Queen Mary
University of London

Rapid and Thorough Exploration of Low Dimensional Phenotypic Landscapes

By

David Peter Smith

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

PhD, Media and Arts Technology

Supervised by:

Dr. Laurissa Tokarchuk, Primary Supervisor

Professor Geraint Wiggins, Secondary Supervisor

Queen Mary
University of London

August 24, 2017

Statement of Originality

I, David Peter Smith, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature:

Date: August 24, 2017

Rapid and Thorough Exploration of Low Dimensional Phenotypic Landscapes

Davy Smith

Abstract

This thesis presents two novel algorithms for the evolutionary optimisation of agent populations through divergent search of low dimensional phenotypic landscapes. As the field of Evolutionary Robotics (ER) develops towards more complex domains, which often involve deception and uncertainty, the promotion of phenotypic diversity has become of increasing interest. Divergent exploration of the phenotypic feature space has been shown to avoid convergence towards local optima and to provide diverse sets of solutions to a given objective. Novelty Search (NS) and the more recent Multi-dimensional Archive of Phenotypic Elites (MAP-Elites), are two state of the art algorithms which utilise divergent phenotypic search. In this thesis, the individual merits and weaknesses of these algorithms are built upon in order to further develop the study of divergent phenotypic search within ER.

An observation that the diverse range of individuals produced through the optimisation of novelty will likely contain solutions to multiple independent objectives is utilised to develop Multiple Assessment Directed Novelty Search (MADNS). The MADNS algorithm is introduced as an extension to NS for the simultaneous optimisation of multiple independent objectives, and is shown to become more effective than NS as the size of the state space increases.

The central contribution of this thesis is the introduction of a novel algorithm for rapid and thorough divergent search of low dimensional phenotypic landscapes. The Spatial, Hierarchical, Illuminated NeuroEvolution (SHINE) algorithm differs from previous divergent search algorithms, in that it utilises a tree structure for the maintenance and selection of potential candidates. Unlike previous approaches, SHINE iteratively focusses upon sparsely visited areas of the phenotypic landscape without the computationally expensive distance comparison required by NS; rather, the sparseness of the area within the landscape where a potential solution resides is inferred through its depth within the tree. Experimental results in a range of domains show that SHINE significantly outperforms NS and MAP-Elites in both performance and exploration.

Acknowledgements

First and foremost, I would like to offer my sincere thanks to my supervisory team, whose expertise, enthusiasm and guidance has been invaluable throught the PhD. It has been a privilege to work at QMUL under their supervision. Dr. Laurissa Tokarchuk, alongside providing an incredible amount of support and guidence, for always letting me know when the newest shiny thing I'd uncovered that day was perhaps just a distraction, and when it was potentially worth investigating further. Dr. Chrisantha Fernando for introducing the field of evolutionary computing to me with such excitement and passion, and Professor Geraint Wiggins for his willingness to supply wisdom whenever I required a push, and his support when difficulties arose.

Thanks are also due to all of my collegues on the MAT program, but in particular Joe Lyske, Duncan Menzies and Louis McCallum who were there from the beginning and supported me throughout. I would like to thank Professor Pat Healey for leading me back onto the path of research when I began to stray.

Most of all, this thesis would never have been completed if it wasn't for the unbelievble support given to me by my family, no matter what difficulties they were facing. I would like to thank my mam and dad for their strength and understanding, and my beautiful wife Shauna, who has travelled this journey alongside me, for her continued love and support, even when her own journey was proving arduous.

Publications

Davy Smith, Laurissa Tokarchuk, and Chrisantha Fernando. Evolving Diverse Strategies Through Combined Phenotypic Novelty and Objective Function Search. In *Applications of Evolutionary Computation*, pages 344–354. Springer, 2015.

Davy Smith, Laurissa Tokarchuk, and Geraint Wiggins. Exploring Conflicting Objectives with MADNS: Multiple Assessment Directed Novelty Search. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, GECCO '16 Companion, pages 23–24, Denver, Colorado, USA, 2016a. ACM. ISBN 978-1-4503-4323-7. doi: 10.1145/2908961.2908975. URL `http://doi.acm.org/10.1145/2908961.2908975`.

Davy Smith, Laurissa Tokarchuk, and Geraint Wiggins. Harnessing Phenotypic Diversity Towards Multiple Independent Objectives. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, GECCO '16 Companion, pages 961–968, Denver, Colorado, USA, 2016b. ACM. ISBN 978-1-4503-4323-7. doi: 10.1145/2908961.2931654. URL `http://doi.acm.org/10.1145/2908961.2931654`.

Davy Smith, Laurissa Tokarchuk, and Geraint Wiggins. Rapid Phenotypic Landscape Exploration Through Hierarchical Spatial Partitioning. In Julia Handl, Emma Hart, Peter R. Lewis, Manuel López-Ibáñez, Gabriela Ochoa, and Ben Paechter, editors, *Parallel Problem Solving from Nature – PPSN XIV*, number 9921 in Lecture Notes in Computer Science, pages 911–920. Springer International Publishing, September 2016c. ISBN 978-3-319-45822-9 978-3-319-45823-6. doi: 10.1007/978-3-319-45823-6. URL `http://link.springer.com/chapter/10.1007/978-3-319-45823-6_85`.

Contents

List of Figures

2: Background

3: Preliminary Experiments

4: MADNS

5: SHINE

B: Parameters For Experimental Studies

List of Tables

2: Background

3: Preliminary Experiments

4: MADNS

5: SHINE

B: Parameters For Experimental Studies

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
CNE	Conventional NeuroEvolution
EA	Evolutionary Algorithm
EC	Evolutionary Computation
EP	Evolutionary Programming
ER	Evolutionary Robotics
ES	Evolutionary Strategies
GA	Genetic Algorithm
IML	Intrinsically Motivated Learning
JSD	Jensen-Shannon Distance
k**-NN**	k-Nearest Neighbours
MADNS	Multiple Assessment Directed Novelty Search
MAP-Elites	Multi-dimensional Archive of Phenotypic Elites
MCMADNS	Minimal Criteria Multiple Assessment Directed Novelty Search
MCNS	Minimal Criteria Novelty Search
ML	Machine Learning
MOEA	Multi-Objective Evolutionary Algorithm
MOLE	Multi-Objective Landscape Exploration
MOO	Multi-Objective Optimisation
NCD	Normalised Compression Distance
NE	NeuroEvolution
NEAT	NeuroEvolution of Augmenting Topologies
NPC	Non-Player Character
NS	Novelty Search
NS-LC	Novelty Search with Local Competition
NSGA	Non-dominated Sorting Genetic Algorithm
PCG	Procedural Content Generation
RL	Reinforcement Learning
SANE	Symbiotic, Adaptive NeuroEvolution
SDT	Self Determination Theory
SHINE	Spatial, Hierarchical, Illuminated NeuroEvolution
SOO	Single-Objective Optimisation
SSE	Sum Squared Error
TWEANN	Topology and Weight Evolving Neural Network

Mathematical Notation

$\mathbb{B}^n$	The set of boolean numbers, $x \in \{0, 1\}$. An n-dimensional boolean number.
$\mathbb{R}$	The set of real numbers.
$\mathbb{Z}$	The set of integers.
$\mathbb{N}^n_{<a}$	The set of natural numbers (inclusive of 0). An n-dimensional integer where $\{x \in \mathbb{Z} \mid 0 \leq x < a\}$.
$\sum\limits_{i=1}^{k}$	Definite Summation within the range $1 \leq i \leq k$.
$a \cdot b$	Multiplication of $a \times b$.
$[a : b]$	Closed interval, the set $\{x \mid a \leq x \leq b\}$.
$[a_1 : b_1] \times \ldots \times [a_n : b_n]$	An n-dimensional bounding area. For each dimension $\{x \mid a_i \leq x \leq b_i\}$.
$\lVert a - b \rVert$	The Euclidean distance between a and b.
$\lfloor x \rfloor$	floor: $\max \{n \in \mathbb{Z} \mid n \leq x\}$.
$\lceil x \rceil$	ceiling: $\min \{n \in \mathbb{Z} \mid n \geq x\}$.
$\lvert A \rvert$	The cardinality of set A.
$\overline{A}$	The mean of A: $\sum\limits_{a \in A} \frac{a}{\lvert A \rvert}$.

1 Introduction

Curiosity is an inherent characteristic of human, and animal, behaviour. Although curiosity, alongside other *intrinsically motivated* activities such as play and creativity, seemingly generate no tangible rewards to an individual, they are beneficial, if not necessary, to future learning and survival. These intrinsically motived behaviours surface as a form of directed exploration within the world, in which strategies may be learned for potential use in future situations (Ryan and Deci, 2000a). The cognitive mechanisms which underpin the emergence of intrinsically motivated behaviours have been widely studied in psychology (Berlyne, 1950, Hull, 1943, Ryan and Deci, 2000a). More recently, these mechanisms have been formally applied to Artificial Intelligence (AI) (Baldassarre and Mirolli, 2013b), termed as Intrinsically Motivated Learning (IML). The application of IML to AI is an area which has begun to produce impressive results, via the algorithmic formalisation of the motivations which drive curious, playful and creative behaviours as directed and structured exploratory processes, inherently useful to solving complex problems and tasks (Schmidhuber, 2013).

Robotics is an area of AI which draws significant inspiration from the embodied nature of animals acting within the physical world. As the systems developed within robotics begin to mature, and their application migrates from simulation and towards physical reality, the conditions for their deployment become increasingly more complex (Silva et al., 2016). Designing functionality which is robust under uncertain and complex conditions is a difficult undertaking, and introduces the potential for fragility within robotic controllers, or deception in the progress made towards their designed function (Lehman and Stanley, 2011a). Therefore, there is a growing concern that the existing traditional engineering approaches applied within robotics will become unsuitable for truly complex and autonomous robotic systems (Silva et al., 2016).

Evolutionary Computation (EC) has been seen as a potential method to alleviate some of the problems faced in the design of highly complex systems, due to its emergent nature, free from low level design requirements (Doncieux et al., 2011). Alongside this, within the field of Evolutionary Robotics (ER), there is a growing body of research which takes inspiration from IML, concerned with the exploration of behavioural diversity rather than the optimisation of objective functionality (Mouret and Doncieux, 2012). Through exploring a diverse range of potential behaviours, it is suggested that more robust controllers, less susceptible to deception, may be developed (Mouret and Clune, 2015).

In this thesis, the current literature concerned with the promotion of behavioural diversity within ER is furthered through both the extension of an existing state-of-the-art algorithm and the introduction of a novel algorithm, which is shown to have the potential to explore low-dimensional phenotypic landscapes more rapidly and thoroughly than the current state of the art (Smith et al., 2016a,c).

The remainder of this chapter is structured as follows. The motivation for the work presented within this thesis is given in Section 1.1 before the specific research questions that are addressed throughout this work (Section 1.2). Next, an outline of the overall structure provided in Section 1.3. The associated publications produced throughout the course of this thesis's development, and the contributions made, are highlighted in Sections 1.4 and 1.5.

1.1 Motivation

In broad terms, the aim of this PhD research is to further develop the recent applications of IML to EC. Models developed in psychology which aim to understand the cognitive mechanisms which underpin IML have been successfully applied to areas of AI, such as Reinforcement Learning (RL), highlighting the potential for this approach to develop complex and autonomous behavioural systems (Schmidhuber, 2013). More recently, research which is influenced by IML has begun to surface within the field of ER, concerned with the exploration of low-dimensional phenotypic landscapes. Again, this is an area of study which shows promise for its potential application to the development of autonomous systems (Mouret and Clune, 2015). Although this complementary work within ER is advancing in a similar direction to the growing body of work within other areas of AI, the study of IML within EC is still relatively embryonic. The overall motivation for undertaking this research is to further develop the current methods applied to the exploration of phenotypic landscapes in EC and ER.

1.2 Research Questions

In this thesis, two state-of-the-art algorithms for the exploration of phenotypic landscapes, the NS algorithm, introduced by Lehman and Stanley (2008) and Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) introduced in Mouret and Clune (2015) are built upon.

This thesis aims to assess whether the application of IML in EC is potentially beneficial to domains in which a range of behavioural strategies are of interest alongside high objective functionality. Initially, this research addresses the following questions:

> Can NS be combined with objective fitness in order to produce a range of high performing solutions which also exhibit high phenotypic diversity?

> Does the addition of NS have a negative effect upon the objective fitness of the solutions?

> Does a population optimised with NS produce solutions to multiple independent objectives?

> Can the combination of novelty and objective search assist with the optimisation of multiple independent objectives as the size of the search space increases?

The MAP-Elites algorithm introduced the concept of an *illumination algorithm* (Section 2.9), taking inspiration from the Multi-Objective Optimisation (MOO) of novelty and objective fitness, most directly Novelty Search with Local Competition (NS-LC). MAP-Elites has been shown to be a successful illumination algorithm (Mouret and Clune, 2015), however, it potentially suffers from limitations in that it does not directly apply pressure for phenotypic exploration. Rather, phenotypic exploration within MAP-Elites is produced solely through the mutation operator. Therefore, building upon the previous questions, this research addresses the following:

> Can the direct promotion of phenotypic exploration be beneficial to the intended functionality of an illumination algorithm?

1.3 Outline of Thesis

This thesis reviews the existing methods and algorithms applied to the optimisation of phenotypic novelty within ER, before providing alternatives for more efficient and fine-grained exploration of these landscapes. The chapters presented within this thesis are structured in the following manner:

Chapter 2

The relevant scientific literature which is built upon within this thesis is presented in Chapter 2. The general fields of EC, NeuroEvolution (NE) and ER are introduced in Sections 2.3 to 2.5. The relevance of IML to the inspiration of the algorithms presented within this thesis is given in Section 2.6. The NS algorithm, which serves as the inspiration for the proposed MADNS variant introduced within this thesis, is described in detail in Section 2.8. A recently formulated class of algorithms, termed *illumination* algorithms, which perform a hybridisation of global divergent phenotypic exploration and local objective exploitation are introduced in Section 2.9. A current state-of-the-art illumination algorithm, MAP-Elites, is presented in Section 2.10. Finally, Section 2.11 summarises the topics discussed within the chapter and highlights the motivation for the experimental studies and algorithmic developments provided in the remainder of the thesis.

Chapter 3

In Chapter 3, the preliminary experimental work which provided the inspiration for the algorithmic developments within this thesis is presented. Initially, in Section 3.2, an experiment is presented in which a linear combination of NS and objective fitness is applied to the evolution of diverse sets of strategies for agents in the video game *Asteroids*.

Secondly, a logic gate experiment was undertaken to assess the ability of NS to simultaneously optimise multiple and conflicting independent objectives. The experimental findings from this experiment are presented in Section 3.3, with the suggestions which lead to the development of the MADNS algorithm discussed in Section 3.3.6.

Chapter 4

MADNS, a novel extension to the NS algorithm is presented in Chapter 4. The background and formal definition of the algorithm are presented in Sections 4.2 and 4.3. Alongside this, a variant of the Minimal Criteria Novelty Search (MCNS) algorithm is developed for domains with unbounded search spaces, named Minimal Criteria Multiple Assessment Directed Novelty Search (MCMADNS).

A series of experimental trials were undertaken to assess the objective performance and phenotypic exploration of MADNS and MCMADNS in a range of simulated maze navigation domains, which are presented in Sections 4.4 and 4.5. A discussion of the findings from these experiments and suggested further directions for the application and development of MADNS and MCMADNS is presented in Section 4.6.

Chapter 5

A substantial body of research exists within mathematics and computational geometry concerned with collision detection and spatial partitioning, with a number of methods developed which efficiently compare large sets of points within low-dimensional spaces. One method, the quadtree (in its 2 dimensional form) (Samet, 1984) is utilised within this chapter for application to a novel algorithm, Spatial, Hierarchical, Illuminated NeuroEvolution (SHINE).

Taking inspiration from the recent MAP-Elites algorithm (Mouret and Clune, 2015), Chapter 5 presents SHINE, a novel divergent algorithm for the exploration of low dimensional phenotypic landscapes. The SHINE algorithm utilises a tree structure for the maintenance and selection of potential candidates. The algorithm is formally defined in Section 5.3.

Initial experimental results from a series of simulated robot navigation tasks, presented in Section 5.4, show that SHINE significantly outperforms NS and MAP-Elites in both performance and exploration.

A second experiment, in which the distance metric of the SHINE algorithm is altered to include objective fitness, validates the ability of SHINE to be constructed as an illumination algorithm. The retina experiment, presented in Section 5.5, shows that SHINE significantly outperforms MAP-Elites in exploration while optimising objective fitness as successfully.

Chapter 6

Concludes the thesis, referring back to the existing literature and outlining the developments made within this work. Suggested applications for the work provided are made, alongside the limitations of the study and proposals for further development.

1.4 Associated Publications

Throughout the course of the development of this thesis, a portion of the algorithmic and experimental work has been presented in international academic publications within the fields of EC and ER. The relevant chapters which include previously published work are as follows:

Chapter 3

The *Asteroids* experiment, (Section 3.2) was presented at the *18th European Conference on the Applications of Evolutionary Computation*, Copenhagen, Denmark, 2015, in the paper Smith et al. (2015).

