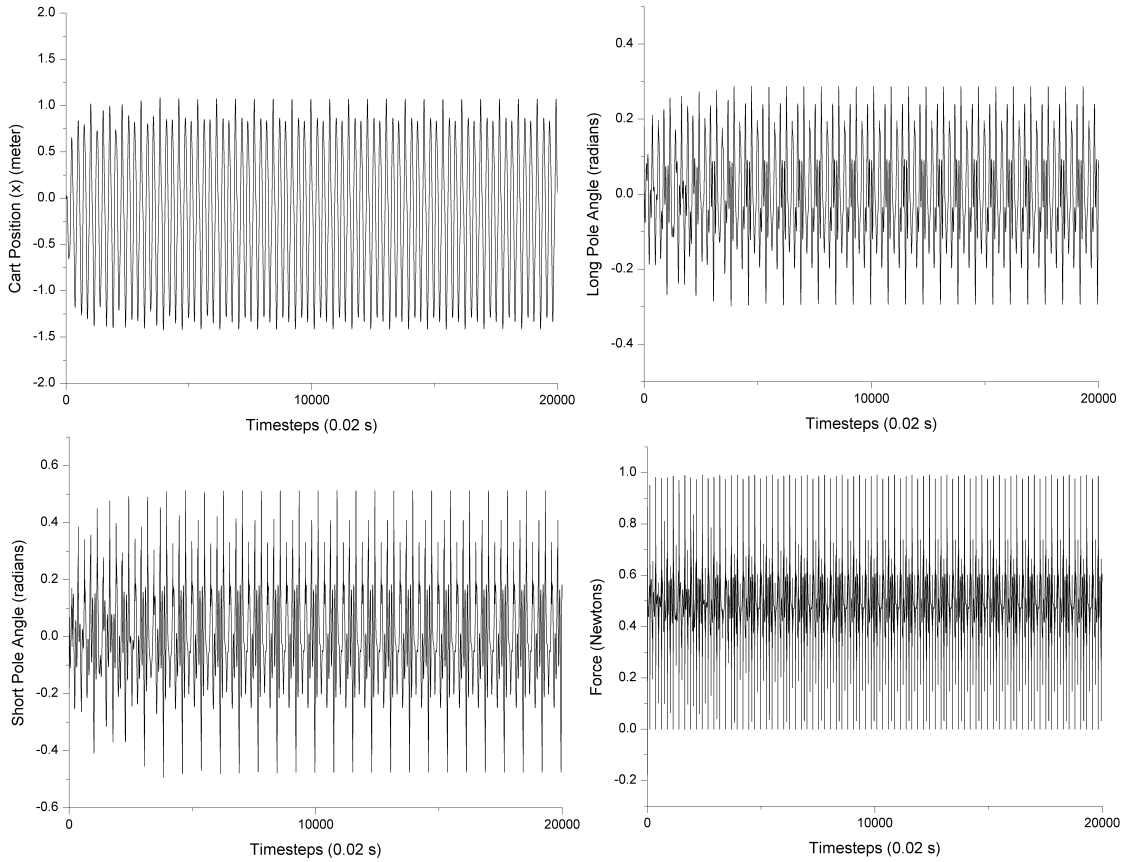


Figure 3.10: Double Pole System. ANT solution evolved with access to velocity information.

function introduced by Gruau *et al.* [64] is used to penalize back and forth oscillations. The Gruau *et al.* fitness measure is a combination of two fitness measures, $f = 0.1f_1 + 0.9f_2$. Where f_1 is defined below, and t is the number of timesteps the pole was balanced.

$$f_1 = t/1000 \tag{3.16}$$

f_2 is given as follows:

$$f_2 = \begin{cases} 0 & \text{if } t < 100 \\ \left(\frac{K}{\sum_{i=t-100}^{t} \left(|x^i| + |\dot{x}^i| + \left|\theta_1^i\right| + \left|\dot{\theta}_1^i\right| \right)} \right) & \text{otherwise,} \end{cases} \tag{3.17}$$

where $K = 0.75$. The denominators are the sum of the absolute values of cart state variables and the large pole over the last 100 timesteps before run completion and f_2 is intended to penalize behaviors that would otherwise simplify the task, namely, penalize swinging of the cart back and forth, without the need to compute the velocities.

In addition, using Gruau *et al.*'s criteria, the population best during each generation undergoes a generalization test. The generalization test involves evaluating how well the controller can balance the poles over

Table 3.1: Double Pole Balancing Task with Velocity Information (avg. of 50 trials, NEAT avg. of 120 trials).

Method	Genetic Evaluations	Standard Deviation	Generations	Population
Genetic Programming [138]	307,200	NA	150	2048
Conventional Neural Nets [166]	80,000	NA	800	100
SANE [113]	12,600	NA	63	200
ESP [59]	3,800	NA	19	200
NEAT [145]	3,600	2,704	24	150
ANT	3,450	2,831	69	50

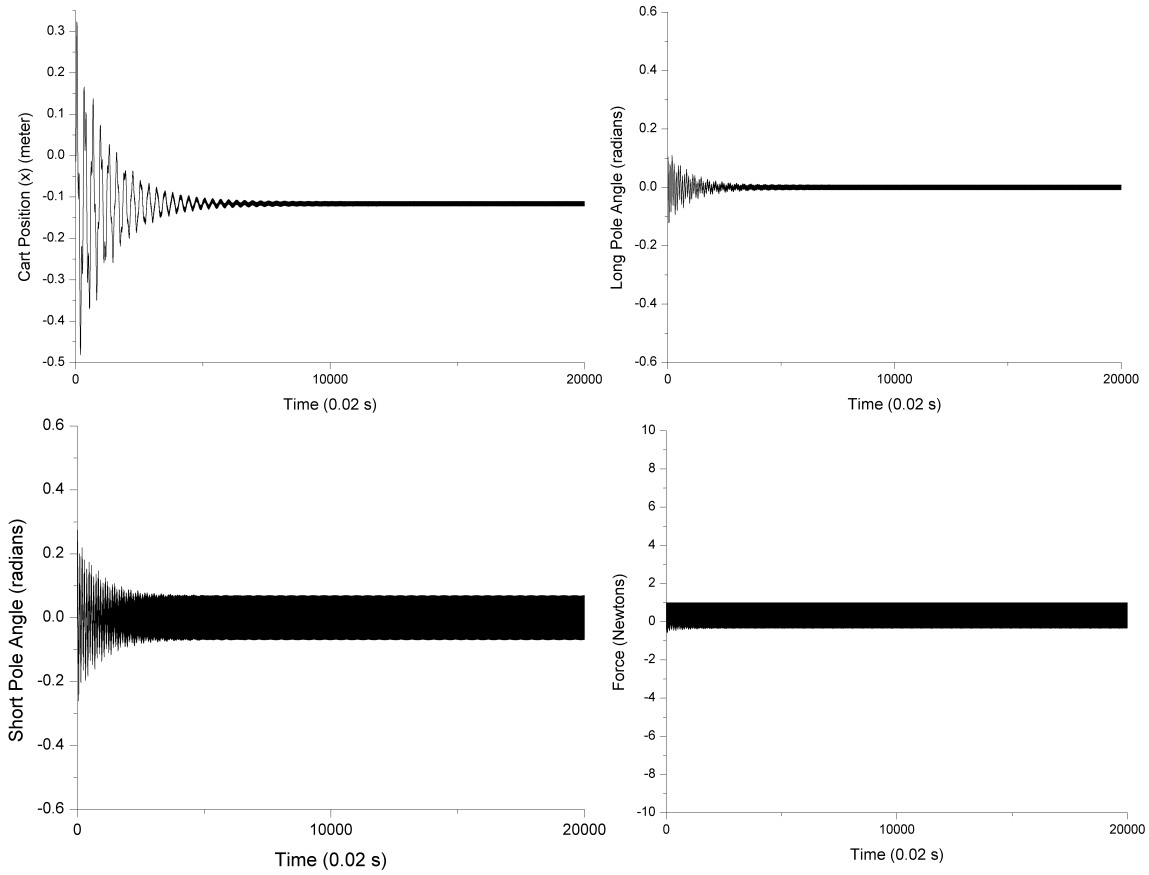


Figure 3.11: Double Pole System. ANT solution evolved using Gruau *et al.* fitness function.

625 different initial states for at least 1000 timesteps. The number of successes is labeled the *generalization performance*. A controller is considered a viable solution if it can generalize for at least 200 of the 625 initial states. Each starting state involves initializing the state variable, including x, $\dot{x}$, θ_1, $\dot{\theta}_1$ to 0.05, 0.25, 0.5, 0.75, 0.95 in the range of $[0, 1]$ rescaled to the range of the input variable giving $5^4 = 625$ initial states.

For this version of the task, motor control neurons within ANT are set to have fully recurrent connections and to use the sigmoid activation function (3.2) for the hidden layer and hyperbolic tangent (3.4) for the output layer to maintain consistency with the other neuroevolutionary approaches. While the ANT framework does not require these constraints be imposed, we are interested in determining the advantages (if any) of

Table 3.2: Double Pole Balancing Task without Velocity Information (avg. of 20 trials)

Method	Genetic Evaluations	Standard Deviation	Generalization	Population	Sucess/Restarts
CE [64]	840,000	NA	300	16384	NA
ESP [59]	169,466	NA	289	1000	NA
NEAT [145]	33,184	21,790	286	1000	NA
AGE [41]	25,065	19,499	317	1000	1/10
ANT	21,282	10,183	289	376	7/10
ANT*	22,241	9,841	317	376	7/10

the regulatory functionality within ANT over the other control approaches.

Among the other control approaches compared, Genetic Programming (GP) techniques used only mutation [138], while the conventional neural network uses both crossover and mutation [166]. SANE and ESP evolve populations of neurons assuming a fixed network topology [113, 59]. Individuals neurons from the population(s) are substituted into the placeholders of the user defined topology and evaluated. The fitness value is then shared among the neurons that participated in evaluation. With SANE, there exists one population of neurons for all the neuron placeholders within the supervisor specified network topology, while ESP maintain separate populations.

Table 3.1 shows previously published performance results of various evolutionary techniques and the performance of ANT for the double-pole balancing task with velocity information. The results are compared in terms of number of genetic evaluations needed before a solution controller (that can solve the task for T=100,000 timesteps) is found. Genetic evaluations are used as a performance measure because we are after techniques that can increase the efficiency of the evolutionary search process. Use of genetic evaluations as a performance measure does not explicitly take into relative computational cost of running each controller. It is assumed that the computational latency associated with running these controllers for a single timestep on hardware is much smaller than reading and/or waiting for an actuator to respond.

Consistent with earlier observations [59, 145], the controllers tend to solve the task by moving the cart back and forth enabling the poles to swing (Figure 3.1). This is a much simpler solution that involves keeping the poles swinging back forth rather than positioning them upright. However, the disadvantage with this strategy is that it is also energy inefficient, requiring moving the cart and poles back and forth constantly.

NEAT, a variable-length neural-network architecture requires the fewest number of genetic evaluations among the published results for the double pole balancing task with velocity information (Table 3.1). ANT requires even fewer genetic evaluations. It should also be noted that the standard deviation for ANT is 2,831 (with velocity information). Based on these results and known standard deviation for NEAT (2,704) [145], it can be concluded that the results are statistically significant except for ANT, ESP and NEAT. ANT requires a smaller population in comparison to NEAT or ESP, but with increased number of generations.

For the double pole balancing task without access to velocity information, ANT and AGE shows comparable performance (within a standard deviation). The standard deviation for ANT, NEAT and AGE are

10,183, 21,790 and 19,499 respectively [145, 41]. Both NEAT and AGE make use of 1000 individuals for the task, while ANT uses merely 376. When the controller population is found to be stuck in a local minimum and showing limited evolutionary progress, the system is restarted. Although ANT and AGE show statistically equivalent performance, AGE produces 1 solution every 10 restarts, while ANT produces solutions 7 times out of 10 restarts.

Using the Gruau fitness function, ANT solutions manage to considerably reduce the back and forth swinging of the cart as shown in Figure 3.11. In addition, ANT performs as well as the best neuroevolutionary approaches and shows comparable generalization performance. Consistent with the earlier observations (double pole balancing with velocity), ANT requires smaller populations but with increased number of generations.

3.5.2 Tile Formation

We also evaluated the evolutionary performance of the ANT architecture on the multirobot tiling pattern formation task discussed in Chapter 2 and in [152]. The task involves multiple simulated robots redistributing objects (blocks) placed in a two-dimensional grid world into a desired tiling lattice structure (Figure 2.8). The simulated robots need to come to a consensus and form one 'perfect' tiling pattern. This task is similar to a segment of the termite nest construction task that involves redistributing pheromone filled pellets on the nest floor for construction of support pillars [37].

The tile formation task may be manually decomposed into a set of subtasks such as foraging for blocks, redistributing block piles, arranging blocks in the desired tiling structure locally, merging local lattice structures, reaching a collective consensus and finding/correcting mistakes. However we are after a controls approach that performs task-decomposition through self-organization. A global fitness measure is used to facilitate self-organized task decomposition. We apply the Shannon's entropy function (see Section 2.6) to measure the arrangement of objects (blocks). When the objects are equally distributed in the desired tiling pattern, the fitness, $f = 1$.

For the task, each neuron uses a modular activation function described by (2.1) and motor control neurons are restricted to feed-forward connections. ANT is initialized to contain between 40 to 400 neurons (uniform distribution). This is intended to determine if there exists any adverse effect due to large topologies when using the regulatory framework. Figure 3.12 shows the performance of the ANT architecture for the 3×3 tile formation task. ANT's evolutionary performance exceeds the performance of the fixed topology BRL architecture with 16 expert networks (found to be best performing among several fixed topology architecture compared in Chapter 2) during earlier stages of evolution. As shown from the results, ANT is able to find the desired solutions within fewer genetic evaluations and with fewer task specific parameters than the BRL architectures (averaged over 120 evolutionary runs). The BRL architectures requires the experimenter predetermine network architecture (number of hidden neurons) and number of expert networks. With ANT these additional instance of human intervention is not required since the topology is a product of evolution.

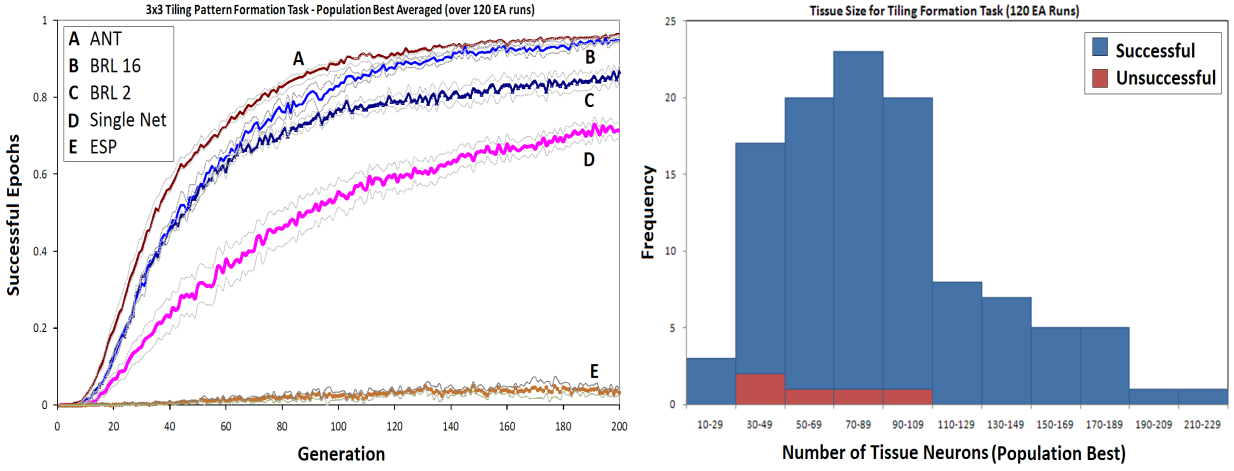


Figure 3.12: Evolutionary performance comparison for the tiling-formation task (left) and histogram of tissue size (right). Standard deviation is shown in gray.

Table 3.3: Sensor Input for the Phototaxis Task.

Sensor Variables	Function	Description
$V_1 \ldots V_4$	Object detection	Obstacle, block, empty space
G_1	Gripper status	Holding object, no object
L_1,L_2	Light Intensity	[0,10]

However, after a 100 generations, the performance of BRL (with 16 expert networks) converges to within one standard deviation. In addition, we also compare a fixed topology neural network consisting of four hidden neurons and one output neuron and BRL architecture with 2 expert networks. For the tile formation task, ANT can at least match the best performing fixed-network topology but without requiring additional task specific or experimenter intervention, making this approach more desirable.

3.5.3 Phototaxis

Biology is replete with examples of phototaxis or chemotaxis (chemical gradient) homing behavior among single cell organisms and insects. Such behaviors are often necessary for the survival of the organism. With these artificial controllers we are interested in determining if the methodologies developed here have the capacity solve what would be considered as a basic survival task for many organisms. Our robotic adaptation of the phototaxis task involves positioning a light source in the middle of a two-dimensional grid world as shown in Figure 3.13. A simulated robot has access to sensors shown in Figure 3.13 (right) and described in Table 3.3. $V_1 \ldots V_4$ are vision sensors that output discrete states to indicate obstacles, objects or empty space. B_1 is also a discrete sensor that indicates whether a robot is carrying an object or not. L_1 and L_2 are light sensors that return discrete normalized light intensity values within the range of $[0, 10]$.

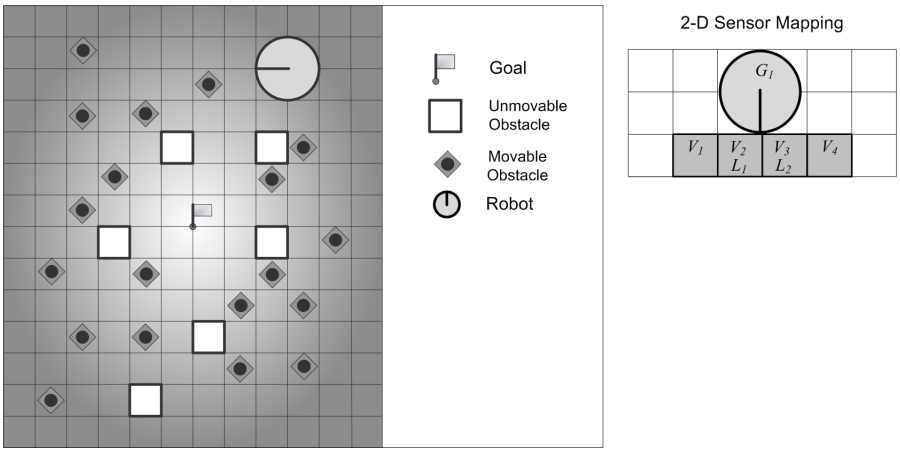


Figure 3.13: (Left) 2D grid world model for the phototaxis task. (Right) Input sensor mapping.

Table 3.4: Basis Behaviors for the Phototaxis Task.

Order	Behavior	Description
1	Pick-Up/Put-Down	Pick up or put down obstacle
2	Move forward	Move one square forward
3	Turn right	Turn 90° right
4	Turn left	Turn 90° left
5	Bit set	Set memory bit to 1
6	Bit clear	Set memory bit to 0

The simulated robot is positioned at a random starting position with a distance, d, from the light source (marked as the goal location) and has access to the set of output behaviors defined in Table 3.4. The light source is placed at the center of the grid world. The intention is for the controller to compute the light intensity gradient emanating from the source and accordingly follow a path leading to the goal location. The two-dimensional grid world also contains randomly distributed objects. There consist of two types of objects, including obstacles that cannot be manipulated by the simulated robots and other objects that can be picked up and moved around. The global fitness function, f, over n initial conditions for this task is simply $f = \sum_{I=1}^{n} \frac{f_i}{n}$ where $f_i \in \{0, 1\}$. $f_i = 1$ is when the robot reaches and stays at the goal location measured after T timesteps and $f_i = 0$ otherwise. T, the total time allowed for completion of the phototaxis task for an initial condition is reduced to ensure only a light homing behavior is a viable means of traversing to the goal location. Otherwise, the controller can reach the goal location by simply performing a random walk. Thus for this scenario, T is also dependent on the density of obstacles placed in the grid world. For the simulation experiments, the controller is evaluated on $n = 100$ different initial condition and each consists of a 20×20 world, 40 movable objects and 40 obstacles placed randomly, $T = 1.3d$ and d is in the range of $[5, 11]$.

For the controller to complete the task under the given time constraint, it needs to orient and home in on the light source. This functionality involves comparing the light intensity from L_1 and L_2 and turning toward direction of higher light intensity. Comparison of the evolutionary controllers show ANT controllers

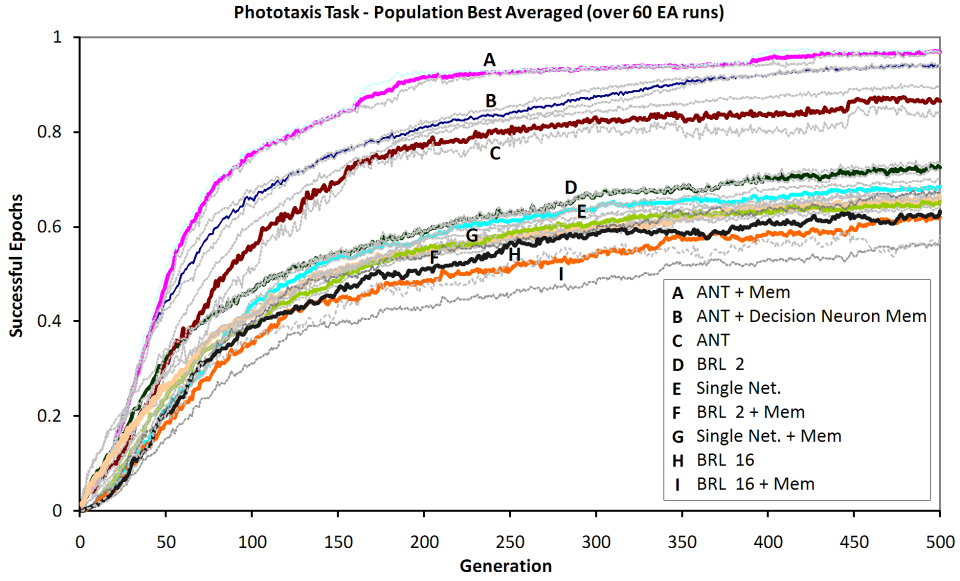


Figure 3.14: Evolutionary performance comparison for the phototaxis task.

with access to memory variables outperforming the BRL (with both 2 and 16 expert networks) and standard neural network topology consisting of 12 hidden neurons and 12 output neurons (Figure 3.14). The emergent behaviors are organized in two or more modes, one set of behaviors for light homing and another for obstacle avoidance. One bit of memory maybe used to keep track and help coordinate switching between modes among the neuron ensembles within the tissue. It should be noted that this ability to exploit memory occurs through self-organization. Access to a memory variable provides advantages particularly in dealing with detouring around obstacles that cannot be manipulated. In such cases, the controller may momentarily (over a few timesteps) transition into an obstacle avoidance mode, in turn triggering a series of behaviors to move away from the light source.

For the standard network topology, 4 of the 12 output neurons trigger one of the basis behaviors while remaining 8 output neurons are assigned to setting or clearing four memory bits. It is apparent that the fixed-topology networks tend to perform better without access to memory consistent with experiments by Nolfi for a robotic garbage collection task [115]. This is owing to a larger search space, when introducing the memory variables. Furthermore, it is apparent that this performance advantage of having access to memory variables is exploited by the decision neurons, that either inhibit or activate the motor control neurons. ANT shows noticeable improvement in fitness performance due to access to memory variables, while the fixed-topology networks appear unable to exploit this capability. This seems to indicate that the memory variables are used for neuroregulation activity rather than to store/retrieve data.

3.5.4 Sign Following

The evolutionary performance of the ANT framework is also demonstrated in simulation and hardware for an unlabeled sign-following task. The workspace is modeled as a two-dimensional grid environment with one holonomic robot (equivalent to a Khepera^{TM} robot, equipped with a gripper and camera) occupying four grid squares. For these tasks, the controller must possess several behaviours including the ability to decipher signs relative to the robot's current frame of reference, to remember the current sign while looking for the next one, and to negotiate obstacles (Figure 3.15). Each sign is color-coded and represents a waypoint (posted in a fixed frame of reference) that gives direction using one of four cardinal points to the next waypoint leading ultimately to the goal location.

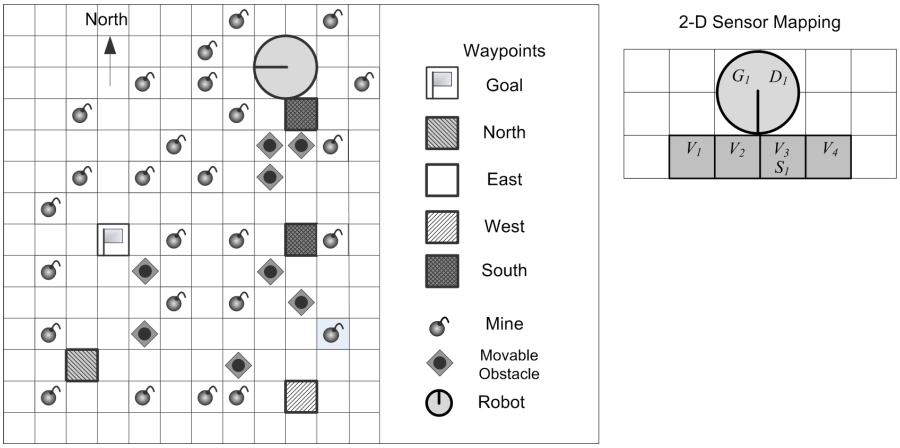


Figure 3.15: (Left) 2D grid world model for the sign-following task. (Right) Input sensor mapping.

Mines undetectable by the robot are randomly laid throughout the floor except along the pathway. Once a robot encounters a mine, it remains disabled for the remainder of its lifetime. The sensory input map is shown in Table 3.5 (see also Figure 3.15 right). The task has to be accomplished using a discrete set of basis behaviors specified in Table 3.6. These behaviors are activated based on controller output, are preordered and all occur within a single timestep. The robot is initially positioned next to the first sign, but the initial heading is randomly set to one of the four cardinal directions. Since the robot can only detect signs placed in front, it needs to go into a 'sign searching' mode and perform a sequence of 'turn left' or 'turn right' behaviors to detect the first sign. Once the first sign is detected, the robot then needs to transition into a 'sign following' mode requiring one bit of memory.

Deciphering signs relative to the robot's current frame of reference makes these tasks particularly difficult given a fitness function that measures success in terms of reaching the goal location. So when the goal is reached $f_i = 1$; otherwise, $f_i = 0$. The simulated robot's world is a 20×20 grid with 80 uniformly distributed obstacles and 40 randomly distributed mines (except along the path to the goal). The fitness is averaged over 100 runs with different initial conditions, the elapsed time for each run being limited to $T = 100$ timesteps.

The evolutionary performance of various control system architectures is compared for the sign-following

Table 3.5: Sensor Input for the Sign-Following Task.

Sensor Variables	Function	Description
$V_1 \ldots V_4$	Object detection	Robot, block, no obstacle
G_1	Gripper status	Holding block, no block
S_1	Sign detection	Red, blue, orange, pink, green
D_1	Heading	North, east, west, south

Table 3.6: Basis Behaviors for the Sign-Following Task.

Order	Behavior	Description
1	Pick-Up/Put-Down	Pick up from V_2 or V_3 or put down obstacle on V_1 or V_4.
2	Move forward	Move one square forward
3	Turn right	Turn 90° right
4	Turn left	Turn 90° left
5, 7, 9, 11	Bit set	Set memory bit i to 1, $i = 1 \cdots 4$
6, 8, 10, 12	Bit clear	Set memory bit i to 0, $i = 1 \cdots 4$

task (Figure 3.16). ANT with access to memory variables is found to outperform all other controllers tested. In addition, as we see in Figure 3.16, the task is found to be intractable (due to performance stagnation) for predetermined fixed-network topologies that lack regulation (see Results and Discussion). ANT solutions (population best) evolved offline have been applied on a holonomic LEGO® NXT (Figure 3.18) and a Khepera™ robot (Figure 3.17).

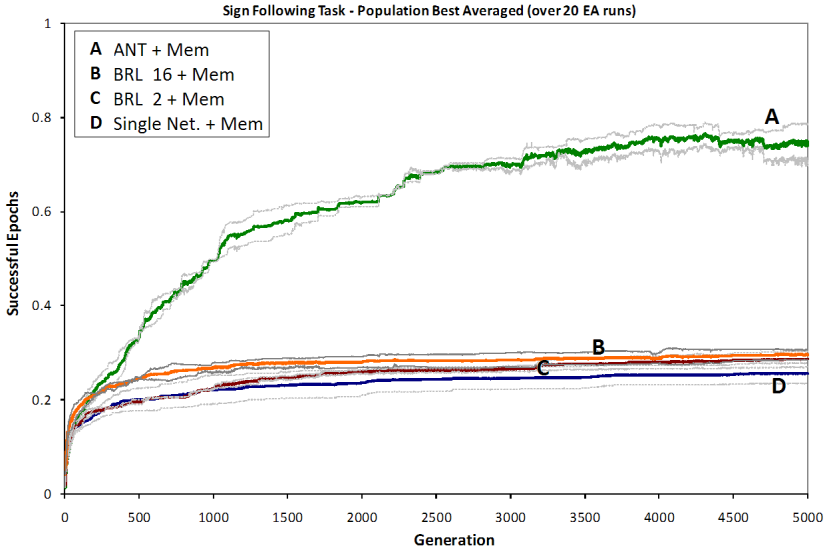


Figure 3.16: Evolutionary performance comparison for the sign-following task.

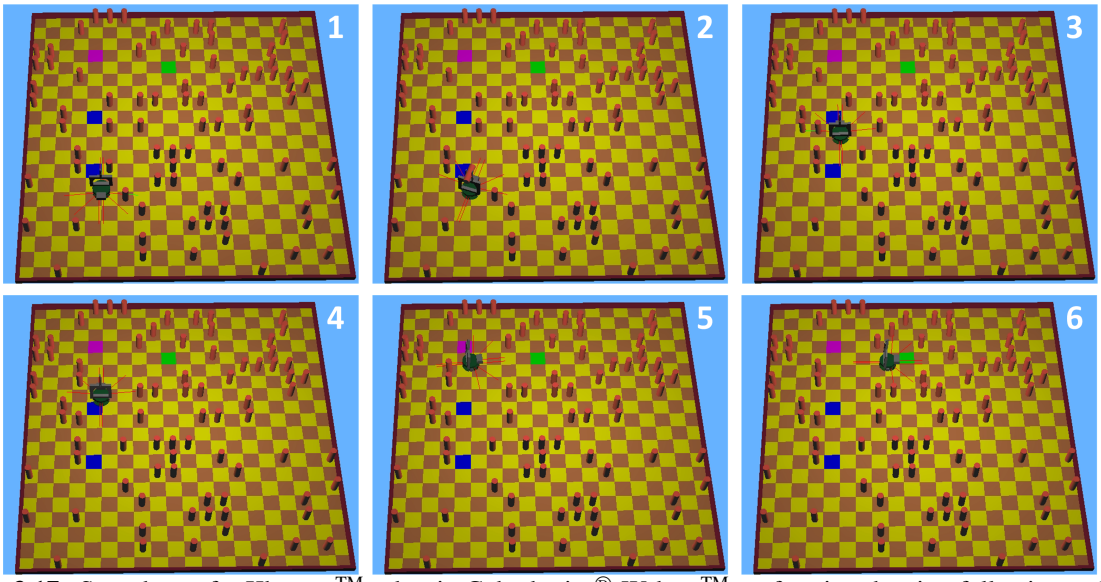


Figure 3.17: Snapshots of a Khepera™ robot in Cyberbotics® Webots™ performing the sign following task using an ANT controller. Frame 1 shows the robot pointing south next to a blue sign (coding for North). Frame 2 shows the robot having switched from a 'sign searching' to 'sign following' mode. The robot continues heading north after having interpreted the blue sign and turns right at the pink sign (East) and stops (Frame 6) having sensed the green sign (goal).

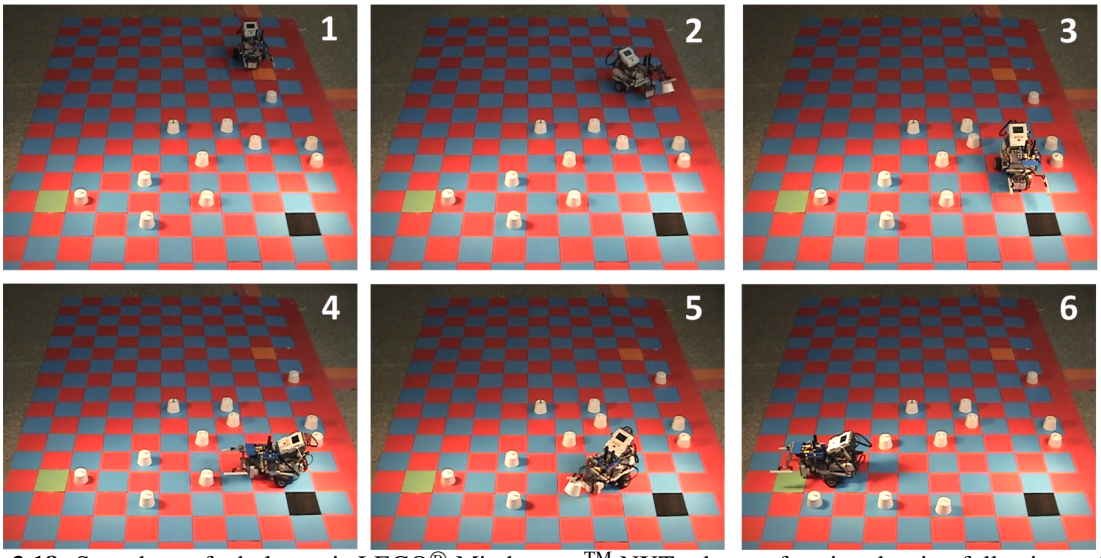


Figure 3.18: Snapshots of a holonomic LEGO® Mindstorms™ NXT robot performing the sign following task using an ANT controller. Frame 1 shows the robot interpreting an orange sign (coding for South) and turns right (robot frame of reference) once having sensed the black sign in Frame 4 (West). Frame 6 shows the robot stopping at the green sign (goal).

3.6 Results and Discussion

The evolutionary performance of the ANT architecture exceeds smaller fixed network topologies for the four different tasks presented here (Figure 3.16 , 3.14, 3.12; Table 3.2, 3.1). In addition, ANT with its

ability to grow in complexity (depending on the task) and exploit modularity managed to find solutions to the sign-following task (memory dependent) where fixed topologies appear to fail.

ANT based solutions require fewer genetic evaluations on average than NEAT or ESP for the two variants of the benchmark double pole balancing task. However, for the double pole balancing task without velocity information, ANT, NEAT and AGE are within the statistical variance. In addition, for this harder variant of the task, ANT showed comparable generalization performance to other algorithms. ANT also outperformed Cellular Encoding (an example of an variable length indirect encoding scheme) and genetic programming (GP) techniques for both tasks. AGE shows better generalization than ANT but require increased number of restarts. NEAT is the next closest performer and supposedly had two distinct advantages, a variable length topology that facilitates complexification, subfunctionalization and an encoding scheme that maintained 'innovation markers' for use during crossover. AGE includes a distributed encoding for gene sites, something that is comparable in functionality to the distributed organization of networks of neurons within ANT.

Both ANT and NEAT use different encoding schemes and for that matter the crossover operator is different from what is defined in standard GA literature [58]. Within NEAT, innovation markers as had been earlier analyzed, requires reading of gene content to allow for proper alignment during crossover. This poses several problems in terms of biological plausibility, including breaking with a decentralized system hypothesis. In contrast, the modular nature of the ANT genome, in addition to the genotype to phenotype mapping, shows that innovation can be protected without the need for explicit markers to keep track of these genetic trends. Subfunctionalization is also featured in ANT, except this is evident through a process of cell division (due to gene miscopy).

Although ANT can find solutions to the double-pole balancing task with comparable if not fewer genetic evaluations than ESP or NEAT, this is accomplished with increased number of generations but with smaller populations. ANT in comparison to ESP and NEAT includes additional overhead, owing to the presence of gene and neural regulatory networks. Coordination of these additional layers of control understandably would require increased number of selection cycles, but a decrease in the population size is generally expected to reduce search performance. Reducing the population size while keeping the selection ratio (tournament size) constant tends to reduce the diversity among the population. Reduced diversity beyond a critical threshold in turn is expected to result in search stagnation.