Chapter 4

The MADNS algorithm (Section 4.3) and the experiments in Sections 4.4 and 4.5 were presented at the *Genetic and Evolutionary Computation Conference* (GECCO) 2016, Denver, Colorado, USA, in the papers Smith et al. (2016a,b).

Chapter 5

The SHINE algorithm (Section 5.3) and the HARD maze experiment (Section 5.4) were presented at the *14th International Conference on Parallel Problem Solving from Nature* (PPSN), Edinburgh, UK, 2016 in the paper Smith et al. (2016c).

1.5 Contributions

The principal contributions of this thesis are:

Chapter 3

The application of combined novelty-objective search to the learning of diverse strategies within a video game domain. An experiment is presented in which a linearly combined MOO of NS and objective fitness is applied to a video game domain. The experiment demonstrates the applicability of such a method for the optimisation of diverse and useful game playing agent controllers.

Chapter 4

The MADNS algorithm, a novel extension to NS for the simultaneous optimisation of multiple independent and potentially conflicting objectives in large phenotypic landscapes, is introduced. MADNS is tested in a number of simulated maze navigation tasks with multiple exits which are both partially and directly conflicting in objective assessment. The MADNS algorithm is shown to outperform NS as the size of the maze increases. Alongside this, an extension to MCNS, the MCMADNS algorithm is presented for unbounded domains.

Chapter 5

A novel *illumination algorithm* for rapid exploration of low dimensional phenotypic landscapes is presented. The SHINE algorithm is shown to outperform a current state-of-the-art illumination algorithm, MAP-Elites in a series of maze navigation domains, and an artificial retina experiment. The SHINE algorithm is shown to explore large phenotypic landscapes in fewer generations, and with higher *quality-diversity* measures, than MAP-Elites.

2 Background

2.1 Introduction

As introduced in Chapter 1, this thesis provides an algorithmic development to the application of IML within EC. In this chapter, the relevant scientific literature is introduced to situate the motivations behind, and the contributions presented in, this thesis. Initially, to provide coherence and clarity to the specific usage of the terms found within the literature, the biologically inspired terminology applied within EC, and used throughout this thesis, is defined in Section 2.2. The general field of EC, which is the main area of focus for the developments presented within this thesis is introduced in Section 2.3.

The experimental assessments presented in Chapters 3 to 5 involve the optimisation of simulated robot controllers, utilising methods from NE, most specifically the NeuroEvolution of Augmenting Topologies (NEAT) algorithm (Stanley and Miikkulainen, 2002). Therefore, in Section 2.4 the major developments in NE are outlined, with an in-depth discussion of the NEAT algorithm in Section 2.4.3.

The algorithms developed within this thesis both take inspiration from, and may be applied to, frequently used domains in ER. A detailed introduction to ER is therefore given in Section 2.5. IML, initially established within the field of psychology (Ryan and Deci, 2000a), is a growing area of study within AI for its potential application to complex domains and to the development of autonomous agents (Baldassarre and Mirolli, 2013a). In Section 2.6, IML is introduced, with its application to the field of AI covered in Section 2.6.1 and ER in Section 2.6.2. The utilisation of IML within the field of ER, through the optimisation of phenotypic diversity, is discussed in Section 2.7.

In Chapter 4 of this thesis, a novel extension to the NS algorithm is presented – MADNS, alongside a variant which extends MCNS. In Section 2.8 the NS algorithm is introduced, with an overview of the numerous extensions and studies recently developed within the field.

Recent developments within the study of divergent phenotypic search in ER have led to the introduction of a new class of Evolutionary Algorithms (EAs) (Mouret and Clune, 2015, Pugh et al., 2016b). Deriving inspiration, in part, from NS-LC, this new set of algorithms have been termed as *illumination algorithms* (Mouret and Clune, 2015). The specific nature of the illumination algorithm is explored in Section 2.9. In Chapter 5 of this thesis, a novel illumination algorithm, named SHINE, is introduced which draws inspiration from the MAP-Elites algorithm. The MAP-Elites algorithm is described in Section 2.10, highlighting suggested future directions for the algorithm, which were explored in the development of SHINE.

Finally, Section 2.11 summarises the literature presented within this chapter to provide the motivation behind the experimental and algorithmic work presented in the remainder of this thesis.

2.2 Biological Terminology

As a biologically-inspired method, EC applies a number of abstractions of concepts and terminology from the field of biology. The biological terminology adopted by EC is used throughout this thesis. However, within the literature, the same term may be applied to different abstractions in different contexts, in particular the *phenotype* (Mouret and Doncieux, 2012). The main biological concepts found within EC, and used throughout this thesis, are defined in this section, in order to provide clarity of their intended meaning for our purposes.

Population

In general, the algorithms applied within EC are population-based search techniques. A population is a finite set of candidate solutions which are optimised towards a single, or multiple, objective(s). Generally, within this thesis, the finite set $\mathcal{P}$ is used to represent a population. The size of the population is defined as the cardinality of this set, $|\mathcal{P}|$.

Individual

An individual is a candidate solution, a member of the population. We define an individual as $\rho \in \mathcal{P}$.

Fitness Function

In EC, the fitness function is used to assess each individual in a population. The goal of a Genetic Algorithm (GA) is to maximise this function. Although traditionally, a fitness function aims to maximise a specific objective, or, in the case of MOO, a set of objectives, the work presented in this thesis is generally concerned with the optimisation of behavioural, or *phenotypic*, diversity. The fitness of an individual is defined as $f(\rho)$. When multiple fitness assessments are compared against one another, a specific definition is provided for each. For example, we define the fitness of an individual calculated with NS as $f_{nov}(\rho)$.

Genotype

In EC, the term *genotype* is a biological abstraction used to refer to the encoding of an individual solution in a population. Generally, a genotype is a string of parameters, with each parameter representing a dimension within the search space. Therefore, a genotype exists as a point within this parameter space. Again, in line with the biological counterpart, each parameter within the genotype is termed a *gene* (for example, a bit in a binary encoding). Also, in EC, a genotype may also be referred to as a *chromosome*, again abstracted from the biological concept.

A simple binary encoding scheme may be illustrated through the *Knapsack problem*, in which a knapsack of finite size must be filled with a maximum possible number of items, each with differing size. The genotypes outlined in

Table 2.1 are used to represent whether an item is held within the knapsack or not (1 or 0), with each bit representing a possible item.

	Item							
	A	B	C	D	E	F	G	H
Genotype A	1	0	1	1	0	1	1	0
Genotype B	0	0	0	1	1	1	1	0
Genotype C	0	0	1	1	1	0	1	1

Table 2.1: *Binary encoded genotypes in the Knapsack problem*

The values used to represent each gene may be any searchable encoding scheme, for example, binary, real, discrete or a tree structure. As it is the genotypic space that search is performed upon, the specific genetic encoding used can greatly impact the effectiveness of a GA. Although the original formulations of the GA presented in Holland (1975) utilised binary encoding, it has since been shown that real number encodings are more optimal for certain optimisation problems. A comprehensive comparison of real-coded GAs and genetic operators was undertaken by Herrera et al. (1998).

Within this thesis, the specific genetic encoding scheme applied is the NEAT algorithm (Stanley and Miikkulainen, 2002). This is a widely used NE method, in which the genotype represents an Artificial Neural Network (ANN). A more detailed introduction to NE and the NEAT algorithm are provided in Section 2.4.

Phenotype

The phenotypic landscape is the major focus of study within this thesis. In biology, the term *phenotype* was introduced by Johannsen (1911) to derive a clear distinction between the heredity of an organism (its genotype) and how this heredity manifests itself through observable characteristics of the organism. Therefore, the term phenotype is used to describe any observable trait of an individual. These traits may include morphological, alongside behavioural, characteristics.

Traditionally in EC, the term phenotype has been used to describe the morphological structure produced through the decoding of a genotype. For example, in NE, the decoding of a genotype into an ANN (Stanley and

Miikkulainen, 2002). As NE is often applied to tasks such as classification and feature selection, with the problem domain remaining separated from the biological world, any further expansion of the term phenotype into behavioural space makes little sense. However, in ER, where a candidate solution is a robot controller, the problem domain is commonly (a simulation of) the physical world. Thus, the behaviour of an agent may be observed in a closer manner to its biological counterpart, much like the observation of animal behaviours (Lehman and Stanley, 2011b). Therefore, the biological abstraction has been expanded to encompass not only the topological structure produced from decoding the genotype, but also the behaviour of an agent within the domain; as there are potentially infinite measures of a controller's behaviour, a single genotype may have multiple phenotypes.

In order to avoid confusion with the specific meaning when referring to a phenotype in ER, the term is sometimes applied *only* in the traditional morphological sense, separated from the behaviour of an agent. For example, Mouret and Doncieux (2012) draws a specific separation between *phenotype space* and *behaviour space*:

> Some authors consider that the behaviour is part of the phenotype. We will here separate the behaviour from the phenotype to make our description of behavioural diversity as clear as possible (Mouret and Doncieux 2012).

Conversely, in Mouret and Clune (2015), the term phenotype is used to also encompass behavioural space. In the MAP-Elites algorithm, *phenotypic elites* are measured through observed low-dimensional phenotypic descriptors, which, dependent upon the experimental domain, may include behavioural characteristics (Cully et al., 2015, Mouret and Clune, 2015). Similarly to Lehman and Stanley (2008), the majority of the experimental work within this thesis is concerned with the optimisation of diverse behaviours within a population. However, in line with Mouret and Clune (2015), the experimental domain presented in Section 5.5 involves diversifying the connections and modularity of ANN controllers. Therefore, within this thesis the biological definition is employed; the term phenotype is used to describe *any* observable trait of an individual, which may include both an observed behaviour, or the underlying structure of an individual's ANN controller.

A phenotypic descriptor of an individual is defined as μ , an n-tuple, where $\mu = (\rho_1, \ldots, \rho_n)$. Each ρ_i refers to a single dimension of an observed characteristic of an individual, with n being dependent upon the specific trait being observed. For example, in the robot navigation mazes presented in Chapters 4 and 5, Maze Ending Position (MEP$_\mu$) is the recorded resting position of the agent at the end of the trial. Therefore, we define the phenotypic descriptor as $\text{MEP}_\mu = (\rho_x, \rho_y)$. In the artificial retina experiment, presented in Section 5.5, the phenotypic descriptor is used to observe the connection weights and the modularity of the ANN controllers. Therefore, in this instance, the Modularity and Connection Cost (MCC$_\mu$) phenotypic descriptor is defined as $\text{MCC}_\mu = (\rho_{modularity}, \rho_{ccost})$.

2.3 Evolutionary Computation

EC is a biologically inspired area of study within AI which applies EAs to Machine Learning (ML) tasks. Taking inspiration from the Darwinian principle of evolution, EAs perform computational search of a problem space through the guided random optimisation of a population of candidate solutions. Although the term EA may be used to describe a range of evolution-inspired algorithms, such as Evolutionary Programming (EP) (Fogel et al., 1966) and Evolutionary Strategies (ES) (Beyer and Schwefel, 2002), it is the GA, introduced by Holland (1975), that is the most commonly applied method in EC, and the focus for the algorithms developed within this thesis.

2.3.1 The Genetic Algorithm

In the traditional GA, as outlined in Algorithm 1, a population of randomly chosen candidate solutions are assessed against a sample problem through the use of a fitness function. Two parents are selected from the original population, with higher probability given to better performing individuals. Dependent upon a crossover probability, the parents are crossed-over at a randomly chosen point, forming two children solutions. The offspring are then mutated, dependent upon a mutation probability, before being inserted into a new population. Once the new population is complete, the old population is replaced and the process repeated for either a fixed number of generations, or until a suitable solution is located.

The GA is a widely applied method in EC and there is a substantial body of research which explores the various characteristics and potential applications of the GA.

Algorithm 1 The Traditional Genetic Algorithm

```
procedure GA
    $\mathcal{P} \leftarrow \varnothing$                          ▷ initialise an empty population set $\mathcal{P}$
    while $|\mathcal{P}| < n$ do                  ▷ Populate $\mathcal{P}$ with $n$ random solutions, $\rho$
        $\rho \leftarrow$ RandomSolution
        $\mathcal{P} \leftarrow \mathcal{P} \cup \rho$
    end while
    do
        $\forall \rho \in \mathcal{P}$ do                        ▷ Evaluate all members of $\mathcal{P}$
            Evaluate($\rho$)
        end for
        $\mathcal{P}' \leftarrow \varnothing$                  ▷ initialise a new empty population set $\mathcal{P}'$
        while $|\mathcal{P}'| < n$ do                ▷ Populate offspring population $\mathcal{P}'$
            $\rho_1, \rho_2 \leftarrow$ WeightedSelection($\mathcal{P}$)     ▷ Select 2 children from $\mathcal{P}$
            $\rho'_1, \rho'_2 \leftarrow$ Crossover($\rho_1, \rho_2$)
            $\rho''_1, \rho''_2 \leftarrow$ Mutate($\rho'_1, \rho'_2$)
            $\mathcal{P}' \leftarrow \mathcal{P}' \cup \rho''_1 \cup \rho''_2$
        end while
        $\mathcal{P} \leftarrow \mathcal{P}'$            ▷ Replace old population $\mathcal{P}$ with offspring $\mathcal{P}'$
    while $terminate$ = false
end procedure
```

The focus within this thesis lies in the development of autonomous systems, most specifically within the field of ER. Due to the aim of ER to develop systems for the autonomous control of robots, the genotypic encodings most commonly used within the field are schemas which are used to encode ANNs. The use of GAs to optimise ANNs, termed NeuroEvolution (NE), is described in the following section.

2.4 NeuroEvolution

NeuroEvolution (NE) is a ML method which applies EAs to the optimisation of ANNs. Unlike more traditional supervised methods for the training of ANNs, such as backpropagation (Hecht-Nielsen, 1989) which require explicitly labelled input-output pairs for feedback to the training algorithm, NE requires no explicit feedback during optimisation. Thus, NE is a generalised training method and may be applied to domains where only limited feedback is available, for example RL problems (Moriarty and Miikkulainen, 1996, Whiteson and Stone, 2006). The only requirement for training through NE, as in a standard GA, is a fitness function, which may be applied to the assessment of an individual after complex task. Thus, NE is often applied to the optimisation of neural controllers where optimal actions at each timestep are unknown — rather only the success of a large sequence of actions is available for assessment, such as in games and autonomous behaviour, e. g., robotics.

In the majority of methods utilised in NE, training progresses in the same manner as a standard GA, with the genotype representing an encoding that may be translated into an ANN. Dependent upon the particular genetic representation, NE can be used to optimise a range of ANNs, including feedfoward and recurrent networks; the weights, neuron activation functions and network topology of the ANNs do not need to be explicitly defined prior to optimisation. In NE, the search process is able to locate promising network topologies and parameters. There exist a number of schemes for the representation of ANNs within NE, which are broadly categorised into Conventional NeuroEvolution (CNE) and Topology and Weight Evolving Neural Networks (TWEANNs). In the following sections, both of these systems are highlighted.

2.4.1 Conventional NeuroEvolution

The most simple method utilised in NE is referred to within the literature as CNE. In CNE, only the weights of an ANN are altered during evolution (Schaffer et al., 1992). Therefore, in CNE, the size and topology of the network must be determined prior to training. In the most simple of CNE methods, a fixed topology network is represented through its concatenated weights, e. g., in (Whitley, 1989). This encoding may then be optimised through the use of a standard GA. In addition to this simple concatenated weight-based encoding scheme, more complex CNE encodings have been developed, for example the Symbiotic, Adaptive NeuroEvolution (SANE) algorithm (Moriarty and Miikkulainen, 1996). In SANE, populations of neurons are evolved rather than full networks. The neurons are then combined as hidden layers in a fully connected network, and their fitness evaluated as the average of the network (Moriarty and Miikkulainen, 1996, Stanley and Miikkulainen, 2002).

CNE is a straight-forward method to implement — there are often no, or few, extra parameters to consider than in a traditional GA. Studies have shown CNE to be suitable method for optimising ANNs in a range of domains (Floreano and

Mondada, 1994, Moriarty and Miikkulainen, 1996, Whitley, 1989). However, it has also been shown that CNE is limited when expanding to more difficult and larger problems; CNE has been shown to be susceptible to convergence towards local optima and suffers from parameter explosion when increasing network size (Stanley and Miikkulainen, 2002). Therefore, when optimising larger ANNs on difficult problems, TWEANN encoding schemes are more commonly applied.

2.4.2 Topology and Weight Evolving Neural Networks

TWEANN methods allow both the connection weights and the topology of a network to be encoded within a genotype. The information required to produce an ANN from a genotype may be either directly encoded, in which all nodes and connections are explicitly stored within the genotype, and indirectly encoded, in which the genotype encodes rules for the expression of an ANN phenotype. Both methods have been successfully applied to NE.

The most simple of TWEANN encodings apply a binary encoding scheme, such as the *structured genetic algorithm* (Dasgupta and McGregor, 1992), allowing genotypes to be optimised using a traditional GA (Algorithm 1). In this system, a binary connection matrix is represented as a string, with the value 1 indicating a connection between nodes. Alongside this, a connectivity matrix is evolved to represent the weights of the connections. Although simple to implement, this method suffers from a number of problems. Firstly, the maximum number of nodes need to be specified prior to running the experiment, as the connection matrices for all genomes need to be the same length. Secondly, the genotype length expands quadratically with the expansion of the maximal number of nodes, irrespective of the number of expressed nodes. Finally, a high proportion of the randomly assigned initial connections within the ANN may have no viable path from input to output nodes, therefore the method suffers from a large number of individuals being infeasible.

The use of graph encoding is a more commonly applied TWEANN schema. Dependent upon the particular encoding schema, graph encoding does not suffer from the problems faced by binary encoding. A number of graph encoding methods have been proposed, including Species Adaptation Genetic Algorithm (SAGA) (Harvey, 1993), and the use of Cartesian genetic programming in Khan et al. (2013). However, the crossover operator can cause a significant problem for many graph encoding based schemes due to the high probability of producing morphological structures which are incompatible. Therefore, many graph based TWEANN methods remove the crossover operator completely.

Perhaps the most widely applied TWEANN method within the field of ER, alongside more general application in EC, is the NEAT algorithm, developed by Stanley and Miikkulainen (2002), which is introduced in the following section.

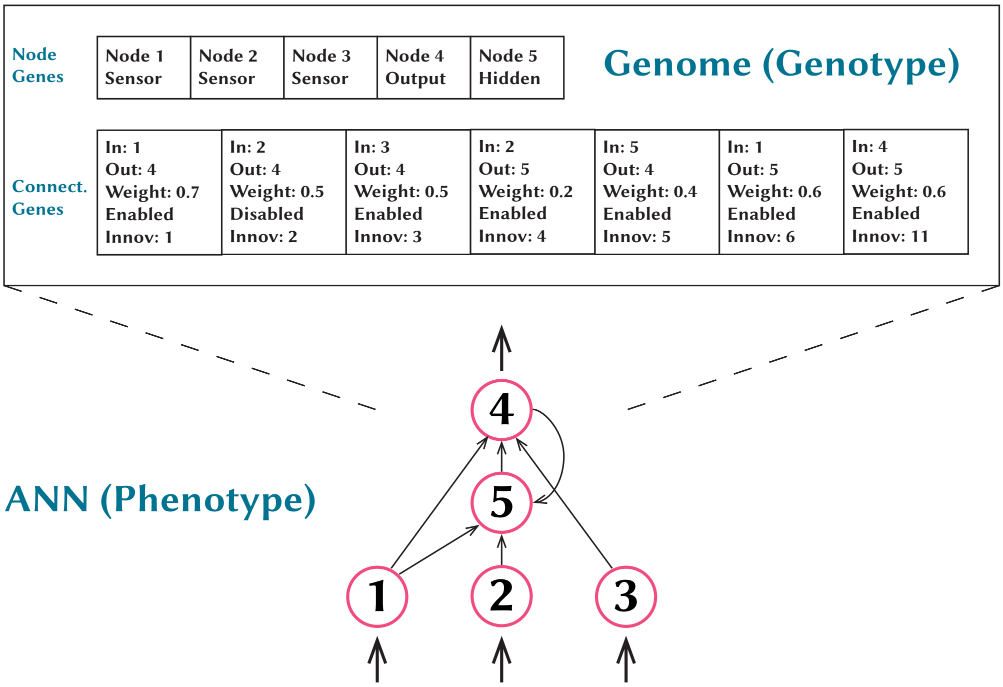
2.4.3 NeuroEvolution of Augmenting Topologies

Since its introduction in Stanley and Miikkulainen (2002), NEAT has become a widely used NE method within ER. NEAT has been successfully applied to a number of domains, including the optimisation of auto-mobile warning systems (Stanley and Miikkulainen, 2002), the real-time evolution of Non-Player Character (NPC) game agents (Stanley et al., 2005), Procedural Content Generation (PCG) within video games (Hastings et al., 2009), evolving competitive car-racing video game agents (Cardamone et al., 2009), competitive co-evolution (Stanley and Miikkulainen, 2002), the evolution of artworks Secretan et al. (2011) and RL (Whiteson and Stone, 2006).

In a similar manner to the SAGA algorithm (Harvey, 1993), the approach taken in NEAT is to begin with a population of neural controllers with minimal topology and to increase complexity as evolution progresses. This process of increasing complexity is inspired by the biological trajectory of evolution observed in natural systems (Martin, 1999) and has been shown to be a promising approach within other areas of EC (Harvey, 1993, Koza, 1995). Due to its wide and successful application, the NEAT algorithm is utilised for the experimental work presented within this thesis. An overview of the NEAT algorithm is provided in the following section.

2.4.4 Overview of the NEAT Algorithm

The NEAT algorithm utilises a direct encoding scheme for the evolution of ANNs with differing topologies. In NEAT, each gene is encoded with two variable length lists. The first list is used to encode the nodes within the ANN and the second list encodes the connections. These are both decoded to produce an ANN. Figure 2.1 provides an example NEAT genotype to phenotype mapping utilising the NEAT encoding scheme.



Node 1	Node 2	Node 3	Node 4	Node 5
Sensor	Sensor	Sensor	Output	Hidden

In: 1	In: 2	In: 3	In: 2	In: 5	In: 1	In: 4
Out: 4	Out: 4	Out: 4	Out: 5	Out: 4	Out: 5	Out: 5
Weight: 0.7	Weight: 0.5	Weight: 0.5	Weight: 0.2	Weight: 0.4	Weight: 0.6	Weight: 0.6
Enabled	Disabled	Enabled	Enabled	Enabled	Enabled	Enabled
Innov: 1	Innov: 2	Innov: 3	Innov: 4	Innov: 5	Innov: 6	Innov: 11

Figure 2.1: *NEAT genotype to phenotype mapping (Stanley and Miikkulainen, 2002)*

An ANN in NEAT begins with minimal topology, i. e., only connected inputs and outputs. Hidden neurons and connections are added through mutation as evolution progresses. To simplify the process of tracking new ANN structures, NEAT marks the historical origin of genes through the use of *innovations*. When a new gene is produced through mutation, a global innovation counter is incremented, and the new gene assigned this innovation number. This allows genes to be compared through historical markings, and through the crossover employed employed by NEAT, enables networks of differing sizes and organisation to remain compatible after reproduction. Although crossover is not fundamental to produce a functioning GA, the NEAT method has shown that, when the encoding scheme allows for a crossover operator without causing significant damage to offspring, its inclusion enhances performance (Stanley and Miikkulainen, 2002).

However, increasing complexification during reproduction of an ANN is likely to initially decrease fitness within offspring. Therefore, NEAT also applies speciation to both encourage a diverse range of topologies and to protect new innovations from being prematurely lost. The mechanism applied to species within NEAT is *explicit fitness sharing* (Goldberg and Richardson, 1987), a commonly used method for diversity preservation in EAs, which is further discussed in Section 2.7.

2.5 Evolutionary Robotics

The field of Evolutionary Robotics (ER) applies methods from EC to develop controllers for the autonomous sensory, navigational and cognitive systems of robots, both in simulation and for real-world application. ER is an area of study still within relative infancy, drawing inspiration from a range of fields. There have been a number of surveys which consider the main aspects of ER and possible future directions for development of the field (Doncieux et al., 2011, Katić and Vukobratović, 2003, Meyer et al., 1998, Nolfi and Floreano, 2004, Silva et al., 2016).

Much in the same way that the interests within the field of EC diverge between the study of artificial life and the applied use of EAs to engineering, ER takes inspiration from, and aims to assist with, the areas of cognitive science, biology and engineering. ER is a rapidly developing field, and has achieved significant progress throughout its short lifespan. However, the results produced through experiments within ER are still relatively simplistic compared to the work produced in mainstream robotics. Therefore, ER has currently not seen widespread adoption within the larger, more general field of robotics. However, practitioners within ER argue that the field has potential due to the robustness of the evolved controllers. The general argument for the potential of ER to assist with the development of autonomous and robust robotic systems, is that the current specifically modelled and preprogrammed approaches are fragile to the uncertainty and complexity of the real world. As EC assesses only the resulting emergent behaviours of the system, rather than requiring hand-designed, detailed specifications of specific behavioural models, its application to robotics is seen by researchers within ER as a more robust method as complexity increases. Rather than specifically designing a desired behaviour, ER may rely upon the holistic evaluation and optimisation of the emergent, self-organization process of evolution. As stated by Stanley (2011):

> …if we can evolve a controller that wakes up inside any body and learns to make it work, all without the need for any traditional analysis whatsoever, there is the potential to revolutionize mainstream robotics. It happened in nature and it should therefore be possible in ER. So while today some in the mainstream may see ER as unnecessary or suboptimal, its promise is in its inspiration, which encompasses the most robust robotic systems on Earth: *nature*. (Stanley 2011, pp. 40)

Much of the study within ER is concerned with the development of robot controllers that are scalable to real-world domains, which involves the consideration of multiple potential objectives under complex conditions of uncertainty. As the field has started to progress, there are a number of open issues which have arisen, and have been suggested as particular areas which may assist with the development of ER to more complex domains as the field progresses.

In Silva et al. (2016), the main open issues faced within ER are highlighted alongside suggested directions for their research. These open issues outlined by Silva et al. (2016) are: *crossing the reality gap*, *prohibitive time required for application to real robots*, *bootstrapping to complex tasks*, *prohibitive time required for application to real robots*, *genotypic encoding and genotype-phenotype mapping*, *lack of standard research practices within ER* and *deception*. Of the open issues suggested by Silva et al. (2016), it is deception which is most relevant to the work developed in this thesis. Deception is a problem which is general to EC, rather than specific to ER. The problem of deception, and the methods developed for overcoming such a problem are discussed in detail within Sections 2.7 and 2.8 of this chapter, and are the focus of much of the work presented throughout this thesis.

There have been a number of recently introduced algorithms which aim to directly address the problem of deception. These algorithms, which include NS (Lehman and Stanley, 2008) and, more recently, MAP-Elites (Mouret and Clune, 2015) share similarities to an area of AI which takes inspiration from IML. In the following section, IML is outlined, starting from its development within psychology, before highlighting its relationship to AI and ER.

2.6 Intrinsically Motivated Learning

Intrinsically Motivated Learning (IML) is the study of the motivations which underly the pursuit of activities for their own sake, rather than from separable consequences . There are numerous activities which can be described as being intrinsically motivated, including but not limited to play, creativity, exploration and curiosity. Due to the nature of intrinsically motivated activities as being removed from any direct rewards or goals, aiming to understand why humans and animals choose to purse such activities is still an area of research within a diverse range of fields. IML has been widely studied in psychology, gaining traction in the 1940s - 60s through two conflicting theories, Operant theory (Skinner, 1953) and Drive theory (Hull, 1943). Operant theory posits that all behaviours are directed by reward, therefore it is the inherent qualities within the task itself that make it enjoyable, or intrinsically motivating. In contrast to this, Drive theory is based on the fulfilling of psychological needs, and directs its focus towards the fulfilment of these needs through the pursuit of an activity.

Self Determination Theory

Self Determination Theory (SDT) is a branch of experimental psychology that is concerned with the "'What' and 'Why' of goal pursuits" (Ryan and Deci, 2000b). SDT posits that human nature tends towards the satisfaction of psychological needs. This active process gives rise to motivations for exploratory, creative or playful behaviours. SDT characterises three universal basic needs: *competence*, *autonomy* and *relatedness*. A situation that provides these three needs will promote intrinsically motivated behaviours. Conversely, situations that are either controlling or offer external rewards create extrinsically motivated behaviours. Extrinsic motivation may become partially or fully internalised by the individual, but as such behaviour has not resulted from intrinsic means, it will ultimately provide limited needs satisfaction. SDT has linked goal-orientated tasks and reward systems with extrinsic motivation, showing that in various situations, extrinsic rewards have detrimental effects on motivation, sustained interest and general psychological well-being.

Flow

Introduced by Csikszentmihalyi (1996), *flow* is a theory concerned with the immersive qualities of undergoing autotelic activities. There have been multiple studies of immersive experiences and technologies in understanding users' feeling of flow. Being in a state of flow has been compared to the feeling of being *in the zone* experienced by people undertaking activities such a sports or creative pursuits.

Aesthetics and Psychobiology

An ambitious project into experimental aesthetics was attempted by D. E. Berlyne, providing a substantial body of research into intrinsic motivation (Berlyne, 1950, 1970, 1971). For Berlyne, the reward from exploratory behaviour is not generated from the activities themselves, but rather from the resulting cognitive processes that they produce (Berlyne, 1971, p. 118). In a similar manner to Hull (1943), exploratory activities are seen by Berlyne as the result of an internal drive towards a series of particular actions that will result in the generation of intrinsic rewards (Berlyne, 1971, p. 277-290). When not immediately faced by the satisfaction of biological needs (e.g. hunger), it is the search for intrinsic reward that dictates our actions.

Berlyne separates the mechanisms that are used to provide the sufficient stimuli for generating intrinsic rewards into three distinct behaviours; *Receptor-adjusting responses* are the most basic and frequent, and are characterised by repositioning the sensory organs to provide required stimuli, for example turning to look at a painting. *Locomotor exploration* involves actively moving towards new sources of stimulation. The third, *investigatory responses*, is used for all other forms of exploratory behaviour and describes the movement and manipulation of external objects and environments to illuminate unseen perspectives of them, or to create transformations that provide new stimuli, for example purchasing a theatre ticket or tuning in a television set (Berlyne, 1971, p. 99).

Thus, informational needs are satisfied, in part, through direct action. Not only does one have the need to gain knowledge about a particular object or environment, but also which actions can be taken to select useful objects and environments. “It is not only necessary to find answers to questions. It is necessary to select questions” (Berlyne, 1971, p. 295).

Berlyne (1971), remains one of the most comprehensive experimental analyses of intrinsic motivation. The term *collative variable* was introduced by Berlyne to describe the hedonic response to external stimuli, which were measured for values that include novelty and complexity. Berlyne attempted to experimentally assess aesthetic arousal from external stimuli, including levels of pleasingness and interestingness in relation to these collative variables (Berlyne, 1970). For Berlyne, interestingness was seen as a measure of how much intrinsic reward is predicted to be received from selecting a particular set of actions that will result in exposure to moderate levels of these stimuli (Berlyne, 1970). However, what is deemed *useful* is not a direct reflection of the immediate absorption and understanding of exposed information. Rather, there exists in *interesting* observations both understandable information and currently incoherent, or incompressible data, which may only be interpreted through either further observations or mutations of internal interrelationships (Berlyne, 1971, p. 215).

2.6.1 Intrinsically Motivated Learning in Artificial Intelligence

Through the application and formalisation of models developed in psychology, researchers in AI have began to focus upon the study of IML. The study of IML is an area within the field of AI which aims to model the autonomous exploratory and curious behaviour exhibited by humans and animals for the application to machines. It has been suggested, and a growing body of empirical results have shown, that the application of intrinsically motivated models to computational learning may produce systems capable of autonomous learning and the capacity for the acquisition of independent and hierarchical skill sets. Baldassarre and Mirolli (2013b), a major survey of research within the area of IML, outlines three major mechanisms relevant within the field — *prediction*, *competence* and *novelty*.

Much of the current research dedicated to modelling IML systems has developed from the work in psychology and cognitive science. Generally, an intrinsically motivated system will generate an internal reward to an agent. In the *prediction progress* model, internal rewards are generated through the agent making progress in the predicted outcome of the world after performing an action (Oudeyer et al., 2007b). In Schmidhuber (2013), reward is generated through *compression progress* — an action is deemed to be interesting if it results in a state which increases the current internal compression of the agent. Due to this focus upon internal reward generation, it is generally the area of RL, rather than EC which has focussed upon IML.

2.6.2 Intrinsically Motivated Learning in Evolutionary Robotics

Unlike other approaches applied to IML, for example reinforcement learning, in EC an individuals' behaviour results from the underlying genotype of the controller, rather than being acquired during the agent's lifetime due to an (internally) generated reward. However, the study of intrinsic motivation is still of great relevance to the current focus within ER upon the optimisation of behavioural diversity. In the case of the NS algorithm, for example, there is no objective goal defined. Rather the aim is to explore as many novel behaviours as possible. Therefore, a population optimised through NS should exhibit intrinsically motivated activity in the sense that the behaviours rewarded are not directly beneficial to the objective.

IML, in this sense, can be considered a process of *directed* exploration, in which optimisation is directly focussed upon the encouragement of population dynamics which exhibit phenotypic diversity.

2.7 Diversity in Evolutionary Computation

Historically, diversity encouragement and preservation have been a central focus for practitioners in EC, as they are fundamental to the successful functioning of EAs. More recently, these previously explored methods have been adapted and applied to an area which shares similarities to the focus of IML. In the following sections the topic of diversity preservation and encouragement in EC is explored.