Nevertheless, the advantage of ANT appears to be that a smaller population of controllers doesn't appear to impact evolutionary performance as drastically as the other approaches. ANT with its regulatory functionality helps nurture neutral evolution with augmented functionality that is not readily expressed. With a selectionist regulatory functionality, it possible for a single ANT controller to consist of competing control approaches (not to be confused with the competing conventions problem), with some lying dormant some of the time. Multiple competing representations also offer some level of redundancy. While each individual exists as a composite of multiple control approaches, deleterious mutations to one representation can result

in another representation augmenting its reduced capability instead of resulting in total loss of functionality. It is hypothesized that regulatory functionality can augment the need for bigger populations by helping to maintain sufficient diversity within fewer individuals.

3.6.1 Network Size and Morphology

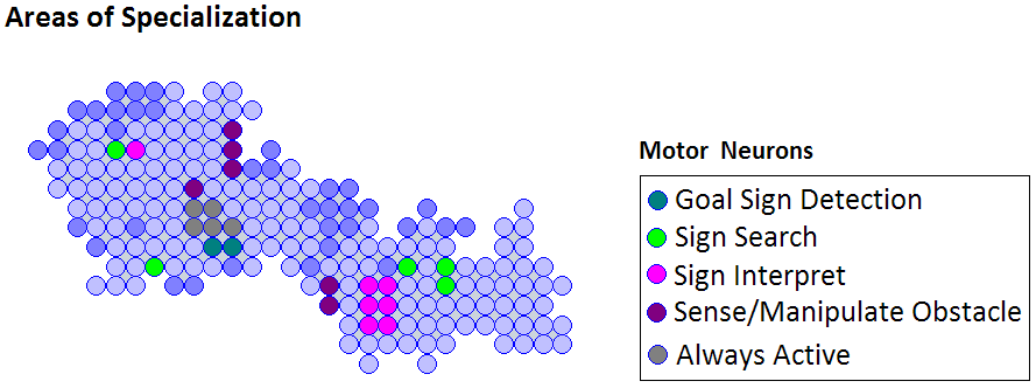


Figure 3.19: Areas of specialization shown for a typical ANT controller solution for the sign-following task. It should be noted that not all motor neurons are active simultaneously within the clusters.

Analyzing the 3-D morphology, active segments (consisting of specialized networks) are distributed sparsely throughout the tissue (Figure 3.19). These networks do not appear to decompose the task according to 'recognizable' distal behaviors but as a set of emergent proximal behaviors (proximity in sensor space) [115]. Larger tissue structures do not seem detrimental to the evolutionary performance because the regulatory system appears to evolve the ability to suppress unnecessary/disruptive neurons within the tissue quickly. There appears to be a steady reduction in the number of active neurons as it converges to a solution (Figure 3.20). During the early phase of evolution, there is greater network activity enabling sampling of more neurons followed by a steady reduction. Convergence to a solution appears to result in the inhibition of cells that add noise to the system (reduction in spatial crosstalk [81]) indicative of a regressive selection process [43, 131].

There appears to be a noticeable improvement in evolutionary performance of larger tissue structures over smaller ones. A histogram of population best during successful runs (with a success rate of less than 0.9) compared to population best during unsuccessful runs (success rate less than 0.1) appears to confirm these observations (Figure 3.12 right). The ANT architecture is more effective at being able to decompose a task and handle the subtasks using simpler, specialized networks consisting of fewer neurons than the fixed architectures. This results in fewer weights having to be tuned and thus a smaller search space. It should be noted that the number of active or selected neurons in the tissue consists of a small fraction of the total. The additional inactive neurons, much like inactive genes, act as neutral sites. As had been discussed earlier, most changes in evolution are theorized to be the result of neutral changes [86].

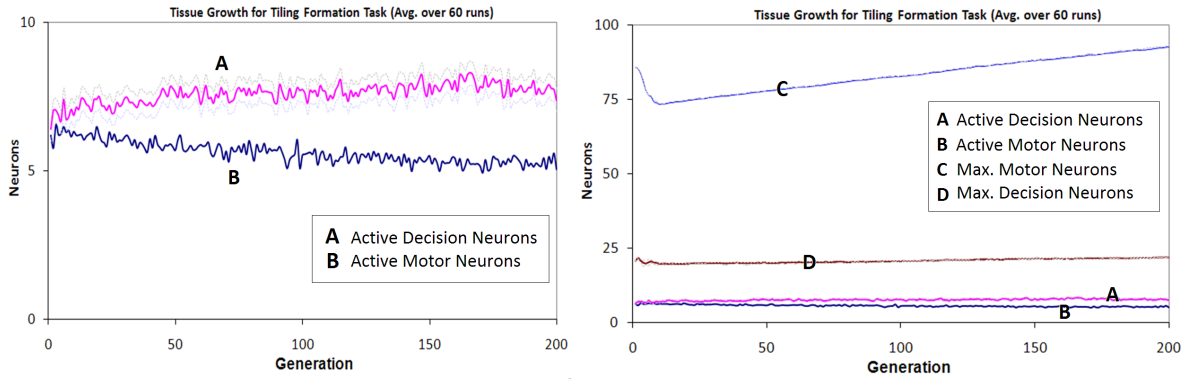


Figure 3.20: (Left) Number of active neurons (population best for tiling-formation task). (Right) Comparison of max tissue size.

3.6.2 Use of Memory Access Behaviors

The most promising result from our experiments is the successful exploitation of memory access behaviors by the ANT architecture. ANT is able to exploit the use of memory variables for the phototaxis and sign-following task. For the phototaxis task, the use of memory variables is not necessary to complete the task, while it is necessary for the sign-following task, where the controller needs to keep track of going from sign-searching to sign-following mode. In contrast, a standard fixed topology, feedforward network and the BRL architectures are unable to exploit this functionality resulting in lower performance (owing to a larger search space). This observation corroborates with Nolfi's experiments for a robotic garbage collection task, where fixed network topologies with access to memory showed reduced evolutionary performance [115]. This is owing to a larger search space, when introducing the memory variables. Furthermore, it is apparent that this performance advantage from having access to memory variables is mostly exploited by the decision neurons, that either inhibit or activate the motor control neurons. For the phototaxis task, ANT shows noticeable improvement in fitness performance due to access to memory variables, while the fixed-topology networks appear unable to exploit this capability. This seems to indicate that the memory variables are used for neuroregulation activity rather than to store/retrieve data.

3.6.3 Basis Behaviours and Coupled Motor Primitives

For the sign following task experiments presented in Section 3.5.4, a set of ordered basis behaviours as shown in Table 3.6 were used. The preordered basis behaviours, while not necessarily task specific does require experimenter input. In addition, human intervention is involved in limiting the number of basis behaviours available for the controllers. What we wish to understand is whether preordered basis behaviours are necessary for finding successful solutions for the sign following task.

In this section we consider an alternative setup, where the ANT controllers are provided building blocks for the basis behaviours in the form of motor primitive [103] sequences. The motor primitives are taken as discrete voltage signals over a discrete time window applied on DC motors as shown in Figure 3.21 and as

Table 3.7: Coupled Motor Primitives for the Sign-Following Task.

Neuron ID	Behavior	Coupled Motor Signals
1	Pick-Up Square V_1	(Grip Pan -1 || Grip tilt -1 (Grip Latch 1 (Grip Pan 1 || Grip tilt 1)))
2	Pick-Up Square V_2	(Grip Pan -1 (Grip Pan -1 || Grip tilt -1 (Grip Latch 1 (Grip Pan 1 (Grip Pan 1 || Grip tilt 1)))))
3	Pick-Up Square V_3	(Grip Pan 1 || Grip tilt -1 Grip Latch 1 (Grip Pan -1 || Grip tilt 1)))
4	Pick-Up Square V_4	(Grip Pan 1 (Grip Pan 1 || Grip tilt -1 (Grip Latch 1 (Grip Pan -1 (Grip Pan -1 || Grip tilt 1)))))
5	Put-Down Square V_1	(Grip Pan -1 (Grip Latch -1 (Grip Pan 1 || Grip tilt 1)))
6	Put-Down Square V_2	(Grip Pan -1 (Grip Latch -1 (Grip Pan 1 (Grip Pan 1 || Grip tilt 1)))))
7	Put-Down Square V_3	(Grip Pan 1 (Grip Latch -1 (Grip Pan -1 || Grip tilt 1)))
8	Put-Down Square V_4	(Grip Pan 1 (Grip Latch -1 (Grip Pan -1 (Grip Pan -1 || Grip tilt 1))))
9	Move Forward	Left Motor 1 || Right Motor 1
10	Turn Right 90°	Left Motor 1 || Right Motor -1
11	Turn Left 90°	Left Motor -1 || Right Motor 1
13	Pivot Right	Left Motor 0 || Right Motor -1
14	Pivot Left	Left Motor 0 || Right Motor 1
15	Pivot Right	Left Motor 1 || Right Motor 0
16	Pivot Left	Left Motor -1 || Right Motor 0
17, 19, 21, 23	Bit set	Set memory bit i to 1, $i = 1 \cdots 4$
18, 20, 22, 24	Bit clear	Set memory bit i to 0, $i = 1 \cdots 4$

Table 3.8: Motor Signals for the Sign-Following Task.

Neuron ID	Motor Sequence or Signal	Net Behaviour
1	Left Motor 1	Left Motor Forward
2	Left Motor -1	Left Motor Reverse
3	Right Motor 1	Right Motor Forward
4	Right Motor -1	Right Motor Reverse
5	Grip Pan Position Bit 1	Pan Position Bit 1
6	Grip Pan Position Bit 2	Pan Position Bit 2
7	Grip Pan Forward	Gripper Pan Forward
8	Grip Pan Reverse	Gripper Pan Reverse
7	Grip Tilt 1	Gripper Tilt Down
8	Grip Tilt -1	Gripper Tilt Up
9	Grip Latch 1	Gripper Latch
10	Grip Latch -1	Gripper Unlatch
11	Drive Motor First	Power drive motors in parallel first
12, 14, 16, 18	Bit set	Set memory bit i to 1, $i = 1 \cdots 4$
13, 15, 17, 19	Bit clear	Set memory bit i to 0, $i = 1 \cdots 4$

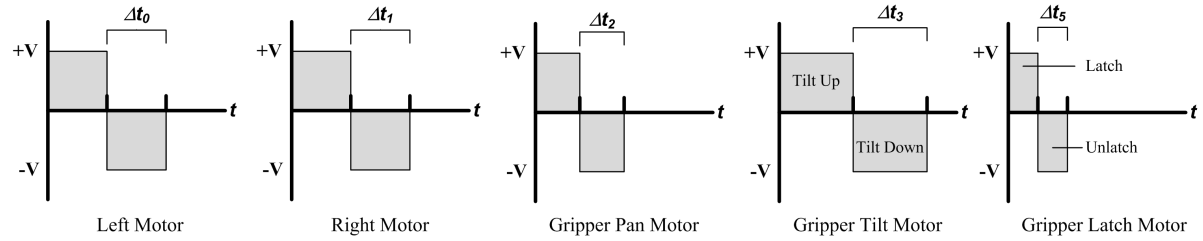


Figure 3.21: Motor primitives composed of discretized voltage signals shown for a simulated robot (used for the sign following task.)

arguments to the motor primitive commands in Table 3.7. These voltage output signals feed to the actuators and can be in one of three states, $\{1, 0, -1\}V$ for a discrete time window, Δt_n, $n \in \{0, 1, 2, 3, 4, 5\}$ as shown. In addition, each actuator takes on a default voltage value of 0. The actual value of V, the voltage constant, is dependent on the actuator.

Tissue Gene

Specifier	Neuron Replication Prob.	Neuron Replication Ratios		Seed Address	Output Signal Order						
D	T_r	n_d	n_m	T_s	o_1	o_2	o_3	...	$o_{\varepsilon-2}$	$o_{\varepsilon-1}$	o_ε
Integer [0,2]	Real [0.001,0.1]	Integers [1,10]		Integer	Integer [1, ε]						

Figure 3.22: Modified tissue gene that includes order of execution of motor primitive sequences.

The ANT controller also needs to determine the order of execution of these motor primitive sequences. The modified tissue gene is shown in Figure 3.22. The order of the output motor primitive sequences are evolved as additional parameters in the tissue gene and is read starting from the left. The elements of the table, $o_1, \cdots, o_\epsilon$ contain the Neuron ID values. The order is randomly initialized when starting the evolutionary process and with each Neuron ID occupying one spot on the gene. Point mutations to this section of the tissue gene involves swapping Neuron ID values between sites. Table 3.7 shows the repertoire of coupled motor primitives provided for the ANT controllers and thus $\epsilon = 24$ for this particular setup (Figure 3.22). In contrast to the higher level behaviors such as 'pickup' and 'putdown' from Table 3.6, behaviours listed here are further broken down into specific sequences of motor primitives such as 'pickup from square 1' or 'putdown on square 2'. Furthermore, the motor primitives are coupled, where for example the left drive motor and the right drive motor are executed in parallel (indicated using $||$). Under this setup, it is still possible for the controller to execute a sequence of motor primitives in a serial fashion. For example the controller may execute 'pivot right' (Neuron ID 13) followed by a 'pivot left' (Neuron ID 16).

Table 3.8 shows a second setup evaluated using the ANT controller. In this setup each Neuron ID refers to a specific motor/actuator or position signal instead of a coupled motor signal. As in the previous setup, the default voltage signal is 0. The controller executes all the motor signal in parallel during each timestep, with the voltage and signal values being specified by the active Neuron IDs. Thus unlike the previous setup the tissue gene doesn't contain a table of Neuron ID entries specifying order of execution.

Figure 3.23 shows the evolutionary performance results using ANT ordered (coupled) motor primitives

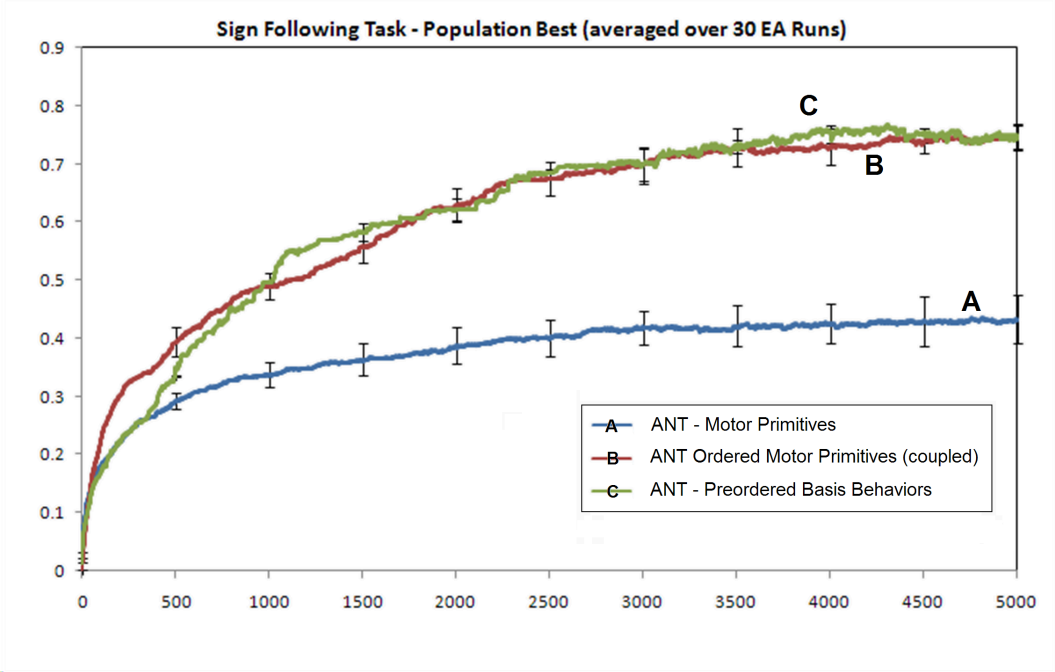


Figure 3.23: Evolutionary performance comparison of ANT-based solutions (using motor primitives and basis behaviours) for the sign-following task.

and using motor primitives (all executed in parallel). Surprisingly the results using the coupled motor primitives show comparable performance to the setup using ordered basis behaviours. This indicates neither preordering the basis behaviours nor specifying sufficiently higher level basis behaviours is necessary for evolving successful solutions. However, when the output neurons within ANT represent specific motor signals (Figure 3.23) we see a substantial decrease in evolutionary performance (with no successful solutions). This suggests that 'physical' coupling of the motor primitives is a sufficient method for successful solutions to emerge. In theory, one could have couplings between the motor primitives emerge from the setup presented in Table 3.8, but this would entail coupling and synchronization of neuronal pathways within the ANT architecture. However, evolving and maintaining synchronization among multiple networks of output neurons (specifying motor signals) appears more difficult to perform than selecting from a repertoire of motor primitive sequences.

The motor primitive sequences shown in Table 3.7 are structured and constant, while the setup presented in Table 3.8 allows for dynamics execution of the primitives. The dynamic setup also allows for a much greater choice of motor primitive being activated and disabled in parallel but this limits the system's ability to sample and develop the necessary 'building blocks' (motor primitive sequences) in addition to solving the task (This is presuming that a good strategy in solving the task involves searching for and assembling 'building blocks' consisting of coupled motor primitive sequences). While in the case of the coupled motor primitive sequences, many of the necessary 'building blocks' needed for solving the task is already present and hence it is a matter of selectively executing the 'building blocks' (motor primitive sequences) in order

to solve the task.

3.6.4 Ablation Experiments and Coarseness

Figure 3.24 shows the evolutionary performance of the ANT topology (for the sign following task), along with ablated topologies for a range of coarseness. The intention in this set of experiments is to determine the role of both coarseness (diameter of chemical concentration field) and superpositioning of coarse chemical fields in the overall performance of ANT. It should be noted that both of these features affect the regulatory functionality within ANT and is thus indicative of the role of neural regulation in evolution.

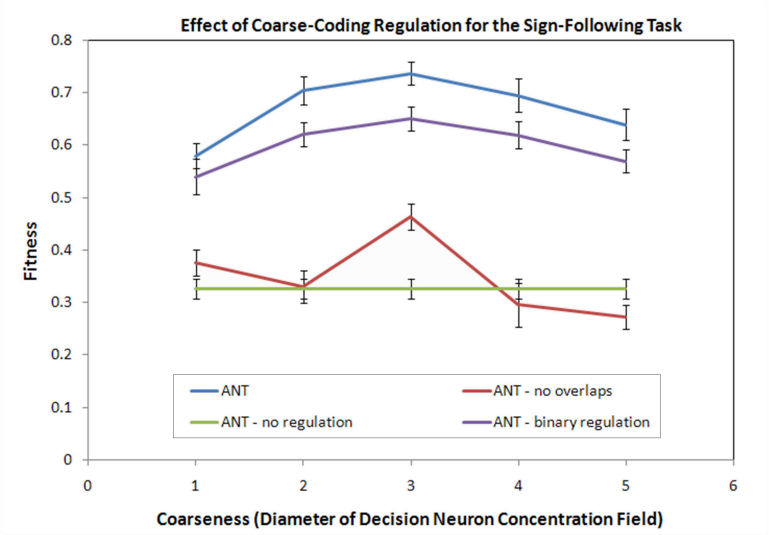


Figure 3.24: Effect of coarse-coding regulation for the sign-following task (population best, averaged over 60 EA runs). Error bars indicate standard deviation.

What is evident is that when the decision neurons are always active (neural regulation turned off), the controllers show substantial drop in evolutionary performance compared to ANT with neural regulation and this appears invariant to the coarseness parameter. It was found that none of the controllers evolved a solution with fitness of 1.0 when neural regulation is turned off and all the neurons remain active after development. This is equivalent to a fully expressed, variable-length topology and hence lacks any mechanism to selectively shut off components of the phenotype during its lifetime. Next we compare the performance with decision neurons able to perform neural regulation, but by disabling neurons selected from multiple overlapping concentration fields. This is to ascertain the role of coarse coding in the regulatory process. It is expected that with increased coarseness, more overlaps of concentration field would occur, given fixed tissue topology and thus more neurons will be inhibited. At the other extreme, with fine concentration fields, chance of overlap will be less but selection of neurons will be disparately distributed, hence it will be rather improbable to select groups of contiguous/interconnected neurons. For this optimum level of coarseness, while coarse coding interactions are inhibited, coarse selection fields will nevertheless select local clumps of interconnected neurons.

Significantly improved performance is evident when no restrictions are imposed on overlapping concentration fields. For binary regulation, concentration of neurotransmitter chemical has an upper bound thresholded to 1. This would result in selection of more neuron ensembles, as the threshold would result in a greater number of ties for the maximum concentration. As with other population coding methods, this approach offers greater redundancy; the information is distributed among multiple neurons but individual selection fields are more prone to alteration/damage since the concentration fields cannot be further strengthened beyond the upper bound.

3.7 Summary

Artificial neural tissue (ANT) addresses several limitations with fixed topology ETDNs presented in Chapter 2 that includes the need to specify the topology and number of expert networks. ANT is composed of gene regulatory and neural regulatory networks. Regulation is used to help coordinate limited resources toward performing control. The gene regulatory networks are used in the development process while the neural regulatory system dynamically activates and inhibits parts of the tissue through sensory preprocessing.

ANT can evolve solutions to control tasks without the need for complexification, where a phenotype starts with a single cell and incrementally increases in size as it is evolved. The neural regulatory framework developed within ANT is based on a coarse coding framework, originally developed for data representation in vision [72] and low-level motor control [2]. The ANT framework also has several biologically plausible enhancements that address a number of open problems in the field. ANT includes a homologous mapping system that addresses issues related to unbounded intron growth due to crossover and the competing conventions problem. It has been analytically shown that by increasing phenotypic neutrality, that damage caused from the competing conventions problem could be reduced. Furthermore, it has been analytically shown that maximization of phenotypic neutrality (Section 3.3.6) increases the probability of finding solution networks. It is argued that the coarse coding regulatory framework within ANT increases phenotypic neutrality by keeping potentially disruptive/unnecessary parts of the tissue neutral.

The evolutionary performance of ANT has been shown to exceed other EA based techniques for benchmark control problems such as the double pole balancing task. In addition, ANT shows better performance over other fixed topology architectures including BRLs for the multirobot tiling formation, single robot phototaxis and sign-following tasks. For the three latter tasks, ANT demonstrates the ability to perform task decomposition through limited supervision, in which different parts of the tissue perform specialized functionality. Unlike the use of *shaping* techniques, only a global fitness function is specified and it does not explicitly bias for particular solution strategies. Only a predefined set of sensory input and a set allowable basis behaviors are defined for each task. ANT is found to produce solutions to the sign-following task, an instance where predefined fixed-topology networks appear to fail. ANT is also able to better exploit memory variables to complete the phototaxis and sign-following tasks in comparison to the other approaches. Ablation tests of ANT's regulatory functionality for the sign-following task shows that the evolutionary

performance is dependent on the coarseness of the regulatory selection fields. Furthermore, ANT, without neural regulation, (consistent with the fixed topology networks) appears unable to find solutions to the sign following task.

In the following chapter, we shall consider the applicability of ANT controllers towards real-world control tasks in mining and in-situ resource utilization (ISRU). By providing limited supervision through use of a global fitness function, we are interested in determining whether ANT controllers can evolve the necessary control and coordination behaviors creatively. We further explore how these controllers maybe extended to control and coordinate multiple systems and determine the advantage of this approach in robotics.

The ad-hoc schemes of AI might also produce intelligent robots. But I think that with the aid of principles seen in neuroscience, we can build a computer that talks like a human, is as endearing as our pets, thinks in metaphor, and manages multiple levels of abstraction.
—William H. Calvin

Chapter 4

COLLECTIVE ROBOTICS

4.1 Introduction

One must marvel at the collective behavior of a colony of ants excavating a network of tunnels or a swarm of termites building towering cathedral mounds with internal heating and cooling shafts [20]. Such behavior in the natural world can both awe and inspire the roboticist [108] when considering the control of multiagent systems.

The Artificial Neural Tissue (ANT) superimposes on a typical feedforward neural-network structure a coarse-coding mechanism inspired by the work of Albus [3, 2]. This coarse coding allows areas of specialization to develop in the tissue which in turn facilitate task decomposition. The mechanism has a biological analogy as neurons can communicate not only electrically by exchanging signals along axons but also through chemical diffusion which can selectively activate neurons. The "genome" of the tissue is evolved in an artificial Darwinian fashion.

With minimal task-specific assumptions and limited supervision, an ANT controller can exploit *templates* (unlabeled environmental cues), *stigmergy* (indirect communication mediated through the environment), and *self-organization*. Because little preprogrammed knowledge is given, ANT may discover novel solutions that might otherwise be overlooked by a human supervisor.

ANT is particularly advantageous for multirobot tasks in which some global behavior must be achieved without a centralized controller. Even though each individual has no perception of the overall task and must

operate with only local sensor data, an ensemble of robots is able to complete a mission requiring a global consensus. The desired global behavior emerges from the local interactions of individual robots. Designing controllers of this sort by hand can be very difficult because the process by which local behaviors work to form a global consensus can be difficult to understand and even counterintuitive. Previous work in the field such as [102, 123, 168, 163] rely on task-specific human knowledge to develop simple "if-then" rules or equivalent coordination behaviors to solve multirobot tasks. In contrast, the approach outlined here provides for a generic framework that allows, by evolutionary computation, decomposition of the task to emerge naturally.

In this chapter, we look at the resource-collection task, which is motivated by plans to collect and process raw material on the lunar surface. In addition we also look at an excavation task, where teams of robots need to excavate a hole in which to bury a nuclear power source. For both tasks, we address the issue of *scalability*; that is, how does a controller evolved on a single robot or a small group of robots but intended for use in a larger collective of agents scale? We also investigate the associated problem of *antagonism*, where one robot undoes the actions of other robots. The end effect is reduced productivity of the entire multirobot system.

In the following section, we present background to the tasks at hand. The reader may refer back to Section 3.3 for details on the Artificial Neural Tissue model. This is followed by a description of the simulation experiments conducted for the resource-collection and excavation tasks. The results are followed by discussions and summary in Section 4.6.

4.2 Background

In 2004, NASA set in place a roadmap to return human astronauts to the Moon by the year 2020. The plan includes development of a human habitat for extended duration stays. Part of the strategy in building a viable long-term habitat on the moon is to implement ways to exploit the in-situ resources [137]. It is argued that the discovery of water-ice on the lunar south pole and abundance of titanium oxide on the surface makes an in-situ resource utilization roadmap viable. Water can be used to produce oxygen and hydrogen for rocket refueling and for powering essential life support systems. These options can significantly reduce payload transportation requirements to the moon.

One of the first steps in base construction will involve installing a power source such as a nuclear reactor. Other tasks will require clearing landing pads, site preparation for base facilities, and construction of transport and communication networks. The use of multipurpose teams of robots for construction of key elements of a human habitat and for resource extraction and processing offers many benefits. Infact, it is very likely an imperative due to concerns of health and safety limiting the productivity of human astronauts. These teams of autonomous robots could work continuously in harsh environments making them very productive and appealing for extended duration tasks without the need for a teleoperation infrastructure.

An earth-based teleoperation infrastructure would require robust operational procedures with the ability to handle intermittent communication interruptions, bandwidth disruptions/traffic limitations and latency issues. Such a system has been demonstrated successfully with the Lunakhod 1 and 2 rover missions; however, operator fatigue is of concern [110] especially when coordinating actions with teams of robots over extended duration missions.

Alternatively, a lunar-based teleoperation system will still require a dedicated human habitat infrastructure for on-site operation. Operation of multiple robots will either require multiple operators or single operator using an automated scheduling system [106]. A scheduling system would be less efficient than continuous operation since it is limited by saturation (when a human operator cannot attend to any more robot tasks) in contrast to an autonomous robotic system that is not limited by these concerns. These attributes make an autonomous robotic system more appealing, with the possibility of having a base deployed and operational in time for astronauts to arrive from Earth. There currently exist two major approaches to developing autonomous control systems: human knowledge/model-based controllers and systems based on machine learning techniques.

Human knowledge/model-based behavior control strategies rely on human input in the form of *ad hoc* control rules/heuristics, task-specific assumptions, and human experience and knowledge. In contrast, machine learning systems presented here perform task decomposition through 'emergent' self-organization behavior. In lunar and planetary environments, task-specific assumptions may not always be valid in situ and may require operational reassessment during a mission. There is also a growing necessity for the development of generic teams of multipurpose 'utility' robots that can facilitate in-situ resource utilization and perform specific tasks that may never have been envisioned during mission planning and modeling stages.

The need for multiple smaller robots instead of one large one stems from constraint on mass and volume of payloads that can be transported to the moon with current launch vehicles. Use of a large vehicle would require multiple launches to ferry all the components and some form of assembly in-situ. Such a requirement implies astronauts need to be on the ground before/during the arrival of the equipment.

Risk of launch failure requires redundant launches of critical equipment where possible. The third critical factor is the need for redundancy in-situ. If a lunar base were to be dependent on a single robotic vehicle for construction, any technical difficulties could bring to a halt the entire mission. All these factors could be addressed by launching a fleet of smaller vehicles instead of one large one.

With a team of multipurpose robotic platforms, base construction can be initiated and completed before arrival of the astronauts, instead of diverting valuable astronaut time toward base construction. Use of astronauts in this regard carries risk of accidents and involves having a habitable structure already in place.

4.3 Related Work

Collective robotic tasks can benefit from some of the same mechanisms that are used by social insects. These include the use of templates, stigmergy, and self-organization. *Templates* are environmental features perceptible to the individuals within the collective [17]. *Stigmergy* is a form of indirect communication mediated through the environment [62]. *Self-organization* describes how local or microscopic behaviors give rise to a macroscopic structure in systems which are not in equilibrium [16]. These are positive traits of natural systems which would be advantageous to implement in robotic systems. However, many existing approaches suffer from another emergent feature called *antagonism* [26]. This describes the effect that arises when multiple agents trying to perform the same task interfere with one another and reduce the overall efficiency of the group. Because our approach is evolutionary in nature, it is able to "learn" how to take advantage of the techniques identified above. As we will show, the systems also learn how to mitigate the effects of antagonism, which is something that is difficult to do in hand-crafted systems.

In insect colonies, templates may be a natural phenomenon or they may be created by the colonies themselves. They may include temperature, humidity, chemical, or light gradients. In robotic applications, template-based approaches include the use of light fields to direct the creation of circular [148] and linear walls [162] and planar annulus structures [168]. Spatiotemporally varying templates (*e.g.*, adjusting or moving a light gradient over time) allow the creation of more complex structures [149].

Stigmergy describes the use of changes in the environment as a means of indirect communication between agents. In the natural world, one way in which ants and termites do this is through the use of pheromone trails. Stigmergy has been used extensively in collective-robotic construction tasks, including blind bull dozing [123], box pushing [102], heap formation [13] and tiling pattern formation [151].

Most of the works cited above rely on either user-defined deterministic "if-then" rules or on stochastic behaviors. The goal is to design a simple controller for agents that have access only to local information, but that are nevertheless able to work together to achieve an overall objective. This is difficult to do by hand, since the global effect of these local interactions is often hard to determine. Existing approaches can be categorized in one of three groups. The simplest method is to design a controller that will allow a single agent to complete the task and have it treat any other agent as an "obstacle" to be avoided. A more sophisticated solution will incorporate an extra set of rules to handle the interaction with other agents gracefully. This usually involves adding some kind of explicit communication between the agents. Even in this case, the rules handling interactions between agents are added as a second step in order to "fix" problems that would occur if they were omitted. It is rarer to find applications that fall into a third category, wherein the controllers are designed from the start with cooperation and interaction in mind.

4.3.1 Collective Robotics

Within collective robotics, self-organization describes how local or microscopic behaviors give rise to a macroscopic (global) structure in systems which are not in equilibrium [16]. With decentralized self-organizing systems, no individual possesses knowledge of the overall environment or the end goal. Individuals merely react to local sensor data.

In [13], a group of robots make use of stigmergy in performing a clustering task. In [123], robots perform the opposite task, area clearing, and [148] uses templates to direct the construction of a wall. In all three cases, a single robot is capable of performing the whole task. The controllers are 'if-then' rules which treat other robots as obstacles to be avoided. The authors of [13] note that for more than three robots, the efficiency begins to decrease. In [123], although increasing the number of robots increases the efficiency, there is a reduction in the ability to control the final size and shape of the area cleared.

The wall-building task described in [162] was originally designed with a single robot in mind. Because only one robot can add a block to the end of the wall at a time, additional arbitration rules had to be added to handle the situation in which multiple robots arrive with a block to add at the same time. Here again, performance begins to decrease after a certain point as more agents are added. The end of the wall becomes a bottleneck, with multiple agents waiting for their turn to access the end of the wall. One could argue that a more efficient solution would have some robots fetching blocks and depositing them near the end of the wall while fewer robots stayed near the wall and moved those blocks into their final positions. In Section 4.4.1, we describe a task in which an ANT-based controller is able to evolve a similar solution.

For the box-pushing task described in [102], a controller for a two-robot system was designed from the start with cooperation in mind. This controller is shown to exhibit superior performance to two non-cooperating robots trying to perform the same task. However, in this case, the controllers make use of explicit communication to share complete state information with one another at each step. This does not scale well to larger groups of agents.

As we can see, most existing controllers are designed first with a single agent in mind, and then are enhanced (based on human knowledge) with arbitration rules when multiple agents must interact. The result is that the interactions between agents are more often antagonistic than cooperative. It is more difficult to design controllers by hand with cooperation in place, because it is difficult to predict or control the global behaviors that will result from local interactions. Designing successful controllers by hand can devolve into a process of trial and error, especially in the case of the first two categories described above.

4.3.2 Excavation

Previous work into autonomous excavation [147] has been limited to single robotic excavation platforms and separate loading/unloading vehicles for terrestrial applications. Digging operations is performed through use of human defined automated scripts (behaviors) that simplify repetitive excavation/truck loading cycles.

These are a more elaborate form of 'if-then' rules, defined specifically for the task at hand. The scripts are used to position and unload an excavator bucket relative to a dump truck, based on a suite of sensors onboard the vehicles. The scripts are developed based on input from expert human operators and model vehicle specific limitations such as load handling capacity. These scripts also incorporate a coarse and refined planner to sequence digging operations within a localized area.

Such systems have been comparable in efficiency to human operated systems. Kinematic modeling-based techniques have been used to automate the digging operations of an electric rope shovel [40]. Sensing of the digging terrain is done using laser ranger sensors. Both systems are explicitly designed for specific vehicle platforms and lack any longer term task planning capability. A control system such as LUCIE [19], apart from identifying and automating cyclic excavation related subtasks, incorporates a whole sequence of intermediate goals that need to be achieved to complete a trench digging task. The task is decomposed and prioritized by a human operator and the controller attends to automatic sequencing of subtasks to achieve each intermediate goal.

4.3.3 Machine Learning

A means of reducing the amount of effort required in designing controllers by hand is to encode controllers as behavioral look-up tables and allow a genetic algorithm to evolve the table entries. This approach is used to solve a heap formation task in [9] and a 2×2 tiling formation task in [151].

A limitation with look-up tables is that they have poor sensor scalability, as the size of the look-up table is exponential in the number of inputs. Look-up tables also have poor generalization. Neural network controllers perform better generalization since they effectively encode a compressed representation of the table. For excavation, poor sensor scalability imposes severe constraints on the choice of digging/resource collection vehicle and number of sensors allowed. As an action must be encoded for each combination of sensor inputs, the controller does not generalize from one state to another one with similar inputs. Neural-network controllers can often overcome this second limitation by effectively implementing a compressed representation of the problem space through generalization.

Neural controllers have been used to solve 3×3 tiling tasks [152] and to build walls, corridors, and briar patches [29], and have been used for multirobot communication and coordination [161]. When using fixed-topology networks, the size of the network must be specified ahead of time. Choosing the wrong size may lead to a network that is either unable to solve the problem or is difficult to train [154].

The collective robotic and autonomous excavation works cited earlier excluding [151] rely on either user-defined, deterministic 'if-then' rules, or on stochastic behaviors. In both cases, designing these controllers is an *ad hoc* procedure that relies on the experimenter's or operator's knowledge of the task at hand. However, for collective robotic tasks, the global effect of local interactions is often difficult to gauge, and the specific interactions required to achieve a global consensus may even be counterintuitive. Thus, at this stage of the field's development at least, designing successful controllers by hand is a process of trial and error.

With autonomous excavation, only one vehicle is considered and thus the impact of digging behaviors when introducing multiple interacting vehicles is also bound to the same problems encountered for collective robotic tasks.

The ANT framework presented here simultaneously addresses both the problems in designing rule-based systems by hand and the limitations inherent in fixed-topology evolutionary systems. Since it is a variable-length neurocontroller model, it provides good scalability and generalization of sensory input [154]. The ANT framework also supports both stochastic and deterministic arbitration schemes. Another advantage with the ANT framework is that it allows for both sensor and behavior extensibility (taking into account future needs/growth) and is not constrained to a specific robotic platform. In addition, ANT does not rely on detailed task-specific knowledge. It evolves controllers tuned towards a user-specified global fitness function. The evolutionary selection process is able to discover for itself how to make use of templates and stigmergy and to mitigate the effects of antagonism.