2.7.1 Exploitation vs. Exploration

The processes of exploration and exploitation are fundamental to the successful function of an EA (Črepinšek et al., 2013). As a class of search algorithms, EAs optimise populations through searching a landscape of potential solutions. Exploration is the process of traversing the landscape, where as exploitation is the process of focussing upon promising solutions within the landscape. Exploration and exploitation are fundamental to search, Eiben and Schippers (1998) go as far as to state that: "exploration and exploitation are the two cornerstones of problem solving by search". The commonly held belief that, in EAs, exploration is directed by search operators (e. g., mutation and crossover), and exploitation is produced by selection is questioned by Eiben and Schippers (1998), highlighting that the processes are not directly antagonistic forces; rather, there exists a complex interplay between the two processes. For example, in NS exploration is directly promoted through selection (Lehman et al., 2013).

2.7.2 Multi-Objective Optimisation

MOO is a mathematical area of study concerned with the development of methods for the optimisation of multiple objective functions simultaneously. MOO has found application in areas which require optimal decision making in domains where trade-offs occur, for example in economics and finance alongside biology, engineering and computer science. In EC, MOO is an area of active study, and a number of methods have been suggested. Formally, a MOO problem may be defined as in Equation [2.1]:

$$\begin{aligned} & max\,(f_1(\rho), f_2(\rho), \ldots, f_k(\rho)), \\ & \text{where } \rho \in \mathcal{P} \wedge k \geq 2 \end{aligned} \qquad [2.1]$$

The main focus for Multi-Objective Evolutionary Algorithms (MOEAs) is to locate sets of solutions which are *Pareto* optimal. Given 2 individuals, ρ_1 and ρ_2, ρ_1 can be said to dominate ρ_2 if the conditions given in Equation [2.2] hold true:

$$\begin{aligned} & f_i(\rho_1) \geq f_i(\rho_2) \;\; \forall\, i \in \{1, 2, \ldots, k\} \wedge \\ & f_j(\rho_1) > f_j(\rho_2) \;\; \exists\, j \in \{1, 2, \ldots, k\}, \\ & \text{where } \rho_1, \rho_2 \in \mathcal{P}. \end{aligned} \qquad [2.2]$$

As the aim in a MOEA is to optimise a set of Pareto optimal solutions, rather than an individual solution as in tradition EAs, successful MOEAs require a high amount of diversity within the population. Therefore, one of the major considerations in MOO is the preservation of diversity within a population. The Non-dominated Sorting Genetic Algorithm (NSGA) algorithm, introduced by Deb et al. (2002), one of the most widely applied MOEAs, for example, utilises fitness sharing intrinsically within the algorithm.

Alongside the methods developed for the encouragement of diversity within MOEAs, it has also been shown that MOO inherently preserves genetic diversity more effectively than Single-Objective Optimisation (SOO), even when applied to single-objective problems (Abbass and Deb, 2003). Knowles et al. (2001) utilised the inherent diversity preservation of MOO for single-objective problems, in order to avoid premature convergence towards local optima experienced by SOO. In the *genetic diversity* EA, introduced by Toffolo and Benini (2003), genetic diversity is directly assessed as an objective within a MOEA. Bui et al. (2005) applied a similar technique for optimisation problems in dynamic environments.

2.7.3 Multimodal Optimisation

Genetic diversity is inherently produced within the population of a GA through the mutation and crossover operators Holland (1975). However, GAs are still susceptible to premature convergence, and in real-world problems, there are often multiple optima in the search space (Sareni and Krahenbuhl, 1998). In multimodal optimisation, the population needs to exploit a number of potential optima, rather than making progress towards a single optima. One method is to niche the population into multiple, diverse sub-populations. The most commonly applied niching techniques are *fitness sharing*, *crowding* and *clearing*. In Friedrich et al. (2008), a theoretical analysis of the effect of deterministic crowding and fitness sharing on a Twomax landscape proved that without diversification, a simple EA would optimise towards a local optima with probability of $1/2$. The probability of finding both optima was significantly increased with inclusion of either diversity mechanism. These methods are commonly used in multimodel optimisation, but have also been applied as diversity preservation mechanisms in other areas of EC.

Fitness Sharing

Fitness sharing is a simple method used for diversity maintenance. Fitness sharing penalises individuals in densely populated areas of the landscape. The population is divided into niches based on a similarity function. Within each niche, the fitness of each individual is then divided by the number of individuals in the niche. This penalises individuals within large niches, which is analogous to densely populated areas of the landscape in which similarity is measured. Similarity can be measured at either the genotypic or phenotypic level; in binary representations, genotypic distance is usually measured using the Hamming distance. Deb and Goldberg (1989) shows that measuring similarity at the phenotypic level may produce more successful results (Sareni and Krahenbuhl, 1998).

The NEAT algorithm applies fitness sharing directly in the speciation mechanism of the algorithm in order to both encourage diversity and protect new innovations Stanley and Miikkulainen (2002).

Clearing

Clearing is a similar method to fitness sharing. The main difference in clearing is that each niche has a maximum size. Therefore, if a niche has a capacity k, then the fitness of the k best individuals is preserved and the fitness is set to 0 for the rest of the (dominated) individuals within the niche (Pétrowski, 1996).

Crowding

Initially introduced by De Jong (1975), crowding is method of diversity preservation in which the general concept is to insert individuals in the population by replacing similar individuals. Since the introduction of *standard crowding* by De Jong (1975), there have been a number of crowding methods introduced, including *deterministic crowding* (Mahfoud, 1995), *restricted tournament selection* (Harik, 1995).

2.7.4 Phenotypic Diversity

More recently within the field of EC, and most directly in ER, there has been a growing body of research studying the importance of preserving and promoting *phenotypic* diversity within an evolutionary trajectory. A number of studies have translated previously applied diversity preservation mechanisms towards phenotypic diversity preservation. Moriguchi and Honiden (2010), Trujillo et al. (2008a,b) expanded the method of speciation within the NEAT algorithm to measure phenotypic, rather than genotypic space, introducing a *behaviour based speciation* mechanism. It was shown that this behaviour based speciation mechanism was capable of producing multiple strategies within a population without significant decrease in fitness. Ollion and Doncieux (2011) showed that measuring phenotypic exploration early on in an evolutionary trial serves as a good indicator for projected objective performance.

The process of optimising phenotypic diversity is the focussed exploration of possible behaviours. Unlike traditional EAs which aim to converge *towards* a single, or set of, objective(s), the optimisation of phenotypic diversity is inherently *divergent*, instead exploring the phenotypic landscape for possibilities, irrespective of their objective functionality. However, these algorithms are often applied with the aim of utilising the discovered behaviours for assistance to objective search, e. g., in overcoming deceptive and complex problems where objective search struggles. The work within this thesis focusses directly upon this currently developing area within ER. Therefore, the major developments, algorithms, and the current areas of focus are outlined below and throughout the following sections.

Lehman and Stanley (2008) introduced NS, which adopted a unique approach, in that it removed fitness pressure, instead measuring individuals *only* by behavioural distance. NS, more comprehensively discussed in Section 2.8, has been widely studied and developed upon since its introduction, and serves as the inspiration for much of the work presented in this thesis.

Mouret and Doncieux (2009a) introduced the *behavioural diversity* algorithm to assist with overcoming the bootstrap problem. The behavioural diversity algorithm shares similarities with NS, in that it measures individuals by behavioural distance. However, the behavioural diversity algorithm still applies fitness pressure through the MOO of fitness and phenotypic diversity. Also, unlike NS, behavioural diversity does not include an archive of previous behaviours (Section 2.8). Mouret and Doncieux (2012) provides a comprehensive study of behavioural diversity, again concluding that the MOO of fitness and diversity outperforms fitness sharing and genotypic diversity measures. Mouret and Doncieux (2009a) highlights the potential problem faced by NS in completely removing fitness pressure, illustrating an easily discovered and initially highly novel behaviour, which would have a fitness value of 0, and would potentially pollute the population, hindering the evolutionary process from starting. However, this illustration is limited in its approach, as in the example provided by Mouret and Doncieux (2009a), the novelty calculation *only* includes the archive. However, it has since been shown that NS is susceptible to producing early populations of highly novel individuals with low fitness, thus severely affecting the evolutionary process. This observation directly led to the development of MCNS, which is introduced formally in Section 2.8.3.1.

2.8 Novelty Search

Fitness function design in EC is a subjective process. Although there is often the space for a number of fitness functions to be designed and tested upon small sample sets before selecting the appropriate measure for the problem, as domains increase in complexity, this may not always be the case. Fitness function design is therefore subject to human error. What may seem to be initially a good indicator of objective fitness, may, in fact be optimising a population towards locally optimal solutions. This can be illustrated through the use of an example which served as the inspiration for the development of the Novelty Search (NS) algorithm (Lehman and Stanley, 2008).

In maze navigation tasks, the general aim is to optimise a robot controller to navigate a path from a starting position to an exit. Usually, the fitness can be easily determined by calculating the Euclidean distance from the robot to the exit; the closer the robot has managed to get towards the exit, the higher its fitness. In most maze domains, this approximation serves as an appropriate measure of fitness to allow the population to optimise solutions which are able to navigate to the exit. However, a simple maze can be constructed in which this fitness function actually directs search towards an incorrect path, a local optima. The HARD maze domain, illustrated in Figure 2.2, is such a domain.

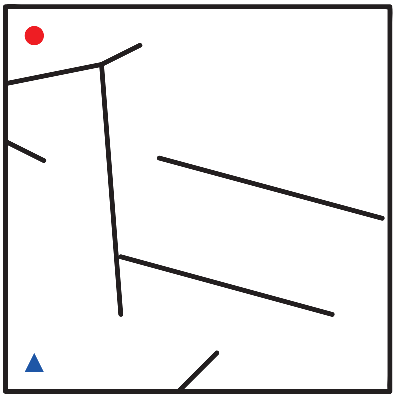


Figure 2.2: *The HARD maze domain*

Figure 2.2 depicts a maze domain in which an agent must navigate from the starting position (triangle), towards the exit (circle). As can bee seen, in order to successfully navigate to the exit, the agent must first move *away* from it, toward the bottom-right corner of the maze. When using linear distance as a fitness function, this will actually *decrease* the fitness of an agent which is making progress towards the exit. Conversely, an agent which moves upwards towards the exit, will be seemingly making progress, but will, in fact, get trapped within the local optima. Because of this deception, an EA with objective fitness performs extremely sub-optimally on this task, managing to locate successful solutions in only a small percentage of trials (Lehman and Stanley, 2008).

This simple example serves to highlight the problem of *deception*, which is often encountered in EC, and something with which EAs are particularly susceptible to. Introduced into the EC literature by Goldberg et al. (1989), the problem of deception has since received wide interest by researchers in the field (Lehman and Stanley, 2008, Stanley and Lehman, 2015, Vose, 2014, Whitley, 2014). The amount with which the level of deception increases the complexity of a problem is still contested within ER. However, Whitley (2014) goes as far as to claim that "the only challenging problems are deceptive", providing a mathematical proof of this theorem. Stanley and Lehman (2015) frames the problem of deception beyond EC, illustrating that it is encountered in much of human life; that paradoxically, sometimes making small advancements towards a larger overall objective can actually hinder performance in the long term:

> Objectives are well and good when they are sufficiently modest, but things get a lot more complicated when they're more ambitious. In fact, objectives actually become *obstacles* towards more exciting achievements, like those involving discovery, creativity, invention or innovation — or even achieving true happiness. (Stanley and Lehman 2015, pp. 7-8)

It is this observation that inspired Joel Lehman and Ken Stanley to "abandon objectives", developing the NS algorithm (Lehman and Stanley, 2008). In NS, the objective fitness of an individual is ignored; the search instead focusses on exploring the phenotypic landscape, with the aim of producing as many novel behaviours as possible. Since the introduction of NS, there has been growing interest in the application of promoting phenotypic diversity to assist with the development of ER beyond simulation and towards more complex domains, alongside the more intrinsically motivated goals of computational creativity and open-ended evolution (Bedau et al., 1998, Secretan et al., 2011). In the next section, the NS algorithm is introduced formally.

2.8.1 The Algorithm

NS is an algorithm which removes the need for an objective fitness function through the assignment of high fitness values to novel behaviours in a population (Lehman and Stanley, 2008). NS does not perform any objective calculation between individuals, rather it is a specific *phenotype* of an individual that is considered. The sole aim of NS is to diversify behaviour within a population. Individuals are measured by observed phenotypic traits; individuals which exhibit newly discovered phenotypes are deemed to be highly novel. In NS, the *novelty* of an individual is calculated as the average distance from a predefined phenotype to the phenotypes of both the other individuals within the population and an archive of previously highly novel phenotypes

Formally, let a phenotypic descriptor be defined as an n-tuple, $\mu = (b_1, b_2, \ldots, b_n)$, where each b_i is a dimension of an observed behaviour of an individual ρ.

The behavioural novelty $f_{nov}(\rho)$ of an individual is defined as the mean distance between μ and its k-Nearest Neighbours (k-NN), where k is a user defined parameter and x_i is the ith nearest neighbour of μ with respect to a distance function, $dist(\mu, x_i)$. The phenotypic descriptors represented by x_i include both the behaviours of the current population and an archive of previously discovered novel behaviours, which is represented by $\mathcal{X}$. The assignment of novelty to an individual is thus given as in Equation [2.3].

$$f_{nov}(\rho) = \frac{1}{k}\sum_{i=1}^{k} dist(\mu, x_i) \qquad [2.3]$$

If the novelty value is above a predefined threshold, t, the then phenotypic descriptor is added to $\mathcal{X}$. The value for t may be dynamically altered if no phenotypes have been added for a number of generations. The update rule for the archive is defined in Equation [2.4].

$$\mathcal{X}' = \begin{cases} \mathcal{X} \bigcup \mu & \text{if } f_{nov}(\rho) > t \\ \mathcal{X} & \text{if } f_{nov}(\rho) \leq t \end{cases} \qquad [2.4]$$

The archive maintenance method presented in Equation [2.4] is proposed by Lehman and Stanley (2008), and is the most commonly used configuration. However other methods for archive maintenance have been suggested. In Lehman and Stanley (2010a) the n most novel individuals of each generation are selected for inclusion within the archive. In Liapis et al. (2013) n random individuals are selected from the population, irrespective of their novelty score. Mouret and Doncieux (2009b) remove the archive completely, therefore only considering the diversity of the current population.

Although NS requires no objective function, there are a number of factors required to take into consideration when applying the algorithm to a particular domain. Gomes et al. (2015) provides a comprehensive study of the effects produced in altering the various parameters of NS. The study presents the following suggestions for the application of NS:

- The k parameter (k-NN) is generally robust, but dependent upon the configuration of the archive. A value of $k = 15$ produces relatively good performance irrespective of the archive type.

- A randomly composed archive is universally preferable over the traditional, novelty-based archive configuration.
- When combining NS with fitness-based search, there is no significant difference between the use of NS and objective fitness as objectives in the NSGA-II MOEA and an equally weighted linear combination of NS and objective fitness.
- High mutation rates penalise exploration of the phenotypic landscape

However, it must be noted that Gomes et al. (2015) only applies NS to various maze navigation domains, with the phenotypic descriptor MEP_{μ} used throughout the experimental assessments. Therefore, the guidelines presented by Gomes et al. (2015) may not be applicable across domains. Alongside the suggestions made by Gomes et al. (2015), the definition of the phenotypic descriptor, μ , is perhaps the most important consideration when applying NS (Pugh et al., 2016b). The effects of using differing phenotypic descriptors in the application of NS is an active area of research, and is further discussed in the following section.

2.8.2 Measuring Phenotypes

As previously stated in Section 2.2, the term phenotype was introduced by Johannsen (1911) to describe any observable trait of an individual; a single individual may potentially have an unlimited number of phenotypes. Therefore, when applying an algorithm which optimises phenotypic diversity, such as NS, careful consideration needs to be taken when selecting *which* phenotype to diversify. The effect of measuring one particular phenotypic descriptor rather than another is an area which has received much recent study in ER, with a number of different conclusions.

Gomez (2009) showed that encouraging diversity by measuring the ANN output vectors and using Normalised Compression Distance (NCD) as a distance metric produces significantly higher behavioural diversity than genotypic or fitness based measures. A surprisingly effective approach taken by Doncieux and Mouret (2012), involved measuring a number of phenotypic descriptors[1], and then randomly switching between them during evolution. This method was shown to outperform both the individual measurements and produce comparable results to the mean of all measurements with lower computational cost. The

[1] Doncieux and Mouret (2012) term a phenotypic descriptor as a *behavioural descriptor*.

suggestion given in Doncieux and Mouret (2012) is that, rather than devising a number of phenotypic descriptors and then trying each of them to discover the most successful, to instead use all of them, applying each randomly throughout the evolution. Gomes and Christensen (2013) showed that domain agnostic, generic phenotypic descriptors may be suitable for optimisation with NS, removing the need for hand-designing domain specific measures.