4.4 Resource Gathering

The effectiveness of the ANT controller is demonstrated in simulation on a resource-collection task. A team of robots collect resource material distributed throughout its work space and deposits it in a designated dumping area. The workspace is modeled as a two-dimensional gridworld environment with one robot occupying four grid squares. For this task, the controller must possess a number of capabilities including gathering resource material, avoiding the workspace perimeter, avoiding collisions with other robots, and forming resources into a berm at the designated location. (In the present experiment, a berm is simply a mound of the resource material.) The berm location has perimeter markings on the floor and a light beacon mounted nearby. The two colors on the border are intended to allow the controller to determine whether the robot is inside or outside the berm location. Though solutions can be found without the light beacon, its presence improves the efficiency of the solutions found, as it allows the robots to track the target location from a distance instead of randomly searching the workspace for the perimeter. The global fitness function for the task measures the amount of resource material accumulated in the designated location within a finite number of time steps, in this case $T = 300$. Darwinian selection is performed based on the fitness value of each controller averaged over 100 different initial conditions.

For this task, inputs to the ANT controller are shown in Table 4.1. The robots are modeled on a fleet of rovers designed and built at the University of Toronto Institute for Aerospace Studies. They have access to a pair of webcams and a set of sonar sensors. All raw input data are discretized. The sonar sensors are used to determine the values of S_1 and S_2. One of the cameras is used to detect resource material and colored floor templates (Figure 4.1). The other camera is used to track the light beacon. In order to identify resources and colored floor templates, a naïve Bayes classifier [69] is used to perform color recognition [1]. Simple feature-

[1]Colour based classification system and light beacon detection procedures for the Argo Rovers implemented by Alex Smith.

Table 4.1: Resource Collection: Sensor Inputs

Sensor Variables	Function	Description
$V_1 \ldots V_4$	Resource Detection	Resource, No Resource
$C_1 \ldots C_4$	Template Detection	Blue, Red, Orange, Floor
S_1, S_2	Obstacle Detection	Obstacle, No Obstacle
L_1	Light Position	Left, Right, Center
Q_1	Light Range	0-10 (distance to light)

Table 4.2: Resource Collection: Basis Behaviors

Order	Behavior	Description
1	Dump Resource	Move one grid square back; turn left
2	Move Forward	Move one grid square forward
3	Turn Right	Turn 90° right
4	Turn Left	Turn 90° left
5, 7, 9, 11	Bit Set	Set memory bit i to 1, $i = 1 \ldots 4$
6, 8, 10, 12	Bit Clear	Set memory bit i to 0, $i = 1 \ldots 4$

detection heuristics are used to determine the values of $V_1 \ldots V_4$ and $C_1 \ldots C_4$ based on the grid locations shown. For detection of the light beacon, the electronic shutter speed and gain are adjusted to ensure that the light source is visible while other background features are underexposed. The position of the light L_1 is determined based on the pan angle of the camera. The distance to the light source Q_1 is estimated based on its size in the image. The robots also have access to four memory bits, which can be manipulated using some of the basis behaviors. Table 4.2 lists the basis behaviors the robot can perform. These behaviors are activated based on the output of the ANT controller, are preordered and all occur within a single timestep. The evolutionary algorithm population size for the experiments is $P = 100$, with crossover probability $p_c = 0.7$, mutation probability $p_m = 0.025$ and a tournament size of $0.06P$. The tissue is initialized as a "seed culture," with 3×6 motor control neurons in one layer. After this, the tissue is grown to include 70–110 neurons (selected from a uniform random distribution) before starting the evolutionary process.

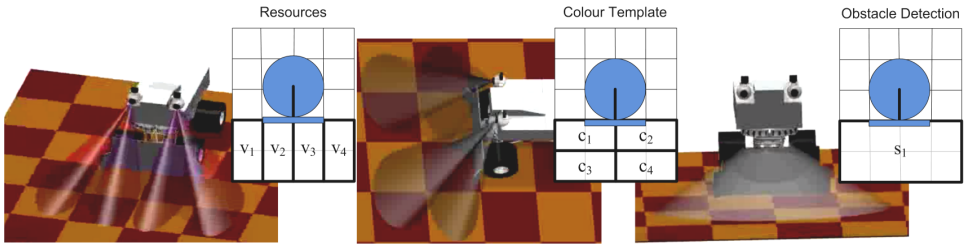


Figure 4.1: Input sensor mapping, with simulation model inset.

See acknowledgements.

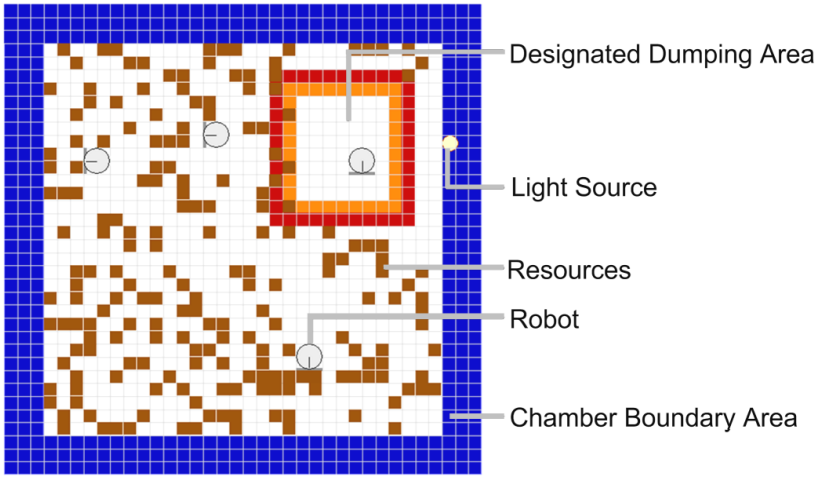


Figure 4.2: 2D grid world model of experiment chamber.

4.4.1 Results and Discussion

Figure 4.3 (left) shows the fitness (population best) of the overall system evaluated at each generation of the artificial evolutionary process. The performance of a fixed-topology, fully connected network with 12 hidden and output neurons is also shown in Figure 4.3 (right). While this is not intended as a benchmark network, in a fixed-topology network there tends to be more "active" synaptic connections present (since all neurons are active), and thus it takes longer for each neuron to tune these connections for all sensory inputs.

The results with ANT controllers in Figure 4.3 (left) show that performance increases with the number of robots. With more robots, each robot has a smaller area to cover in trying to gather and dump resources. The simulation runs indicate that a point of diminishing returns is eventually reached. Beyond this point, additional robots have a minimal effect on system performance with the initial resource density and robot density kept constant. The evolutionary process enables the decomposition of a goal task based on a global fitness function, and the tuning of behaviors depending on the robot density.

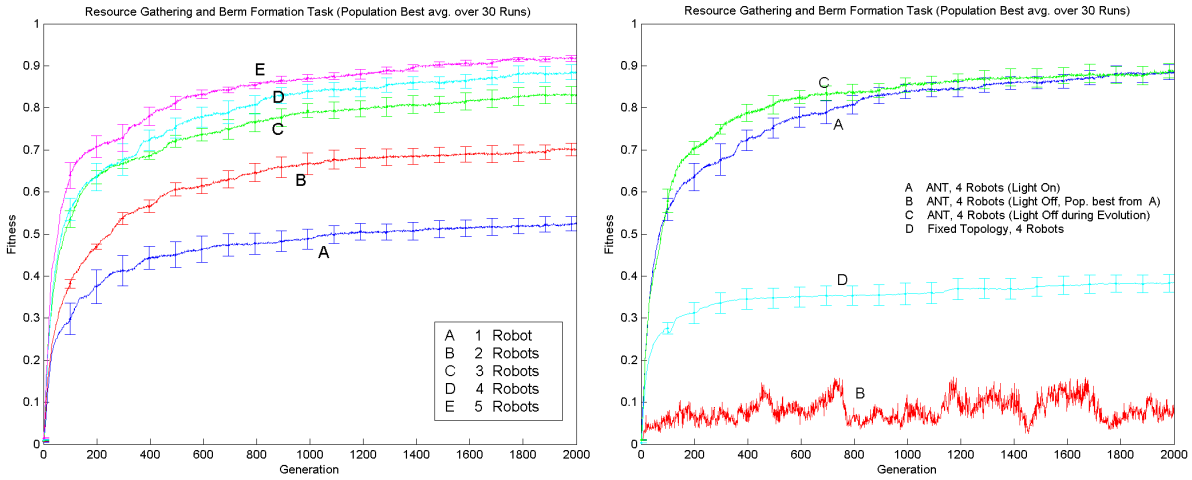


Figure 4.3: Evolutionary performance comparison of ANT-based solutions for one to five robots (left) and performance with four robots for fixed topology and with light beacon off (right). Error bars indicate standard deviation.

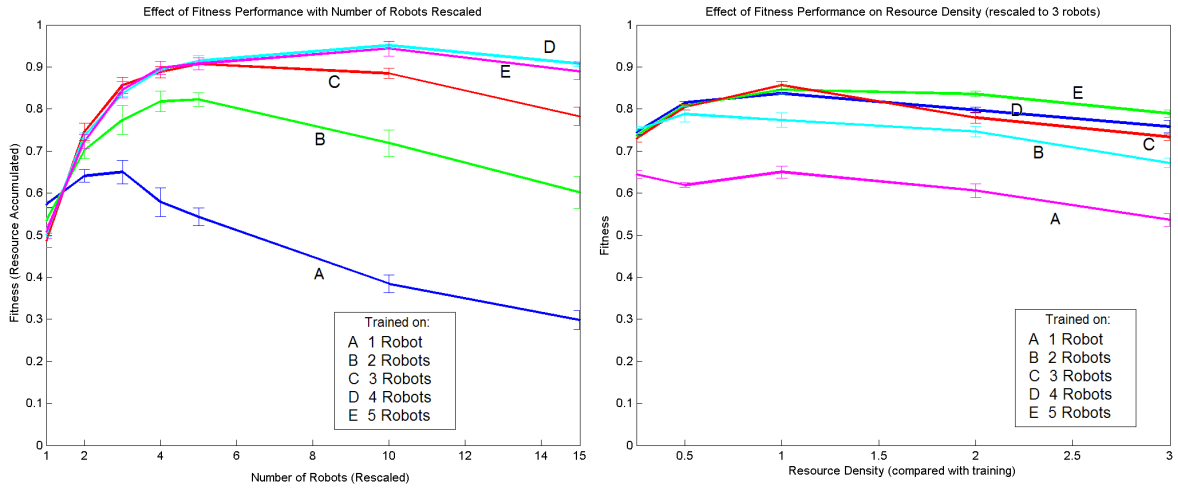


Figure 4.4: Scaling of ANT-based solutions from one to five robots (left) and change in resource (right).

In an ANT-based architecture, networks are dynamically formed with decision neurons processing the sensory input and in turn "selecting" motor-control neurons through the coarse-coding framework [154]. The behavioral activity of the controllers (Figure 4.6) shows the formation of small networks of neurons, each of which handles an individual behaviors, such as dumping resources or detecting visual templates (boundary perimeters, target area markings, etc.). Localized regions within the tissue do not exclusively handle these specific user-defined, distal behaviors. Instead, the activity of the decision neurons indicate distribution of specialized "feature detectors" among independent networks.

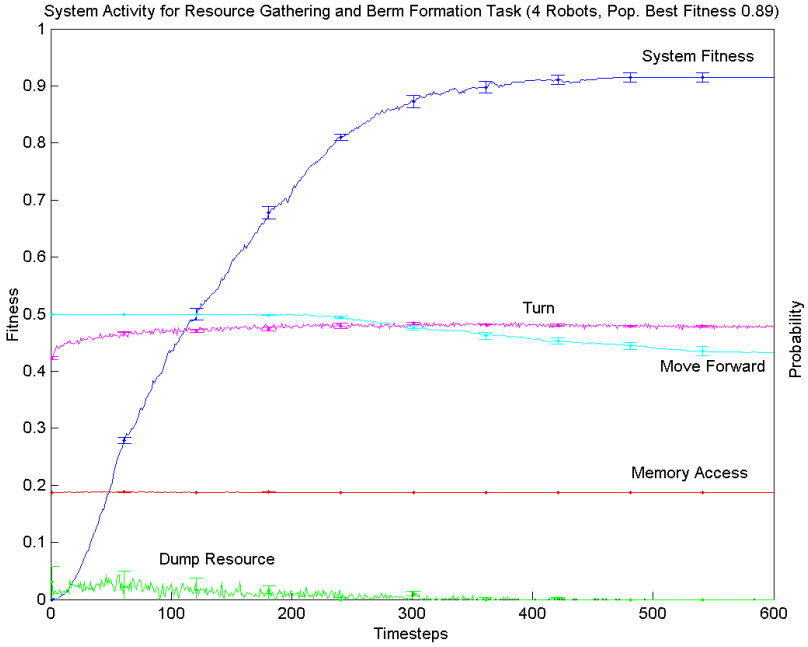


Figure 4.5: System activity for the resource-collection task.

4.4.2 Behavioral Adaptations

Some of the emergent solutions evolved indicate that the individual robots all figure out how to dump nearby resources into the designated berm area, that but not all robots deliver resource all the way to the dumping area every time. Instead, the robots learn to pass the resource material from one individual to another during an encounter, forming a "bucket brigade" (Figure 4.7). This technique improves the overall efficiency of system as less time is spent traveling to and from the dumping area. Since the robots cannot explicitly communicate with one another, these encounters happen by chance rather than through preplanning. As with other multiagent systems, communication between robots occurs through the manipulation of the environment in the form of stigmergy. The task in [162] is similar in that distributed objects must be delivered to a confined area; however, the hand-designed controller does not scale as well as the "bucket brigade" solution that the ANT framework discovered here.

While the robot controllers can detect and home in on a light beacon (exhibiting "phototaxis"), this source of navigation is not always used. Although not necessary, the light beacon helps in navigation by allowing the robots to locate the dumping area. It is notable that the overall fitness is unaffected when the controllers are evolved with the light source turned off. However, when the light source is turned on, the controllers do make use of it to navigate to the dumping area even though this does not appear to translate into a higher fitness (Figure 4.3 right). When a controller evolved with the light source on is tested with the light source off, the fitness is substantially lower.

Phototaxis requires a means to determine the light gradient and intensity, both of which are made available in these simulations. The robot model assumes the light-detection sensor is directional and has a limited field of view. Hence, once a robot faces away from the light source, the light detection sensor is in the "Not Visible" state. Although the light intensity input appears to be unused, light direction values appears to be used intermittently (Figure 4.7) in combination with other visual templates. When a robot faces a boundary region, the direction of the light source is used to determine whether to "Turn Right" or "Turn Left." This behavior in turn is triggered when the robot had accumulated resources along the way.

In our simulations, the dumping area is usually next to the boundary region. Where possible, the controllers appear to perform "sensor fusion," making use of data from multiple sensors to perform navigation to and from the dumping area.

In these simulation experiments, the robots have no way to measure the remaining time available; hence, the system cannot greedily accumulate resource materials without periodically dumping the material at the designated area. This explains why we see a steady increase in the amount of resource material gathered over time (Figure 4.5).

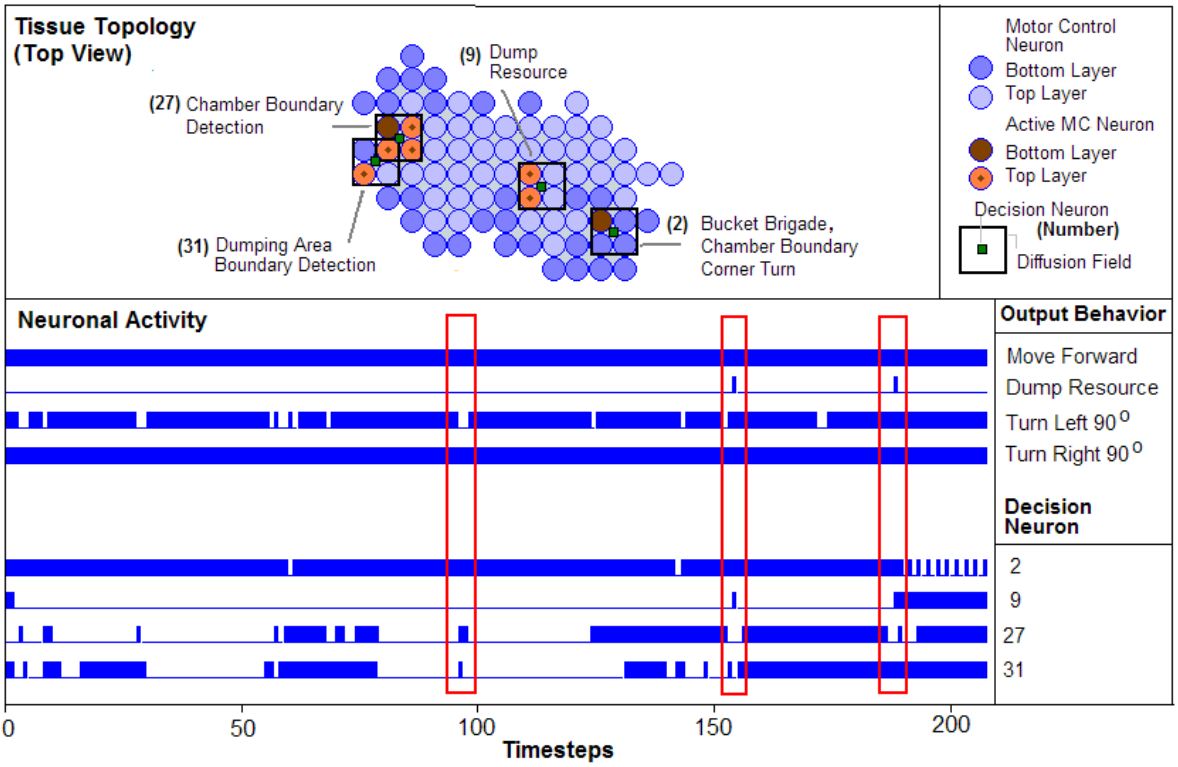


Figure 4.6: Tissue Topology and neuronal activity of a select number of decision neurons. These decision neurons in turn "select" (excite into operation) motor control neurons within its diffusion field.

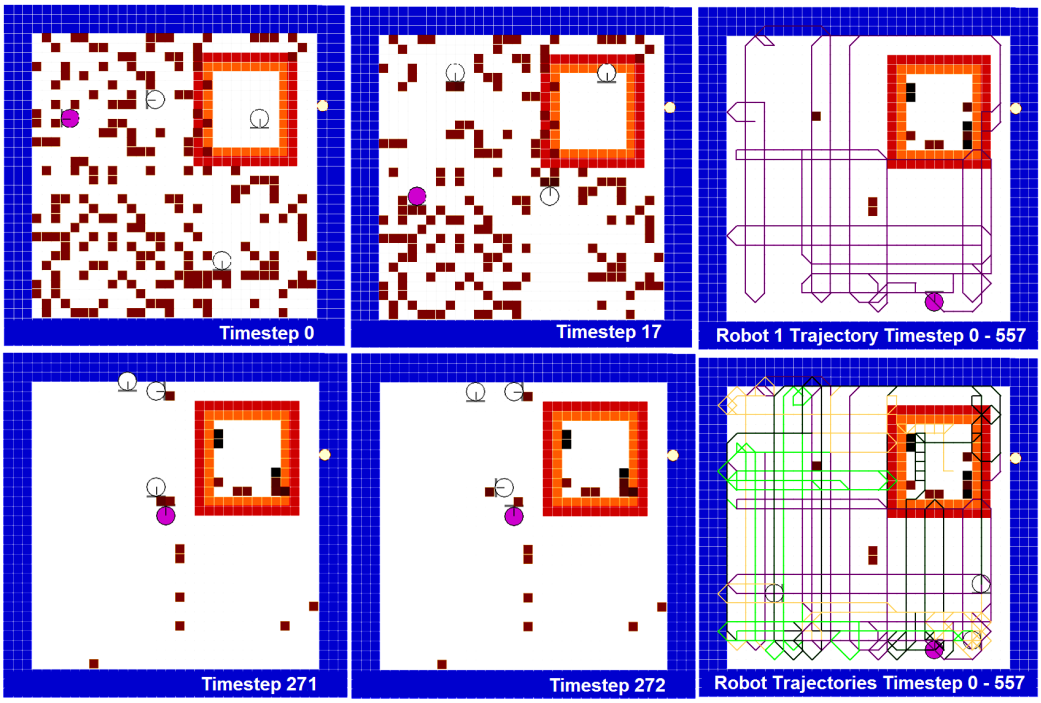


Figure 4.7: Snapshots of robots and trajectories of a task simulation (4 robots).

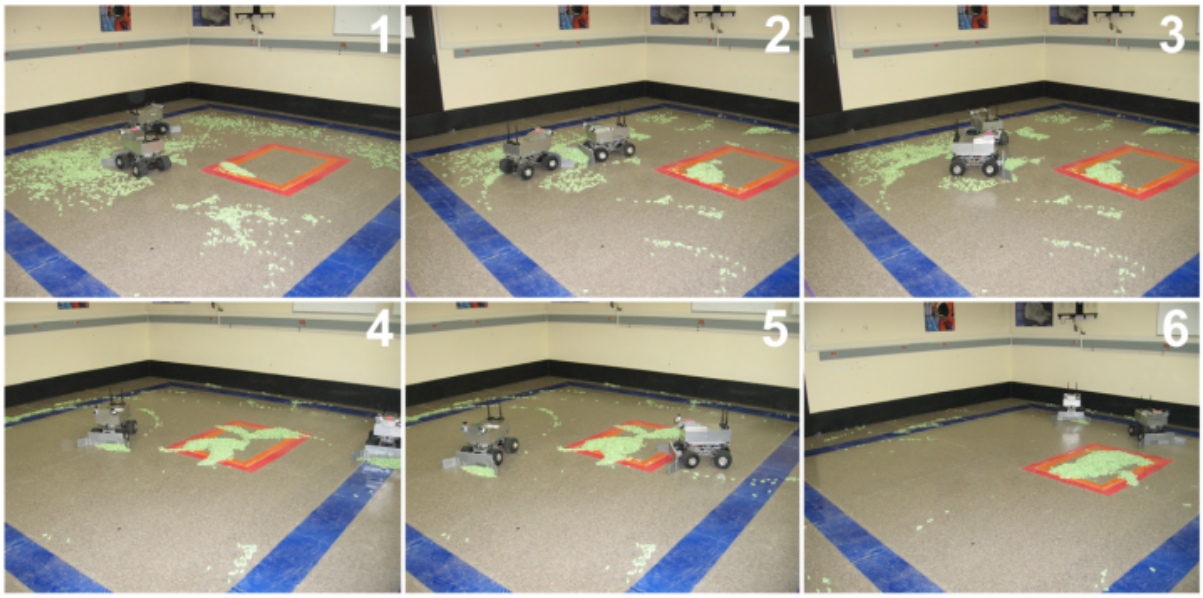


Figure 4.8: Snapshots of two rovers performing the resource collection task using an ANT controller. Frames 2 and 3 show the "bucket brigade" behaviour, while frames 4 and 5 show the boundary avoidance behaviour.

4.4.3 Controller Scalability

We examine the fittest solutions from the simulation runs shown in Figure 4.4 (right) for scalability in the number of robots while holding the amount of resources constant. Taking the controller evolved for a single robot and running it on multirobot system shows limited performance improvement. In fact, using four or more robots results in a *decrease* in performance, owing to the increased antagonism created.

The scalability of the evolved solution depends in large part on the number of robots used during the training runs. The single-robot controller expectedly lacks the cooperative behavior necessary to function well within a multiagent setting. For example, such controllers fail to develop "robot collision avoidance" or "bucket brigade" behaviors. Similarly, the robot controllers evolved with two or more robots perform demonstrably worse when scaled down to a single robot, showing that the solutions are dependent on cooperation among the robots. To limit the effects of antagonism, controllers would need to be trained under conditions in which the probability of encounters among robots is sufficiently high.

The effect of changes in resource density is also examined in the evolved solutions (Figure 4.4 right). When keeping the number of robots and the maximum number of timesteps constant while increasing the amount of resource material, the percentage of resource material accumulated in the dumping area appears to decline steadily. The peak performance appears to occur when the resource density is at or near the training conditions.

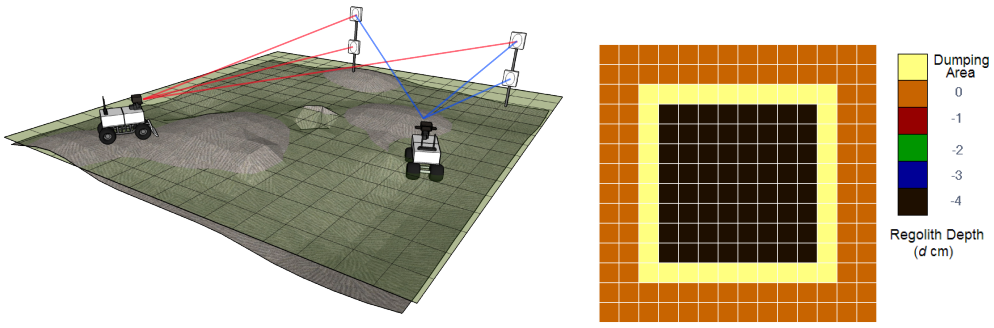
4.5 Excavation

4.5.1 Simulation Experiments

The excavation task is intended to demonstrate the feasibility of teams of robots digging a hole to bury a nuclear reactor in the lunar regolith. This is intended as the first step in setting up a lunar base. It could be argued that emergent task decomposition maybe necessary to accomplish the task given a global fitness function. A layout of the simulation experiment area used for training is shown in Figure 4.9 (right). The experiment region or workspace is modeled as a two-dimensional grid environment with the size of four squares in the grid being just able to accommodate one robot. For this task, the controller needs to accomplish a number of subtasks including interpreting excavation 'blue prints', perform layered digging, avoid burying or trapping other robots, clearing and maintaining excavation routes. Each robot controller has access to a goal map that defines the location of the dumping area and target depth of the excavation area. The global fitness function f for the task is given as follows:

$$f = \frac{\sum_{j=1}^{J} \sum_{i=1}^{I} \vartheta_{i,j} \cdot e^{-2|g_{i,j} - h_{i,j}|}}{\sum_{j=1}^{J} \sum_{i=1}^{I} \vartheta_{i,j}} \tag{4.1}$$

where I and J are the dimensions of the workspace and $\sum_{j=1}^{J} \sum_{i=1}^{I} \vartheta_{i,j} > 0$ and $\vartheta_{i,j} = 1$ if grid square i, j is to be excavated and 0 otherwise; $g_{i,j}$ is the target depth and $h_{i,j}$ is the current regolith depth. For the evolutionary runs, fitness f is calculated after $T = 100$ timesteps, for an excavation area of 8×8 squares surrounded by the dumping area.



(a) Localization (b) Goal Map

Figure 4.9: Localization setup for hardware demonstrations and an example goal map.

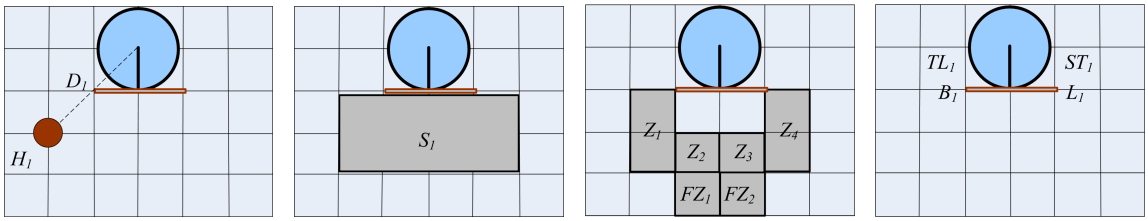
4.5.2 Sensor Inputs and Basis Behaviors

For this task, inputs to the ANT controller are shown in Table 4.3 (right). Two different localization techniques have been used with the robots. For one, the robots have access to a pair of webcams and a laser range finder mounted on a pan tilt unit [2]. The webcams are used to detect and center the pan tilt units on

[2]Laser range finder base localization system developed by Alex Ho. See acknowledgements.

Table 4.3: Excavation: Sensor Inputs

Sensor Variables	Function	Description
$Z_1 \ldots Z_4$	Depth Sensing Relative to Goal Depth	Level, Above, Below, Don't Care, Dump
E_1, E_2	Depth Sensing Relative to Ground	Above, Below or Level
B_1	Blade Position	Below, Level, Above, Home
F_1	Blade Load	$0-4$
S_1	Front Obstacle Detection	Obstacle, No Obstacle
D_1	Separation Distance From Nearest Robot	$0-3$
H_1	Heading From Nearest Robot	North, East, West, South
R_1	Robot Tilted Downwards	True, False
U_1	Robot Stuck	True, False



(a) Virtual Radar (b) Sonar (c) Depth Sensing (d) Motion Sensing

Figure 4.10: Robot Input Sensor Mapping for the Simulation Model.

up to four localization targets consisting of ellipses with contrasting patterns (Figure 4.9). A laser range finder is used to measure the distance from each target to robot and triangulation is performed to determine (x, y, z) coordinates of the robot within the target workspace. The discretized x and y coordinates are use to lookup the goal depth $g_{x,y}$ of each grid square region in front of the robot (Figure 4.10). An alternate localization system used involves mounting different colored LED lights on the robots and using a stationary overhead camera to monitors the lights. The system is calibrated by determining the edges of the workspace and identifying the coordinates on the camera image. By mapping the workspace boundary, one can then easily transform the pixel positions of the light sources to the x, y position of the robots in the workspace for a given z value. Under this setup, z is predicted by keeping track of excavation behaviours executed by each robot and estimating the change in z value within a given area in the map.

In addition, a laser scan of the ground in front of the robot is performed to measure depth of regolith in relation to the wheel depth to determine $Z_1 \ldots Z_3$ and $E_1 \ldots E_2$ (Figure 4.9). The current x and y positions of each robot are used to determine the relative position of the nearest robot and its relative heading through map sharing. All raw input data are discretized. The sonar sensors are used to determine the values of S_1. A pair of load cells on the blade is used to determine F_1 and an onboard tilt sensor is used to determine R_1. Frame 'differencing' of two consecutive webcam images of the ground is used to determine U_1 (whether the rover is stuck or not). The robots also have access to one memory bit, which can be manipulated using some of the basis behaviors. Table 4.4 lists the basis behaviors the robot can perform (in order) within a

Table 4.4: Excavation Basis Behaviors

Order	Behavior	Description
1	Set Throttle	Move one grid square back; turn left
2	Move Forward	Move one grid square forward
3	Move Backward	Move one grid square forward
4	Random Turn	Randomly turn 90° right or left.
5	Turn Right	Turn 90° right
6	Turn Left	Turn 90° left
7	Blade Above	Set blade above ground d cm
8	Blade Below	Set blade below ground d cm
9	Blade Level	Set blade level to ground d cm
10	Blade Home	Retract blade to home position (no contact with regolith).
11	Bit Set	Set memory bit 1 to 1
12	Bit Clear	Set memory bit 1 to 0

single timestep. Darwinian selection is performed based on the fitness value of each controller averaged over 50 different initial conditions, within an 8×8 excavation area (Figure 4.11). The evolutionary algorithm population size for the experiments is $P = 100$, crossover probability $p_c = 0.7$, mutation probability $p_m = 0.025$ and tournament size of $0.06P$.

4.5.3 Results and Discussion

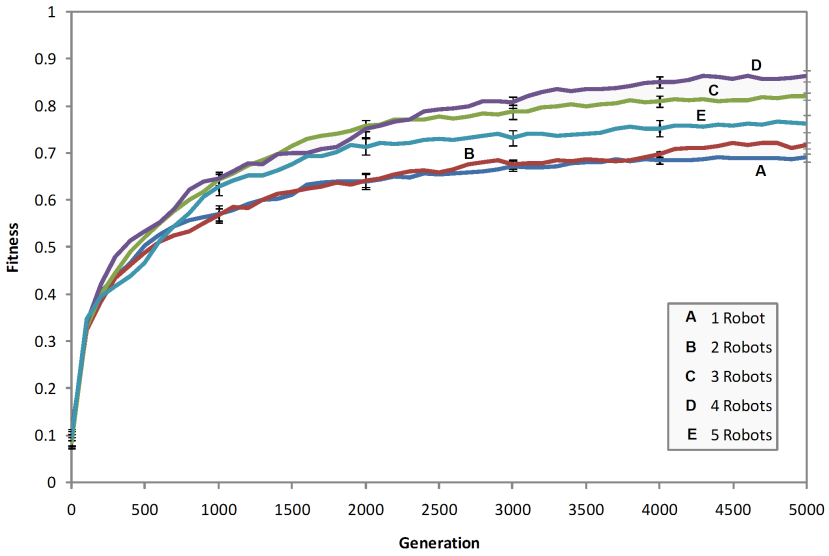


Figure 4.11: Evolutionary Performance Comparison of ANT Based Solutions for between 1 and 5 Robots.

Figure 4.11 shows the population best fitness of the overall system evaluated at each generation of the artificial evolutionary process. It is apparent that the system performance is affected by the density of robots

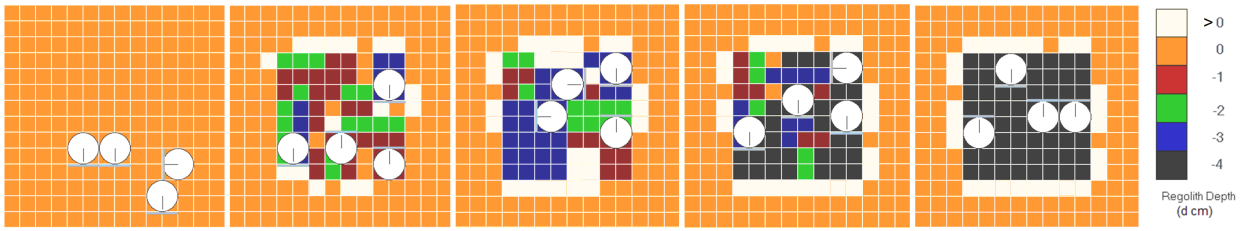


Figure 4.12: Simulation snapshots of an excavation task simulation (4 robots) after 0,50,75,100,170 timesteps.

per digging area (8×8 squares). A single robot is not as efficient as 4 robots, as the excavation can be performed in parallel, with each robot having a smaller area to cover. However, with more than 4 robots for a 8×8 area, the problem of antagonism arises, when multiple robots trying to perform the same task interfere with one another and reduce the overall efficiency of the group (Figures 4.13, 4.14 left). The emergent solutions indicate that the individual robot exploits templates by learning to sense depth relative to the specified goal map and determine whether to lower, level or raise the blade (Figure 4.12). Once the robot senses a dumping area in front, the controller executes a combination of 'move backward' followed by a 'turn left' or 'turn right' to offload the excavated material.

As with other multiagent systems, communication between robots occurs through manipulation of the environment in the form of stigmergy. The manipulation of the environment involves either excavating a region or dumping off excavated material. Controllers also exploit the ability to sense the depth of soil relative to wheel depth (Figure 4.12). This enables each robot controller to sense whether it is excavating deeper or backfilling at the current depth. The ability to backfill, although useful in some situations, can also undo the effort of other robots excavating at different depths within the system. The robots also have the ability to sense the relative position of a nearby robot much like a virtual form of radar. This sensing capability appears to be exploited particularly to avoid collisions when a series of output behaviors such as 'move forward' and 'turn left' is applied in sequence. Although obstacles can be detected using sonars directly in front of the robot, there exists blind spots to the extreme right and left making it difficult to detect and react to obstacles when a sequences of behaviors are executed.

We examine the fittest solutions from the simulation runs shown in Figure 4.11 for scalability in the number of robots while holding the size of the digging area constant (Figure 4.13). Taking the controller evolved for a single robot and running it on a multirobot system shows a steep degradation in performance. This is expected since the single-robot controller lacks the cooperative behavior necessary to function well within a multirobot setting, showing similarity to the resource gathering task [155]. For example, such controllers fail to develop 'robot collision avoidance' behaviors. Similarly, a multirobot system scaled down to a single robot setting also shows a degradation in system performance. With the multirobot systems, controllers have evolved to exploit and depend on cooperative actions to complete the task; thus when the environment is abruptly changed to exclude such a possibility the controller performs poorly.

It is interesting to note that the controllers trained with 4 robots for an 8×8 digging area perform

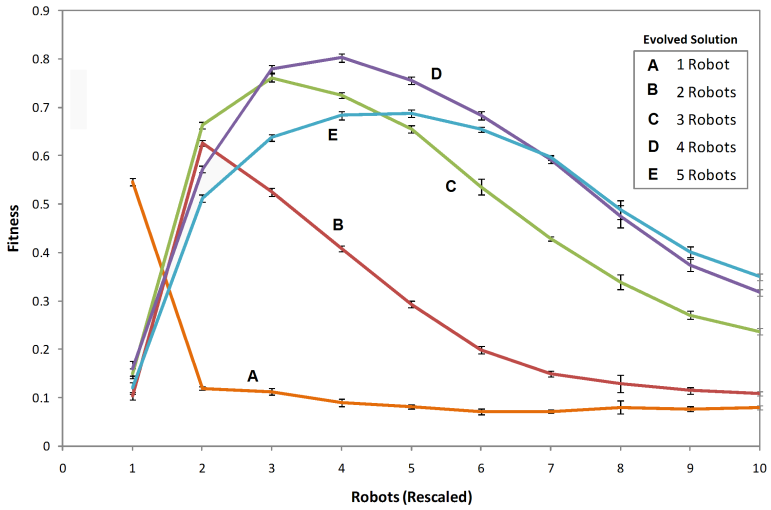


Figure 4.13: Scaling of ANT based Solutions from 1 to 5 robots (8×8 excavation area, average $1.5d$ cm depth).