Most recently, Pugh et al. (2016a,b) provide a comprehensive study in a number of complex maze navigation tasks with two differing phenotypic descriptors[2]. The first is the MEP_μ phenotypic descriptor, introduced in the original NS experiments by Lehman and Stanley (2008) and often applied since, including the experiments presented in Sections 4.4, 4.5 and 5.4 of this thesis. The MEP_μ descriptor is 2-dimensional, where each dimension records the ending x and y coordinate of the agent respectively. The second decriptor used in Pugh et al. (2016a) is a 5-dimensional descriptor, where each dimension records the rotation of the agent at each fifth of the trial time. These descriptors were selected for their *quality aligment*. The first, MEP_μ is direclty aligned with the objective fitness landscape, i. e., the ending position of the agent may be directly translate to the distance from the exit. The second descriptor is less directly able to provide information of the quality of the solution in terms of whether the exit has been reached. The results of Pugh et al. (2016a,b) show that the unaligned phenotypic descriptor performs significantly sub-optimally compared to the traditional MEP_μ descriptor. This further highlights the importance of domain specificity in selecting a phenotype to diversify.

2.8.3 Novelty Search Variants

Since the introduction of NS, various hybrid variants have been proposed to assist the phenotypic exploration potential of the algorithm towards particular objectives. Hybrid NS variants include MCNS (Lehman and Stanley, 2010b), progressive MCNS (Gomes et al., 2012), NS-LC (Lehman and Stanley, 2011b) and constrained novelty search (Liapis et al., 2013, Preuss et al., 2014). NS has also been combined with objective fitness search, both through linear mixing of the novelty and objective fitness values (Smith et al., 2015), and as separate objective in a MOEA (Mouret and Doncieux, 2012). Some of the more commonly used NS variants are discussed in the following sections.

[2] Pugh et al. (2016a) term a phenotypic descriptor as a *behavioural characterisation*.

2.8.3.1 Minimal Criteria Novelty Search

Minimal Criteria Novelty Search (MCNS) was introduced in Lehman and Stanley (2010b) to restrict the exploration potential of NS in domains with large phenotypic landscapes. Through penalising solutions which do not meet a predefined criteria, MCNS directs exploration towards *useful* areas of the phenotypic landscape. In domains where the potential for exploration is high, MCNS has been shown to outperform traditional NS (Lehman and Stanley, 2010b). MCNS is identical to NS, with the addition of a minimal criteria that each individual must meet, $f_{mc}(\rho) : \mathcal{P} \mapsto \mathbb{B}$. Any individual which does not meet this criteria is assigned a novelty score of 0. Therefore, MCNS is defined as in Equation [2.5]:

$$f_{mcns}(p) = \begin{cases} f_{nov}(\rho) & \text{if } f_{mc}(\rho) \ is \ true \\ 0 & \text{if } f_{mc}(\rho) \ is \ false \end{cases} \qquad [2.5]$$

It was noted in Gomes et al. (2012) that, dependent upon the minimal criteria, initial populations may contain few, or no, successful individuals. This would result in (almost) all individuals receiving a fitness score of 0 and evolution would be severely affected, or unable to progress. Therefore, Gomes et al. (2012) introduced progressive MCNS, in which the minimal criteria is made progressively more difficult throughout the evolution, as the fitness of the population increases.

2.8.3.2 Combining Novelty Search with Objective Search

Mouret (2011) highlights two of the major problems eoncountered when optimising through NS as the inspiration for applying a MOO of phenotypic novelty and objective fitness. Firstly, when the search space of the doamin is large, NS either struggles, or requires a exhaustive amount of generations, to locate interesting solutions. The second problem highlighted is that, once an interesting area of the landscape is encountered, NS does not fine tune the solutions because that area of the landscape will not be regarded as novel any longer; therefore NS will continue exploration into unvisited, but potentially less interesting areas of the search space. Mouret (2011) shows that, even with a deceptive fitness function[3] the MOO of novelty and objective fitness produces solutions with higher objective fitness; in this experiment, the exit point has a *optimal* radius. Although NS produces solutions in all trials within the exit radius, only the MOO produces *optimal* solutions.

It is also potentially unnecessary to implement a dedicated Pareto-based MOO of novelty and objective fitness, the simple hybridisation of NS and objective search through linear mixing is a sufficient method to exploit the *best of both worlds.* As previously described in Section 2.8.1, Gomes et al. (2015) shows no significant difference between the MOO of NS using a dedicated multi-objective algorithm and through linearly mixing novelty and objective fitness. However, when applying linear mixing, the ratio of novelty and objective fitness may need to be fine tuned – experiments in different domains report different optimal mixing ratios. Cuccu and Gomez (2011) apply a linearly mixed MOO of novelty and objective search to the *deceptive Tartarus* problem, reporting an optimal ratio of 4/5 novelty to 1/5 objective fitness. In a range of maze domains, Gomes et al. (2015) reports an optimal equal ratio of 1/2 novelty and 1/2 objective fitness. In a preliminary experiment presented within this thesis in Section 3.2, the optimal ratio was discovered to be in the range of 1/6 — 1/3 novelty, dependent upon the phenotypic descriptor used.

2.8.3.3 Novelty Search with Local Competition

A different variation on the standard MOO of NS and objective search is Novelty Search with Local Competition (NS-LC), introduced in Lehman and Stanley (2011b). NS-LC takes inspiration from the local competition observed within different species in nature:

> For example, in the context of a virtual world, a medium-size, medium-mass organism may be more optimal for traveling quickly than a tall, low-mass organism; a global trade-off between fitness (i.e. locomotion speed) and novelty means that most resources will be spent on the medium morphology even though the way that the tall morphology would eventually perform at its best relative to nearby niches may ultimately be just as interesting. (Lehman 2012, p. 138)

As stated above, unlike a standard MOO between global fitness and novelty, the aim in NS-LC is to locate individuals with high objective fitness within *local*

[3] The experimental domain presented in Mouret (2011) is the HARD maze domain, as depicted in Figure 2.2

phenotypic niches. Due to the k-nearest neighbours calculation performed by NS, this is a relatively straightforward addition. As previously defined in Section 2.8 and Equation [2.3], $f_{nov}(\rho)$ is calculated by comparing the distances between ρ and its k-NN in phenotypic space. Formally, we define the k-NN from $\mathcal{P} \bigcup \mathcal{X}$ as the set $\mathcal{X}_{knn}$, which is identical to set used for the measurement of novelty in Section 2.8 and Equation [2.3]. The local competitiveness of an individual, $f_{lc}(\rho)$ is then calculated as the number of individuals with a lower objective fitness than ρ, i. e., the cardinality of the dominated set $\mathcal{D}$, as given in Equation [2.6]:

$$\begin{aligned} \mathcal{D} &= \{x \in \mathcal{X}_{knn} \mid f_{obj}(\rho) > f_{obj}(x)\}, \\ f_{lc}(\rho) &= |\mathcal{D}| \,. \end{aligned} \tag{2.6}$$

NS-LC is then applied through the MOO of $max\ (f_{nov}(\rho), f_{lc}(\rho))$.

2.8.4 Applications of Novelty Search

Alongside wide application to the field of ER, due to its ability to avoid premature convergence through diversity, the application of NS to multiple areas within both EC and ML is an active area of study. Alongside other domains, the algorithm has been applied to data clustering, (Naredo and Trujillo, 2013), the evolution of ANNs (Risi et al., 2009, 2010) and genetic programming Lehman and Stanley (2010a). In Gomes et al. (2014), NS is applied to cooperative co-evolution in order to assist with the avoidance of premature convergence to equilibrium states, a common problem in objective-based co-evolution.

The NS algorithm has also been recently applied to PCG in video games (Preuss et al., 2014) and the evolution of game playing agents. A 2-population variant of NS, named *Feasible-Infeasible Novelty Search* is introduced in Liapis et al. (2013) for the generation of diverse and feasible video game levels. Smith et al. (2015) uses a linear combination of NS and objective search to evolve agents for the video game *Asteroids* (Section 3.2). NS has also been applied to the generation of evolutionary art (Vinhas et al., 2016).

2.8.5 Novelty Search Inspired Algorithms

Alongside the numerous extensions and hybridisations of NS, there are a growing number of IML inspired EAs recently introduced which, although differ functionally from NS, are conceptually inspired by the algorithm. In Stanton and

Clune (2016), the *curiosity search* algorithm is introduced. Taking inspiration from NS, curiosity search adds an intra-life novelty score, which determines the number of different behaviours that an individual has exhibited *during* a trial. Unlike NS, which considers the population dynamics as the intrinsically motivated system, curiosity search promotes exploration of the phenotypic landscape at an individual level. For example, in a maze navigation task with multiple exits, NS would produce a population of controllers, each of which would navigate to a single exit (this is further explored in Chapter 4 and lays the foundations for the presented MADNS algorithm). Curiosity search, however, would produce individuals within the population which would navigate to all of the exits. The fundamental motivation behind curiosity search is that, it has been shown, as prior selection of potential stepping stones towards a more complex task are extremely prone to deception, and it actually beneficial to optimise towards as many stepping stones as possible. As with other iIML inspired ERs, the concept of curiosity differs from the cognitive models developed by Schmidhuber (2013) or Oudeyer et al. (2007a). No actual learning takes place during an individual's lifetime — rather, curiosity is passed on through heredity, the expression of a genome leads to curious behaviour.

Evolvability search is another recent algorithm inspired by NS (Mengistu et al., 2016). In evolvability search, the novelty of an individual is determined through the phenotypic diversity of its direct offspring. The motivation behind evolvability search is to test the hypothesis that directly promoting phenotypic diversity in offspring would produce more variation in the population than indirect methods such as NS. This assumption is shown to be correct in Mengistu et al. (2016). Although evolvability search shows promise, it is a computationally expensive metric, adding further complexity to the already expensive calculation performed by NS.

Surprise search is an EA developed for computational creativity (Gravina et al., 2016). Similarly to NS, surprise search is a divergent, phenotypic exploration based EA. Surprise search is identical to NS, however it aims to optimise divergence from the *predicted* future behaviours of the population (Gravina et al., 2016).

2.9 Quality Diversity and Illumination

Although completely *abandoning objectives* has been shown, through the introduction of NS, to be capable of avoiding deception and to produce diverse sets of strategies, the method has also been found to be ineffective in large solution spaces (Cuccu and Gomez, 2011), and when the measurement of behaviour is unaligned with the fitness landscape (Pugh et al., 2016b). In response to these limitations of NS to produce high performing solutions in certain tasks, various methods for the combination of NS and objective search have been suggested and applied to a range of domains with various levels of success, as previously outlined in Section 2.8.3.1. However, algorithms which involve the MOO of behavioural novelty and objective fitness have generally focussed upon a global fitness assessment, rather than well performing individuals in similar phenotypic niches. NS-LC, outlined in Section 2.8.3.3, differs from the standard methods for combining novelty and objective fitness, in that it considers the local fitness of similarly behaving individuals (Lehman and Stanley, 2011b).

This combination of global phenotypic exploration with local fitness has been the focus of a recently proposed direction within ER. This new direction concentrates on the development of EAs which aim to optimise high performing individuals within phenotypic landscapes; such algorithms tend to focus upon exploration of the phenotypic landscape, whist simultaneously optimising local niches for high objective functionality. The aim of such algorithms is to ignore global fitness, instead focussing on producing a wide range of behavioural niches, where the highest performing solution(s) for each niche is optimised. This class of EAs have been termed as both *quality diversity* algorithms (Pugh et al., 2015), due to their hybrid approach to optimising high quality, diverse populations, and *illumination* algorithms (Mouret and Clune, 2015), referring to their ability of illuminating high performing areas of the phenotypic landscape. The term *illumination*, rather than quality diversity, is adopted Throughout this thesis, and in particular in the introduction of the SHINE algorithm, (Chapter 5).

Due to the study of localised objective optimisation within phenotypic niches being a recent development within ER, there are few methods which have been specifically designed to perform as *illumination* algorithms. However, NS-LC can be described as an illumination algorithm, and formed the inspiration for the development of two algorithms by Jeff Clune and Jean-Baptiste Mouret, the

Multi-Objective Landscape Exploration (MOLE) and MAP-Elites algorithms (Clune et al., 2013, Cully et al., 2015, Mouret and Clune, 2015).

2.9.1 Measuring Illumination

One of the difficulties encountered in measuring the performance o illumination algorithms is that, due to their hybrid phenotypic exploration and local objective functional nature, traditional fitness and diversity measures may not give a good indication of intended performance. Therefore, a range of empirical measurements have been suggested for assessing performance of an illumination algorithm. Pugh et al. (2016b) suggests a *quality diversity*, or QD-score. The method for calculating the QD-score of an illumination algorithm was inspired by the MAP-Elites algorithm, and is logically similar to the function of that algorithm. As in MAP-Elites, the QD-score is calculated by dividing the phenotypic landscape into t discrete bins, represented by the set $\{\mathcal{N}_1, \ldots, \mathcal{N}_t\}$. The QD-score of an evolutionary trial is then obtained by calculating the sum of the maximum fitness achieved in each area of the phenotypic landscape, as defined in Equation [2.7]:

$$\text{QD-Score} = \sum_{i=1}^{t} \max_{\mu \in \mathcal{N}_i} f(\mu) \qquad [2.7]$$

Cully et al. (2015), Mouret and Clune (2015) introduce a more comprehensive set of criteria for the evaluation of illumination algorithms. Their measures, *global performance*, *global reliability*, *precision* and *coverage* aim to give an holistic view of the performance of an illumination algorithm. The four measures suggested by Mouret and Clune (2015) are applied within this thesis to assess the performance of the SHINE algorithm, and are formally outlined in Section 5.5.4.

— 2.10 Multi-Dimensional Archive of Phenotypic Elites —

The MAP-Elites algorithm (Cully et al., 2015, Mouret and Clune, 2015) is the first to be specifically designed as an illumination algorithm. Two algorithms formed the inspiration for MAP-Elites — NS-LC (Lehman and Stanley, 2011b) and the MOLE algorithm, introduced by Clune et al. (2013), which is a MOEA conceptually similar to the MOO of objective fitness and NS (Section 2.8.3.2). Recent studies have shown that, although NS-LC and MOLE both conceptually function as illumination algorithms, MAP-Elites significantly outperforms both in all of the criteria suggested by Mouret and Clune (2015) (Cully et al., 2015, Mouret and Clune, 2015). MAP-Elites is relatively simple to implement; the main procedure of the algorithm is defined in Algorithm 2.

Algorithm 2 MAP-Elites algorithm main procedure (simple, default version).

```
Precondition:
    $\mathcal{X} \leftarrow \varnothing$        ▷ initialise an empty set of representative solutions $\mathcal{X}$
    $\mathcal{O} \leftarrow \varnothing$        ▷ initialise an empty set $\mathcal{O}$ to contain performance

procedure MAP-Elites
    do
        if $iter < G$ then        ▷ first $G$ iterations, generate random solutions
            $\rho' \leftarrow$ RandomSolution()
        else
            $\rho \leftarrow$ RandomSelection($\mathcal{X}$)        ▷ otherwise randomly select from $\mathcal{X}$
            $\rho' \leftarrow$ RandomVariation($\rho$)        ▷ create a randomly modified copy $\rho$'
        end if
        $\mu \leftarrow$ PhenotypicDescriptor($\rho'$)        ▷ simulate $\rho'$, record phenotype $\mu$
        $f \leftarrow$ Performance($\rho'$)        ▷ record performance $f$
        if $\mathcal{O}(\mu)$ is 0 or $\mathcal{O}(\mu) < f$ then
            $\mathcal{O}(\mu) \leftarrow \mathcal{O} \cup f$        ▷ store performance $f$ in $\mathcal{O}$ according to $\mu$
            $\mathcal{X}(\mu) \leftarrow \mathcal{X} \cup \rho'$        ▷ store the solution $\rho'$ in $\mathcal{X}$ according to $\mu$
        end if
        $iter \leftarrow iter + 1$
    while $iter < I$        ▷ repeat for $I$ iterations
end procedure
```

In the original, simple version of the MAP-Elites algorithm (Mouret and Clune, 2015), evolution proceeds by dividing a low-dimensional phenotypic landscape into a matrix of n bins. During evolution, each individual's phenotype is measured and assigned to the relevant bin. If the individual has a higher objective fitness than the current phenotype within the relevant bin, the lower performing individual is replaced. Offspring are generated by assigning an equal probability of selection to each of the elites within the whole phenotypic matrix. MAP-Elites is a recent algorithm, and therefore only few studies exist to empirically validate its performance, however, initial results from the algorithm are highly promising. Since its introduction, MAP-Elites has been applied to the retina problem, soft robot morphologies and a physical robot arm in Mouret and Clune (2015) and produced remarkable results when applied to adaptability in the damage recovery of hexapod robots in Cully et al. (2015).

The MAP-Elites algorithm benefits from its simplicity in implementation, however, there a number of limitations which are noted by the authors (Mouret and Clune, 2015), providing the follow suggestions for future variants of the algorithm:

- Storing more than one individual per feature cell to promote diversity
- Biasing the probability of offspring selection, e. g., biasing towards cells with empty adjacent cells[4]
- Including crossover

All three of these suggestions are incorporated into the SHINE algorithm, introduced in Smith et al. (2016c) and within Chapter 5 of this thesis. SHINE applies an hierarchical structure to the discretised phenotypic landscape, in order to provide a computationally inexpensive method for assigning high fitness to unvisited and sparse areas of the landscape.

2.11 Summary

This chapter has reviewed and discussed the relevant literature to provide a background for the work which is presented in the remainder of this thesis. The research presented in the remaining chapters is developed from the current methods used for the application of IML to EC. The research highlights areas which are currently unexplored in the literature, before presenting a number of algorithms which further develop the field.

In Section 2.8, the NS algorithm was introduced. Although NS has been shown to be an effective method for overcoming deception (Lehman and Stanley, 2011c), it has also been shown that the algorithm may struggle to optimise solutions in non-deceptive domains and large search spaces (Cuccu and Gomez, 2011). The

[4] it is noted by Mouret and Clune (2015) that this method of biasing in preliminary experiments did not increase performance

MOO of NS and objective fitness has been shown to be a suitable method for the optimisation of diverse sets of high performing solutions (Lehman and Stanley, 2011b). However, the application of such techniques beyond maze navigation domains or creature morphologies remains unexplored. Therefore, the suitability for such an approach to a video game domain is assessed in Chapter 3.

In the experiment presented in Section 3.2, game playing agents are optimised to learn diverse strategies for the video game *Asteroids*. Specifically, the experiment applies a linearly-mixed combination of NS and objective fitness, (as introduced in Section 2.8.3.2) with five differing phenotypic descriptors (as described in Section 2.8.2).

Populations optimised through NS may be successful in locating solution(s) to a particular objective, but it may also be the case, due to the high diversity of the resulting populations, that solutions to independent objective(s) are also located. This is a potential characteristic of the algorithm which remains untested in the literature. Therefore, an experiment, presented in Section 3.3, was designed to assess this possible characteristic of NS. It is shown in Section 3.3 that, in fact, when optimising through NS, solutions to multiple independent objectives are located. Resulting from the findings of this experiment, in Chapter 4, a novel extension to the NS algorithm is introduced, MADNS. Variants for both NS and MCNS (Section 2.8.3.1) are tested in a range of maze domains. The MADNS algorithm is shown to assist NS in locating solutions to multiple independent objectives in domains with large search spaces.

In Section 2.9 a recent class of algorithms, termed *illumination* algorithms were introduced. This area of ER is an embryonic, yet promising, direction for the field. The MAP-Elites algorithm is a current state of the art algorithm designed specifically as an illumination algorithm. However, the algorithm suffers from a lack of direct promotion of phenotypic diversity. Through adding a pressure for selection of diverse phenotypes within the algorithm. Through the utilisation of techniques from computational geometry, a method is presented in Chapter 5 which allows for direct pressure in the selection of diverse phenotypes without the computationally expensive calcualtion required by NS. Resulting from this, in Chapter 5, a novel illumination algorithm, SHINE, is introduced. Through applying the measurement criteria highlighted in Section 2.9, the SHINE algorithm is shown to significantly outperform both MAP-Elites and NS-LC.

3 Preliminary Experiments

3.1 Introduction

The training of agents through the optimisation of diverse behavioural characteristics is a recent area of study within ER that shows a promising direction towards the application of algorithms to more complex and uncertain domains (Cully and Mouret, 2013, Doncieux and Mouret, 2012, Mouret and Doncieux, 2012). Initially, a series of preliminary experiments were undertaken in order to further understand the particular characteristics of the NS algorithm and its potential applications.

In complex tasks, and when faced with uncertainty, objectives may be solved in multiple ways, therefore a number of diverse strategies may emerge. In nature, the behavioural sciences observe not only a vast array of different species, but also a scale of behavioural differences within the same species (Martin et al., 1993, Tinbergen, 1963).

In Section 3.2, the NS algorithm is applied to a domain in which the *interestingness* of a controller is of high value, namely within video games. When optimising agents which will perform as NPCs, the ability to exhibit multiple strategies and characteristics of play, whilst also being high performing, is of importance to provide the player with a varied and enjoyable play experience. However, when using a traditional objective fitness approach in EC to optimise NPC agents, there is a tendency for the resulting controllers to exhibit similar strategies for play. Although the evolved strategies may be high performing, they will be of limited enjoyment for players, who will encounter agents which exhibit similar behaviours both throughout a single play of the game and upon replaying multiple times. Therefore, the evolution of high performing agents which exhibit a wide range of strategies is of great benefit to such a domain. The application of IML may therefore potentially be of benefit to the optimisation of video game agents. An experiment was designed that utilised a video game domain in order to address the initial research questions (Section 1.2):

> Can NS be combined with objective fitness in order to produce a range of high performing solutions which also exhibit high phenotypic diversity?
>
> Does the addition of NS have a negative effect upon the objective fitness of the solutions?

When applying NS, it has been also shown that, by varying either the phenotypic descriptor or the distance metric used in the fitness assessment of an evolutionary task, widely different strategies may emerge amongst the population. This suggests the importance of both the definition of the phenotypic descriptor and the distance metric selected when optimising with NS (Mouret and Doncieux, 2012, Pugh et al., 2016b). However, there has been little research which addresses the effect that particular phenotypic definitions play on both objective fitness and the resulting useful strategies of the agents in a complex and uncertain domain. Therefore, an initial experiment was designed to assess a MOO of objective fitness and NS over a number of phenotypic descriptors.

In this chapter, the preliminary experimental findings which underpin the algorithmic developments in this thesis are presented. This chapter is structured as follows. An initial experiment, in which diverse sets of agent controllers for the video game *Asteroids* ore optimised through the linear combination of objective fitness and NS, over a range of phenotypic descriptors, is presented in Section 3.2.

Resulting from the findings of this *Asteroids* experiment, a second study was designed in order to determine the capability of NS to simultaneously optimise multiple independent objectives. In this experiment, which is presented in Section 3.3, the ability of NS to optimise a complete suite of logic gates was assessed. It was shown that, unlike optimising with objective fitness or random search, NS is able to simultaneously optimise *all* logic gates.

Finally, the implications from this preliminary experimental work are discussed in Section 3.4, alongside the reasoning behind the development of an extension to the NS algorithm presented within this thesis.

3.2 Evolving Diverse Strategies in *Asteroids*

3.2.1 Motivation

The combination of objective fitness and phenotypic novelty as an optimisation procedure has been shown to produce multiple solutions to a single objective, for example, in the evolution of diverse sets of virtual creatures (Lehman and Stanley, 2011b). This study investigated whether a similar method could be applied to the optimisation of diverse strategies in more complex and uncertain domains, and what effect altering the phenotypic descriptor would have upon both the objective performance and the diversity of strategies produced. An experiment was designed to assess the effect of varying phenotypic definitions on objective performance at an uncertain task, the classic video game *Asteroids*.

The decision behind the use of a classic video game as a test domain, rather than a more standard domain often used in ER, was as follows. Video games serve as useful testing domains for ML, and as applied in this context, ER (Yannakakis and Togelius, 2015); they often involve spatial navigation and manipulation of environments, under uncertain conditions and with pressure of survival from enemies. Games are often also easily objectively assessed through the number of points scored (Yannakakis and Togelius, 2015). Video game domains have been widely applied to other areas of ML such as RL (Mnih et al., 2013). Although not as commonly applied to ER, their use as test domains has expanded recently, especially in the context of general game playing AI (Hausknecht et al., 2012, 2014).

The video game *Asteroids* (Figure 3.1) was specifically chosen as the domain in this experiment for the following reason. An effective strategy for performing well in the game, which is often employed by expert human players, is to stay in the centre of the screen while rotating and shooting the asteroids, without the use of any thrust. Although perhaps not initially obvious to a human player, this is a common strategy observed when training agents through the use of EAs, as highlighted in Hausknecht et al. (2014). Therefore, it would be expected that, in the case of objective optimisation, this strategy would be frequently observed. Thus, it should also be the case that agents optimised for objective fitness would perform little movement around the playing field, instead remaining static in the centre of the map. The aim of this experiment was to assess whether the inclusion of

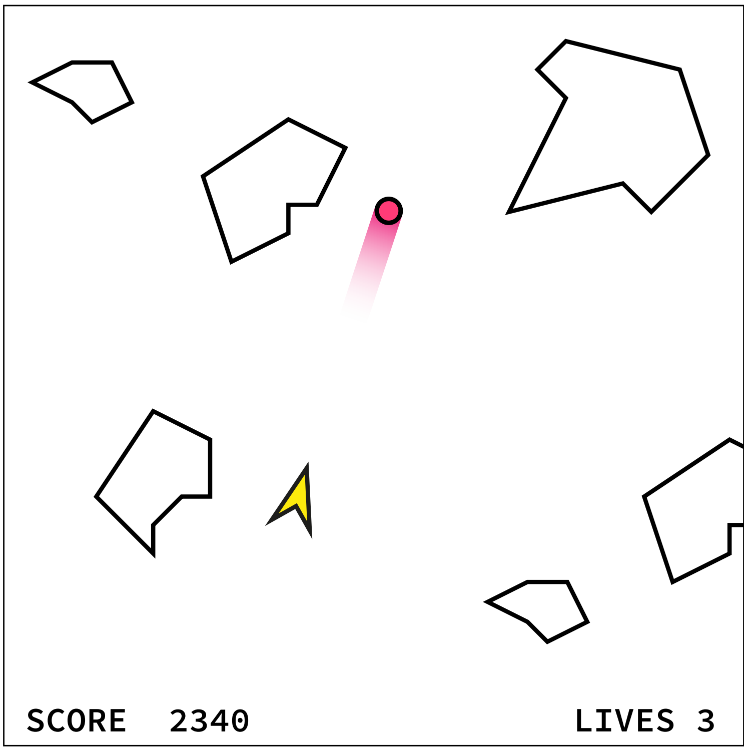


Figure 3.1: *Screenshot of the* Asteroids *video game*

phenotypic novelty as a parameter of the fitness assessment would encourage the optimisation of controllers with high scoring, but also diverse strategies of play.

As shown by Gomes et al. (2015), when combining NS and objective function, there is no significant difference between using a dedicated MOEA and linear combination. However, it has also been shown that the ratio of NS to objective fitness is highly domain dependent (Section 2.8.3.2). It would also be expected that altering the ratio of NS and objective fitness would effect the diversity of the strategies produced, therefore an experiment can be designed which observes the direct relationship between phenotypic novelty and objective function – whether the inclusion of an optimal proportion of novelty may actually increase objective performance, and what effect it has upon the diversity of the successful controllers. Therefore it was decided to linearly combine novelty and objective fitness, across a range of mixing ratios.

In the presented experiment, agent controllers were trained to play the video game *Asteroids*, with fitness assessed using a linear combination of points scored and NS, using a range of mixing ratios. Five different phenotypic descriptors were used to measure phenotypic novelty.

3.2.2 Domain

A bespoke version of the classic arcade game *Asteroids* implemented in the Processing environment for the Java programming language was developed as the experimental domain (Figure 3.1). The aim of *Asteroids* is to score as many points as possible by shooting asteroids and avoiding collisions. The player controls a spaceship that has a left and right rotation, a forward thrust and the ability to shoot. Due to *Asteroids* being set in deep space, the spaceship is not effected by friction or gravity, therefore takes a long time to slow down after thrust is applied. As the name suggests, the enemies in the game are asteroids, which appear on the screen in waves, with random velocity and direction. Asteroids appear in three different sizes: starting off as large, and when hit splitting into two medium size asteroids, which in turn each split into a further two small asteroids. The first wave consists of three large asteroids. After the player has cleared all of the asteroids, the next wave begins, with one more large asteroid than the last. The playing field in *Asteroids* is constructed as a toroidal space, i.e. if asteroids or the ship move off the edge of the screen, they reappear on the opposite side.

Agents were allocated three lives in each round of the game. To remove the possibility of an agent discovering a linear trajectory which avoids collisions with all asteroids, therefore making the game last infinitely long, a timer was added to the task, requiring agents to hit an asteroid every 1000 update loops, or approximately 15 seconds.

3.2.3 Agent Model

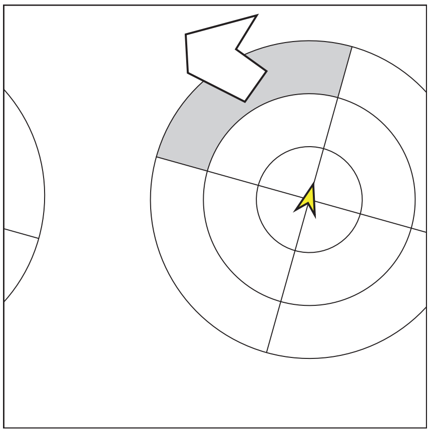


Figure 3.2: *Agent perceptual state map*

The perceptual field for the agent was constructed as a dartboard-style map with binary inputs centred on its position and rotation, providing a discrete representation of polar coordinates relative to the agent (Figure 3.2). Inputs to the map were assigned a value of zero if no asteroids were present within the related coordinate, and one if any number of asteroids appeared within the bounds. A series of trial experiments were conducted using a range of input maps with differing resolutions and sizes. The final perceptual field used consisted of four segments and three slices and a diameter of 0.8 of the world's length. In order to enable the agent to adapt to the toroidal nature of the playing space, it was also decided to allow the agents' perception to extend beyond the edges of the screen, overlapping to the opposite side (Figure 3.2).

3.2.3.1 Controller Network

The NEAT (Stanley and Miikkulainen, 2002) algorithm was used to train the agent controllers throughout this experiment (for an overview of the NEAT algorithm, please refer to Section 2.4.3). The NEAT algorithm has previously been succesfully applied to the evolution of NPCs in real-time video games (Stanley et al., 2005), therefore is a suitable NE method for the optimisation of the agent controllers in this experiment.

The dartboard state map was passed to the inputs of the network as a 12-dimensional binary array (with a value of 1 if at least one asteroid was detected in the position and zero otherwise) at each time step in the game. The networks were assigned three floating point outputs used to control left and right rotation and thrust, and one binary output for shooting.

Figure 3.3 shows the controller topology used in the experiment. The NEAT algorithm's parameters were set to enable the evolution of recurrent nodes within the networks, allowing for the possibility of a short term memory to develop within the controllers.

3.2.4 Objective Assessment

To assess the agents during a trial, the objective fitness was measured using the agent's score at the end of a game, defined as ρ_{score}. The novelty of the agent, which is defined as $f_{nov}(\rho)$, was calculated using the standard NS measurement, as previously defined in Section 2.8 and Equation [2.3]. The phenotypic descriptor, μ, used to measure novelty is dependent upon the assigned condition, selected from the descriptors in Section 3.2.5.

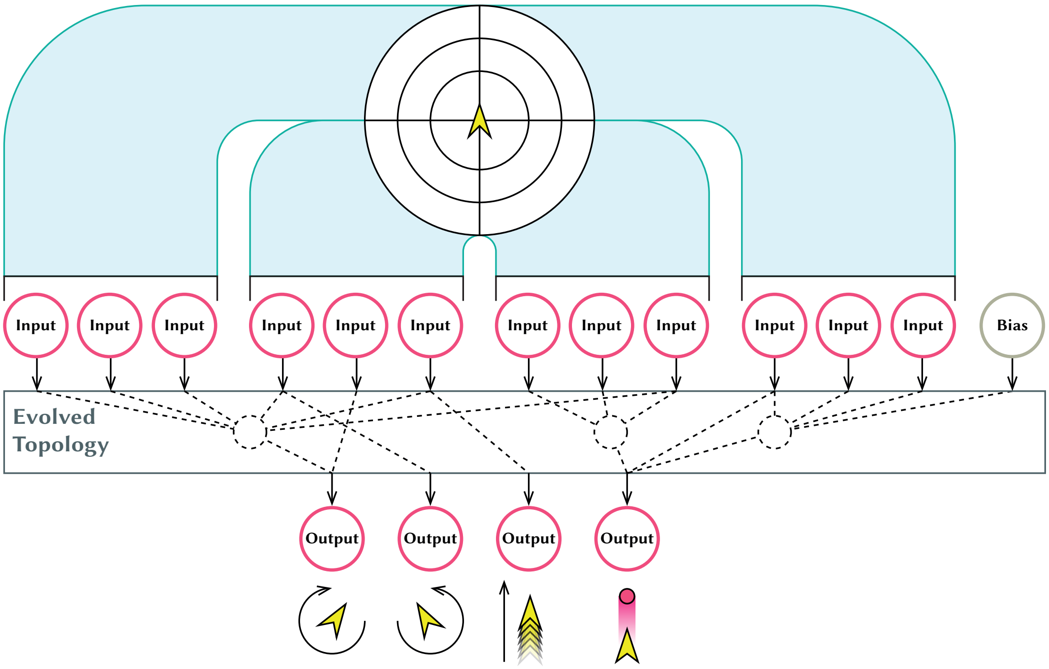


Figure 3.3: Asteroids *agent topology*

Although relatively simple for a human player to grasp, *Asteroids* is a particularly noisy domain. The agents' actions in the game directly effect the trajectory of the task, therefore introducing a level of uncertainty. The same controller can receive a wide range of scores in different games, subsequently effecting fitness measures. Therefore, 10 games were played by each individual in each generation, and the mean objective fitness was then assigned, as given by $f_{obj}(\rho) = \overline{\rho}_{score}$. Due to the computationally exhaustive nature of novelty search over multiple assessments, the behavioural values were evaluated on one random game per generation.