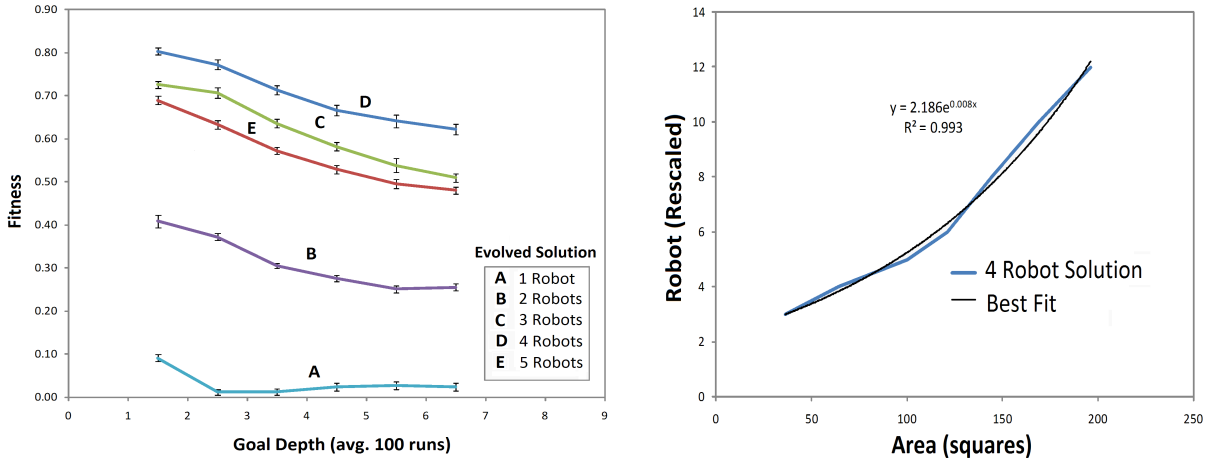


Figure 4.14: Scaling of ANT based solutions for varying depth (left) and 4 robot solution for varying excavation area (right).

considerably better overall than solutions trained for other densities. It is also apparent that this solution scales better for increased goal depth (Figure 4.14 left) and was found to scale better than other solutions for increased excavation area. The optimal ratio (best fit) of robots to digging area using solution trained with 4 robots is shown in Figure 4.14 (right). However, when the number of robots is set higher than the optimal number, solutions trained under higher robot densities perform better, although the system performance falls short compared to the optimal setting (Figure 4.13).

It is apparent from these simulation experiments that there exists an optimal set of training conditions under which solutions show improved scalability. Although the controllers maybe better adapted to antagonism under higher training densities with improved obstacle avoidance techniques, these behaviors may not be as well tuned to completing the overall objectives (excavation) effectively. It should also be noted that

this optimal condition is dependent on task duration. Furthermore the optimal density is beneficial when the task is time limited. Given enough time, the suboptimal solution can also attain similar end results but at increased cost, namely energy expenditure.

4.5.4 Selection for Multirobot Behaviour

Based on Figures 4.13 and 4.14, it is apparent the controller trained for a 4-robot solution, produces solution that show better rescalability than other initial conditions for a 8×8 excavation area. Can evolutionary techniques be used to determine both a multirobot controller and determine system parameters (such as robot density) simultaneously? For the experiment conducted in Figure 4.15 (right), we also include number of robots N as a variable within ANT. This is akin to a population reproduction rate, where certain individuals have a greater number of children at a time than others. A histogram in Figure 4.15 shows the distribution of the N. If there were to be no selection pressure on N, then the histogram should show evidence for a uniform distribution. Based on these results, it is apparent evolutionary selection leans toward the optimal density of robots over other robot densities (Figure 4.11).

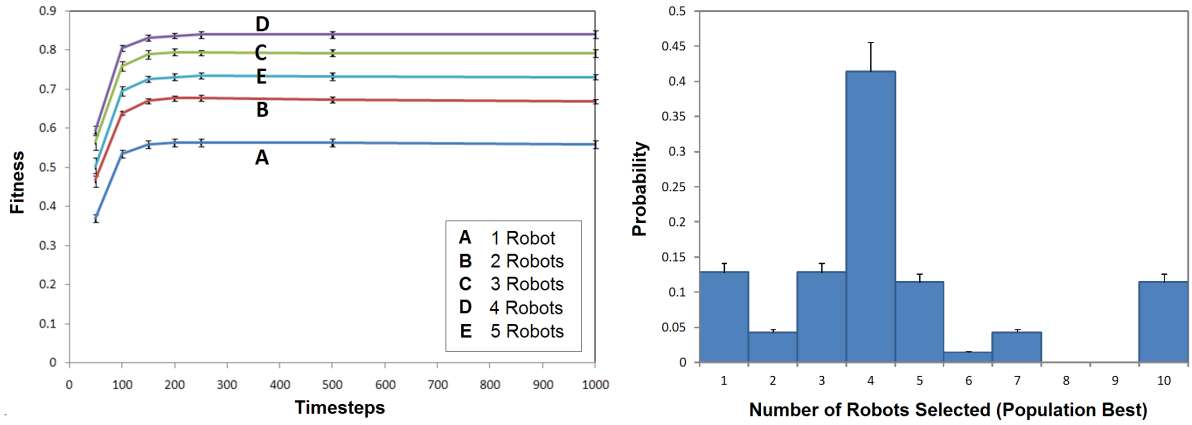


Figure 4.15: Fitness Performance Comparison of ANT Based Solutions for between 1 and 5 Robots with varying time (left). Histogram of number of robots selected, where N the number of robots is evolved as parameter (right).

The question then is why is the system tending toward the optimal density? It is not always obvious that controllers will gravitate towards an optimal setting and stay there. For one if the optimal solution were to be sensitive to damaging mutations, then the population may gravitate to a more stable suboptimal solution. However it has been found that solutions evolved under this optimal set of condition is better rescalable in general. Both traits provide an advantage, helping to gravitate solution towards this setting and remaining there. An individual having adapted under this optimal setting is better adept at handling a deleterious mutation of N robots that could either result in increased or decreased robot densities.

From a practical viewpoint, this technique of evolving a controller and other system parameters concurrently reduces the need to analyze the rescalability performance. This approach allows for simultaneously determining the need for a multirobot controller and its contents. However owing to the stochastic na-

ture of the search process, an optimal density can be obtained with some statistical certainty provided the evolutionary runs are sufficiently repeated.

Demonstration of the ANT-based excavation controllers using higher-fidelity simulations within Digital Spaces^{TM}, an off-the-shelf commercial 3-D Lunar simulation tool shows very promising results[3]. The simulated rovers are equipped with a front loader and able to deliver up to $300W$ of sustained output power. Three rovers can excavate a $4 \times 4 \times 1$ m pit within a 30 minute timeframe (Figure 4.16).

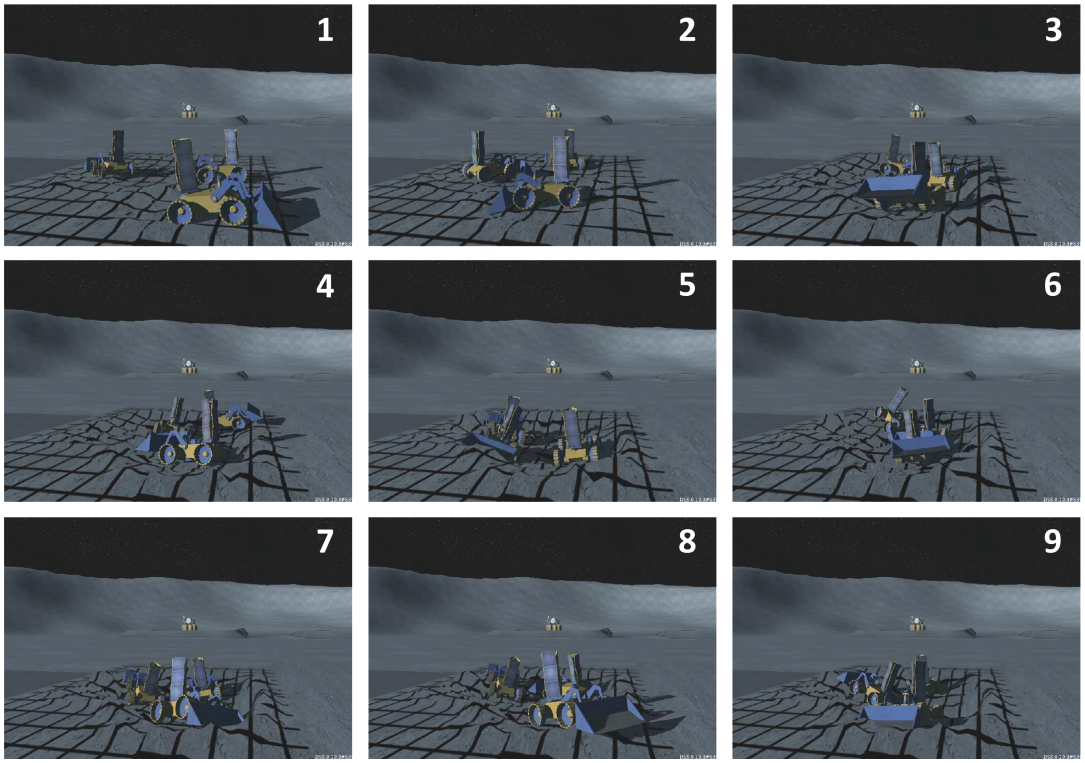


Figure 4.16: Digital Spaces^{TM} simulation of three rovers using a four-robot ANT controller solution to perform excavation on simulated lunar terrain. The rovers unlike under training conditions use a front-loader bucket instead of a two-way bulldozer blade. The excavation blue print includes a hole and a ramp for the rover to enter/exit surrounded by the dumping area.

Hardware demonstrations of the ANT-based excavation controllers have also been conducted and show feasibility of the overall approach. Similar to the resource gathering task, controllers have been ported onto the Argo rover platforms. Although the rovers are severely underpowered for the task, the controller performance validates many behaviors seen using the low fidelity simulations. Evidence of collision avoidance and use of templates are apparent (Figure 4.17 right,4.19,). The rovers also correctly interpret the goal map and dump in the specified dumping areas on most occasions. However on a few occasions, the overhead system looses track of rover position and has been a source of error. The controllers show evidence of rocking behavior as a strategy to get out of sand traps and deep 'pot holes' (Figure 4.18). The controllers also learn

[3]Rover 3-D models and soil interaction dynamics were developed by Nader Abu El Samid. See acknowledgements.

to stop repeated excavation moves and 'back out' when it is unfruitful or gets the robot further buried in sand. This is accomplished by exploiting the memory bit and 'stuck' state. Laser scans (Figure 4.17 right) [4] and video (Figure 4.19) show evidence of multiple holes forming, that in turn merge into larger holes.

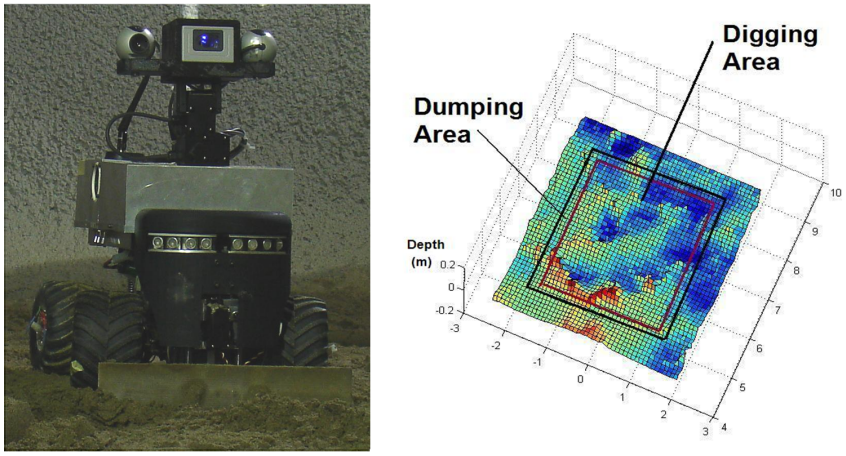


Figure 4.17: (Left) An Argo rover equipped with a two-way bulldozer blade, with a laser range finder and a pair of Logitech® Quickcams™ affixed to the pan-tilt unit. (Right) 3-D laser scan of the work area after nearly 40 minutes of excavation. The digging area and dumping area are highlighted. The rovers reach a maximum depth of 15 cm.

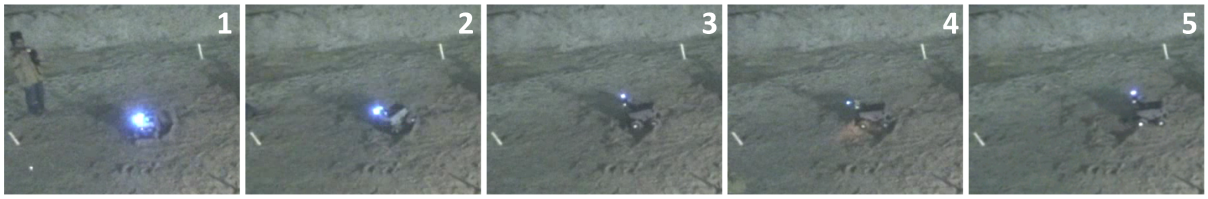


Figure 4.18: Video snapshots of rocking behaviour followed by a 'backout' behaviour triggered after the rover gets stuck trying to push the blade forward.

4.6 Summary

The artificial neural tissue framework presented in Chapter 3 has been applied on two robotic tasks with application in mining and in-situ resource utilization. ANT is shown to evolve solutions to a resource gathering task, akin to open pit mining. The framework uses a global fitness function, that offers no explicit bias toward a particular task decomposition strategy. The intention is for the evolutionary process to yield creative solutions given a set of generic basis behaviors and sensory input. ANT controllers are shown to exploit *templates* (unlabeled environmental cues), *stigmergy* (indirect communication mediated through the environment), and *self-organization*.

Analyzing neuronal activity of a four-robot ANT solution controller shows evidence of areas of specialization within the tissue. Furthermore, the emergent behaviors do not appear to be organized in a distal manner. The controllers evolve several emergent behaviors include use of 'bucket brigades' to transport

[4] 3-D Laser scanning system developed by Ken Law. See acknowledgements.

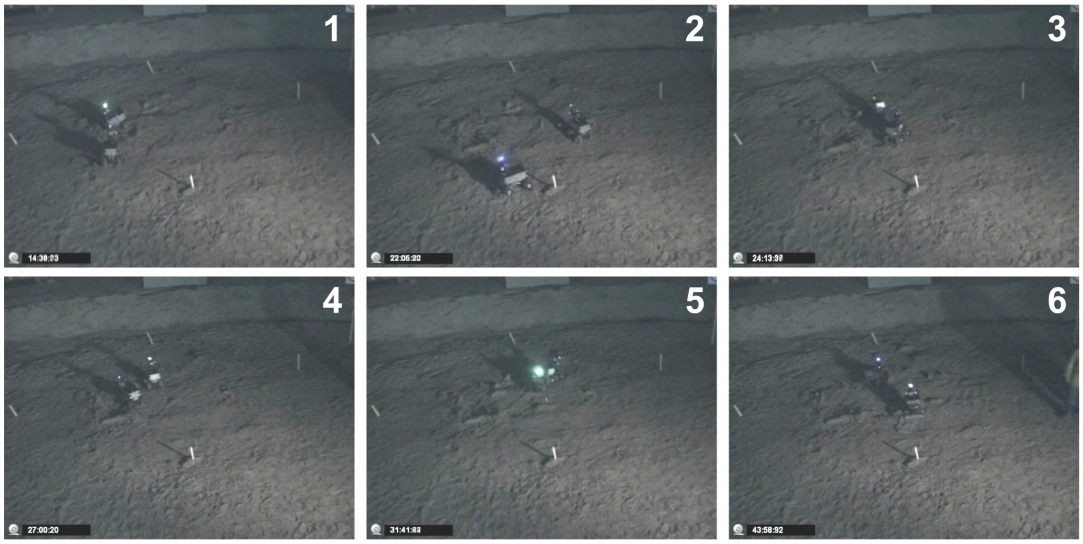


Figure 4.19: Video snapshots of two Argo rovers performing excavation over a 40 minute timeframe. Each rover is equipped with an LED light beacon for localization by the overhead camera system.

resource to the dumping area, exploit color templates to identify dumping area/boundary regions in the workspace and direction of a light beacon. Although ANT controllers learn to exploit use of a light beacon to home in on the resource dumping area, the controllers also attain a dependency to the light beacon and hence show poor performance when the beacon is turned-off.

Increased density of robots within the work area is shown to produce higher fitness. However, what is also evident is diminishing return with higher robot densities. Controllers evolved at higher densities showed better rescalability when applied on a larger number of robots. While solutions evolved with a single robot lacks multirobot cooperative behaviors such as 'collision avoidance' or 'bucket brigade' and hence show poor performance when applied on multiple robots. Similarly, ANT controllers evolved in a multirobot setting show decreased performance over a single robot solution when scaled down to a single robot setting. However multirobot controllers show better rescalability performance overall than a single robot solution for with increased resource density and increase robot density. Rescalability performance is also indicative of how the controllers perform when one or a few are damaged/non operational. Both the multirobot and single robot controllers have been successfully applied on hardware using the Argo-class rovers. Multiple hardware runs demonstrate the feasibility of the approach and robustness to sensor and actuator noise.

Multirobot ANT controller solutions also show advantages over a single robot solution for the excavation task. The multirobot controllers require less time to complete the task by better exploiting parallelism. However, what is apparent from the excavation task is that there is an optimal training setting with respect to the density of robots. ANT controllers evolved at higher or lower than optimal robot density setting shows reduced performance in the rescaled scenario. While the controllers evolved at higher robot densities maybe better adapt to collision avoidance or other capabilities, they may not be as well tuned to completing the overall objectives (excavation) effectively, hence there are competing (multiple) objectives. ANT con-

trollers have also been used to evolve multirobot controllers and determine the optimal robot density setting concurrently. However owing to the stochastic nature of the search process, the runs need to be repeated in order attain the optimal settings with some degree of statistical certainty.

ANT controllers evolved at the optimal robot density show better performance overall for increased excavation depth and evolve a desired solution with fewer genetic evaluations. For the excavation task, the ANT controllers not only develop short-range obstacle avoidance behavior but show evidence of synchronizing actions by exploiting the 'virtual radar' sensor for the excavation task. Furthermore the controllers can successfully interpret and follow user-defined excavation blueprints, learn to perform cooperative excavation, in which the robots avoid backfilling previously excavated regions. High-fidelity simulation using a three-dimensional off-the-shelf lunar simulator and hardware demonstration of the ANT-based excavation controllers show feasibility of the overall approach. Although hardware rover platforms were severely underpowered, the demonstrations validated many behaviors observed in simulation. Hardware demonstrations show evidence of 'rocking' behaviors to avoid deep pot holes and ability to back out of excavation moves when it is found to be unfruitful.

The evidence of evolution pours in, not only from geology, paleontology, biogeography, and anatomy (Darwin's chief sources), but from molecular biology and every other branch of the life sciences. To put it bluntly but fairly, anyone today who doubts that the variety of life on this planet was produced by a process of evolution is simply ignorant, inexcusably ignorant, in a world where three out of four people have learned to read and write.
—Daniel C. Dennett

Chapter 5

NEURAL CODING AND BIOLOGICAL PLAUSIBILITY

5.1 Introduction

The Nouvelle AI movement highlights the need to physically 'realize' architectures and concepts in robotics. When presenting models of biology, it is not sufficient merely to present these concepts as symbolic manipulations in a computer but determine whether these systems are biologically plausible and how these systems maybe physically grounded.

Research into biological nervous systems has uncovered evidence of different neural coding schemes [15, 132, 119, 97, 54, 122, 2, 72]. In particular, we are interested in determining how these neural coding hypotheses fit in with the evolution and operation of ANT controllers. Determining how neural systems encode information can give valuable insight into their organization and functionality. By encoding information, we mean how both data and program is organized (in computational terms) in neural systems. The assumption has been neural structures use a standardized coding scheme as these structures show remarkable regularity throughout the cerebral cortex. Work by Mountcastle [114] has suggested that brain has standardized structural components that are throughout the various brain lobes. He contends that because the various lobes in the cortex look very similar, they use the same computational methods to perform the different brain functions including visual and auditory processing. Recent experiments with ferrets appear

to validate this hypothesis and have shown that transplantation of retinal fibers to the auditory cortex just after birth results in visual orientation modules emerging in the auditory cortex [140].

In addition, it is hypothesized that the underlying principles and conditions that have resulted in the evolution of particular neural data coding schemes also influence cortical organization and regulation. If individual neurons are unreliable and inefficient in storing large quantities of data then it makes sense that stored data is distributed, with many copies and parts stored within many neurons to reduce the risk of data loss. Similarly, if neurons perform control behaviors and are also individually unreliable and inefficient, then it makes sense for their control functionality to be distributed. Thus the underlying factors that affect data coding and representations have similar implications in terms of control.

Neural encoding has traditionally been analyzed from a data representation and storage viewpoint. It is known that axons and dendrites have limited bandwidth, are noisy, unreliable and imprecise. Yet to get around these limitations, it has been theorized by Albus that the data channels are distributed in a coarse coding scheme [2]. Ballard [7] went further and hypothesized that certain centers in the brain encode information in a coarse coding representation. He highlighted how a coarse coding scheme can also account for modularity in the brain. However, his hypothesis unlike Albus[2]; did not give a detail account of how the coarse coding mechanism could be physically-realized.

In this chapter, we address how a coarse coding regulatory scheme maybe physically realized. It has already been suggested in Chapter 3 that chemicals maybe used to diffuse signals when performing neural regulation. Here we consider the role of nitric oxide playing the role of this messenger chemical and address the implication of this chemical communication through diffusion on neural system operation. In particular, the need for chemical communication implies the need for chemical receptors and a slow diffusion process has impact on system operating frequency/response times. We present detailed simulations of nitric oxide diffusion in simulated cell structures and analyze the plausibility of the whole approach. This is compared against known parameters such as size of neurons, operating frequency and independent calculation of neural activity measures.

Chapter 5 begins with a background on different coding schemes observed in neuroscience. Readers may wish to refer to this section where necessary. This is followed by a section analyzing the physical and biological plausibility of nitric oxide playing a role in coarse coding neuroregulatory function followed by implications in computation. Section 5.4 includes analysis of ANT controller solutions in terms of redundancy, sparseness and information theoretic measures.

5.2 Background

In the quest to determine how neural systems encode information (that is what format the data and program are organized/stored), several competing coding schemes have been proposed including sparse coding [119], population coding [6] and coarse coding (subset of population coding) [2, 72]. This section provides a brief

background into these coding schemes.

5.2.1 Sparse Coding

Various studies suggesting that information is represented by a few active neurons within a neuronal population [119]. This is termed as *sparse coding*. Evidence for sparse coding is apparent in higher stages of visual processing streams across many different organisms [15, 132, 119]. Sparse coding schemes are effective in storing pattern representations and maximizing memory capacity by reducing cross-talk between stored contents [167]. Furthermore, sparse coding has been shown to be applicable with local, biologically plausible Hebbian learning rules [120, 84, 170, 48, 121]. Sparse coding has also been found to be energy efficient [95]. It has also been estimated that the average firing rate is rather low because of energy constraints and sparse coding is considered a necessity in this regard [5, 94]. At its extreme, sparse coding is a localized encoding scheme in which data is encoded within a single neuron (similar to the grandmother cell hypothesis [89]). This is in contrast to a population coding where data is distributed over an ensemble of neurons.

5.2.2 Population Coding

Population coding involves combining the activity of multiple neurons to encode data. As given below, $\hat{s} = \kappa \sum_{i=1}^{n} r_i s_i$ and encodes data by summing the activity s_i of the ith neuron in a population of n neurons and r_i is the weighting constant. Population coding schemes have been observed from analyzing several brain functions including control of eye movement [97], muscle movements [54] and visual discrimination in the primary visual cortex (V1) [122, 6].

The basic premise of population coding is that the individual neurons are noisy or broadly tuned. In order for the nervous system to reconstruct or recall information, many neurons need to participate. Thus, information is laid out in a distributed fashion. Several measures have been developed to characterize population coding, including Fisher information and Kullback-Leibler divergence. A particular type of population coding is coarse coding. Unlike generic population coding and sparse coding that benefit from use of statistical or information theoretic measures, coarse coding relies on spatial geometry. More details on coarse coding can be found in the following section.

5.2.3 Coarse coding

Coarse coding is a distributed means of representation that involves use of multiple coarse receptive fields to represent a 'finer' field (Figure 5.1 left). For this example, the coarse receptive fields are circles and the area represented by 'finer' field more accurately encodes for the position q than the individual coarse receptive fields. Hinton had proposed this mechanism as means of visual perception [72]. A form of coarse coding called tile coding was developed earlier by Albus [2] and involves combining multiple coarse exhaustive

partitions of input space (known as coarse tiles) to form finer tiles (Figure 5.1 right). Albus had developed tile coding as part of a mathematical model of the cerebellum, known as Cerebellar Model Arithmetic Computer (CMAC). Both works consider coarse coding as a means of data representation. With the CMAC, the coding scheme is hypothesized to overcome bandwidth and reliability limitations of individual sensory channels within neurons in the cerebellum, while Hinton described coarse coding as an efficient means of coding visual representations within neuron-like ensembles.

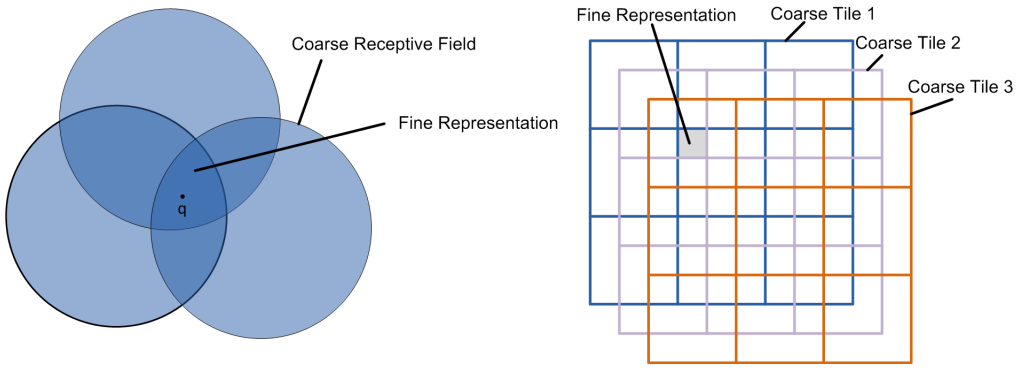


Figure 5.1: (Left) Coarse coding involves use of multiple overlapping coarse receptive fields to form a finer representation that more accurately encode for position of point q in two-dimensions. (Right) Tile coding is a form of coarse coding, where multiple overlapping coarse tiles are combined to form a finer representation of a feature.

5.3 Physical Basis

We wish also to address evaluate the physical plausibility of chemical communication occurring in the nervous system. A chemical suggested for this role is nitric oxide. The molecule is small and nonpolar suggesting it can diffuse from a source and can pass through cell and lipid structures rapidly [91, 99]. Nitric oxide (NO) was first suggested as a messenger molecule by Garthwaite *et al.* [52]. More recent reviews by Hölscher confirms the ubiquitous role of NO in the nervous system although there remains some debate over the results [76]. Numerous other studies have extended these observation to both vertebrates and invertebrates [51, 44, 53, 117].

Studies by Chapman *et al.* [27] showed learning impairment in young chicks and evidence for amnesia/learning impairment in rats for a water maze task when injected with NO inhibitors. In contrast, if the rats were pretrained on the same spatial learning tasks, the effect of NO inhibitors was found to be negligible. It has been suggested that pretraining give the animals acquired skills at solving a task that can be used in a 'recall mode' without being adversely affected by NO inhibition [76]. This suggests NO plays a role in spatial learning, but has little role in recall and execution of learned traits/skills.

It is suggested that a wireless chemical communication medium has a role to play when it serves a complimentary role with the wired medium. Neurons for example rely on exploratory schemes (trial and error based search) to form synaptic connections with neighboring neurons, something that can be initiated

by chemical signaling [24, 139, 67]. Montague *et al.* [112] have suggested that NO and similar chemicals help to initiate development and function of synaptic connections.

5.3.1 Coarse Coding Regulation

In this section we generalize the coarse coding interaction of chemical diffusion fields emitted from decision neurons (Section 3.3.2) so that we may determine if such conditions can occur in a physical system. We may imagine the diffusion of a chemical concentration field from a chemical *emitting* neuron as equivalent to a coarse receptive field (see Figure 5.2, 5.1). This chemical acts as a stimulus used to trigger into operation (or transition to a certain state) a target neuron. The chemical diffusion field emitted by a single neuron is not precise, in that it can spill over into a neighboring volume. However, the assumption here is that the chemical emitting neurons (senders) need to trigger/activate into operation a specific group of neurons (receiver) at a particular 'fine' position. Thus multiple chemical emitting neurons can coordinate their actions by emitting overlapping chemical fields that *encode* for the position of the intended receiver (individual or groups of neurons). The receiver neurons then need to sense it is positioned at a chemical concentration peak and be triggered into operation. This act of chemical emitting neurons selectively activating or inhibiting receiver neurons is a form of neural regulation. Therefore we label this as a *coarse coding regulatory process*, since a coarse coding mechanism is used to activate/inhibit other neurons. Here we mathematically define the conditions outlining this process for a physical medium.

We define, $\mathbf{x} = [x_1, x_2, x_3]$, a position vector in three dimensions. Within three dimensional space we define two concentration fields, $\rho_{A,\alpha}$, $\rho_{B,\beta}$ of chemical substances α and β respectively. Also, $\rho_{A,\alpha}(\mathbf{x}), \rho_{B,\beta}(\mathbf{x}) > \rho_{\min} \geq 0$ for connected volumes A and B respectively. $\rho_{\min}$ is a constant and is the minimum concentration that can be sensed in the system. A neuron located in this system can be in any one of $h \in [0, 1, .., H]$ states.

We also define $p^h_{C,\xi}(\bar{\rho}_{C,\xi})$, probability of setting neurons into a particular state h (i.e. trigger into operation) located in volume C and is dependent on $\bar{\rho}_{C,\xi}$, the average concentration of a stimulus substance ξ in volume C. Thus,

$$\bar{\rho}_{C,\xi} = \frac{1}{V_c} \int_C \rho_{C,\xi}(\mathbf{x}) d\mathbf{x} \tag{5.1}$$

where $V_C = \int_C d\mathbf{x}$ is the volume and $d\mathbf{x} = dx_1 dx_2 dx_3$. It should be noted that $p^h_{C,\xi}$ is assumed to be proportional to a monotonically increasing positive function of the concentration ρ_ξ as substance ξ is classified as a stimulus.

For this system, we define a coarse coding regulatory process (see Figure 5.2) as the following:

$$p^h_{C,\xi} > \max\left\{p^h_{A^*,\xi}, p^h_{B^*,\xi}\right\} \tag{5.2}$$

In addition, the following condition also has to be met:

$$V_c < \min\{V_A, V_B\} \tag{5.3}$$

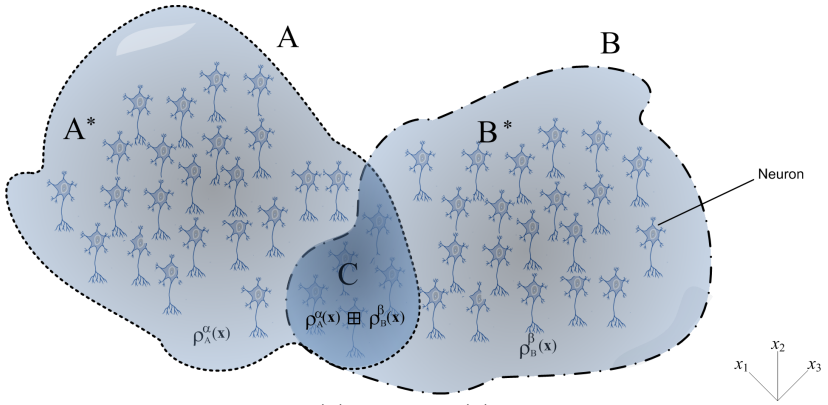


Figure 5.2: Chemical concentration fields $\rho_{A,\alpha}(\mathbf{x})$ and $\rho_{B,\beta}(\mathbf{x})$ within volumes A and B respectively intersecting over volume C.

where volume $C = A \cap B$ and regions A^* and B^* are the non-overlapping subregions of A and B respectively. That is the probability of triggering receiver neurons into state h by a stimulus chemical ξ must be higher in volume C (that overlaps between volume A and B) than in the non-overlapped region and the volume C must be smaller than volume A or B. Therefore we say that multiple concentration fields coarse code for the position of receiver neurons that are to be stimulated.

To determine how substances α, β relate to stimulus ξ, we first denote the interaction of two concentration fields of α and β at position $\mathbf{x}$ within an enclosed volume C forming substance ξ:

$$\rho_{C,\xi}(\mathbf{x}) = \rho_{A,\alpha}(\mathbf{x}) \boxplus \rho_{B,\beta}(\mathbf{x}) \tag{5.4}$$

When $\xi = \alpha = \beta$ indicates a situation where no chemical reactions occur and simply involves mixing of two concentration fields, thus the concentration of substance ξ at position $\mathbf{x}$ is $\rho_{A,\alpha}(\mathbf{x}) + \rho_{B,\beta}(\mathbf{x}) = \rho_{C,\xi}(\mathbf{x})$ and $\boxplus$ is substituted with the $+$ operator. In order for the inequality condition (5.3) to be satisfied, then it follows that $\rho_{C,\xi}(\mathbf{x}) > \rho_{\min C} > \rho_{\min} \geq 0$ and where $\rho_{\min C}$ is a constant. For the case where $\xi \neq \alpha \neq \beta$, then it follows that the interaction of substance α and β results in a chemical reaction forming substance ξ. Lets assume the substances undergo the following chemical reaction:

$$\alpha_n + \beta_m \rightarrow n\xi + (m-n)\beta \tag{5.5}$$

where substance α and β initially take on a molecular form α_n and β_m, leaving an excess amount of substance β left over. In this case, the concentration of α molecules in volume A is given by $\rho_{A,\alpha}$ and β molecule in volume B as $\rho_{B,\beta}$ respectively, with $m > n > 0$ and $m, n \in \mathbb{I}$. It follows from the chemical reaction process that molecule $\xi = (\alpha\beta)$. Therefore the interaction of the two substances results in the following:

$$\int_C \rho_{C,\xi}(\mathbf{x}) d\mathbf{x} = \int_C \left[\rho_{A,\alpha}(\mathbf{x}) \boxplus \rho_{B,\beta}(\mathbf{x})\right] d\mathbf{x} = \min\left\{ \frac{n}{m} \int_C \rho_{B,\beta}(\mathbf{x}) d\mathbf{x},\ n \int_C \rho_{A,\alpha}(\mathbf{x}) d\mathbf{x}, \right\} \tag{5.6}$$

where we assume the two molecules of α and β are thoroughly mixed over volume C.

5.3.2 Diffusion Simulation

Based on the work by Philippides *et al.* [124], we simulate NO interaction in neural tissue as a chemical diffusion process and determine if the conditions outlined in Section 5.3.1 can be physically realized and establish the plausibility of the decision neuron characteristics outlined in Chapter 3. The dynamics of the diffusion process is governed by the differential equation of the following form:

$$\frac{\partial c}{\partial t} - D\nabla^2 c = -\lambda c \tag{5.7}$$

where c is the concentration D is the diffusion coefficient and λ is modeled to account for loss of chemical due to processes such as oxidation and binding [124]. This process has been observed to decay exponentially with respect to time [111].

For a solid spherical source, 5.7 is solved to get the following concentration field:

$$\begin{aligned} c_{\text{SolidSphere}}(a,r,t) = Q\rho e^{-\lambda t}\Bigg[\tfrac{1}{2}\left(\operatorname{erf}\left(\tfrac{a+r}{2\sqrt{Dt}}\right) + \operatorname{erf}\left(\tfrac{a-r}{2\sqrt{Dt}}\right)\right) \\ - \tfrac{1}{r}\sqrt{\tfrac{Dt}{\pi}}\left(\exp\left(\tfrac{-(a-r)^2}{4Dt}\right) - \exp\left(\tfrac{-(a+r)^2}{4Dt}\right)\right)\Bigg] \end{aligned} \tag{5.8}$$

where a is the radius of the spherical source, and is the radial distance from the source center, $Q\rho$ is the concentration of NO produced per second (in units of moles per volume per second) and t is time. The value for $Q\rho$ is independent of the cell shape and is based on Wood and Garthwaite [52]. Earlier works in the field assumed cells to be point sources. The problem with such an assumption is the problem of physical plausibility, where the concentration is infinity at the source origin. The works of Philippides *et al.* [124] overcame this issue by assuming the cells are hollow spherical source. Based on principle of superposition, one can then easily express a hollow spherical field as combination of two solid fields as shown below:

$$c_{\text{HollowSphere}}(a,b,r,t) = c_{\text{SolidSphere}}(a,r,t) - c_{\text{SolidSphere}}(b,r,t) \tag{5.9}$$

where here a is the outer radius and b is the inner radius.