In each of the experiments the fitness of an individual was determined as a linear combination of novelty and objective fitness, with the ratio of each dependent upon a multiplier, $\lambda = \{0, {}^1/_6, \ldots, {}^5/_6, 1\}$. The values assigned for phenotypic novelty $f_{nov}(\rho)$ and points scored $f_{obj}(\rho)$ were normalised against the maximum and minimum scoring individuals in $\mathcal{P}$. The calculation for the normalisation of objective fitness is given in Equation [3.1]:

$$f_{obj}(\rho) = \frac{f_{obj}(\rho) - \min_{x \in \mathcal{P}} f_{obj}(x)}{\max_{x \in \mathcal{P}} f_{obj}(x) - \min_{x \in \mathcal{P}} f_{obj}(x)} \qquad [3.1]$$

This results in the final fitness value $f(\rho),\ [0:1]$, as defined in Equation [3.2]:

$$f(\rho) = \lambda \cdot f_{nov}(\rho) + (1 - \lambda) \cdot f_{obj}(\rho) \qquad [3.2]$$

3.2.5 Phenotypic Descriptors

In the presented experiment, a range of both domain agnostic and domain specific phenotypic definitions, with a varying number of dimensions, were devised. The five phenotypic definitions, as highlighted in Table 3.1, and formally defined in the following sections, were evaluated in the experiment:

μ	Domain **S**pecific/ **A**gnostic	n-dimensions
AC_μ	**A**	100
GC_μ	**S**	1
$\overline{\mathrm{AD}}_\mu$	**S**	1
$\overline{\mathrm{TR}}_\mu$	**S**	2
NA_μ	**S**	100

Table 3.1: *Details of phenotypic descriptors used in* Asteroids *experiment*

Action Count (AC_μ)

In order to calculate AC_μ, the controller network was given a randomly generated set of hypothetical game states. Formally, let us define a game state as a 12-tuple, $S = (s_1, \ldots, s_{12})$ where each $s_i \in \mathbb{B}$. a set of n random game states were produced at the beginning of each generation and presented to all of the controllers in the population before gameplay began. The outputs from each state were observed and the highest value output's index used to generate a string of actions. As each controller network has four outputs (Figure 3.3), let $\mathbb{N}^n_{<4}$ represent the set of output states, where each string has n digits and $\mathbb{N}_{<4}$ represents the integers $\{0, 1, 2, 3\}$. This results in an n-dimensional phenotypic descriptor. Throughout the experiment, the value $n = 100$ (Table 3.1).

Novelty is calculated using the standard NS calculation as defined in Equation [2.3]. Although Gomez (2009) suggests NCD as being a potentially optimal distance metric, due to the computational complexity of both NCD, which requires a compression to be calculated with each distance measurement, and NS, which introduces its own computational load due to the maintenance of an expanding archive, we forgo the NCD measurement, comparing instead the Hamming distance of action vectors, which has previously produced comparably similar results (Gomez, 2009, Mouret and Doncieux, 2012).

The Hamming distance between the phenotypic descriptor strings is used to calculate the novelty of an individual. Given two descriptors, $\mu, x \in \mathbb{N}^n_{<4}$ the Hamming distance may be described as given in Equation [3.3]:

$$dist(\mu, x) = |\{i \in \mathbb{N}_{<4} \,|\, \mu_i \neq x_i\}| \qquad [3.3]$$

As the AC_μ phenotypic descriptor is domain agnostic, and measured against hypothetical game states, when evaluating through NS alone, i. e., $\lambda = 1.0$, the game does not need to be played.

Ground Covered (GC_μ)

The play area, with size $w \times h$ is divided into a 2-dimensional matrix, as shown in Figure 3.4. Formally, let the matrix be defined as M, with size $n \times n$, where the indices of the cells are given by $M_{x,y} : x, y \in \mathbb{N}_{<n}$. At each time step, the agent's position, given by (ρ_x, ρ_y), is used to update the corresponding index within the matrix, where $x = \lfloor n \cdot \frac{\rho_x}{w} \rfloor$, $y = \lfloor n \cdot \frac{\rho_y}{h} \rfloor$. $M_{x,y} = 1$.

At the end of the game, GC_μ is calculated as defined in Equation [3.4].

$$\mathrm{GC}_\mu = \frac{1}{n^2} \cdot \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} M_{x,y} \qquad [3.4]$$

This results in a 1-dimensional phenotypic descriptor, where GC_μ is in the range $[0 : 1]$. Throughout this experiment, $n = 20$.

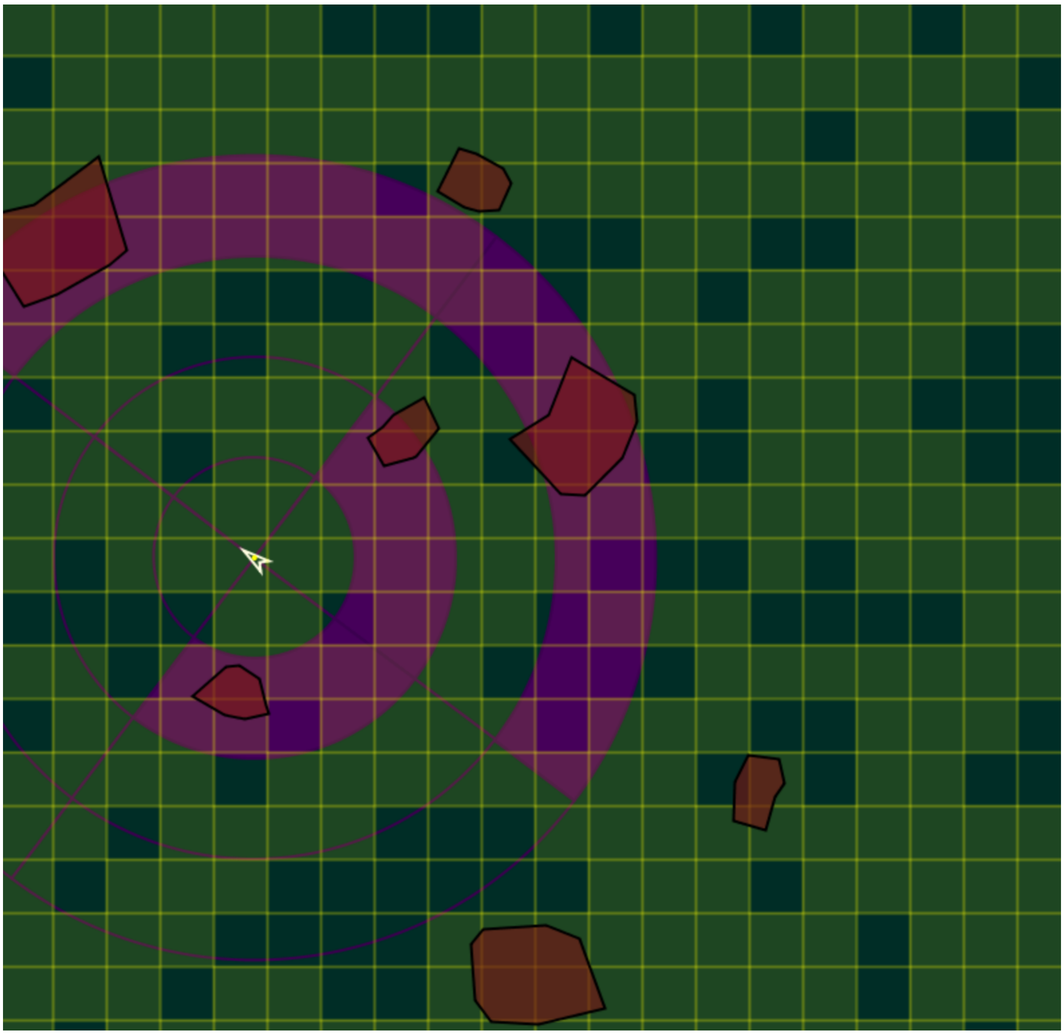


Figure 3.4: GC_{μ} *phenotypic measurement*

Mean Asteroid Distance ($\overline{\mathrm{AD}}_{\mu}$)

The $\overline{\mathrm{AD}}_{\mu}$ phenotypic descriptor is calculated as follows. Let all of the current asteroids be represented by the set A_t. At each time-step during gameplay, the mean Euclidean distance is calculated between the agent's current position, defined as ρ_t, and each asteroid, represented as $a \in A_t$. The maximal distance, i. e., the diagonal length of the domain is represented by $l = \sqrt{w^2 + h^2}$. Thus, the mean asteroid distance at time t is represented by $d_t(\rho_t, A_t)$ and is calculated as in Equation [3.5]:

$$d_t(\rho_t, A_t) = \frac{1}{|A_t|} \cdot \sum_{a \in A_t} \frac{\|\rho_t - a\|}{l} \qquad [3.5]$$

The phenotypic descriptor is then defined as the mean distance during the full trial, with n timesteps, as given in Equation [3.6]

$$\overline{\mathrm{AD}}_\mu = \frac{1}{n} \cdot \sum_{t=1}^{n} d_t(\rho_t, A_t) \tag{3.6}$$

This results in a 1-dimensional phenotypic descriptor, where $\overline{\mathrm{AD}}_\mu$ is in the range $[0:1]$.

Mean Thrust and Rotation ($\overline{\mathrm{TR}}_\mu$)

Let the agent's thrust be represented by ρ_{thrust}. The agent's rotation is defined as ρ_{rot}. At each timestep during the game, the thrust and rotation were measured. At the end of the trial, the mean value of each are calculated. This results in a 2-dimensional phenotypic descriptor, as defined in Equation [3.7]:

$$\overline{\mathrm{TR}}_\mu = (\overline{\rho}_{thrust}, \overline{\rho}_{rot}) \tag{3.7}$$

N-Actions (NA_μ)

The NA_μ phenotypic descriptor is calculated in an identical manner as AC_μ. However, the hypothetical game states are replaced with the first n states of the game. Therefore, NA_μ is a domain specific variant of AC_μ. The NA_μ descriptor is therefore an n-dimensional descriptor. As in AC_μ, $n = 100$ throughout the experiment.

3.2.6 Experimental Parameters

The experiment was performed as follows[1]. For each of the 5 phenotypic descriptors, 40 treatments were performed for each the mixing ratios $\lambda = \{0, 1/6, 2/6, 3/6, 4/6, 5/6, 1\}$. This results in a total of 1400 treatments.

Treatments were restricted to 1000 generations. The average score for 10 games in each generation, previously defined as $f_{obj}(\rho) = \overline{\rho}_{score}$, was recorded. The mean values of the maximum f_{obj} located within the allocated 1000 generations for each treatment is presented in Section 3.2.7.1.

[1] A comprehensive description of the experimental parameters used throughout this experiment is presented in Appendix B.1

3.2.7 Results

In the following sections, the objective performance and the level of diversity for all of the evaluated conditions is discussed.

3.2.7.1 Performance

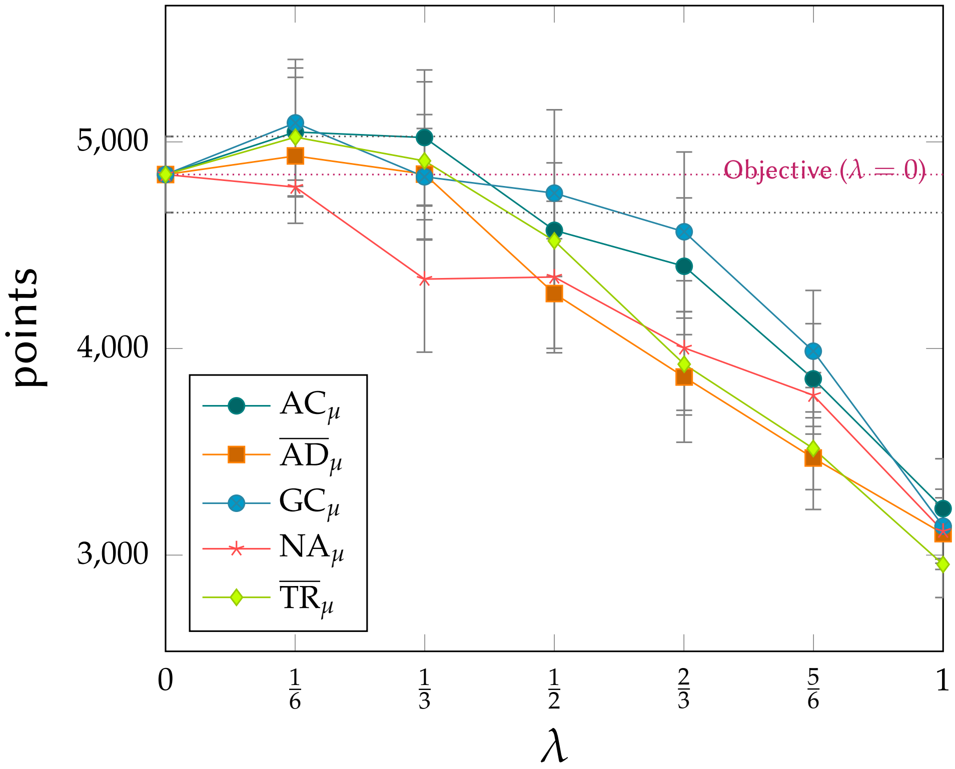


Figure 3.5: *Mean points scored using a linear combination of objective fitness and novelty search*

Figure 3.5 outlines the maximum objective fitness achieved by an agent over 40 evolutionary runs of 1000 generations, for varying mixing ratios of objective fitness and NS, $\lambda = \{0, 1/6, \cdots, 5/6, 1\}$. In each generation of the experiments, the mean score over 10 games was assessed. Our results show that, for mixed objective fitness and NS $(0 < \lambda < 1)$, the phenotypic descriptor has no significant effect upon the objective performance of the NS algorithm. Comparisons of significant difference between two algorithms were calculated throughout using a two-tailed Mann-Whitney U test (Gomes et al., 2015).

There is no difference in performance between phenotypic descriptors for pure objective fitness ($\lambda = 0$, all values = 4842), which is expected, as NS is not included at this value. Unexpectedly, there is no significant difference in points

μ	Median Points
$\lambda = 0$	4842
AC_μ	5092
GC_μ	5023
$\overline{\text{AD}}_\mu$	4932
$\overline{\text{TR}}_\mu$	5048
NA_μ	4781

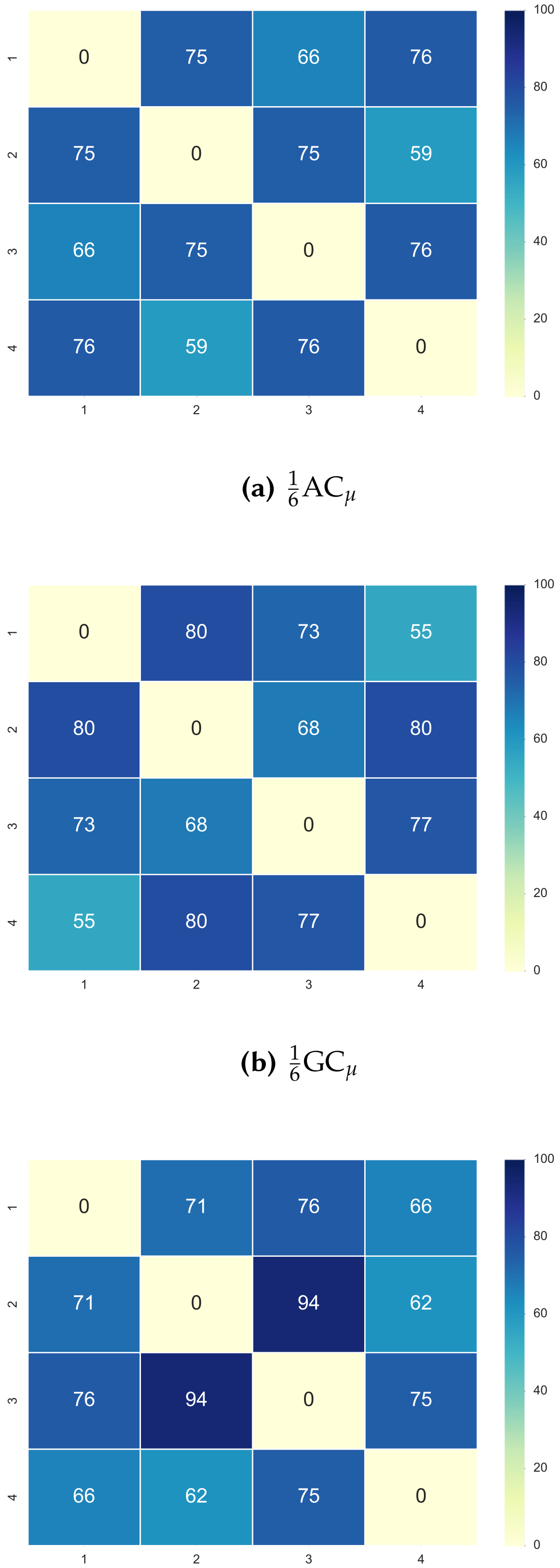
Table 3.2: *Median points scored for* $\frac{1}{6}\lambda$.

scored between phenotypic descriptors for pure NS, ($\lambda = 1$). Although outperforming random search, pure NS ($\lambda = 1$) performs significantly sub-optimally over all tested phenotypic descriptors ($p < 0.001$), compared to objective fitness search (NS = $[2955 : 3225]$, objective fitness = 4842). The linear combination of NS and objective fitness increases the performance of the agents compared with NS ($\lambda < 1$). The results show no significant improvement in objective fitness with the addition of NS. However, of the five tested behaviours, AC_μ, $\overline{\text{AD}}_\mu$, GC_μ and $\overline{\text{TR}}_\mu$ produced median fitnesses which outperformed the median objective fitness for $\frac{1}{6}\lambda$, with only NA_μ under performing (median $= 4842$, $\text{AC}_\mu = 5092$, $\overline{\text{AD}}_\mu = 4932$, $\text{GC}_\mu = 5023$, $\overline{\text{TR}}_\mu = 5048$, $\text{NA}_\mu = 4781$) (Table 3.2). Of all tested behaviours, AC_μ with a mixing ratio of $\frac{1}{6}\lambda$ produced the most successful results ($\text{AC}_\mu\ \frac{1}{6}\lambda = 5092$).