In addition to defining the geometry of the chemical source, we need to consider the temporal characteristics of the diffusion process. We can express a continuous diffusive source using temporal superpositioning as follows:

$$C_{\text{cont}}(a,b,r,t) = \int_0^t S(t-t')c(a,b,r,t')dt' \tag{5.10}$$

where $c(a,b,r,t')$ is the field from a hollow sphere and $S(t)$ governs the emission function of the chemical. The emission function $S(t)$ for this model consists of a finite length square wave of the concentration field and is the following:

$$C_{\text{square}}(a,b,r,t_1+T) = \int_{t_1}^{t_1+T} S(t_1+T-t')c(a,b,r,t')dt' \tag{5.11}$$

where the source synthesizes a chemical for T seconds and the concentration is given after t_1 sec after it has stopped synthesis.

5.3.3 Superpositioning and Coarse Coding

A neuron for these simulations is modeled as a hollow sphere for simplicity sake and the net interaction from two neurons is shown in Figure 5.6. A diffusion coefficient, $D = 3300\ \mu m^2 s^{-1}$ for an aqueous salt solution and $\lambda = 0.1386\ s^{-1}$ is used for the simulations (consistent with [124, 111]). The time evolution of the diffusion process is shown in Figure 5.6, columns 1-4 for 5 different initial conditions. The resulting two dimensional NO concentration fields are then discretized into $20 \times 20\ \mu m$ squares as shown in Figure 5.6 columns 5. Due to the discretization process, we consider several configurations, namely two neurons aligned side by side at 20 and $40 \mu m$ apart and diagonally at 28.3 and $56.6\ \mu m$ apart.

Based on the criteria outlined in Section 5.3.1, what is important is determining the region of highest concentration. As noted from these simulations the region of highest concentration form at the center of the two concentration field sources and steadily remain so until any future disturbance (Figure 5.6, 5.5 left). Besides this higher fidelity NO concentration field model, we also consider a low fidelity binary source concentration field model (Section 3.3.2) as shown in column 6. The net interaction of the coarse coding process for the high fidelity and low fidelity models match for the 2×2 cases immediately after the chemical burst from the sources (Figure 5.6, rows 1-2).

For larger separation distances, the desired coarse-coded configuration is reached after a finite delay in time, which is in proportion to the square of the separation distance (as shown in Figure 5.3). The binary model of the 3×3 configuration (column 6, row 4) shows noticeable differences over the discretized results (column 5, row 4). For the discretized results, we see that the column of squares at the center is not of equal concentration so although, the interaction satisfies our definition of a coarse coding selection process, it does differ with the lower fidelity model. One workaround toward matching the binary model to the discretized results is to assume that the slight difference in concentration between the squares in the central column cannot be detected.

Knowing that biological neurons operate within the 0.1 to 100 Hz range [5], this puts an upper bound of 25 cell radii on the operational distance of such chemicals signaling mechanisms (using best-fit from Figure 5.3). Figure 5.4 labels discrete squares of interest measured in Figure 5.5 (left). For the spike train signal shown at the bottom of Figure 5.5 (left). We see concentration spikes occurring at C_1, with reduced amplitude for C_0. However within an action potential cycle, we see that concentration of $C_0 > C_1$ for about 50% of the time, at a frequency of 5 Hz. The percentage of time $C_0 > C_1$ within an action potential cycle is dependent on the action potential period as shown in Figure 5.3 (right). Furthermore it should be noted, that difference in concentration values between the neighboring squares, C_0 and C_1 remains minute. To increase this absolute difference in concentration values further, the burst time length could be increased as shown in Figure 5.5 (right). Increase in sustained burst time results in a linear increase in the concentration values of

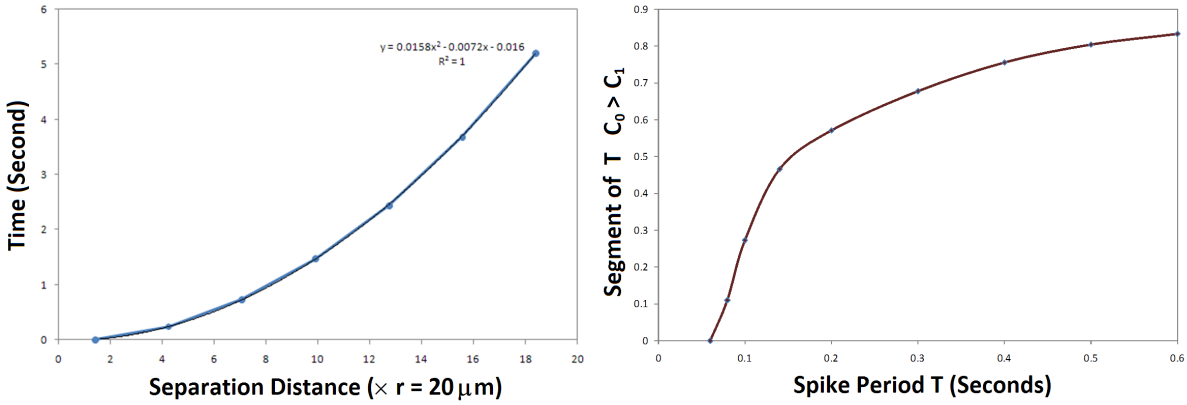


Figure 5.3: Effect of separation distance on time taken to reach steady-state coarse-coding configuration using NO (left). Effect of action potential cycle on fraction of period (T) for system in steady-state coarse-coding configuration using NO (right).

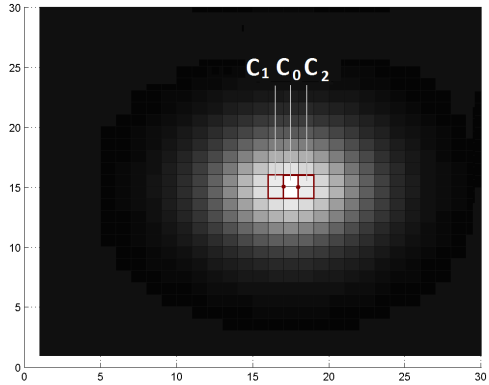


Figure 5.4: Configuration showing two nitric oxide chemical diffusion sources being 'merged' once having reached the steady-state coarse-coding configuration. Grid squares of interest labeled C_0, C_1 and C_2.

both C_0 and C_1. A chemical signaling mechanism could use this property to ensure the concentration signal has sufficient amplitude at the intended destination.

Based on this analysis, it is observed that chemical signaling is a feasible means of communication over relatively short distance (up to 25 cell radii) between neighboring neurons to operate within an operating frequency of 0.1 to 100 Hz.

Two possibilities as to how chemical communication could impact regulation of neuron ensembles are summarized in Table 5.1. One involves using chemical communication as a postprocessing approach while the other involves chemical communication in a preprocessing approach. As shown in Figure 5.7, we make this comparison for a feedforward network with n_{input} neurons at the input layer, n_{hidden} neurons at the hidden layer and n_{output} neurons at the output layer. In total, there are α layers in the network.

The postprocessing approach involves use of chemical communication as a filter mechanism that can be used to override or reweigh network output signals. The chemical concentration is used to amplify/deamplify the neuron output signal as shown in Table 5.1 in a distributed manner. In a regulatory context, the process is binary and results in either shutting-off/overiding the signal from each output neuron or not.

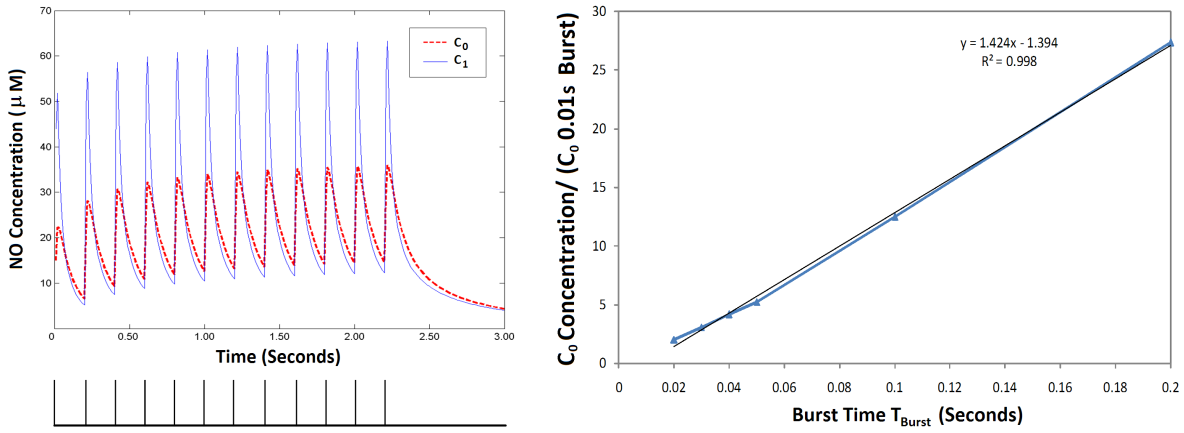


Figure 5.5: NO concentration within grid squares C_0 and $C_1 = C_2$ (by symmetry) (left) for the given spike train (bottom left). Effect of sustained burst time on concentration magnitude of grid square C_0 (right).

The advantage of this postprocessing approach is that the process of filtering and transmission of a chemical signal can be done in parallel to computation done by the nonchemical emitting neurons. Thus the time needed for computation is αt_{neuron} instead of $t_{\text{diffusion}} + \alpha t_{\text{neuron}}$ for the preprocessing step. $t_{\text{diffusion}}$ is the total time needed for chemical producing neurons to compute its sensory input and diffuse a messenger chemical to the intended destination, while t_{neuron} is the time needed for each neuron to process sensory input and produce a spike. Thus if the transmission of the chemical signal involves a high degree of latency, then performing the chemical communication step in parallel with other computational processes is temporally efficient as in the postprocessing approach. Parallel execution of events depends on the fact that the activity between the chemical emitting and receiving neurons are coordinated. This is to ensure the chemical signal arrives in time for the output neurons to sense the regulatory message. If there is an unexpected delay in the transmission of the chemical signal, then the output neurons will miss the regulatory message and act in an intended fashion.

One of the other possibilities is that chemical communication could be used as a preprocessing regulatory mechanism used to initiate activity among specified neurons. The question remains how the chemical neurons can sense regions of high concentration and be activated. Unlike the postprocessing approach where chemical sensing neurons would rely on the amplitude of chemical concentration field, in this approach the neurons need to measure concentration gradients. To measure concentration gradients, each neuron needs to have multiple concentration readings. Therefore neurons at local concentration peaks get triggered while those at the slopes remain inhibited. This can result in situations where neurons at multiple local peaks of different amplitude get activated when according to (5.2), only neurons at the highest concentration peaks should always be selected in the deterministic case and have higher probability of being selected for the stochastic case. Thus, the neurons also need to rely on the absolute concentration value besides the concentration gradient information.

It would appear that neurons having to sense concentration gradients using multiple concentration sen-

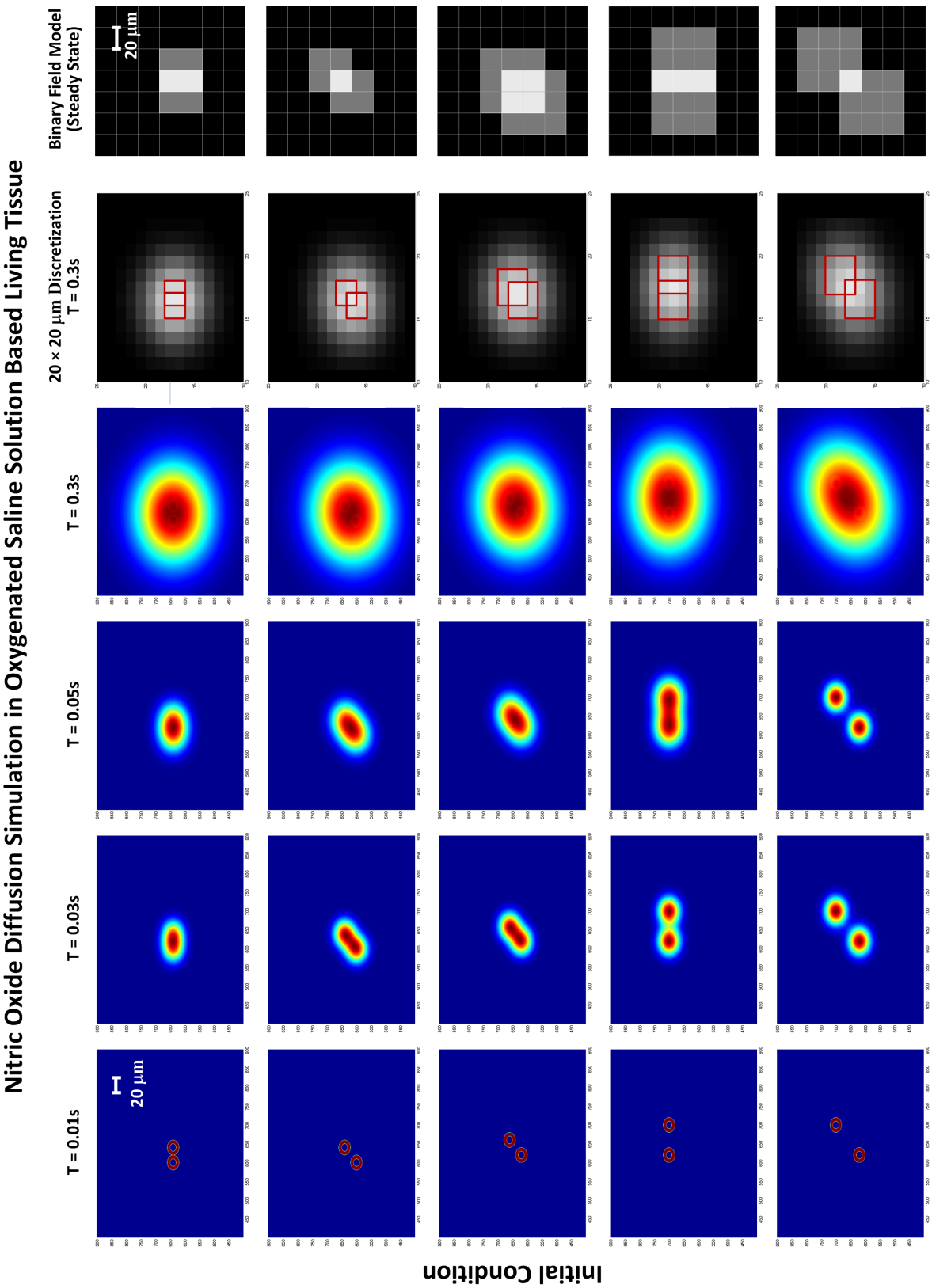


Figure 5.6: Nitric oxide diffusion simulation showing time evolution of two diffusion sources for various initial configurations (columns 1-4). Discretization of resultant field at 0.2 seconds (column 5) and binary field model (column 6).

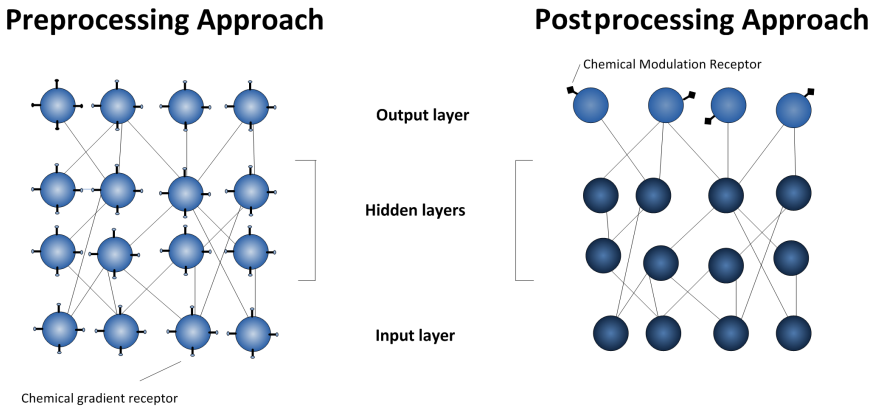


Figure 5.7: Schematic representations of a preprocessing (left) and postprocessing (right) approach to neuronal volume signalling using chemicals.

sors are outwardly more complicated than relying on chemical modulation. However, it should be emphasized that many prokaryotes and single-celled eukaryotes can perform both chemotaxis and phototaxis, (since these are necessary survival behaviors). Therefore it is plausible that such behaviors have evolved in the nervous systems of multicellular organisms.

Table 5.1: Comparison of postprocessing and preprocessing approach to neuronal volume signalling using chemicals.

Chemical Communication	**Preprocessing**	**Postprocessing**
Number of Neurons:		
Chemical Gradient Sensing	$n_{\text{input}} + n_{\text{hidden}} + n_{\text{output}}$	0
Chemical Modulation	0	n_{output}
In Active Mode	$\varrho n_{\text{input}} + \eta n_{\text{hidden}} + \kappa n_{\text{output}}$	$n_{\text{input}} + n_{\text{hidden}} + n_{\text{output}}$
In Dormant Mode	$(1-\varrho) n_{\text{input}} + (1-\eta) n_{\text{hidden}} + (1-\kappa) n_{\text{output}}$	0
Computation Time	$t_{\text{diffusion}} + \alpha t_{\text{neuron}}$	αt_{neuron}
Activation Function	$\phi = \phi_{\text{pre}}(\rho(\mathbf{x}), \nabla\rho(\mathbf{x})) \cdot \phi_{\text{activation}}$	$\chi_{\text{post}}(\rho(\mathbf{x})) \cdot \phi_{\text{activation}}$

The preprocessing approach is initiated once a chemical signal diffuses into the network, triggering a chain reaction that activates certain neurons and leaves others dormant. The advantage of this approach over a postprocessing modulatory approach is that it is energetically more efficient: Not all the neurons within the ensemble have to be active all the time. Hence, $\varrho n_{\text{input}} + \eta n_{\text{hidden}} + \kappa n_{\text{output}}$ neurons need to be active for the preprocessing approach, where ϱ, η, $\kappa \in [0, 1]$ and are the fractions of input, hidden and output neurons activated respectively. This is in contrast to having all the neurons active for the postprocessing approach. Sensory preprocessing can be used to coordinate and activate neurons specialized to handle specific sensory input as summarized in ANT. However, this adds a time delay dependency in the whole process. Over shorter distances, this time delay according to Figure 5.3 (left) could be kept under 1 s. In considering these regulatory approaches, the preprocessing approach is energetically more efficient and does not require chemical emitting and sensing neurons to be acting in synchrony. Both of these factors are important considering that the human brain accounts for 20% of the resting metabolism [5, 135] and neurons

are known to be asynchronous. However, with other organisms where the nervous system metabolism maybe less, a postprocessing approach could also be feasible. From a computational standpoint, the preprocessing approach is also more efficient in that fewer neurons need be evaluated than without.

It should be noted that a wired medium of communication relies largely on electrical pulses to perform signal transfer. However, this medium of communication relies on (significantly) shorter range chemical diffusion between axons clefts and dendrites of two neighboring neurons. A wireless communication medium using chemicals is expected to have a slower reaction time as a chemical needs to be diffused over larger distances with sufficient 'signal strength' so that the receiver can reliably distinguish this signal from 'background' noise. We argue that a chemical (wireless) communication medium serves a complementary role to a wired one when trying to synchronize neighboring neurons that are unconnected and operate at different frequencies. In order for the neurons to be synchronized, (i.e., start a particular operation in parallel), there needs to be a delay/waiting mechanism to bring the neurons 'to attention.' A slower reaction time is sufficient in this circumstance as this process needs to guarantee that all the neighboring neurons are acting in synchrony and the 'waiting' period needs to be longer than the operating action potential cycle of the slowest neuron.

5.3.4 Computation

After having explored conditions under which a coarse coding regulatory process may occur through diffusion of nitric oxide in organic tissue, we wish to determine whether a coarse coding process can be exploited to perform logical operations including NOT, AND, OR and XOR gates. Coarse coding selection involves superpositioning of coarse noisy fields that in combination form finer fields. This superpositiong of fields is used to perform *selection* of the finer fields based on the coarse coding selection criteria. We use the word 'selection' to describe this process based on a selectionist view of computation. Within the selectionist view, there exist multiple competing control sequences. A process of deliberation (selection) occurs, where certain control sequences or representations are favored resulting in an outcome [43].

Let us define a two dimensional $m \times m$ array, **b** (see Figure 5.8), representing a two dimensional plane of size $l \times l$, where grid square, $b_{i,j} = \frac{1+(-1)^{i+j}}{2}, b_{i,j} \in \{0, 1\}$ for $i = 1 \ldots m$ and $j = 1 \ldots m$ and is in a checkerboard pattern of 1s and 0s. Let us label the $m \times m$ array as the output layer. In addition, let us also include a second array of $m \times 2$ elements, **a**, representing a plane, $l \times \frac{2l}{m}$. The contents of the array form a checkerboard pattern described earlier and let us label it the input layer. Let each column of the input layer represent a binary variable $s_i \in \{a_{i,1}, a_{i,2}\}$. We also introduce the 'diffuse process,' by which a chemical is diffused from the nodes of this two-dimensional plane and form a binary concentration field. Setting $s_i = 1$ thus involves selecting the grid square region $a_{i,2}$ through a coarse coding selection process, whereby we select an overlapping region from multiple diffusion sources according to Section 5.3.1. Furthermore, we impose some constraints, whereby a diffusion process forms a binary, coarse concentration field over a 2×2 array in the output layer. In addition, let us assume selection of a particular value of s_i can trigger one or more chemical diffusion bursts in the output layer.

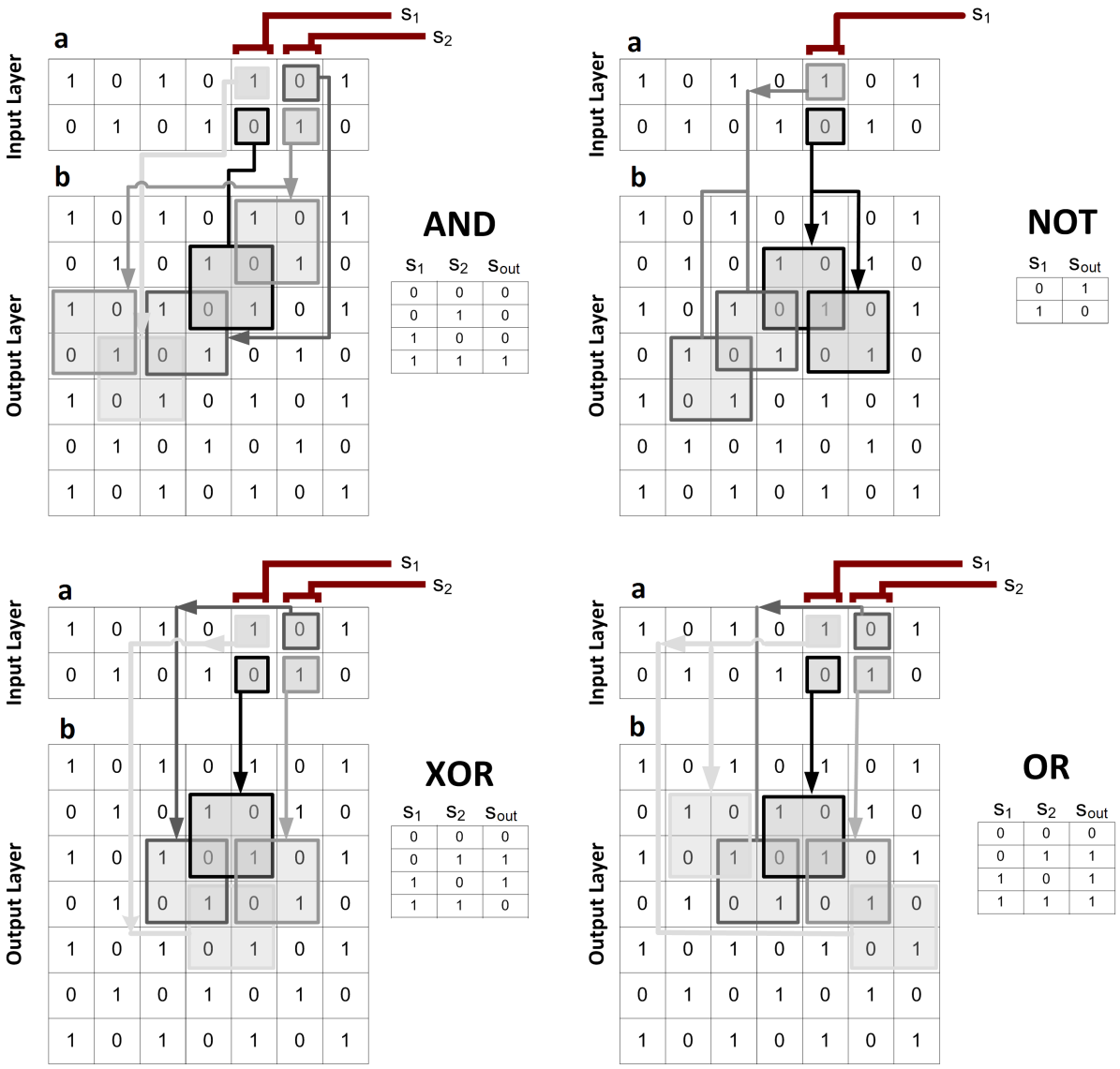


Figure 5.8: Coarse coding of multiple chemical diffusion sources to perform the AND, NOT, XOR and OR logic operations.

Since the variables are binary, a 2×1 selection of s_i results in a stochastic result for the variable s_i, with 1 or 0 being selected with equal probability. Within the output layer, we let $b_{i,j}$ be one possible representation for a single binary variable s_{out}. It is important to emphasize that we have defined a physical system and a means of representing this physical system through this mapping of the physical system to binary variables. Having defined the physical system and representation of it using binary variables, we can then build circuits that describe the logical NOT, OR, AND and XOR operators (Figure 5.8). Selection of a 2×2 region in the output layer for s_{out} can be interpreted as a noisy output, with 1 or 0 being selected with equal probability. Apart from the circuit description shown in Figure 5.8, one may have additional redundant selection field acting in parallel. The need for redundancy may exist where if individual mechanisms involved in the

diffusion process (i.e., to pump the chemical) were damaged or act unreliably. Let p_{rel} be the probability that the chemical is successfully diffused from a node. Then if there is n identical mechanisms centered on the same node, the overall probability that a chemical is successfully diffused from the node is $1-(1-p_{\text{rel}})^n$ for $n \geq 0$. With n sets of diffusion fields in place of each diffusion field shown for the logical operators, the overall probability of getting the desired output state also becomes $1-(1-p_{\text{rel}})^n$. Thus by including redundant selection fields we can improve the overall reliability of the mechanisms. As we have shown, not only can a coarse-coding selection framework work serve as a signalling mechanism, but coordinated use of such a scheme can be used perform computation.

5.4 Degeneracy, Coordination and Sparseness

In this section we analyze ANT controllers both using neural coding measures and phenotypic activity. It had been earlier observed that when given the opportunity, the number of active neurons increases even after convergence to a solution suggesting evidence of degenerative protective mechanisms against adverse mutation (Figure 3.20). Degeneracy is where multiple topologies with the same function are present in the controller [43] and is a form of populating coding. A special case of degeneracy is redundancy, where the topologies have the same topology and functionality. Degeneracy is an important characteristic of biological systems that enable a system as a whole to be robust despite the fragility of its parts [43]. From an evolutionary standpoint, the artificial tissues evolve not only solve the intended task but also steadily develop ways to ensure its survival and successful transfer of its progeny. Figure 5.10 showing the sparseness measure of the population best for the sign-following task corroborates with this observation. This sparseness measure was developed by Rolls and Tovee [136] and is used for one-sided distributions (i.e., either positive or negative) responses as given below:

$$a = \frac{\left(\frac{1}{n}\sum_{i=1}^{n} s_i\right)^2}{\frac{1}{n}\sum_{i=1}^{n} s_i^2} \tag{5.12}$$

The measure involves taking the activation ratio from i different stimuli taken over n trials where $s_i \in \{0, 1\}$ and a approaches 0, implies the distribution is tending towards high sparsity, while $a = 1$ indicates neurons are fully redundant. It is interesting to note that the ANT solutions lean toward a sparse distribution (with regards to neuronal activity) although such a distribution is not imposed on the architecture. When mutation to the growth program is abruptly stopped after 1000 generations, we see evidence of more sparseness than without. This indicates that without mutations to the growth program, neurons are assigned to handle very specific sets of sensory input and thus quickly specialize. It is also indicative of a competitive process, where very few neurons are active at any one time and with 'bits and pieces' of the controller functionality localized among a few neurons.

There is evidence that mutation to the growth program results in degenerate features being added to the

tissue. Increased degeneracy implies activity is distributed among more neurons, thus trending away from a sparse distribution and towards population coding. With population coding, functionality is shared between neurons providing functional redundancy. This is a cooperative trend where multiple neurons cooperate to represent functionality. Intuitively speaking the controllers have opted for greater redundancy in the face of increased mutation in the system since redundancy better protects the controller against mutational damage.

Figure 5.9 shows that as ANT controllers converge to a solution, the total fraction of active motor neurons exponentially decreases in a 'regressive' trend, although the number of genes added to the tissue increase in a linear trend. Selectionist models account for such an outcome particularly since the system starts with a large repertoire of functionality and is pruned down to using 'necessary' functionality through selection. Based on this analysis, ANT controllers show biologically consistent trends with evidence of sparse-coded activity, increased degeneracy/evidence of populating coding due to steady mutations to the growth program and 'regressive' trends in neuronal activity.

Under these conditions, the controllers are tending towards less than 20% neurons simultaneously active where the decision neurons emit 3×3 chemical fields. It takes 0.3 seconds for a 3×3 fields to reach a coarse coding configuration (Figure 5.6). If we were to assume a preprocessing regulatory approach (Section 5.3.3), then the population of wired neurons need to be able to react to the chemical stimulus and fire output signals within at least a 0.3 s time period (3.3 Hz) before the next burst of chemicals from the chemical emitting neurons.

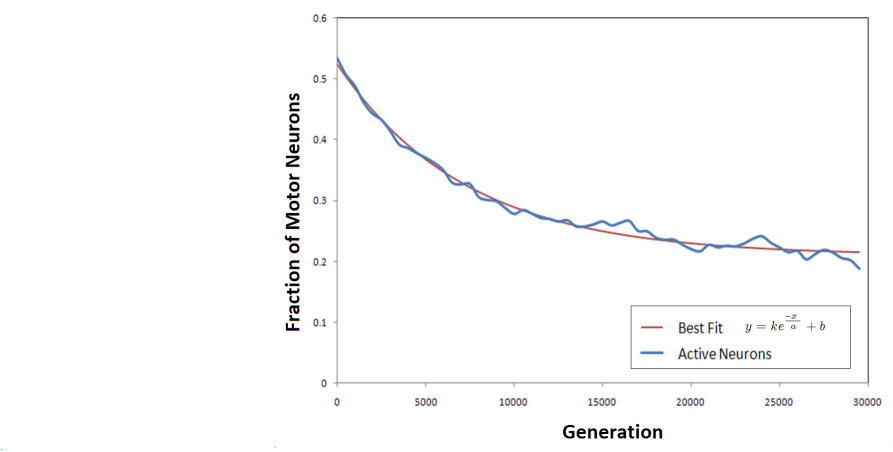


Figure 5.9: (Right) Fraction of tissue neurons active among population best averaged over 20 EA runs for for the sign-following with diameter of decision neuron concentration field of 3 squares for a neuron 'gene' addition rate of 0.07 genes/generation. Best fit parameters: $a = 7277.4$, $b = 0.209$, $k = 0.31$.

5.4.1 Information Theory

In this section we analyze the content of ANT controller solutions using information theory. The intention is to quantify factors at play during the artificial evolutionary process, particularly the emergence of new functionality. For this analysis we make the assumption that a genome maybe represented by the probability

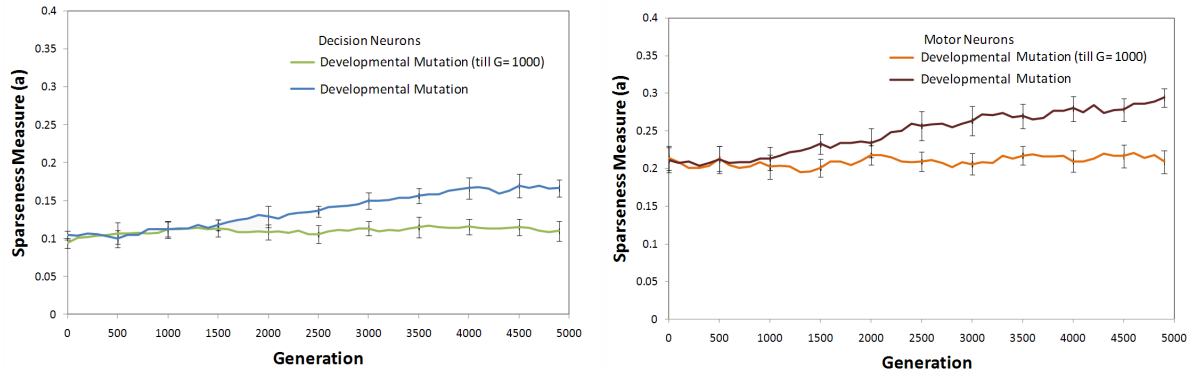


Figure 5.10: Sparseness measure of decision (right) and motor (left) neuron activity among ANT solutions (population best for the sign following task, averaged over 60 EA runs).

distribution of its actions. This is in contrast to comparing the raw content of genes between individuals where it is not possible to determine whether expressed genes play a phenotypically neutral role or not. Hence an alternative is to use a frequentist approach and measure the information content of the activity of the individual as it interacts with its environment.

Shannon's Entropy of Phenotypic Activity

To measure the phenotypic activity, we arrange the set of q unique activity vectors into a matrix, $\mathbf{B} = [b_1, \ldots, b_q]$. Each $b_i = [c_1, \ldots, c_k]^{\mathrm{T}}$ is a column vector with k elements, representing the state of k neurons, where $c_k \in \{0, 1\}$. By unique we mean that the column vectors $b_1 \neq b_2 \neq \ldots \neq b_q$. The activity of the phenotype at timestep t can be also expressed as a column vector $S(t) = [s_1 t, \ldots, s_k(t)]^{\mathrm{T}}$, with k elements, where $s_k(t) \in \{0, 1\}$ and indicates whether the kth neuron spikes or not at time t. We are interested in calculating the frequency, F_i of the activity vector b_i triggered during a controller's lifetime. By calculating frequencies, we may then proceed to calculate Shannon's entropy with respect to phenotypic activity. We compute F_i as follows:

$$F_i = \sum_{t=1}^{T} \beta([\vec{1} - S(t)] \cdot [\vec{1} - b_i] + S(t) \cdot b_i,\ k) \tag{5.13}$$

where T is the total number of timesteps during a controllers's lifetime. $\beta(d, k)$ is a thresholding function defined below:

$$\beta(d, k) = \begin{cases} 1, & \text{if } d = k \\ 0, & \text{otherwise} \end{cases} \tag{5.14}$$

where $d \in \mathbb{I}$, k is the number of elements in column vectors b_i, $S(t)$ and $\vec{1}$. Each element within $\vec{1}$ is a 1. From computing F_i, the frequency of the ith unique activity vector, we may then proceed to compute $p(F_i)$. It follows that the probability of the ith activity vector $p(F_i) = \frac{F_i}{\sum_{j=1}^{q} F_j}$. Having computed the probabilities for the q unique activity vectors, Shannon's entropy $H(F)$ is as follows:

$$H(F) = -\sum_{i=1}^{q} p\left(F_i\right) \log_2 p\left(F_i\right) \tag{5.15}$$

This procedure can be generalized to measure the activity of any discrete set of time varying variables and hence can also be used to measure the information content of output behaviour (activity) of the ANT controllers, in addition to the discrete sets of sensory inputs encountered during a controller's lifetime. Using this procedure we determine the Shannon's entropy of the input, decision, motor control neurons and output behaviours as shown in Figure 5.11.

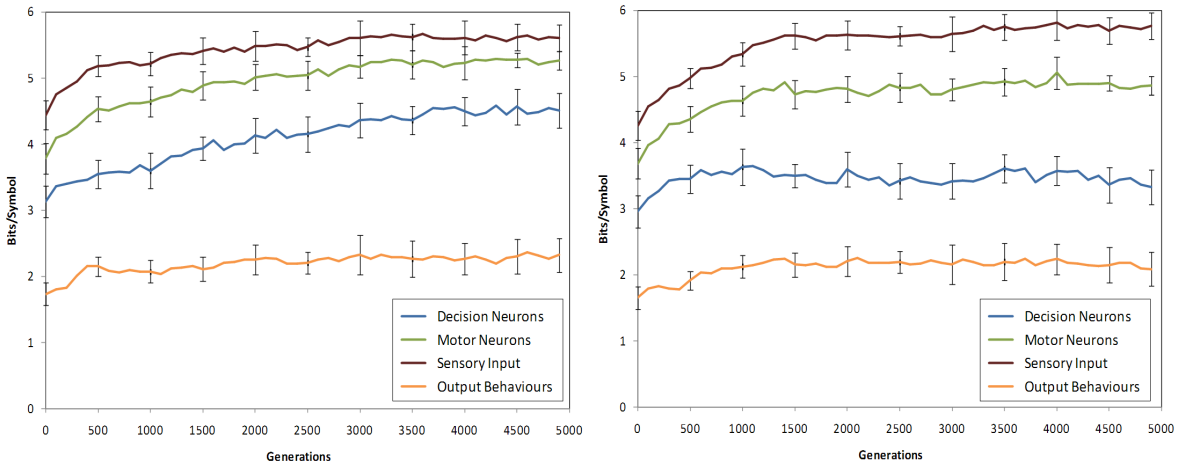


Figure 5.11: Shannon's entropy measure of activity among ANT controllers with (right) and without (left) mutations to the growth program after 1000 generations (for the sign-following task, population best averaged over 60 EA runs).

Mutual Information of Phenotypic Activity

Based on the Shannon entropy measure of phenotypic activity developed earlier, we proceed to determine the mutual information of the phenotypic activity of two individuals. The mutual information from this frequentist approach is the following:

$$I(F^u; F^v) = \sum_{i=1}^{q}\sum_{j=1}^{q} p(F_{i,j}^{u,v}) \log_2 \left(\frac{p(F_{i,j}^{u,v})}{p(F_i^u)p(F_j^v)} \right) \tag{5.16}$$

where F^u and F^v are the frequency vector for individual u and v respectively. i and j are unique activity vectors for individuals u and y respectively. Thus F^u and F^v are computed using the procedure outlined earlier to determine Shannon's entropy. In addition, we need to compute the joint probability, $p(F_{i,j}^{u,v})$. To do this we determine $\mathbf{B}^{\text{inp}} = [b_1^{\text{inp}}, \ldots, b_q^{\text{inp}}]$, that is the unique combination of sensory input states encountered by both individuals, u and v. For each one of these sensory input combinations, we determine the joint frequency of the neuron activity as follows:

$$F_{i,j}^{u,v} = \sum_{t=0, S_{input}^u(t)=S_{input}^v(t)}^{T} \begin{array}{l} \beta([\vec{1} - S^u(t)] \cdot [\vec{1} - b_i^u(t)] + S^u(t) \cdot b_i^u(t), k_u) \cdot \\ \beta([\vec{1} - S^v(t)] \cdot [\vec{1} - b_j^v(t)] + S^v(t) \cdot b_j^v(t), k_v) \end{array} \tag{5.17}$$

where T is the total number of timesteps during a controllers's lifetime, k_u and k_v is the number of elements in the column vectors for individual u and v respectively. Since its not assured both individuals have the same

number of potential active neurons, we set the number of columns for both individuals be $\bar{q} = \max(q^u, q^y)$ and fill in 0s for these additional entries. It follows the joint probability of the activity is as follows:

$$p(F_{i,j}^{u,v}) = \frac{F_{i,j}^{u,v}}{\sum_{j=1}^{\bar{q}} \sum_{i=1}^{\bar{q}} F_{i,j}^{u,v}} \tag{5.18}$$

Figure 5.12 shows the average mutual information between every unique pairing of individuals within the population of ANT controllers. The comparison made in terms of the decision neuron, motor neuron and output behaviour activity with and without mutation to the growth program after 1000 generations.

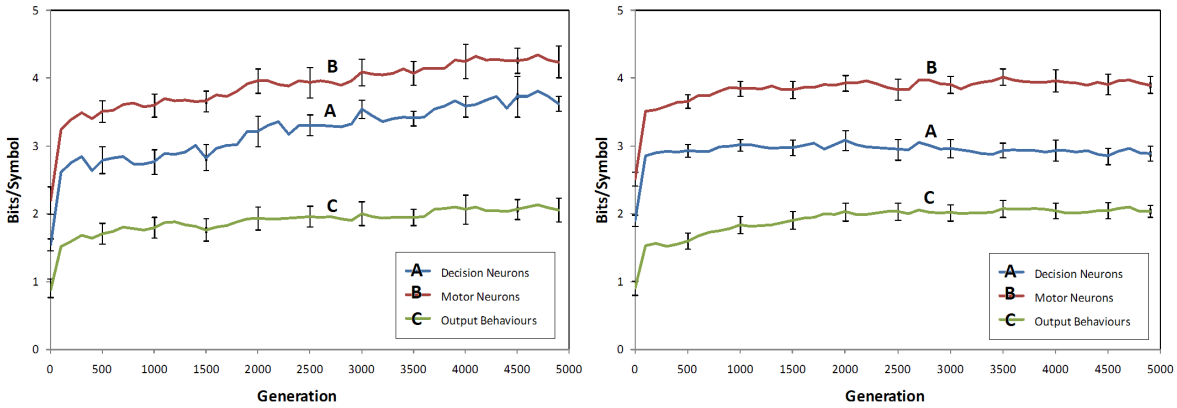


Figure 5.12: Mutual information measure of activity among ANT controller populations with (right) and without (left) mutation to growth program after 1000 generations (for the sign following task, averaged over 60 EA runs).

Information Theoretic Analysis

Figure 5.11 shows the Shannon information content of ANT controllers (population best) undergoing evolution for the sign following task. The interaction of the ANT controller with its test environment could be represented in the sender-receiver communication model shown in Figure 5.13. ANT in this scenario *filters* information from its sensory input and triggers a set of output behaviours. Thus it would be expected that the information content closer to the information source (activity of motor neurons, decision neurons) is higher than the information content of the controller output (Figure 5.11). The ability for artificial neurons to generalize their sensory input from an information theoretic point of view implies information filtering. Filtering processes results in the loss of information going from the sensory input (source) to the output behaviours for each controller and confirmed in Figure 5.11. However, it should be noticed as the evolutionary process proceeds, the controllers manage to increase the information content of the sensory input representations (from comparison of motor neuron and input neuron activity) as shown in Figure 5.11. This is indicative of the controller acquiring new functionality through evolution, that enables it to better exploit its sensors, gaining more information content and better traverse its environment.

The repertoire of sensory input combinations encountered by the controllers increases, thus increasing their information content. This may seem counterintuitive, but a randomly initialized set of control rules

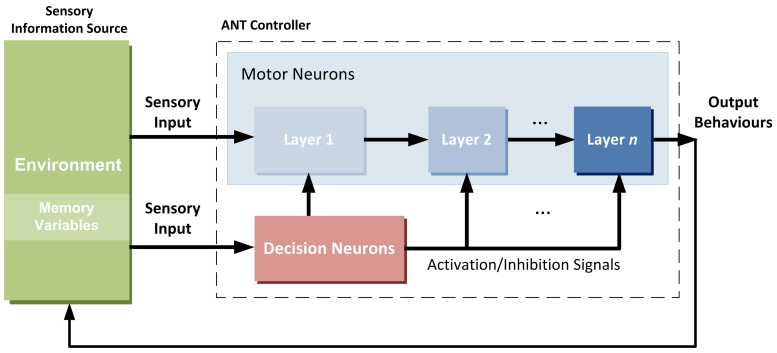


Figure 5.13: Schematic of information flow between an ANT controller and its environment.

won't be expected to be as successful in interpreting and following the signs (for the sign following task). Instead these randomly initialized controllers are bound to get stuck or end up with a shorter lifespan (when encountering hidden mines buried outside the pathway for the sign following task). With these controllers being less probable in traversing further within the training environment, they understandably encounter a smaller repertoire of sensory input combinations. ANT without any restrictions to growth in the development program shows increased neuronal activity and improved sensory input representation than without. Increased number of active, parallel neurons also increases spatial crosstalk and thus has negative consequences. Therefore these trade-offs appear to balance out and show no improved performance either way.

Figure 5.12 shows mutual information among individuals within the ANT population for the sign-following task. The trend shows a steady increase in mutual information of phenotypic activity through the evolutionary process. As individual controllers converge toward a solution, it would be expected that successful solutions dominate the population. This would in turn show increased mutual information between variants of the dominant individual. However, what we do not expect to see is the mutual information increasing without bound, since this has negative consequences, namely loss of genetic diversity. Loss of genetic diversity as with biological analogues could make an entire population susceptible to genetic defects triggered through a series of deleterious mutations. A steady mutation rate also ensures that the individuals within the population maintain differences.

5.5 Summary

In this chapter we have looked at nitric oxide as a potential candidate for performing neural regulation in the nervous system. High fidelity simulations of nitric oxide diffusion show that it is both physically and biologically feasible 'wireless' communication mechanism under known constraints such as typical inter-neuron dimensions and operational frequency. It is also argued that regulatory functionality (using chemical diffusion) can help reduce energy consumption in the nervous systems since not all the neurons have to be active all the time. Instead, the regulatory system can better coordinate the activity of specialized neurons to a task at hand. In addition, the relatively slow diffusion process can hypothetically be used as a 'synchronization mechanism' for neurons operating at different frequencies. Analysis of ANT controller

solutions shows that the neurons are sparsely activated. A steady mutation to the ANT growth program results in increased degeneracy indicating a trend toward population coding. Furthermore, consistent with selectionist theory, the total fraction of active neurons exponentially decreases in a 'regressive' trend (despite the linear growth rate in the number of genes added) as ANT controllers converge to a solution.

Information-theoretic analysis of ANT solutions was performed based on the perspective that a genome can be described by its *actions*. The analysis validated the reasoning that an ANT controller acts as *information filters*. Individual neurons have the ability to perform generalization, which from an information theoretic point of view is *filtering*. It is shown that through a process of evolution, the controller handles a bigger repertoire of sensory input combinations. This process seems to be increased particularly with regards to the information content of neuron activity . A mutual information measure developed based on the earlier premise, that a genome can be described by its actions applied to ANT controller populations shows an increase mutual information content (among a population) followed by a steady-state condition as evolution progressed. The steady state condition is indicative of how the controller population sustains diversity (due to a constant mutation rate) as it is converging toward a solution.

The theory of evolution by cumulative natural selection is the only theory we know of that is in principle capable of explaining the existence of organized complexity.
—Richard Dawkins

Chapter 6

NEURAL HETEROGENEITY AND REGULATION

6.1 Introduction

It has been theorized that exploratory selectionist mechanisms (the process by which selection of parallelized selection of functional outcomes) facilitates *evolutionary adaptability* [87]. Evolutionary adaptability is the ability for genes to be heritable, facilitate formation of selectable phenotypes, are less susceptible to lethal mutations and produce novel traits with fewer mutations [87]. Extensive evidence of selectionist processes has been found in the immune system and this had spurred interest into how these processes might be at work within the brain [43, 87].

A selectionist regulatory system was described in Chapter 3, in which decision neurons emit chemical signals used to activate or inhibit neighboring motor neurons. The selectionist system worked at inter-cellular levels. In this chapter, we extend the selectionist regulatory framework described earlier to the gene/molecular level. By better modeling gene regulatory interactions we hope to explore the notion of population variability and diversity (heterogeneity) in neuronal ensembles. Population variability from the Darwinian viewpoint is at its essence [43]. Increased variability enables the system to stochastically explore changes that are biased through the selection process and is theorized to be advantageous, helping to improve evolutionary adaptability [87]. This is in contrast to a typical mechanical or control system where

as Edelman points out such variability could be catastrophic [43]. The experiment work in this chapter is intended to test this hypothesis.

It should be noted biological evidence hardly points to the notion of a 'typical' homogeneous feedforward binary neurons of the McCulloch-Pitts type. Neurons are, in fact, complex heterogeneous multistate analog systems with memory. A better model of the regulatory interaction at the gene/molecular level with neurons can account for the range of heterogeneous neuronal behaviours observed in biology and not require the use of non-extensible neuron activation functions.

In this chapter, we demonstrate the advantages of these exploratory selection/regulation mechanisms based on a coarse-coding scheme, inspired by Albus [2, 3] and Hinton [72]. We show evidence of emergent task decomposition and specialization occurring both at the cellular and gene/protein level within an artificial neural tissue (ANT) framework [154]. The model exhibits cellular and tissue extensibility through ontogeny, resulting in the emergence of neural heterogeneity, use of memory and multistate functionality within the ANT framework. Within each neuron, multiple networks of protein compete and cooperate for representation through a coarse-coding framework for protein binding sites. We choose to use coarse coding as it is a moderately distributed coding scheme that allows for pooling and redundancy thus helping to render the system robust in the face of noisy variables.

This model, with no explicit supervision and limited task-specific assumptions, produces solutions to a variant of the sign-following task. Fixed-topology homogeneous artificial neural networks (Section 3.5.4) are found to be intractable for this task. In addition, fixed-topology networks that lack regulatory functionality perform poorly in complex tasks with limited supervision owing to the *bootstrapping problem*, which causes premature stagnation of an evolutionary run [116].

6.2 Artificial Neural Tissue and Coarse-Coding Cell Model

The ANT architecture consists of a developmental program, encoded in the genome that constructs a three-dimensional neural tissue and associated regulatory functionality. The tissue consists of two types of neural units, decision neurons and motor-control neurons. Regulation is performed by decision neurons, which dynamically excite or inhibit motor-control neurons within the tissue based on a coarse-coding framework (Figure 6.1).

The ANT model presented in Chapter 3 pertains to the regulation of population of neurons, with a preset neuron activation function model. The coarse-coding cell model presented here allows for structural heterogeneity among neurons by facilitating both competition and cooperation of multiple representations within neurons. This functionality allows for multistate functionality and short-term memory functionality (in contrast to use of recurrent connections). In contrast, within the standard ANT model competition and cooperations occurs due to interactions among neuronal populations.

Our neuron model permits a number n_c of messenger-channel protein networks (Figure 6.2 left). Each

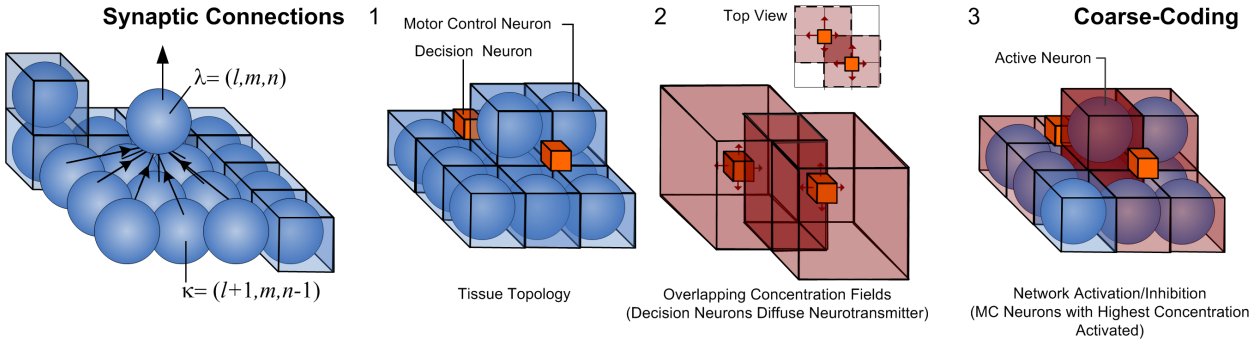


Figure 6.1: Synaptic connections between motor-control (MC) neurons and operation of neurotransmitter field.

protein network receives the same inputs $\mathbf{x}$, an $n \times 1$ real-valued column, which represents either sensory data or inputs from other neurons. These inputs are fed through m_j 'ion channels' that transform 'electrical signals' into various types of ion, the concentration of which are collectively denoted $\mathbf{y}_j$, an $m_j \times 1$ real-valued column. The concentrations are given by

$$\mathbf{y}_j = \mathbf{W}_j \mathbf{x}$$

where $\mathbf{W}_j$ is an $m_j \times n$ real-valued weight matrix associated with the jth protein network.

Each protein network produces an 'activation protein,' whose concentration c_j is determined by a linear combination of p_j basis functions $\psi_{ik}, i = 1 \cdots m_j, k = 1 \cdots p_j$, dependent on the ion concentrations $\mathbf{y}_j$. The ith basis function, in fact, depends only on the ith ion concentration y_{ij}:

$$\psi_{ik}(y_{ij}) = \begin{cases} 1, & \text{if } \tau_{1,k} \leq y_{ij} \leq \tau_{2,k} \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

The boundary parameters, τ_1 and τ_2 for each ψ are evolved. The concentrations c_j that determine the output of the neuron are simply $c_j = \sum_{i=1}^{m_j} \sum_{k=1}^{n_{b,j}} \psi_{ik}(y_{ij})$. (Note that c_j are integers.) This structure is reminiscent of and was inspired by the coarse-coding scheme of Albus [2, 3] and we accordingly refer to it as coarse-coding regulation. The basis functions here square-hat functions (Figure 6.2 right) in one dimension although the kind of tiled functions in Albus's Cerebellar Model Arithmetic Computer can also be used.

Binary-State Neuron. Let us first consider a binary-state neuron, i.e., one where the output is $s(t)$, where t represents the discrete time-step and is either 0 or 1. This output is given by

$$s(t) = \begin{cases} a_j, & \text{if } c_j \text{ is a unique maximum} \\ \phi s(t-1), & \text{otherwise} \end{cases} \tag{6.2}$$

The messenger-channel protein networks compete to determine the neuron's output. If $c_j = \max\{c_1, c_2 \ldots c_{n_c}\}$ and is uniquely determined, i.e., no two networks produce the same maximum concentration, then the output is taken as $a_j \in \{0, 1\}$ (genetically evolved). Otherwise, the output takes the value $\phi s(t-1)$, where

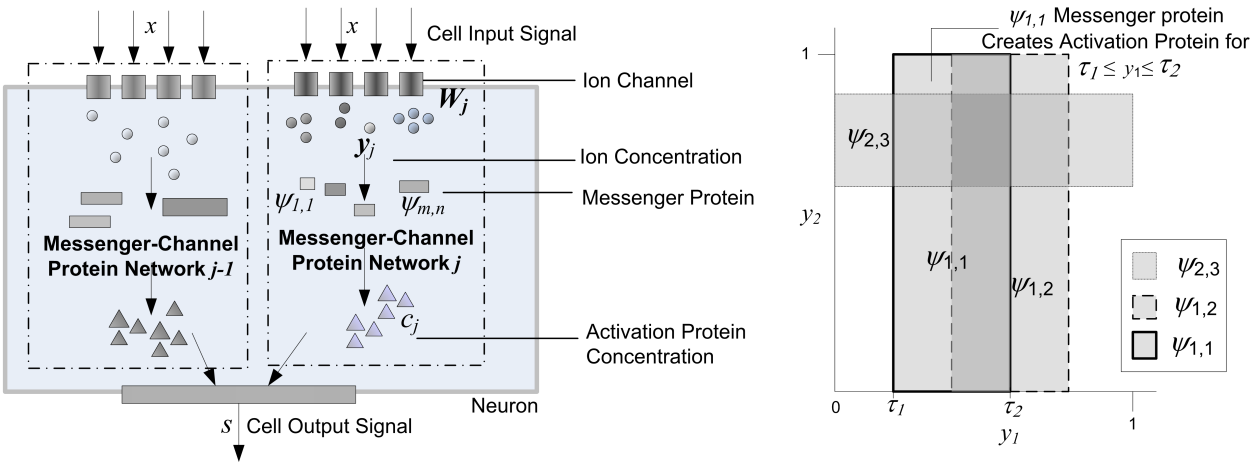


Figure 6.2: (Left) Schematic of competing messenger-channel protein networks. (Right) Coarse-coding interactions between messenger-protein for $m_j = 2$.

$\phi \in \{0, 1\}$. When $\phi = 1$, the output from the previous time-step is maintained. Thus $s(t)$ is intended to model the 'spike' status of the neuron.

Multistate Neuron. Spiking neurons superimpose their spikes on a background signal. We model this aspect of the neuron by allowing for a multiple-state output, $\mathbf{s} = [s_1 \;\; s_2]$ where $s_1 \in \{0, 1\}$ is associated with the spiking signal and s_2 with the background signal. The output s_1 is given again by (6.2). We offer two models for the computation of s_2, a feedforward model and a memory model. In the former, s_2 is given by (6.2) with a_j replaced by $b_j \in \{0, 1/q_b, ..., 1\}$ which is graduated in q_b (an integer greater than one) uniform steps between 0 and 1. In the memory model, $b_j \in \{0, \max\{0, s_2(t-1)-1/q_b\}, \min\{s_2(t-1)+1/q_b, 1\}, 1\}$ allowing for storage, reset, gating and increment/decrement functionality. This multistate model of the neuron is an attempt to better incorporate biological observations of neuron action potential through bottom-up modeling of protein interactions.

Evolution and Development Details of the neuronal development process for the tissue remains identical to previous versions of ANT and can be found in [154, 155]. Unlike previous versions of ANT that used neurons with a modular activation function using two thresholds [152], the coarse-coding cell model allows for a developmental activation function. The activation function is extensible in that additional components can be included with each neuron based on mutations to the development process. Cell and protein genes have a binary 'activation' parameter, used either to express or repress gene contents. (The genome structure is shown in Figure 6.3) Each channel protein references a cell address. Messenger and action proteins in turn reference a channel protein. Since these genes are modular, it is possible for a messenger-channel protein network to be incomplete and thus lacking channel proteins.

Mutations in the genome can perturb existing genetic parameters or addition of new (cell, messenger,

Channel Protein Gene

Specifier	Reference Address	Weights				Gene Activate	Reference Pointer	Neuron Address
D	A	w_1	w_2	...	w_n	G	P	N
Integer [0,6]	Integer	Real [0,1]				Binary	Integer	Integer

Messenger Protein Gene

Specifier	Reference Address	Boundary Params.		Gene Activate	Channel Protein Reference Pointer
D	A	τ_1	τ_2	G	P
Integer [0,6]	Integer	Real [0,1]		Binary	Integer

Action Protein

Specifier	Reference Address	Output State		Gene Activate	Channel Protein Reference Pointer	Neuron Address
D	A	a	b	G	P	N
Integer [0,6]	Integer	Allowable States		Binary	Integer	Integer

Motor Control Neuron Gene

Specifier	Reference Address	Position			Memory		Multi-state	Gene Activate	Cell Death	Replication Prob.	Output Behaviour	Reference Pointer
D	A	x	y	z	ϕ	b	q_b	G	C	R	k	P
Integer [0,6]	Integer	Integers			Binary		Integer	Binary	Binary	Real [0,1]	Integer [0,b]	Integer

Figure 6.3: Genome of messenger-protein network components and a typical motor control neuron.

channel or action) genes caused by random gene transcription errors with a probability of p_{te}. Thus a new cell-protein gene as a result of a transcription error is a copy of an existing cell-protein gene with perturbations starting from a point chosen from a uniform distribution along the gene's length and with the gene activation parameter toggled off by default.

6.3 Related Work

Artificial developmental systems mimic ontogenic processes from biology and have been successfully used, with variable-length genomes, to 'grow' topologies and heterogeneous functionality without explicit intervention. ANT is a morphogenetic system with a directly encoded genome and uses gene regulatory systems (GRNs) for development. The coarse-coding cell model presented here is an extension of the ANT framework, in that extends the capability of representing multiple competing and cooperating representations to within each neuron. The model is also extensible in that components within a neuron can be added as result of future mutations.

Artificial embryogenic systems (L-systems [143] and cellular-encoding systems [64]) use indirect encoding schemes that involve recursive rewriting of the genotype to produce a phenotype. However, it has been argued that indirect encoding schemes introduce a deceptive fitness landscape and result in poor performance for smaller search spaces owing to overhead [134].

Examples of artificial morphogenetic systems include the work of Eggenberger [46] and of Gomez and Eggenberger [45], the latter using 'ligand-receptor interactions' to perform cell aggregation. A morphogenetic system was also used on POEtic by Roggen *et al.* [133]. Developmental tissue models such as NORGEV by Astor and Adami [1] are also morphogenetic and facilitate cellular heterogeneity. Cell replication and synaptic connections are formed through a gene-regulatory network (GRN) based developmental and learning system using a genetic-programming-type command set. However, in our ANT model, regu-

lation continues after development at the gene/protein and cellular levels. Neuroregulatory functionality is performed through coordinated release of diffusive neurochemicals resulting in superpositioning of chemical concentration fields (Figure 6.1). Other models such as GasNet allow for volume signaling between neurons using neurochemicals but lack explicit regulatory functionality [79].

In the coarse-coding cell model presented here, a selection/regulation process is also at work within protein networks resident in each neuron, thus facilitating heterogeneity and open-ended growth in complexity of cells. The use of cellular heterogeneity may be more biologically plausible but more complex, multistate cells do not necessarily present advantages over simpler binary-state cells. However, the use of multistate feedforward and memory neurons may be beneficial.

The need for specialized memory neurons arises from the 'error decay' problem evident with standard recurrent connections for learning processes. A stored signal remains unprotected from spurious inputs decays or grows (without bound) making it difficult to recall a signal after many timesteps [63]. Long Short-Term Memory (LSTM) [74] overcomes this limitation but it is a predefined architecture consisting of a storage neuron, a reset gate, an input gate (to protect memory from spurious inputs) and an output gate. However, for the T-maze task (a simpler variant of the sign-following task), it was found that recurrent networks trained using Enforced Subpopulations (ESP) and Hierarchical ESP (H-ESP) outperformed a LSTM architecture [60]. LSTM also lacks biological plausibility and similar memory functions can be obtained without the use of predefined cell blocks using the coarse-coding cell model.

6.4 Sign-Following Task

The effectiveness of the coarse-coding cell model is demonstrated in simulation on two memory-dependent versions of the unlabeled sign-following task. The workspace is modeled as a two-dimensional grid environment with one holonomic robot (based on a Khepera^{TM}, equipped with a gripper and camera) occupying four grid squares. For these tasks, the controller must possess a number of capabilities including that to decipher signs relative to the robot's current frame of reference, to remember the current sign while looking for the next one, and to negotiate obstacles (see Figure 6.4 right). Each sign is color-coded and represents a waypoint (posted in a fixed frame of reference) that gives direction in one of four cardinal points to the next waypoint leading ultimately to the goal location.

Mines (undetectable by the robot) are randomly laid throughout the floor except along the pathway. Once a robot encounters a mine, it remains disabled for the remainder of its lifetime. The sensory input map is shown in Table 6.2 (see also 6.4 left). The task has to be accomplished using a discrete set of basis behaviors specified in Table 6.2. These behaviors are activated based on controller output and all occur within a single time-step. The robot is initially positioned next to the first sign, but the initial heading is randomly set to one of the four cardinal directions. Since the robot can only detect signs placed in front, it needs to go into a 'sign searching' mode and perform a sequence of 'turn left' or 'turn right' behaviors to

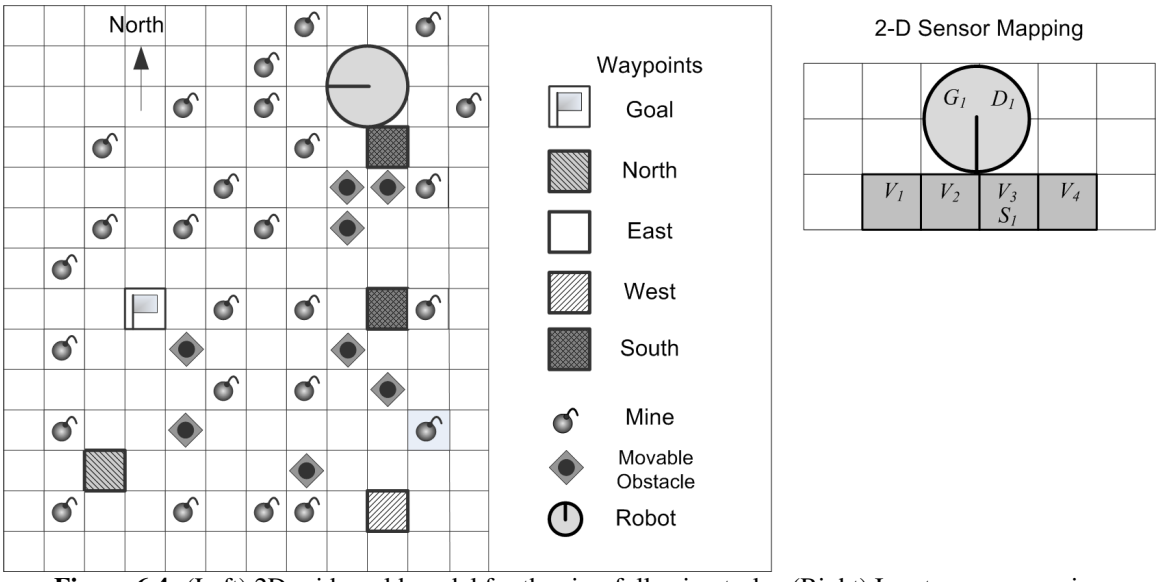


Figure 6.4: (Left) 2D grid world model for the sign-following tasks. (Right) Input sensor mapping.

Table 6.1: Sign-following Tasks: Sensor Input

Sensor Variables	Function	Description
$V_1 \ldots V_4$	Object detection	Robot, block, no obstacle
G_1	Gripper status	Holding block, no block
S_1	Sign detection	Red, blue, orange, pink, green
D_1	Heading	North, east, west, south

detect the first sign. Once the first sign is detected, the robot then needs to transition to a 'sign following' mode, requiring one bit of memory.

Deciphering signs relative to the robot's current frame of reference makes these tasks particularly difficult given a fitness function that measures success in terms of reaching the goal location. The two versions of the task considered here are (1) where the controller has access to a compass sensor at each time-step and (2) where compass sensor readings are penalized or restricted. We shall refer to the former variant as *compass-enabled* and the latter as *compass-restricted*. Even the simpler compass-enabled version is found to be unsolvable for predetermined fixed-network topologies that lack regulation (see Results and Discussion).

In the compass-restricted version, the controller must perform sign following knowing just its initial heading thus requiring the controller to predict and keep track of the robot heading (ego-orientation) in addition to accomplishing the other subtasks described earlier. Keeping track of long term dependencies is acknowledged to be difficult with recurrent connections [60] making the sign-following task a good benchmark for multistate architectures. The robot in this case has access to one additional behavior, the 'get hint' behavior, which interrogates the compass for the 'true' heading. However, the fitness function incrementally

Table 6.2: Sign-following Task: Basis Behaviors

Order	Behavior	Description
1	Pick-Up/Put-Down	Pick up from V_2 or V_3 or put down obstacle on V_1 or V_4
2	Move forward	Move one square forward
3	Turn right	Turn 90° right
4	Turn left	Turn 90° left
5, 7, 9, 11	Bit set[1]	Set memory bit i to 1, $i = 1 \cdots 4$
6, 8, 10, 12	Bit clear[1]	Set memory bit i to 0, $i = 1 \cdots 4$
13	Get hint[2]	Get current heading (D_1)

[1] Behaviors disabled for recurrent and memory neuron architectures

[2] Behaviors disabled under certain conditions (see text)

penalizes and restricts the number of hints used. The fitness function for a given run is defined as

$$f_i = \begin{cases} \dfrac{1}{1 + \beta n_{\text{hint}}/16}, & \text{if goal is reached} \\ 0, & \text{otherwise} \end{cases} \tag{6.3}$$

The total fitness function is averaged over all runs. For the compass-enabled variant, $\beta = 0$ although the robot always knows the compass direction. So when the goal is achieved $f_i = 1$; otherwise, $f_i = 0$. For the compass-restricted one the reward for success is discounted according to the number n_{hint} of hints that have been used. However, for the first 5,000 generations, the robot is not penalized for using hints; hence $\beta = 0$. For the subsequent 10,000 generations, $\beta = 1$ but the hint can only be used in the first four time-steps. (This allows the robot to get the true direction reading as it starts out.) After 15,000 generations, hints are proscribed altogether. These parameters were found through experimentation to work well for this task.

The evolutionary performance of various control system architectures is compared for the two variants of the sign-following task (see Figure 6.7). The robot's world is a 20×20 grid with 80 uniformly distributed obstacles and 40 randomly distributed mines (except along the path to the goal). The fitness is averaged over 100 runs with different initial conditions, the elapsed time for each run being limited to 100 timesteps. Controllers that lack recurrent connections or memory neurons have access to four memory bits, which can be manipulated using the defined basis behaviors. Evolution assumes a population of 100 individuals in each generation and a tournament size of 24. The crossover probability $p_c = 0.7$, the mutation probability $p_m = 0.005$ and the transcription error rate $p_{te} = 0.005$.

6.4.1 LEGO® Hardware Experiments

The LEGO® Mindstorms™ NXT robot used for the hardware experiments is shown in Figure 6.5 and consists of a pair of servo drive motors in a differential drive configuration, with a third drive motor used to actuate the gripper [5]. The ANT controller is executed on a laptop that communicates with the LEGO®

[5] Basis behaviour implementation, sensor integration work and hardware experiments on the LEGO® NXT robotic platforms done with help from Stephen Chee and Yinan Wang. See acknowledgements.

robot using a wireless Bluetooth interface. All sensor input data fed into the ANT controllers are discretized according to Table 6.2. At the beginning of each timestep a sensor scan is performed and data is then fed into the ANT controller. During a sensor scan, the robot pans 120^o and determines the sensor state of the variables $V_1 \ldots V_4$ and S_1 using sonars and a pair of color sensors respectively. G_1, gripper status, is determined by keeping track of pickup/putdown behaviors. The initial heading set for H_1 is hardcoded and remains unchanged for the experiments where the controller is expected to keep track of the heading internally.

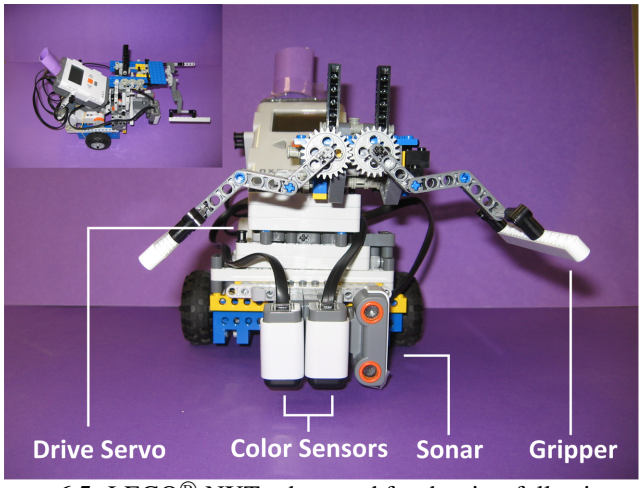


Figure 6.5: LEGO® NXT robot used for the sign-following task.

The basis behaviors implemented for the LEGO® NXT robot are shown in Table 6.2. They have been developed assuming the environment is a checkerboard pattern as shown in Figure 6.6. A robot is always centered at the edge of each grid square and a pair of color sensors are used to maintain alignment as its executing basis behaviors such as 'move forward', 'turn right' or 'turn left'. The color sensors are also used to ensure the robot travels one square forward when executing the 'move forward' by detecting change in color in the square ahead. Execution of 'pickup' and 'putdown' behaviors require turning on the gripper drive servo for a fixed length of time. The throttle values for the drive servos had been reduced to minimize the effects of slippage but this has also meant that a typical sign-following run lasts 30 minutes (Figure 6.6).

6.5 Results and Discussion

Figure 6.7 (left) shows the average (population best) fitness of the controllers evaluated in each generation for the compass-enabled variant of the sign-following task. For some comparison, a fixed-topology recurrent network with 9 hidden and 4 output neurons is also shown. (Although this is typical of the results obtained for such a network, we did not optimize performance in any way.) Fixed-topology networks tend to have more 'active' synaptic connections present (all neurons are active) and thus more spurious neurons need to be dealt with simultaneously. The ANT topology with the coarse-coding neuroregulatory mechanism disabled (using modular activation function) shows better performance than fixed topologies (including H-ESP [60]) but not sufficient to complete the task for all the initial conditions tested.

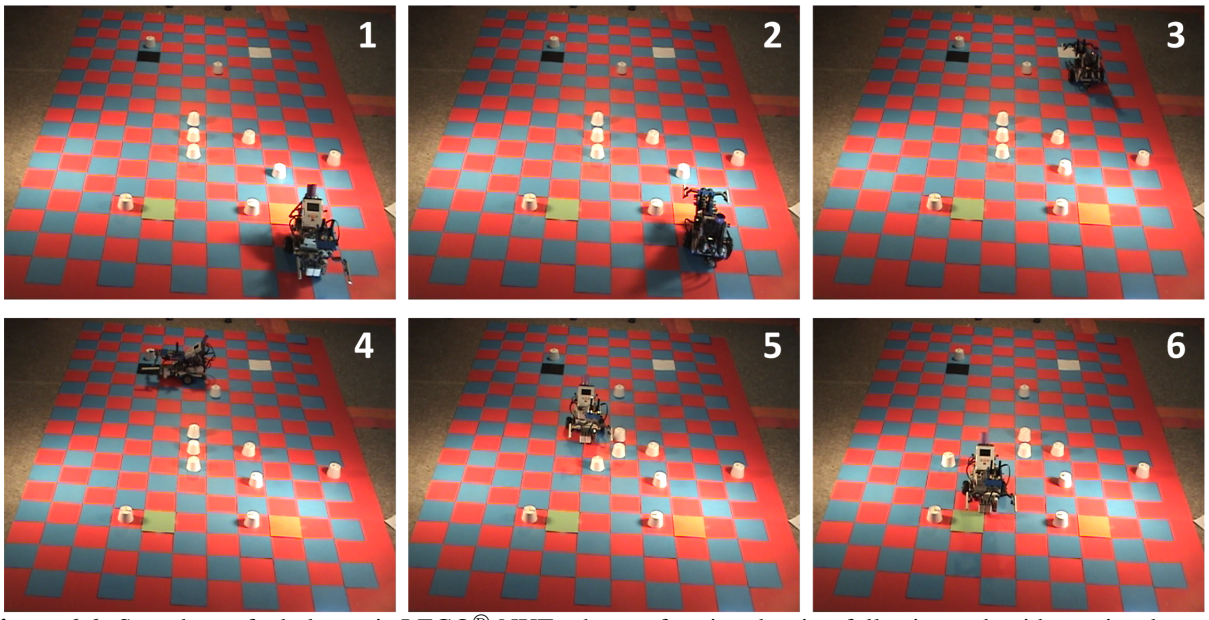


Figure 6.6: Snapshots of a holonomic LEGO® NXT robot performing the sign-following task with restricted access to the compass sensor using an ANT controller (multistate coarse-coding cell model, $\phi = 1$). Frame 1 shows the robot pointing North with access to the compass sensor during the first time step. The robots searches for the first sign by turning right until sensing the orange sign (coding for South) and moves forward. Once at the white sign (East), the robot turns left until reaching the black sign (North) and continues moving forward and picking up/putting down obstructing objects until it reaches the green sign.

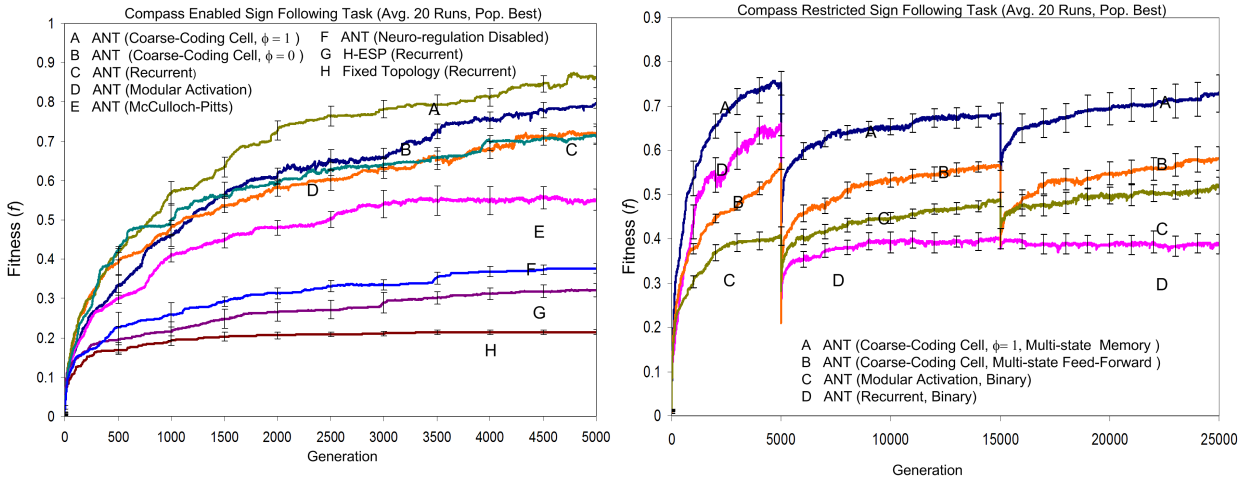


Figure 6.7: Comparison of performance for (left) compass-enabled task and (right) compass-restricted task.

In contrast, heterogeneous architectures, whether predefined (modular activation function) or developmental, outperform other architectures. It should be noted that these heterogeneous architecture provide an increased repertoire (variability) of available activation functions over a standard McCulloch-Pitts model. A larger repertoire (toolbox) of functions provides the system capacity to better match the necessary functionality with one that is available through selection.

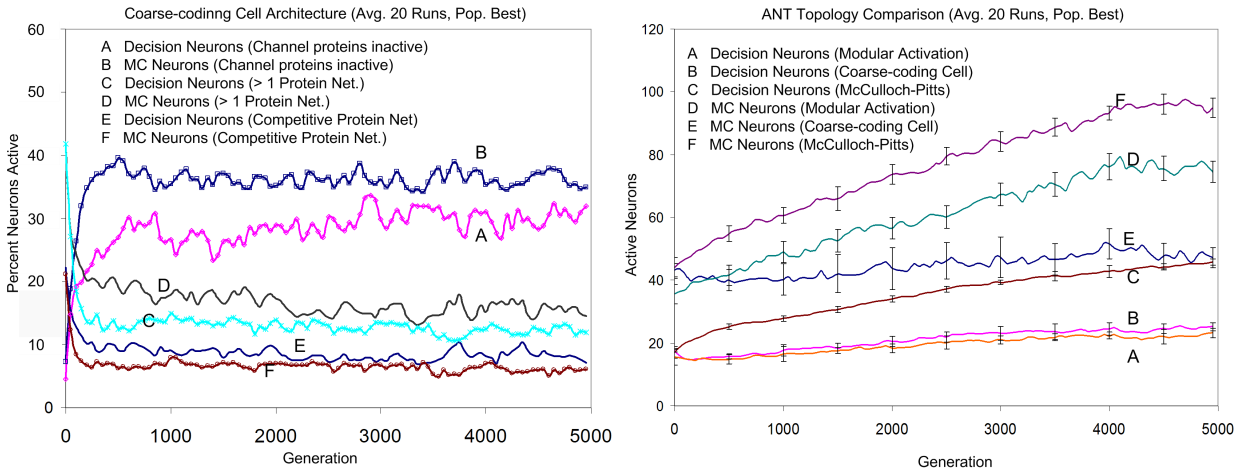


Figure 6.8: Comparison of (left) coarse-coding cell models and (right) ANT topologies.

In the coarse-coding cell model, selection is favoring increased complexity among decision neurons (Figure 6.8 left) but this is followed by a gradual simplification in neuron structure. This simplification is possible with all the channel proteins disabled, resulting in constant output (independent of sensory input).

While the modular activation function also allows for heterogeneity, it is a fixed-cell architecture that does not facilitate intracellular competition nor 'complexification.' Overall, the 'expressed' structure of the coarse-coding cell model is using fewer tuning parameters (both weights and thresholds) than with the modular activation function as it converges to a solution (Figure 6.8 right).

Figure 6.8 (right) also shows that many messenger-channels protein networks have redundant action proteins implying a 'weak' cooperative setup. A weak cooperative setup is advantageous, allowing for co-operative and competitive tendencies and can better facilitate a transition between the two. The added benefit of the heterogeneous coarse-coding cell model is that it also facilities memory functionality through gating. The evolutionary performance with gated-memory functionality ($\phi = 1$) shows a definite improvement over recurrent architectures for both tasks (Figure 6.7 left).

For the compass-restricted version of the task (Figure 6.7 right), only the multistate coarse-coding cell models can reach a fitness of 0.9 within 25,000 generations at least once. The binary-state neuron model with feedforward connections (and access to 4 memory bits) shows comparable results with a fitness of 0.8 at least once while the binary recurrent architecture performs poorly. Among the four architectures, the binary recurrent model (localized memory) lacks built-in competitive network mechanisms for memory representation. For the multistate model, these competitive mechanisms are the protein networks resident in each cell, while similar functionality is present within the feedforward architecture, where neuron ensembles compete for control of the 4 memory bits (see [155]).

These results also show *intra*cellular competition has an advantage over *inter*cellular competition for memory representation. The ANT topology with access to 4 memory bits is an instance where there is

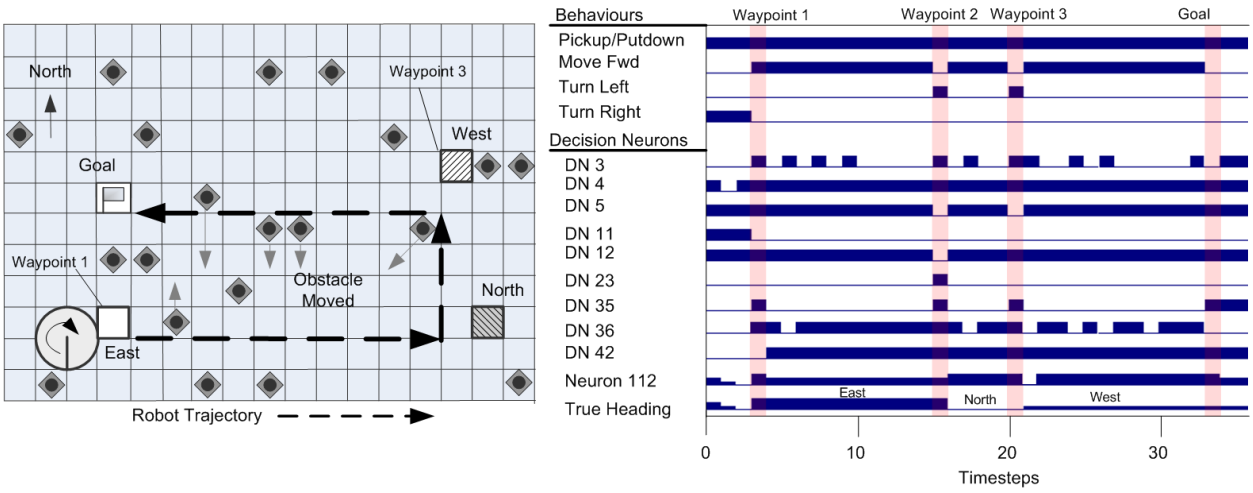


Figure 6.9: (Left) Typical ANT solution (fitness $f = 0.99$) using multistate neurons (memory model) for the compass-restricted sign-following task. (Right) Output behavior and function of memory neurons.

intercellular competition, that competition among motor neurons to write to the memory bits. While within the intracellular model, the competition is internal to each neuron. Multiple local representation may exist and competition is over which representation is considered the 'functional representation.'

The multistate neuron model with memory functionality outperformed the multistate feedforward model (Figure 6.7 right). It should also be noted that the model is expected to be more favorable in helping to map expected compass sensory input due to the inclusion of increment/decrement and reset 'actions' (see Figure 6.9). The solutions shown in Figure 6.9 has also ported onto a holonomic LEGO® robot. See results of a typical run in Figure 6.6.

Comparison of decision neuron activity and the output behaviours (Figure 6.9) shows evidence of distributed representation of some of the behaviour functionality. For example decision neurons 5 and 12 show evidence of handling the 'turn left' behaviour, while decision neurons 11 and 42 keep track of transitions from 'sign searching' to 'sign following' modes. Motor neuron 112 along with decision neuron 24 are found to keep track of robot heading.

Motor neuron 112 interestingly maintains two forms of internal representation of the robot heading. One of these representations is a 1:1 mapping of the true heading and is maintained during the beginning of each trial until the robot detects the first sign. When the controller detects the first sign, it switches from sign searching to the sign following mode and incidently motor neuron 112 also switches representation of the robots heading. The internal representation of the heading after this transition is shown in Figure 6.10. The evidence shows that the motor neurons maintain a unique representation of each quadrant of the axes shown (i.e. s_2 switching to 0 and returning to 1 within the North-West quadrant) instead of maintaining an internal value representation of the cardinal directions (i.e. $s_2 = 1/3$ for North). For the South-East quadrant, neuron 112 doesn't change state, instead decision neuron 24 is triggered.

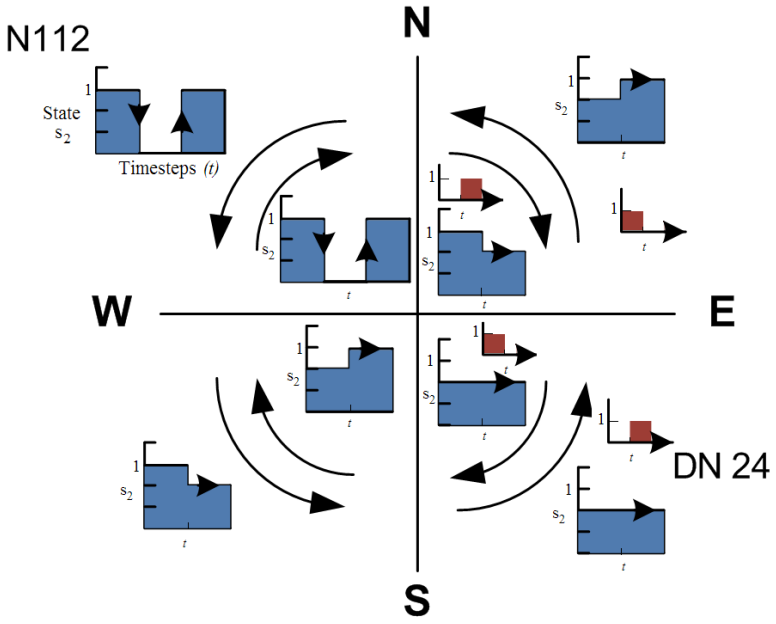


Figure 6.10: Internal representation of motor neuron 112 and decision neuron 24 mapping the robot heading.

6.6 Summary

The ANT architecture has been extended to include dynamic gene-regulation using a coarse-coding framework. The intention has been to see if increased heterogeneity in neuron ensembles produces positive effects on evolutionary performance as would be hypothesized under a selectionist framework. This developmental approach to defining neuron structure allows for evolution to mold neural topology, neuron structure and function. In this chapter, a biologically plausible means of gene regulation is presented, in which coarse coding interaction result in promotion/inhibition of competing representation within each neuron. This in contrast to standard neuron models, where the output is binary and represents the spike state. Neurons are known to be multistate analog systems. Part of the problem with a phenomenological model is that there exist abundance of variability in neural structure, hence each variant may need a separate model. Using a developmental model, the neural structure is generic and may represent any number of neuronal characteristics, including multistate representation provided there is evolutionary advantage for emergence of such a structure.

This extension to the ANT framework has been tested on two variants of the sign-following task, one with access to a compass sensor and other with restricted access to the compass. Both variants of the task require use of short-term (working) memory. Performance comparison for the sign-following task with access to compass sensor readings showed that a coarse-coding cell model of ANT, with $\phi = 1$ (the memory enabled) version outperforming the other control architectures tested. The coarse-coding cell model with $\phi = 1$ allow for neurons to have access to local memory. As opposed to a fixed number of memory bits, this architecture allows for a variable number of neurons with memory functionality being created. This architecture was found to be advantageous over a standard recurrent architecture and McCulloch-Pitts

neuron model with access to memory bits.

Furthermore, it is confirmed that ANT with neuroregulatory functionality turned off shows a steep drop in performance for the sign-following task. Overall the ANT architecture also outperformed H-ESP using recurrent neurons, the best-known architecture applied on the T-maze task. ANT architectures are also compared against the harder sign-following with restricted access to the compass sensor. Here the controllers learn not only to interpret the signs correctly and end up at the goal location, but they also learn to keep track of the heading without access to compass readings.

Evolutionary performance for this harder task shows that a multistate coarse-coding cell model shows improved evolutionary performance over a multistate feed forward and binary neuron models. Furthermore, the architecture manages to successfully transition from situations where it has access to compass information to one without by self-regulating access to the 'get hint' behavior. Successful solutions were validated on hardware, by applying the controllers onto a holonomic LEGO® robot.

The essence of life is statistical improbability on a colossal scale.
—Richard Dawkins

Chapter 7

SYNTHESIS

Central to the development of controllers for solving a particular task is the organization of these controllers. Organization entails how these controllers are both spatially and temporally structured, a relationship between how the components within a controller form a coherent whole. This work is primarily inspired by regulatory control evident in biological nervous system and is intended for application in robotics.

One must truly marvel at the capabilities of biological nervous systems that exhibit computational capabilities in vision, memory, decision making/control, learning/plasticity that remains unmatched by even the most up to data computer hardware today. Regulatory control is an organizational mechanism, used to arbitrate/coordinate the components, allocate sufficient resources and is redundant to component failures. The individual components within these biological systems are noisy, unreliable, have limited computational/control capacity. Yet the system as a whole is robust to the limitations of its components. But how is this accomplished? In this regard, theory of evolution remains a formidable means of explaining how such regulatory systems have arisen [131]. As Dawkins [35] points out:

> The theory of evolution by cumulative natural selection is the only theory we know of that is in principle capable of explaining the existence of organized complexity.

Regulatory systems unlike humanity's creations lack an explicit designer, and is a self-organizing process that has arguably driven it toward greater organizational complexity [35].

A number of theorists argue that biological processes within an organism mimic evolutionary (selectionist) mechanisms. Chief among these is the discovery that the immune system uses exploratory selection

and variability to develop antibodies [43]. It has also been vigorously argued that the brain uses selectionist mechanisms in organization and control [43, 25]. In this work we develop a theoretical framework showing how selectionist mechanisms can exploit superpositioning of chemical concentration fields to perform regulation.

There lacks evidence of explicit distinction between data representation and control (program) representation with these systems. Hence it's reasonable to believe that mechanism of data encoding within nervous system can give us vital clues about how the nervous system is organized. It is expected that the underlying factors that have given rise to a particular data encoding and control schemes should be one and the same. Both data representation and control are performed by the same components, namely interaction of intercellular signals and it is known that these individual components are noisy, unreliable and have limited computational capacity.

It is shown that the superpositioning of chemical concentration fields can be modeled as a coarse coding scheme proposed by Albus [2] and Hinton [72]. This phenomenon of superpositioning of chemical concentration fields to perform neuroregulation is encapsulated within the artificial neural tissue (ANT) architecture. ANT with its ability to exploit neuroregulation shows improved *evolvability* [87] over controllers that lack this capability for several robotic and control tasks including double pole balancing, tile formation, phototaxis, sign-following and resource gathering. Of particular interest is the sign-following task, where controllers that lack neuroregulation are shown unable to find solutions to the task.

Determining a suitable neural-network topology for a task at hand has been considered a difficult problem in the field. Some have gone as far as considering development of neural network controllers that concurrently search for a suitable topology, tune neuronal weights as the 'holy grail' in neural network controller development [129]. In Chapter 2, several fixed topology ensemble approaches have also been presented including the Binary Relative Lookup (BRL) and Binary Decision Tree (BDT) architecture. The limitation with fixed-topology ensemble networks is that it requires supervisor intervention in determining the network topology and number of expert networks *a priori*. This has also limited the applicability of neural network controllers to new and less well understood task domains.

Artificial Neural Tissues (ANT) can overcome the need for supervisor intervention in determining the network topologies through use of artificial regulatory systems. Within ANT, the regulatory systems dynamically activate and inhibit neuronal groups by modeling a biologically plausible coarse coding arbitration scheme. In addition, ANT can overcome tractability concerns affecting other variable length topologies that lack neural regulatory systems. Variable-length topologies that lack neural regulatory mechanisms are grown incrementally starting from a single cell, since the size of the network architecture may inadvertently make training difficult [144].

It is also shown that an advantage of the Artificial Neural Tissue (ANT) framework is its ability to facilitate *phenotypic neutrality*. We define phenotypic neutrality as the ability to set components into a neutral configuration (where actions imparted by these components result in no change to the rest of the system)

post development. This is in contrast to *gene neutrality*, where certain genes are not expressed during an individual's lifetime but a potential series of mutations can activate this gene passed onto an individual's progeny. Using a bottom-up probabilistic framework developed in Chapter 2, it is shown that increasing the probability of having phenotypically neutral neurons in effect increases the probability of finding solution networks. This is assuming the probability of feature neurons remain constant (Section 3.3.6).

Based on this evidence, it is argued that ANT can increase phenotypical neutrality sufficiently to increase the probability of finding a solution network for the sign-following task. This in turn facilitates the artificial evolutionary process in searching for a solution. The assumption that there exists this critical threshold is reasonable and comes from analysis of the tile formation task. The mapping of success rate (binary) value for each initial condition to probability of finding solution networks, χ is found to be a sigmoid function that is equivalent to a smooth threshold (Section 2.9.5). In addition, without sufficient phenotypic neutrality, the evolutionary performance is expected to remain stagnant as confirmed from comparison with other control architectures and with neuroregulation turned off for ANT (Section 3.6.4).

The coarse coding regulatory framework presented allows for phenotypical neutrality, particularly since the chemicals diffused from individual decision neurons can be used to broadcast to many neighbouring neurons. Superposition of these chemical concentration fields (coordination of multiple sources) can then be used to activate and inhibit specific neuronal groups. There exist comparable biological evidence of sparsely scattered nitric oxide synthesizing neurons (equivalent to decision neurons) with a plexus (meshwork) of chemical emitter covering vast volumes of the neuropil (volume between neurons) [125].

While synaptic connections can also do the same function, where a signal from a single neuron is broadcasted to a neuron ensemble, there remains some challenges that need to be addressed with this alternate possibility. For one, each receiver needs to correctly distinguish and prioritize such regulatory signals from the clutter of other synaptic data signals. This again brings us back to the issue of spatial crosstalk, where the influx of synaptic signals from multiple sources makes it difficult for adaptive systems to perform 'exploration' and to pick out signals of interest from the other potential noisy sources.

Use of chemical signaling serves as an alternate channel of communication and it is expected that different mechanisms are used to interpret each type of signal. Using both an analytical and simulation framework, it is shown that use of chemical signaling to perform neuroregulation provides ANT controllers substantial benefits in terms of evolvability. This is shown to be unmatched with synaptic connection alone (via simulation). Furthermore, the number of neurons active within the ANT architecture is found to decrease steadily. A best fit of this trend shows that the controllers are tending towards less than 20% neurons simultaneously active for a frequency of less than or equal to 3.33 Hz based on the calculations presented in Section 5.4. This matches well with neural activity estimates/bounds based on energy consumption by Attwell and Laughlin [5] that show less than or equal to 15% neurons are simultaneously active for less than 4 Hz within rodent brains.

In chapter 5, the coarse coding framework is extended to describing gene-protein interactions. This

coarse coding model of gene regulation results in the emergence of neural heterogeneity and ability for a single neuron to represent multiple states. The model also allows for concurrent evolution of neuronal activation functions, structure, synaptic connection and chemical interaction behaviour. In addition, this developmental model of neuron structure allows for competitive representation of internal structure to emerge.

Some of these additional functions aid in single neurons having memory gating functionality equivalent to the LSTM memory model [74], but without the need to rely on a predefined structure. Both LSTM and coarse coding cell model address the issue of temporal crosstalk inherent within standard recurrent network architectures. Increased variability in neuron structures and neuronal groups is expected to advantageous from a selectionist viewpoint and is confirmed from the evolutionary simulations (Section 6.5). Furthermore, this approach is also shown to discover more robust, memory dependent solutions for the compass restricted sign-following task.

Returning to the analytical framework presented in Chapter 2, there also exists several alternatives to using neuroregulation to increase probability of finding solution networks (Section 3.3.6). One would be to introduce more supervision. By introducing more supervision, one cannot guarantee increased the probability of finding a solution network. This is expected since increased supervision, with the addition of incorrect assumption may make finding a solution network less probable. In contrast, increasing phenotypic neutrality without increasing supervision can increase the probability of finding solution networks. Considering biological organisms often do not require such external intervention in the form supervision and can adapt through self-organization, then the strategies to increase phenotypic neutrality is important. In this regard, the analysis justifies the need for regulatory systems that can increase phenotypic neutrality.

The mapping system within ANT addresses a strategy in reducing the effects of the competing conventions problem. An increase in the ratio of neutral genes to feature genes probabilistically reduces the worst-case chance of incompatible sets of genes being intermixed during crossover of two parents. Nevertheless it should be noted that addressing the competing conventions problem alone is not sufficient to finding solutions for the sign-following task. It should also be noted that the effects of the competing conventions problem increases when there is greater diversity among an evolving population.

Information-theoretic analysis of the evolving ANT population shows increased average mutual information, though this trend stagnates after a few hundred generations for the sign following task. A further increase in mutual information is not expected since a steady mutation rate helps to maintain population diversity. These two trends of a population converging toward a solution (thus reducing diversity) combined with a steady mutation rate (increasing diversity) then is expected to result in a steady-state equilibrium.

It is of further interest to note that neuronal activity with the ANT controllers lean toward a sparse distribution, although the organizational mechanisms rely on a coarse coding selection process. This evidence agrees with biological observation of sparse-coded activity of neurons in the visual cortex and other brain regions [15, 132, 119]. However the evolutionary process also appears to encourage formation of degenerate network components thus showing evidence of a population coding scheme that has also been observed

in biological nervous systems. Degeneracy is a description of variable components having very similar functions [43]. Redundancy is a special case of degeneracy, with all components being identical. While a sparse coding scheme has been shown to be energy efficient [95], a population coding scheme allows for degeneracy.

7.0.1 Applications in Robotics

The evolutionary approach used to 'breed' ANT controllers for a given task involves use of low fidelity, grid world simulation environments. Evolution in a simulation environment is preferred since this involves less risk of damage incurred during training, less time required for training, besides less intervention required in replenishing power for these robotic systems. For application in space robotics, it remains a considerable challenge to have 'real world' conditions that are representative of the low-gravity surface environment of other planets and moons. Analogue sites maybe used for this purpose, but often these sites not readily accessible due to their remote location. As a result, practically speaking, software simulation environments fill this void.

Once a palette of sensory input and output behaviours are defined, it should be noted that the ANT controllers cannot perceive or distinguish between this simulacrum and reality. For that matter, it cannot perceive the additional complexity evident in a 'real' world environment. To ensure the controllers perform as expected in a 'real' world environment (embodied as robotic hardware), one has to only ensure the simulation environment is statistically representative of the 'real' world conditions from the robot controller's perspective.

ANT with the use of a global fitness function shows the possibility of training controllers with limited supervision to perform self-organized task decomposition. This approach involves specifying a global fitness function that improves system performance without explicitly biasing for a particular task decomposition strategy. This is particularly advantageous for use with multirobot controllers; where it is often easier to define the global goals over required local coordination behaviors require task specific information. Furthermore, an artificial evolutionary process has been used to concurrently evolve the multirobot controllers and necessary optimal configurations (i.e., number of robots) for an excavation task. In addition, such an approach facilitates discovery of novel behaviors, that otherwise maybe overlooked by a human designer including use of 'bucket brigade' behaviours for the multirobot resource gathering task, 'error correction' for the tiling formation task and keeping track of heading for the compass restricted sign following task.

This approach to goal specification requires limited supervision to perform task decomposition and has applicability in space robotics. Both bandwidth limitations and latency reduce the ability to do continuous teleoperation during robotic missions on the Moon, Mars and beyond. Hence, there is a real need for autonomous control of these robotic platforms with limited ground control intervention. Instead of communicating a detailed set of task instructions as has been done with rovers for the Mars Pathfinder and at times for the Mars Exploration Rovers, one can simply provide a global fitness function. Using a global fitness

function, ANT controllers can then evolve the necessary strategies to complete the task in-situ. This vastly reduces the instruction set that need be communicated to these robotic platforms at the price of increased energy consumption (due the evolutionary process).

To complement the global fitness function-based approach to controller development, a bottom-up analytical approach is introduced to analyze and predict neural network controller performance. Once the probability of the componental feature neurons necessary to make a solution network is known, one may then determine the optimum topology that maximizes probability of finding a solution network through analytical/numerical means. This is akin to knowing the necessary subtasks to determine probability of finding a solution for the overall task. Although the topology is a two-layer network, with one layer of hidden neurons and one output layer, such a network is sufficient to represent a universal approximator [32].

For some of the tasks presented in Chapter 3 and 4, both the network inputs and outputs are discrete. The outputs for these controllers represent basis behaviours and controllers are decentralized. The use of a predefined palette of basis behaviours is accepted as an *ad hoc* procedure. Alternately, such basis behaviours could be obtained using machine learning techniques such as artificial evolution in stages. Nevertheless, the intent with some of these experiments is to use a generic palette of basis behaviours that are not task-specific thus reducing the need for basis behaviour related design decisions by the experimenter.

Use of a generic set of basis behaviours is also an alternate strategy in dealing with the robot transfer problem. The robot transfer problem refers to tweaking or other forms of manipulation necessary to handle controllers trained on one robotic platform to be usable on another. In this regard these generic behaviours such as 'move forward', 'turn right', 'turn left' can be baselined (show matching characteristics) to different robotic platforms, independent of the controller or task at hand. Hardware evolution of controllers poses a number of challenges in this regard to the robot transfer problem. The effects of wear and tear on particular robot components will make each robot behave differently, (i.e. uneven turning servos, uneven stopping characteristic), therefore some means of calibration/human intervention is required to ensure a controller evolved on one robot can be transferred onto another with matching components. More human intervention is involved in reshaping the controller if the robotic platform and available sensors are of different specifications. With the approach presented here, we can apply ANT controllers without any tweaking on any robotic platform, provided the basis behaviours and sensory input match those of the training environment.

From a multirobot point of view, there exists a trade off between use of hand-coded basis behaviors (microscopic behaviors) versus use of evolutionary search techniques for multirobot coordination and control (resultant macroscopic behaviours). Where design and implementation of basis behaviours is more involved than determining the coordination behaviours, this approach may have limited practical value. Our premise, based on experience with multirobot controllers, is that designing the basis behaviour requires much less effort than determining the un intuitive coordination schemes that gives rise to emergent group coordination behaviours. To further emphasize this point, analysis of the evolved solutions indicates they are not organized according to readily 'recognizable' distal behaviors but as proximal behaviors (organized according

to proximity in sensor space) [115].

One of the more promising features of the evolved multirobot controllers solution is the scalability of these solutions to a larger 'world' size. In contrast, scalability maybe limited with a hand-coded solution, since other robots tend to be treated as obstacles to be avoided or use of heuristics for interaction requiring ad hoc assumptions based on limited knowledge of the task domain. In order to explore the flexibility of the control framework, ANT has also been applied to real-valued inputs/outputs as in the double pole balancing task.

We have only begun to understand the capabilities of biological nervous systems. The ability to 'mine' its capabilities holds much promise with regards to computation, organization and control. A reductionist philosophy towards simulating these regulatory control systems has been demonstrated to be a viable robotics framework, for application in real world tasks/scenarios. It should be noted that the potential applications for this technology extend well beyond robotics.

7.1 Conclusions

The essence of this thesis is summarized below for the sake of clarity. The conclusions for this thesis are as follows:

I Neuroregulation, the dynamic ability to activate and inhibit groups of neurons plays an important part in artificial evolution of neural controllers. Neuroregulation is predicted to increase phenotypic neutrality thus providing an unmatched advantage over neural controllers that lack this capability. Neuroregulation is also shown to be an alternate solution to more supervision in overcoming the *bootstrap problem*.

II Controllers such as the artificial neural tissue (ANT) architecture that exploit neuroregulation show improved evolutionary training performance over a range of robotic and control tasks compared with fixed topology neural network architectures.

III Superpositioning of Nitric Oxide concentration field in a coarse coding scheme is shown to be a viable mechanism of neuroregulation in artificial systems. The operational frequency and neuronal activity among such processes match with independent energy/activity estimates done for rodent brains [5].

IV It is possible to predict evolutionary performance of fixed neural network topology through a statistical framework provided the probability of finding constituent feature neuron (components) that make up a solution network is known *a priori*. This approach may also be used to bound the size of a network topology in order to maximizes the probability of finding a solution network.

V Fixed topology configurations are found to tend towards solutions that minimize spatial crosstalk. A mean of reducing spatial crosstalk is through increasing phenotypic neutrality.

VI Increased genotypic neutrality for a three-dimensional mapping scheme used within ANT is analytically shown to decrease the effects of the competing conventions problem.

VII Controllers that exploit neuroregulation are shown to perform self-organized task decomposition, given a global fitness function that does not bias for particular task decomposition strategies and set of generic palette of basis behaviours. This process facilitates emergence of creative solutions that may never have been envisioned by a designer.

VIII Controllers that exploit neuroregulation are shown to concurrently evolve multirobot solution controllers and select for an optimal number of robots.

IX It is feasible to evolve solutions on single and multirobot tasks, with low fidelity grid world simulations and transfer these solution onto any number of robotic hardware platforms provided the sensory input and output basis behaviours match those described in simulation.

X Artificial Neural Tissue architectures are shown able to produce emergent coordination behaviours in solving for tasks in a decentralized multirobot setting. Resultant emergent behaviours remain un intuitive, particularly for little known task domains. However the system show good rescalability performance and robust system performance when evolved under an optimal robot density.

XI A coarse-coding developmental model of neuron structure show evidence of increased variability among neuron ensembles. This model is shown to produce neurons with memory gating functionality and represent multiple states concurrently. Increased variability as predicted from selectionist theories of cognition is found to improve evolutionary training performance.

7.2 Contributions

Neural Network Evolution Theory

I Produced an original, bottom-up theoretical foundation to predict the effects of topology size on finding a solution neural network for a specified task.

II Theoretical framework has been shown to be consistent with observational evidence for a multirobot tile-formation task.

III Framework is used to predict accurately network topology that will maximize probability of finding solution networks.

IV Developed Shannon's Entropy and Mutual Information measures interpretation of a phenotype as a probability distribution of its actions.

Artificial Neural Tissue Framework: An original neural network control framework that has been applied on a growing list of previously unattempted and challenging real-world robotic tasks.

- **I** A novel approach to neural network control has been developed. It models both chemical ('wireless') and electrical ('wired') signaling mechanisms. The method requires much less supervision than previous methods and with only a specification of a global fitness and a generic set of basis behaviours/primitives. Controller topology is expressed through a gene-regulated artificial development process. A coarse coded neuroregulatory framework dynamically activate/inhibits parts of the phenotype through environment modulated stimulus.
- **II** The ANT model is shown to outperform fixed network topologies for a range of robotic and control task. The ANT framework is also shown to solve certain tasks that are found to be intractable (empirically) for network architectures that lack regulatory functionality.
- **III** Demonstrated the advantages of a neuroregulatory framework through analytical means. Results show that neural network controllers can overcome the bootstrap problem with limited supervision by increasing phenotypic neutrality.
- **IV** First known instance of neuroregulatory control architecture validated on multiple robotic hardware platforms.
- **V** Demonstrated that the effects of the competing conventions problem can be reduced with increased genetic neutrality.

Neuroregulation and Coordinated Volume Signalling

- **I** Demonstrated a physical basis for coarse-coding as a regulatory mechanism.
- **II** Showed coarse-coding selection can be exploited to perform computation.
- **III** Results shown to be consistent with biological observations. In particular, evidence of sparse-coded activity, distributed encoding and neuron ensemble degeneracy.

Robotics

- **I** Demonstrated a viable, practical means to overcoming the robot transfer problem. The approach involves evolving neural network controller solutions through low-fidelity simulations, that allow for interchangeability of robotic platforms, without tweaking the controller. Approach validated on a number of robotic platforms with the help of several others (see Acknowledgements).
- **II** The ANT framework shown to produce creative solutions, that may appear unintuitive particularly for little known task domains. Specially suited for collective robotics, where it difficult to know what set of local behaviours give rise to a desired global behaviour.
- **III** Applicability in space exploration and in-situ resource utilization (ISRU) where a global fitness function can be communicated (requiring less bandwidth, less communication infrastructure) instead of a detailed instruction set or teleoperation of a remote, robotic platform.

Self Organization

- **I** Demonstrated a scalable framework to perform self-organized task decomposition, using a global fitness function and a palette of generic output behaviours for memory dependent tasks.
- **II** Demonstrated self-organization maybe exploited to regulate/reshape a fitness functions. Approach is shown to facilitate evolution in stages through behaviour based self-regulation mechanisms.
- **III** Simulation work has shown increased structural variability (repertoires) aids evolvability.

7.3 Open Questions

In this thesis, focus has been exclusively oriented towards use of an artificial evolution paradigm for training these controllers. It would be of utmost significance to also look at how both classical and stochastic learning algorithms can be used within the ANT framework. Of particular interest is to be able to train multirobot controllers with limited supervision directly on hardware. The question remains whether variability in a population of controllers can be augmented by internal variability resident within an ensemble of neurons. The answer would appear to be positive, but for this hypothesis to have practical benefits, one must be able to implement a selectionist envisioned learning algorithm on a single individual instead of a population. This approach maybe advantageous (from a practical viewpoint) is if it can reduce the number of hardware evaluations/samplings.

Information theoretic analysis of the ANT solutions gives us useful insight into processes occurring during evolution. However, it remains to be seen as to whether generic information theoretic approaches could be used as a guiding mechanism in place of a task-specific global fitness function. For the sign-following task, could increasing the information content of an input sensor ensemble facilitate convergence to a solution?

From a biological perspective, it remains an open question as to why the cerebral cortex on average contains six layers of neurons. From a computational perspective, it has been shown that having two or more layers is advantageous [68]. Yet, this earlier work do not take into account spatial or temporal crosstalk. To minimize crosstalk it would be expected that neural networks structure will form columns with higher density of connections within a column than between columns. The question then is whether limiting spatial or temporal crosstalk can account for columnar organization of biological neurons?

One of the issues with using the ANT architecture is the need to determine what sensors and actuators are necessary for a task at hand. In practise, a sensor-input layout is often constrained due to hardware limitations or based on physical characteristics of a robot and thus relying on body-brain coevolution may not always be practical. Evolving for sensor input layouts in addition to the controller may give useful insight into sensors that maybe necessary for a task versus optional ones. This may help guide a robotic designer in their design processes, by helping them determine what sensors and actuators are necessary to complete a given goal.

Chapter 8

APPENDIX

Feature Detector Probability Calculation

In this section we compute the probability of obtaining p_a, for the 'O' feature detector neuron. This is the probability that the feature detector neuron maybe obtained by random initialization of the neuron parameters. The feature neuron for this example needs to correctly identify a 5×5 pixel uppercase 'O' by outputting a 1 and 0 otherwise (Figure 8.1). The neuron makes use of four feature windows labeled ($F_1 \ldots F_4$) as shown with 2×2 pixels per feature window. Let us assume that each pixel contained within the feature window is monochrome (i.e. black or white).

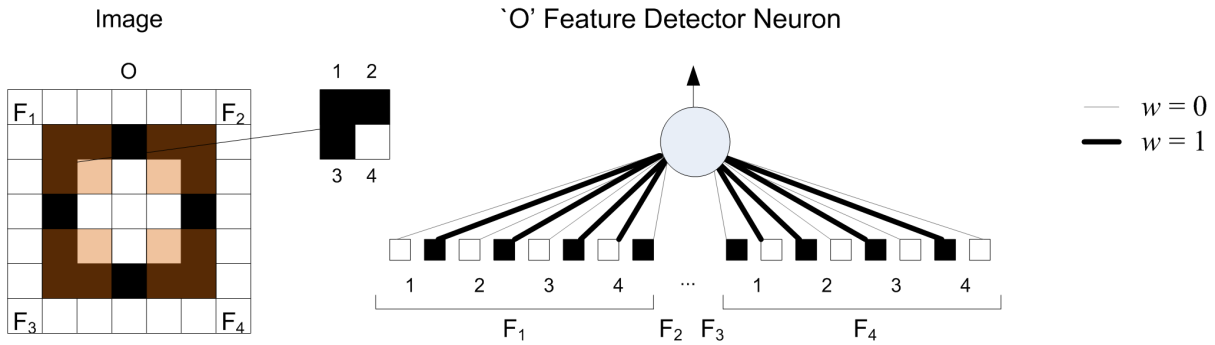


Figure 8.1: The 'O' feature detector neuron makes use of 4 2×2 pixel pre-positioned monochrome feature windows (shaded orange) to identify an uppercase 'O'. A McCulloch-Pitts neuron model is used. In addition the weights and neuron output signal are binary.

The feature detector neuron contains 32 input weights. Let us further assume that each weight is binary (i.e. $w_i \in \{0, 1\}$) and that a McCulloch-Pitts neuron model (equivalent to ϕ_1 from 2.1) is used for each neuron, implying the output is binary. $p(x)$ is computed according to (2.3) and $\sum_i x_i = 16$ and is a constant for this task. Furthermore let us also set the threshold, $\theta \in \{0, 1/16, 1/8, ..., 1\}$.

Based on these constraints, each $w_i = 1$ for the desired colored pixel. In addition, $w_i = 0$ for the inverse of the desired colored pixel. Since each weight is binary, then the probability of obtaining the correct weight for each pixel is $\left(\frac{1}{2}\right)^2 = 0.25$. The probability of obtaining the correct weights for the 16 pixels is then $\left(\frac{1}{2}\right)^{32}$. In addition, $\theta = 1$ to ensure the neuron is activated only when all the pixels match the desired patten and 0 otherwise. Therefore the probability of obtaining this value from random initialization is $\frac{1}{17}$. Then $p_a = \left(\frac{1}{17}\right)\left(\frac{1}{2}\right)^{32}$

Probability Maximization

To find, n, the number of hidden neurons that will maximize the probability of finding a solution network from 2.10, we take the derivative $\frac{dP_1}{dn}$ as follows:

$$\frac{dP_1}{dn} = \log(p_a + p_{\text{neut}}) \cdot (p_a + p_{\text{neut}})^n - \log p_{\text{neut}} \cdot p_{\text{neut}}^n \tag{8.1}$$

and set $\frac{dP_1}{dn} = 0$ to obtain n_{extreme}:

$$n_{\text{extreme}} = \frac{\log\left(\frac{\log p_{\text{neut}}}{\log(p_a + p_{\text{neut}})}\right)}{\log(p_a + p_{\text{neut}}) - \log p_{\text{neut}}} \tag{8.2}$$

Furthermore, taking $\frac{d^2P_1}{dn^2}$, we get:

$$\frac{d^2P_1}{dn^2} = [\log(p_a + p_{\text{neut}})]^2 \cdot (p_a + p_{\text{neut}})^n - [\log p]^2 \cdot p_{\text{neut}}^n \tag{8.3}$$

and setting $\frac{d^2P_1}{dn^2} = 0$ to obtain $n_{\text{inflection}}$:

$$n_{\text{inflection}} = \frac{\log\left(\frac{[\log p_{\text{neut}}]^2}{[\log(p_a + p_{\text{neut}})]^2}\right)}{\log(p_a + p_{\text{neut}}) - \log p_{\text{neut}}} \tag{8.4}$$

For both (8.4) and (8.2), the following conditions apply, $0 < p_a < 1$, $0 < p_{\text{neut}} < 1$ and $0 < p_a + p_{\text{neut}} < 1$.

Expected Topology

Based on the probability $P_1(n)$, we also seek to determine the expected topology. To find the expected topology we make an assumption, namely that selection of an m feature candidate solution is dependent on the solution network probability $P_m(n)$. In order to find the expected topology under this assumption, we

treat $P_m(n)$ as a probability density function. In calculating for the expected topology for $m = 1$, we find K_1:

$$K_1 = \frac{\log p_{\text{neut}} \log(p_a + p_{\text{neut}})}{p_{\text{neut}} \log(p_a + p_{\text{neut}}) - (p_a + p_{\text{neut}}) \log p_{\text{neut}}} \tag{8.5}$$

the normalization constant that satisfies $1 = \int_1^\infty K_1 P_1(n) dn$. This is to renormalize the probability space to determine the most likely number of hidden neurons (under the provided assumption), given the feature detector probability p_a and probability of finding a neutral neuron, p_{neut}, we find expectation value with respect to n, $E_1(n)$:

$$E_1(n) = \int_1^\infty n K_1 P_1(n) dn \tag{8.6}$$

Solving for $E_1(n)$, we obtain the following:

$$E_1(n) = K_1 \left[\frac{(p_a + p_{\text{neut}})(1 - \log(p_a + p_{\text{neut}}))}{(\log(p_a + p_{\text{neut}}))^2} + \frac{p_{\text{neut}}(\log p_{\text{neut}} - 1)}{(\log p_{\text{neut}})^2} \right] \tag{8.7}$$

As expected, if $p_a = 0$, then the solution network probability is also zero, $P_1(n) = 0$, independent of n. As highlighted in this section, varying the network topology has its advantages, namely it could be used to maximize probability of finding a solution. However, what is required with this theoretical framework is the individual probability of obtaining feature neurons. This implies performing task decomposition and determining the components necessary to complete the task and would require some domain knowledge of the task at hand from a human supervisor. In contrast, the techniques we introduce in Chapter 3 and beyond addresses how to modify the network topology to facilitate task decomposition without the need for explicit supervision or *a priori* task-specific information.

BIBLIOGRAPHY

[1] Astor J.C., Adami C., A developmental model for the evolution of artificial neural networks. *ALife* (2000) 6(3):189–218.

[2] Albus, J. S. A theory of cerebellar function. *Mathematical Biosciences*, (1971) 25–61.

[3] Albus, J. S. *Brains, Behavior and Robotics*. BYTE Books, McGraw-Hill, 1981.

[4] Allis, V., Koetsier, T., On Some Paradoxes of the Infinite II. *The British Journal for the Philosophy of Science*, (1995) 46(2):235–247.

[5] Attwell, D., Laughlin, S.B., An Energy Budget for Signaling in the Grey Matter of the Brain, *Journal of Cerebral Blood Flow and Metabolism*, (2001) 21:1133–1145.

[6] Averbeck, B., Latham, P., Pouget, A., Neural Correlations, Population Coding and Computation, *Nature Review Neuroscience*, (2006) 7:358–366.

[7] Ballard, D.H., Cortical Connections and parallel processing. *The Behavioural and Brain Sciences*, (1986) 9:279–284.

[8] Banzhaf, W., (2003) "On the Dynamics of an Artificial Regulatory Network," *Advances in Artificial Life: Proceedings of the 7th European Conference on Artificial Life (ECAL-2003)*, pp. 217–227.

[9] Barfoot, T. D., D'Eleuterio, G. M. T. (1999) "An evolutionary approach to multiagent heap formation," in *Proccedings of the 1999 Congress on Evolutionary Computation (CEC)*, pp. 427–435.

[10] Barlow, H.B., Why have multiple cortical areas? *Vision Research*, (1986) 26:81–90.

[11] Barto, A., Sutton, R., Anderson, C., "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *Artificial Neural Networks: Concept Learning*, SMC-13:834-846. (1983).

[12] Baudrillard, J., *Simulacra and Simulation*, Ann Arbor: The University of Michigan Press, 1994.

[13] Beckers, R. , Holland, O. E., Deneubourg, J. L., (1994) "From local actions to global tasks: Stigmergy and collective robots," in *Fourth International Workshop on the Syntheses and Simulation of Living Systems*, pp. 181–189.

[14] Bentley, P., (2003) "Evolving Fractal Gene Regulatory Networks for Robot," In *Advances in ALife: Proceedings Of the 7th European Conference on Artificial Life*, pp. 753–762.

[15] Berry, M.J., Warland, D.K., Meister, M.: "The structure and precision of retinal spike trains," *Proceedings of the National Academy of Sciences*, USA, (1997) 94:5411–5416.

[16] Bonabeau, E. , Theraulaz, G. , Deneubourg, J.L., Aron, S. , Camazine, S. , "Self-organization in social insects," in *Trends in Ecology and Evolution*, (1997) 12:188–193.

[17] Bonabeau, E. , Dorigo, M. , Theraulaz, G. , *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford Univ. Press, 1999.

[18] Bongard, J., (2002) "Evolving modular genetic regulatory networks," In: *Proceedings of the 2002 Congress on Evolutionary Computation*, 2002, vol 2, pp. 1872–1877.

[19] Bradley, D. A., Seward, D., "The Development, Control and Operation of an Autonomous Robotic Excavator," *Journal of Intelligent and Robotic Systems*, (1998) 21:73–97.

[20] Bristow, K. L. Holt, J. A. "Can termites create local energy sinks to regulate mound temperature?" in *Journal of Thermal Biology*, (1987) 12:19–21.

[21] Brooks, R.A., Elephants don't play chess. *Robotics and Autonomous Systems*, (1990) 6:3–15.

[22] Bullock, T.H., NEUROSCIENCE: The Neuron Doctrine, Redux, *Science*, (2005) 310:791–793.

[23] Burnet, F. M. *Self and not-self*. Cambridge, MA: Cambridge University Press, 1970.

[24] Cain, D.P. *et al.*, *Behavioural Neuroscience*. (1996) 110:86–102.

[25] Calvin, W., *The Cerebral Code: Thinking a Thought in the Mosaics of the Mind*, MIT Press, 1998.

[26] Chantemargue, F., Dagaeff, T., Schumacher, M., Hirsbrunner, B., "Implicit cooperation and antagonism in multi-agent systems," University of Fribourg, IIUF-PAI, Internal Technical Report, 1996.

[27] Chapman, P.F. *et al.* (1992) Inhibition of nitric oxide synthesis impairs two different forms of learning. *NeuroReport 3*, pp. 567–570.

[28] Cowey, A. Why are there so many visual areas? *The Organization of the Cerebral Cortex*. Cambridge, MA: The MIT Press, 1981.

[29] Crabbe, F. L., Dyer, M. G., (1999) "Second-order networks for wall-building agents," in *International Joint Conference on Neural Networks*, vol. 3, pp. 2178–2183.

[30] Cramer, N. L., (1985). "A Representation for the Adaptive Generation of Simple Sequential Programs," In J. J. Grefenstette, ed., *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pp. 183–187.

[31] Crowley, J.L., Coutaz, J., Ray, G., Reignier, P., (2002) "Perceptual Components for Context Aware Computing," In the *Proceedings of the 4th International Conference on Ubiquitous Computing*, pp. 117–134.

[32] Cybenko, G. Approximation by Superposition of a sigmoidal function, *Mathematical Control Signal Systems*, (1989) 2:303–314.

[33] Darwen, P., Yao X., Speciation as automatic categorical modularization. *IEEE Transactions on Evolutionary Computation*, (1997) 1(2):101–108.

[34] Darwin, C. *The variation of animals and plants under domestication*. London: Murray, 1868.

[35] Dawkins, R. *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe without Design*, Norton and Company, New York, NY, 1986.

[36] Dellaert, F., Beer, R. (1994). "Towards an evolvable model of development for autonomous agent synthesis," *Artificial Life IV: Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems*, pp. 246–257, MIT Press, Cambridge, MA.

[37] Deneubourg, J.L., Application de l'ordre par fluctuations 'a la description de certaines 'etapes de la construction du nid chez les termites. *Insectes Sociaux*, (1977) 117–130.

[38] Descartes, R., (1641) Meditations on First Philosophy, in The Philosophical Writings of Ren Descartes, trans. by J. Cottingham, R. Stoothoff and D. Murdoch, Cambridge: Cambridge University Press, 1984, vol. 2, pp. 1–62.

[39] Dorigo, M., Colombetti, M., Robot Shaping: Developing autonomous agents through learning. *Artificial Intelligence*, (1994) (71):321–370.

[40] Dunbabin, M., Corke, P., Autonomous Excavation using a Rope Shovel, *Journal of Field Robotics*, (2006) 23(6/7):379–394.

[41] Dürr, P., Mattiussi, C., Floreano, D. (2006) "Neuroevolution with Analog Genetic Encoding," In: *Parallel Problem Solving From Nature PPSN iX*, vol 9. Springer, Berlin, pp. 671–680.

[42] Edelman, G., Group Selection and Phasic Reentrant Signalling: A Theory of Higher Brain Function, *The Mindful Brain*, MIT Press, 51–95, 1982.

[43] Edelman, G., *Neural Darwinism: The Theory of Neuronal Group Selection*, Basic Books, New York, 1987.

[44] Edelman, G., Gally, J., (1992) "Nitric oxide: linking space and time in the brain," *Proceedings of the National Academy of Sciences*, USA 89:11651–11652.

[45] Eggenberger, P., Gomez, G., Pfeifer, R., Evolving the morphology of a neural net for controlling a foveating retina, *ALife*, vol. 8, (2002) pp. 243-251.

[46] Eggenberger, P., (1997). "Evolving Morphologies of simulated 3d organism based on Differential Gene Expression," *Procedings of the 4th European Conference on ALife*.

[47] Federici, D., Downing, K. Evolution and Development of a Multicellular Organism: Scalability, Resilience, and Neutral Complexification, *Artificial Life*, (2006) vol. 12, pp. 381–409.

[48] Field, D., What is the goal of sensory coding? *Neural Computation*, (1994) 6:559–601.

[49] Fikes, R., Nilson, N., STRIPS: A new approach to Application of theorem proving to problem Solving. *Artificial Intelligence*, 2, (1971) 189–208.

[50] Floreano, D., Dürr, P., Mattiussi, C. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, (2008) 1(1):47–62.

[51] Gally, J., Montague, P., Reeke, G., Edelman, G., "The NO hypothesis: possible effects of a short-lived, rapidly diffusible signal in the development and function of the nervous system," *Procedings of the National Academy of Sciences*, USA (1990) 87:3547–3551.

[52] Garthwaite, J., Charlesm S., Chess-Williams, R., Endothelium-derived relaxing factor release on activation of NMDA receptors suggests role as intercellular messenger in the brain. *Nature*, (1988) 336:385–388.

[53] Gelperin, A., Nitric oxide mediates network oscillations of olfactory interneurons in a terrestrial mollusc. *Nature*, (1994) 369:61–63.

[54] Georgopoulos, A.P., Schwartz, A.B., Kettner, R.E. Neuronal population coding of movement direction. *Science*, (1986) 233:1416–1419.

[55] Gerstein, G., Aertsen, A., *Journal of Neurophysiology*. (1985) 54:1513.

[56] Gecshwind, N., Galaburda A.M., *Cereberal Lateralization : Biological Mechanisms, Associations, and Pathology*, MIT Press 1987.

[57] Gibson, J.I., (1966) The senses constdered as perceptual systems. Boston: Houghton–Mifflin.

[58] Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison–Wesley, NY, 1986.

[59] Gomez, F., Miikkulainen R., (1999) "Solving Non-Markovian Control Tasks with Neuroevolution," In *Proceedings of the International Joint Conference on Artificial Intelligence*.

[60] Gomez, F., Schmidhuber, J., (2005) "CoEvolving Recurrent Neurons Learn Deep Memory POMDPs," In the *Proceedings of Genetic and Evolutionary Computation Conference* pp. 491–498.

[61] Gomez, F., Miikkulainen, R., (2003) "Active Guidance for a Finless Rocket using Neuroevolution," In the *Proceedings of Genetic and Evolutionary Computation Conference*.

[62] Grassé, P., "La reconstruction du nid les coordinations interindividuelles; la theorie de stigmergie," in *Insectes Sociaux*, 1959, 35:41–84.

[63] Graves, A., Beringer, N., Schmidhuber, J., (2004) "A Comparison Between Spiking and Differentiable Recurrent Neural Networks on Spoken Digit Recognition," *Proceedings of the 2nd IASTED International Conference on Neural Networks*, NCI.

[64] Gruau, F., Whitley, D., Pyeatt, L., (1996). A comparison between cellular encoding and direct encoding for genetic neural networks. *Genetic Programming 1996*, 81-89. Cambridge, MA: MIT Press.

[65] Gustafson S., *et al.*, Problem Difficulty and Code Growth in Genetic Programming, *Genetic Programming and Evolvable Machines 5*, (2004) 271–290.

[66] Han, W., Jafari, M.H., (2003) "Controller Synthesis via Mapping Task Sequence to Petri Nets in Multi-Agent Collaboration Applications," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 216–221.

[67] Hargreaves, E.L., Cain, D.P., (1992) Hyperactivity, hyper-reactivity, and sensorimotor deficits induced by low doses of the N-methyl-D-aspartate non-competitive channel blocker MK801, *Behavioral Brain Research*, 47:23-33.

[68] Hassoun, M.H., *Fundamentals of Artificial Neural Networks*, MIT Press, 45–46, 1995.

[69] Hastie, T. , Tibshirani, R. , Friedman, R. , *The Elements of Statistical Learning*. New York: Springer, 2001.

[70] Hawkins, J., Blakeslee, S., *On Intelligence*, Times Books, 2004.

[71] Heyes, C., Four Routes of Cognitive Evolution, *Psychological Review*, (2003) 110:713–727.

[72] Hinton, G., (1981) "Shape Representation in Parallel Systems," *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pp. 1088–1096.

[73] Hinton, G., (1999) "Product of Experts," *Proceedings of the 9th International Conference on Artificial Neural Networks* vol 1, pp. 1–6.

[74] Hochreiter, S., Schmidhuber, J., Long Short-Term Memory, *Neural Computation*, (1997) 9(8):1735–1780.

[75] Holland, J. , *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1975.

[76] Hölscher, C., (1997) Nitric oxide, the enigmatic neuronal messenger: its role in synaptic plasticity. *Trends in Neuroscience*, 20:298–303.

[77] Hornby, G., Pollack, J., Creating High-Level Components with a Generative Representation for Body-Brain Evolution. *ALife 8*, 2002.

[78] Hornby, G.S., Pollack, J.B. (2001) "The advantages of generative grammatical encodings for physical design," In *Proceedings of the IEEE Congress on Evolutionary Computation 2001*, pp. 600–607.

[79] Husbands, P., Evolving Robot Behaviours With Diffusing Gas Networks, *EvoRobots* (1998) 71–86.

[80] Jerne, N.K., Antibodies and Learning: Selection versus Instruction, In the *Neurosciences: A Study Program*, Rockefeller University Press, pp. 200–208.

[81] Jacobs, R., Jordan, M., Barto, A., Task decomposition through competition in a modular connectionist architecture. *Cognitive Science*, (1991) (15):219–250.

[82] Jordan, M., Jacobs, R., Hierarchical Mixtures of Experts and EM Algorithm. *Neural Computation*, (1994) (6):181–214.

[83] Kandel, E.R., Schwartz, J.H., Jessel, T.M., *Principles of Neural Science*, McGraw Hill, New York, 2000.

[84] Kanerva, P., (1993) Sparse distributed memory and related models. In *Associative Neural Memories: Theory and Implementation*. Edited by Hassoun MH. New York: Oxford University Press, pp. 50–76.

[85] Kargupta, H., Editorial: Special Issue on Computation in Gene Expression. *Genetic Programming and Evolvable Machine*, (2002) 3:111–112.

[86] Kimura, M., *The Neutral Theory of Molecular Evolution*. Cambridge, 1983. University Press.

[87] Kirschner, M., Gerhart, J., "Evolvability," *Proceedings of the National Academy of Sciences*, (1998).

[88] Kitano, H., Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, (1990) 4(4):461–476.

[89] Konorski, J., *Integrative activity of the brain: An interdisciplinary approach*. Chicago: University of Chicago Press, 1967.

[90] Kube, R., Zhang, H., (1992) "Collective Robotics Intelligence : From Social Insects to robots," In *Proceedings Of the Simulation of Adaptive Behavior*. pp. 460–468

[91] Lancaster, J.R., A tutorial on the diffusibility and reactivity of free nitric oxide. *Nitric Oxide*, (1997) 1:18–30.

[92] Langdon, W., (2000) "Quadratic bloat in genetic programming," *Proceedings of the Genetic and Evolutionary Computation Conference*.

[93] Lenat, D., Guha, R.V., *Building Large Knowledge-Based Systems*, Addison-Wesley, 1989.

[94] Lennie, P., The cost of cortical computation. *Current Biology*, (2003) 13:493–497.

[95] Levy, W., Baxter, R., Energy efficient neural codes. *Neural Computation*, (1996) 8:531–543.

[96] Lee, Y., *et al.* "Adaptive Stochastic Cellular Automata: Applications," *Los Alamos National Lab Tech Report*: LA-UR 90-1227. (1990).

[97] Lee, C., Rohrer, W.H., Sparks, D.L., Population Coding of saccadic eye movements by neurons in the superior colliculus. *Nature*, (1988) 332:357-360.

[98] Lewontin, R.C., The units of selection. *Annual Review of Ecology and Systematics*, (1970) 1:1–18.

[99] Liu, X., Miller, M., Joshi, M., Sadowska-Krowicka, H., Clark, D., Lancaster, Diffusion-limited reaction of free nitric oxide with erythrocytes. *Journal of Biological Chemistry*, (1998) 273:18709–18713.

[100] Liu, Y., Yao, X., Higuchi, T., Evolutionary Ensembles with Negative Correlation Learning. *IEEE Transactions on Evolutionary Computation* (2000) 4(4):380–387.

[101] Luke, S., Spector, L. (1996) "Evolving graphs and networks with edge encoding: Preliminary report," In Koza, J.R., ed.: *Late Breaking Papers GP 96*. pp. 117–124.

[102] Matarić, M. J., Nilsson, M. , Simsarian, K. T., (1995) "Cooperative multi-robot box-pushing," in *Proceedings of the IEEE/RSJ International Conference on Intelligent RObots and Systems*, pp. 556–561.

[103] Demeris, J., Matarić, M.J., (1998) Perceptuo-Motor Primitives in Imitation, Autonomous Agents '98 Workshop on Agents in Interaction Acquiring Competence.

[104] Mattiussi, C., Floreano, D., (2004) "Evolution of analog networks using local string alignment on highly reorganizable genomes," *NASA/DoD conference on evolvable hardware (EH2004)*, pp. 30–37.

[105] Mattiussi, C., Marbach, D., Dürr, P., Floreano, D., (2007b) The age of analog networks. *AI Magazine*.

[106] Mau, S., Dolan, J., (2007) "Scheduling for Humans in Multirobot Supervisory Control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems*, IEEE, Washington DC.

[107] McCarthy, J., Hayes, P., Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, (1969) (4):463–502.

[108] Melhuish, C. , Welsby, J. , Edwards, C., "Using templates for defensive wall building with autonomous mobile ant- like robots," in *TIMR99 Toward Intelligent Mobile Robots*, 1999.

[109] Michie, D., Chambers, R.A., "BOXES: An Experiment in Adaptive Control," In E. Dale nd D. Michie (eds.) *Machine Intelligence*, Edinburgh: Oliver and Boyd, (1968) (2):137–152.

[110] Miller, D., Machulis, K., "Visual Aids for Lunar Rover Tele-operation," in *Proceedings of 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, ESA Publishing, 2005.

[111] Moncada, S., Palmer, R., Higgs, E., Biosynthesis of nitric oxide from L-arginine. *Biochemical Pharmacology*, (1989) 38:1709-1715.

[112] Montague, P.R., Gally, J., Edelman, G.M., Spatial Signaling in the Development and Function of Neural Connections, *Cerebral Cortex*. (1991) 1:199-220.

[113] Moriarty, D.E., Miikkulainen, R., Forming neural networks through efficient and adaptive co-evolution. *Evolutionary Computation*, (1997) 5:373–399.

[114] Mountcastle, V.B., An Organizing Principle for Cerebral Function: The Unit Module and the Distributed System, *The Mindful Brain*, MIT Press, (1982) 7–50.

[115] Nolfi, S., Using Emergent modularity to develop control systems for mobile robots., *Adaptive Behaviour*, ISAB, (1997) 5(3):343–363.

[116] Nolfi, S., Floreano D., *Evolutionary Robotics : The Biology, Intelligence, and Technology of Self-Organizing Machines*, MIT Press, 2000.

[117] O'Shea, M., Colbert, R., Williams, L., Dunn, S., (1998) Nitric oxide compartments in the mushroom bodies of the locust brain. *NeuroReport* No 3, pp. 333-336.

[118] Ohno, S., Evolution by gene duplication. Berlin: Springer-Verlag, 1970.

[119] Olhausen, B., Field, D.J., Sparse Coding of Sensory Inputs, *Current Opinion in Neurobiology*, (2004) 14:481–487.

[120] Palm, G., On associative memory. *Biological Cybernetics*, (1980) 36:19–31.

[121] Palm, G., Sommer F.T., Associative data storage and retrieval in neural networks. In *Models of Neural Networks III*. Edited by Domany E, van Hemmen J.L., Schulten K. New York: Springer; (1996) pp. 79–118.

[122] Paradiso, M., A theory for the use of visual orientation information which exploits the columnar structure of striate cortex. *Biological Cybernetics*, (1988) 58:35–49.

[123] Parker, C. A., Zhang, H. , Kube, C. R. (2003) "Blind bulldozing: Multiple robot nest construction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Washington DC, pp. 2010–2015.

[124] Philippides, A.O., Husbands, P., O'Shea, M. Four- dimensional neuronal signaling by nitric oxide: A computational analysis. *Journal of Neuroscience*, (2000) 20(3):1199–1207.

[125] Philippides, A.O., Swidbert, R.O., Husbands, P., Lovick, T., O'Shea, M. Modeling Cooperative Volume Signaling in a Plexus of Nitric Oxide Synthase-Expressing Neurons. *Journal of Neuroscience*, (2005) 25(28):6520–6532.

[126] Plaut, D.C., Hinton, G.E., Learning sets of filters using back-propagation. *Computer Speech and Language*, (1987) 2:35–61.

[127] Pujol, J.C.F., Poli, R., Evolving the Topology and the Weights of Neural Networks Using a Dual Representation, *Journal of Applied Intelligence*, (1998) 8(1):73–84.

[128] Quartz, S.R., Sejnowski, T.J., The neural basis of cognitive development: A constructivist manifesto, *Behavioral and Brain Sciences*, (1997) 20:537–596.

[129] Radcliffe, N.J., Genetic set recombination and its application to neural network topology optimisation. *Neural computing and applications*, (1993) 1(1):67–90.

[130] Radding, C.M., Homologous Pairing and Strand Exchange in Genetic Recombination, *Annual Review of Genetics*, (1982) 16:405–437

[131] Reed, S., (1989). Neural Regulation of Adaptive Behaviour. *Ecological Psychology*, 1:97–117

[132] Reinagel, P., How do visual neurons respond in the real world? *Current Opinion on Neurobiology*, (2001) 11:437–442.

[133] Roggen, D., Floreano, D., Mattiussi, C., "A Morphogenetic Evolutionary System: Phylogenesis of the POEtic Tissue," *International Conference on Evolvable Systems* (2003) pp. 153–164

[134] Roggen, D., Federici, D., "Multi-cellular Development: Is There Scalability and Robustnes to Gain?" In *Proceedings of Parallel Problem Solving from Nature* (2004) pp. 391–400

[135] Rolfe, D., Brown, G., Cellular energy utilization and molecular origin of standard metabolic rate in mammals. *Physiological Reviews*, (1997) 77:731-758

[136] Rolls, E., Tovee, M., Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex, *Journal of Neurophysiology* (1995) 73:713–726.

[137] Sander, G.B. *et al.*, In-Situ Resource Utilization (ISRU) Capability Roadmap, NASA Technical Report, 2005

[138] Saravanan, N., Fogel, D., (1995). Evolving neural control systems, *IEEE Expert Magazine*, 23–27.

[139] Saucier, D., Cain, D.P., Spatial learning without NMDA receptor-dependent long-term potentiation, *Nature*, (1995) 378:186–189

[140] Sharma, J., Angelucci, A., Sur, M., (2000) Induction of Visual Orientation Modules in Auditory Cortex. *Nature*, 404:841–847.

[141] Sigal, N., Alberts, B., Genetic Recombination: The nature of a crossed strand-exchange between two homologous DNA molecules, *Journal of Molecular Biology*, (1972) 71(3):789–793.

[142] Simon, H.A., (1969) *The Sciences of The Artificial*, MIT Press, Boston, MA.

[143] Sims, K., (1994) "Evolving 3D Morphology and Behavior by Competition," *Proceedings of Artificial Life IV*, MIT Press, pp. 28–39.

[144] Stanley, K., Miikkulainen, R., (2002) "Continual Coevolution through Complexification," in *Proceedings of the Genetic and Evolutionary Computation Conference*.

[145] Stanley, K., Miikkulainen, R., Evolving Neural Networks Through Augmenting Topologies. *Evolutionary Computation*, (2002) 10(2): 99–127.

[146] Stanley, K., Miikkulainen, R., A taxanomy for aritificial embrogeny., *Artificial Life 9*, (2003) vol 2., pp. 93–130.

[147] Stentz, A., Bares, J., Singh, S., Rowe, P., (1998) "A Robotic Excavator for Autonomous Truck Loading," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robotic Systems*, IEEE, pp. 175–186.

[148] Stewart, R., Russell, A. (2003) "Emergent structures built by a minimalist autonomous robot using a swarm-inspired template mechanism," in *The First Australian Conference on ALife (ACAL2003)*, pp. 216–230.

[149] Stewart, R., Russell, A., (2004) "Building a loose wall structure with a robotic swarm using a spatio-temporal varying template," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 712–716.

[150] Stone, P., Veloso M., (2000) "Layered Learning," *Proceedings of the 11th European Conference on Machine Learning*, pp. 369–381

[151] Thangavelautham, J., Barfoot, T., D'Eleuterio, G.M.T., (2003) "Coevolving communication and cooperation for lattice formation tasks (updated)," in *Advances In Artificial Life: Proceedings of the 7th European Conference on ALife (ECAL)*, pp. 857–864.

[152] Thangavelautham, J., D'Eleuterio, G.M.T., (2004) "A neuroevolutionary approach to emergent task decomposition," in *Proceedings of the 8th Parallel Problem Solving from Nature Conference*, vol. 1, pp. 991–1000.

[153] Thangavelautham, J., D'Eleuterio, G.M.T., (2004) "Learning From Insects: Development of An Emergent Task Decomposition Network for Collective Robotics," *Proceedings of the 6th Dynamics and Control of Systems and Structures in Space Conference*, Riomaggiore, Italy

[154] Thangavelautham, J., D'Eleuterio, G.M.T., (2005) "A coarse-coding framework for a gene-regulatory-based artificial neural tissue," in *Advances In Artificial Life: Proceedings of the 8th European Conference on ALife*, pp. 67–77.

[155] Thangavelautham, J., Alexander S., Boucher D., Richard J., D'Eleuterio G.M.T., (2007) "Evolving a Scalable Multirobot Controller Using an Artificial Neural Tissue Paradigm," *IEEE International Conference on Robotics and Automation*, Washington D.C.

[156] Thangavelautham, J., Smith, A., Boucher, D., Richard, J., D'Eleuterio, G.M.T., (2007) "Application of an Artificial Neural Tissue Controller to Multirobot Lunar ISRU Operations," in *Proceedings of the Space Technology and Applications International Forum*, vol. 880, pp. 389–399.

[157] Thangavelautham, J., D'Eleuterio, G.M.T., (2007) "Developmental Neural Heterogeneity through Coarse-Coding Regulation," *Advances in Artificial Life: Proceedings of the 9th European Conference on Artificial Life*, Lisbon, Portugal,

[158] Thangavelautham, J., Smith, A., Samid, N., Ho, A., Boucher, D., Richard, J., D'Eleuterio, G.M.T., (2008) "Multirobot Lunar Excavation and ISRU Using Artificial-Neural-Tissue Controllers," in *Proceedings of the Space Technology and Applications International Forum* 2008, New Mexico, US

[159] Thangavelautham, J., Barfoot, T., D'Eleuterio, G.M.T., Evolutionary-Based Control Approaches for Multirobot Systems, Chapter 3, *Frontiers in Evolutionary Robotics*, Iba, H, Editor, Advanced Robotics Systems International, Vienna, Austria, 2008.

[160] Tolman, E.C., Honzik, C.H., Insight in rats. *Univ. of California Publications in Psychology 4* (1930) 215–232.

[161] Trianni, V., Dorigo, M., "Self-organisation and communication in groups of simulated and physical robots," in *Biological Cybernetics*, vol. 95. Springer Berlin / Heidelberg, Sep. 2006, pp. 213–231.

[162] Wawerla, J., Sukhatme, G., Matarić, M., (2002) "Collective construction with multiple robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2696–2701.

[163] Werfel, J., Nagpal, R., (2005) "Building patterned structures with robot swarms," in *International Joint Conference on Artificial Intelligence*, pp. 1495–1502.

[164] Whiteson, S. *et al.*, (2003) "Evolving Keep-away Soccer Players through Task Decomposition," In the *Proceedings of the Genetic and Evolutionary Computation Conference*,

[165] Widrow, B., (1987) "The Original Adaptive Neural Net Broom-Balancer," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 351–357.

[166] Wieland, A., (1991) "Evolving neural network controllers for unstable systems," In *Proceedings of the International Joint Conference on Neural Networks* (Seattle, WA), Piscataway, NJ: IEEE pp. 667–673.

[167] Willshaw, D., Buneman, O., Longue, H., Nonholographic associative memory. *Nature*, (1969) 222:960–962.

[168] Wilson, M., Melhuish, C., Sendova-Franks, A., Scholes, S., "Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies," in *Autonomous Robots*, (2004) 17:115–136.

[169] Young, J.Z., Sources of Discovery In Neuroscience, In *Neuroscience: Paths of Discovery*, 1975, MIT Press, pp. 15–46

[170] Zetzsche, C., "Sparse coding: the link between low level vision and associative memory," In *Parallel Processing in Neural Systems and Computers*. (North-Holland): Elsevier Science; (1990) 273–276.

[171] Zhang, S., Bartsch, K., Srinivasan, M., Maze learning by honeybees. *Neurobiology of Learning and Memory* (1996) 66:267–282

[172] Zhang, J., Evolution by gene duplication: An update. *Trends in Ecology and Evolution*, (2003) 18(6):192-198.