The ideal mixing ratio of NS and objective fitness remained relatively consistent throughout the experiments. A small ratio of NS to objective fitness ($\frac{1}{6}\lambda$) produced the highest results for GC_μ, $\overline{\text{TR}}_\mu$, $\overline{\text{AD}}_\mu$ and NA_μ. AC_μ, however, produced comparable results for both $\frac{1}{6}$ and $\frac{2}{6}$ λ ($\frac{1}{6}\lambda = 5048$, $\frac{2}{6}\lambda = 5021$).

3.2.7.2 Diversity

Due to the tested domain being an interactive video game, good progress requires the agent to constantly alter the trajectory of play (i.e. by shooting asteroids). This makes the assessment of play strategies a difficult task. It was therefore decided to compare the full state-action parings for agent controllers to indicate the distance of actions between agents.

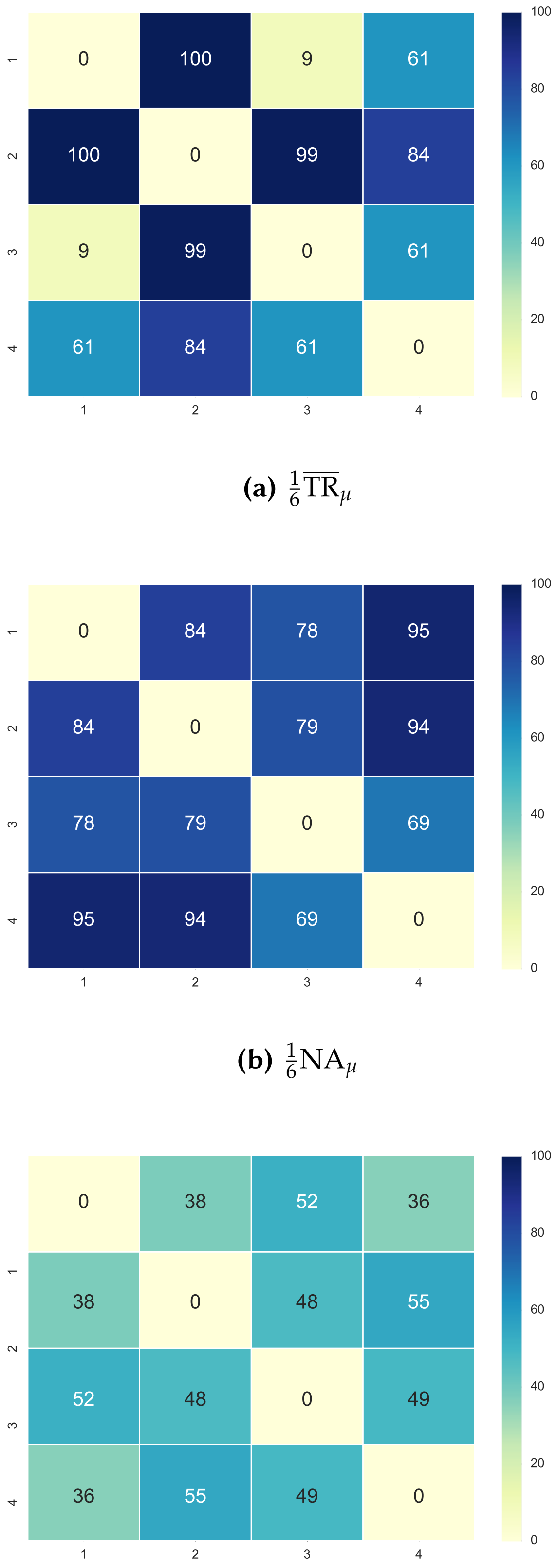


(a) $\frac{1}{6}\mathrm{AC}_{\mu}$

(b) $\frac{1}{6}\mathrm{GC}_{\mu}$

(c) $\frac{1}{6}\overline{\mathrm{AD}}_{\mu}$

Figure 3.6: *Action set distances between high performing controllers in* Asteroids



(a) $\frac{1}{6}\overline{\mathrm{TR}}_\mu$

(b) $\frac{1}{6}\mathrm{NA}_\mu$

(c) $\lambda = 0$

Figure 3.7: *Action set distances between high performing controllers in* Asteroids

In order to assess the diversity of high performing strategies produced within a single evolutionary run, the four highest scoring individuals were stored over the course of one training cycle of 1000 generations for each phenotypic type, using the optimally combined phenotypic-objective ratio ($\frac{1}{6}\lambda$) alongside a separate, purely objective fitness run ($\lambda = 0$). The four highest scoring agents were selected as, when optimising for application to a video game, low scoring agents would be of little use. Therefore, although the phenotypic diversity of lower scoring agents may be potentially higher, they are of no use as game playing agwents.

The actions performed for all combinations of the input state space were compared for these high scoring agents in order to establish the diversity of controller networks produced by the addition of NS. The state input map used in the trials consisted of a 3×4 two dimensional binary input matrix, giving $2^{12} = 4096$ possible state combinations. Each controller network was presented with all of the possible input states. The highest valued output was recorded for each state, and the resulting values converted into a string of length 4096, where each character represents the winning action, e.g. "ULS" $= \{up, left, shoot\}$. The Hamming distances for each action string between agents within the same phenotypic definitions were compared as a percentage

Figures 3.6 and 3.7 show the results of the pairwise comparison between the four highest scoring individuals in each phenotypic descriptor. For example, in Figure 3.6b the outputs for agent 1 have a distance of 80, 73 and 55 from the ouptuts of agents 2, 3 & 4 respectively.

All Agents across all of the phenotypic definitions (Figures 3.6a to 3.6c, 3.7a and 3.7b), with the exception of agents one and three in the Mean Thrust and Rotation ($\overline{\mathrm{TR}}_{\mu}$) phenotypic type (Figure 3.7a), produce state-action parings with equal or higher distances than the objectively trained agents (Figure 3.7c). This strongly indicates a more diverse set of actions for input states within singular evolutionary runs.

3.2.8 Discussion

As shown in Section 3.2.7, training through a combination of objective function and NS is a viable method to produce controllers which are not only high scoring, but also adopt multiple strategies. Alongside this, varying the particular phenotypic definition or metric used may further increase the diversity of strategies adopted. The results show that training through a linear combination of objective fitness and NS with multiple phenotypic definitions is a viable method to produce a range of useful controllers which adopt a diverse range of strategies. A linearly mixed fitness assessment is shown to produce controllers which adopt different strategies of play than controllers trained through standard objective fitness without significant effect on objective performance. The results show that, dependent on an ideal mixing ratio, the linear combination of objective fitness and NS produces diverse, high-scoring solution populations.

Although a linear mixing of objective and NS was specifically chosen in order to assess the direct impact that NS has upon objective fitness, algorithms such as MCNS or a dedicated MOO algorithm, such as NSGA-II (Deb, 1999), may produce similar results without the need for an optimal mixing ratio.

Also, due to the nature of the tested domain, in which agents have the ability to alter the environment and thus the trajectory of play, empirical assessment of the diversity of strategies produced by agents is a difficult task. In this experiment, we chose to quantitatively assess the distance between full state space to action mappings, however more qualitative methods developed in the behavioural sciences may become more applicable to uncertain and changeable domains (Martin et al., 1993). Assessment criteria could be established, for example, through the qualitative human assessment of videos of the agents playing the game.

As shown in the results for points scored (Figure 3.5), when objective fitness is completely ignored, i. e., $\lambda = 0$, the performance may be severely affected. However, although some of the solutions which emerge through optimisation with pure NS are not directly useful to the domain at hand, it does not necessarily follow that these strategies are without use for all domains. An interesting direction to further extend studies analysing the diversity of strategies produced through combined objective fitness and NS, could assess the transferability of trained agents or populations into either different domains, or their adaptability in domains which alter over time.

Further to this, although it was shown that evolution through NS may produce sub-optimal solutions, this may be dependent upon the objective. For example, in the *Asteroids* domain, the objective assessed was the number of points scored. However, if the objective was changed to the amount of the domain covered, NS may have performed more successfully than objective fitness. As the NS algorithm's aim is to produce a diversity of potential solutions through *abandoning objectives*, it may be the case that an evolutionary trajectory optimised through NS is locating solutions to multiple differing objectives which go unobserved. This is a potentially interesting characteristic of the NS algorithm, which is previously unexplored in the literature. Therefore, a simple experiment was designed with the aim of establishing whether a population optimised through pure NS would, in fact, produce solutions to multiple different objectives. The experiment, which involves the simultaneous optimisation of a complete set of logic gate networks, is presented in Section 3.3.

3.3 Experiment 2: Conflicting Logic Gates

3.3.1 Motivation

As indicated in the previous experiment, although it was shown that optimising through pure NS can lead to suboptimal results, it is not necessarily the case that these controllers are suboptimal at *any* objective. When optimising with NS, the exploration of the phenotypic landscape is directed, i. e., it differs from random sampling, and is a focussed, divergent optimisation procedure. Therefore, if a controller is encountered which is good at one particular objective, a highly *novel* controller in subsequent generations would be one which exhibited the opposite phenotype. For example, if an evading behaviour was encountered, then a pursuit behaviour would be deemed highly novel if encountered afterwards. It may follow that, when a population is optimised with NS, it actually directs the search towards multiple independent objectives. A simple experiment was therefore designed to test this hypothesis.

3.3.2 Domain

The experimental domain is a simple logic gates experiment. In this experiment, the controllers are ANNs which are optimised to function as logic gates. The networks are given two truth values as inputs, and they must output the relevant logical truth for the complimentary inputs. This domain was selected for its simplicity, although the task is easily solved, for logic gates such as the XOR gate, it requires the development of hidden neurons within the ANN. Alongside this, with the fitness assessment used in this experiment, achieving a perfect score is unlikely through random sampling (Stanley and Miikkulainen, 2002). Therefore this simple domain was utilised to establish whether NS could be applied to optimise a complete suite of logic gate controllers.

3.3.3 Network Topology

As in Section 3.2.3.1, the NEAT algorithm was used to optimise the logic gate ANNs throughout this experiment. The ANNs evolved each have two inputs and one output, as illustrated in Figure 3.8.

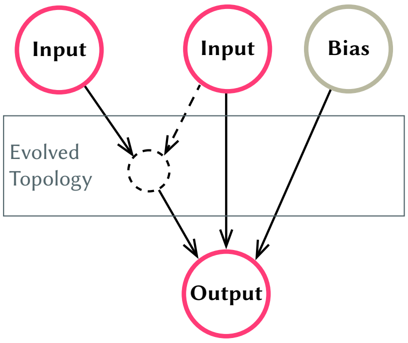


Figure 3.8: *Structure of logic gate network*

3.3.4 Objective Fitness and Phenotypic Descriptor

The ANNs optimised in this experiment were set to utilise a *signed sigmoid* activation function. These activation functions output values in the range $[-1 : 1]$. Therefore, the binary truth values were assigned as **true** $= 1$ and **false** $= -1$. In each trial, the inputs of each network were queried with the four possible truth combinations, resulting in the series of input values: $(1, 1), (1, -1), (-1, 1), (-1, -1)$. The output from the network was recorded for each input pair, resulting in a 4-tuple, $R = (r_1, r_2, r_3, r_4)$, where each r_n falls within the range $[-1 : 1]$. This output vector is used for both the objective fitness assessment, $f(R)$ and the phenotypic descriptor, Logical Truths (LT$_\mu$) $= R$.

INPUT		**OUTPUT**					
1	**2**	**OR**	**AND**	**NOR**	**XOR**	**XNOR**	**NAND**
T	T	T	T	F	F	T	F
T	F	T	F	F	T	F	T
F	T	T	F	F	T	F	T
F	F	F	F	T	F	T	T

Table 3.3: *Truth table for binary logical operators*

The optimal output values for the relevant logic gate to be optimised are determined from the truth values given in Table 3.3. For example, the optimal outputs of an XOR logic gate would be the values $(1, -1, -1, 1)$. These optimal outputs are defined as (t_1, t_2, t_3, t_4).

The fitness of a solution, $f(R)$, is calculated as the Sum Squared Error (SSE) of the network output values and the optimal output values. This value is subtracted from the maximum possible error, which is 16. Therefore the fitness of the result set R is defined as given in Equation [3.8].

$$f(R) = 16 - \sum_{i=1}^{4}(t_i - r_i)^2 \qquad [3.8]$$

3.3.5 Results

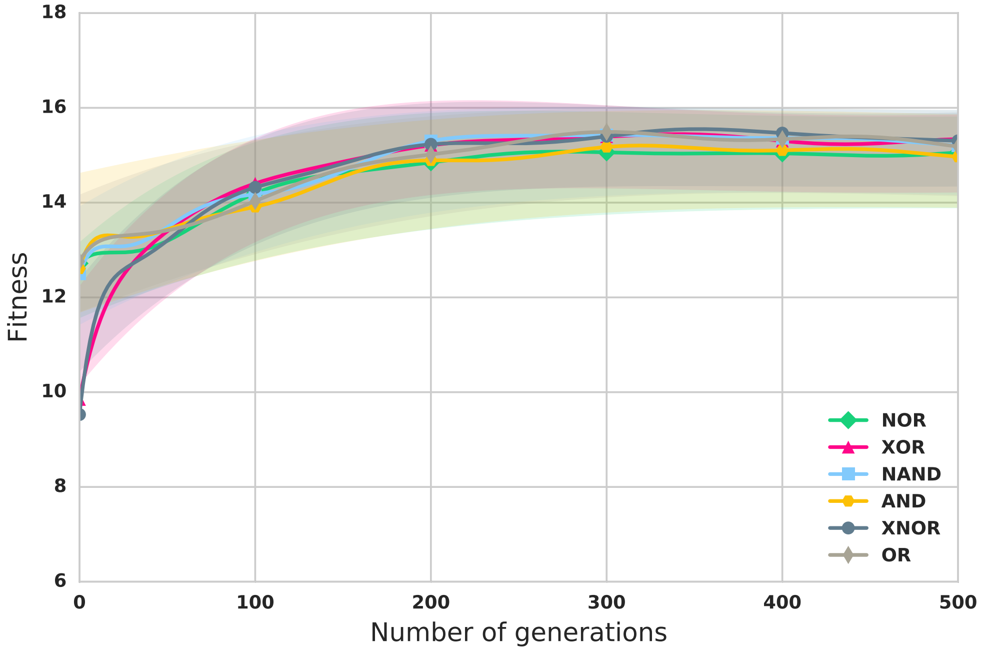


Figure 3.9: *Fitness for evolution of logic gates with NS*

Two objective fitness assessments were conducted, the first to optimise an XOR logic gate and the second, an XNOR logic gate. Each trial was run for 500 generations, with a population of 100 NEAT networks. 20 replicates of each treatment were performed with differing random seed values. Identical seed values were given to each treatment in each replicate. As is expected, the objective fitness assessment produces high scoring solutions for the respective logic gate, i. e., XOR fitness produces optimal solutions for the XOR logic gate. As an optimal XNOR logic gate in the XOR trial would, by definition, score a fitness value of 0, the trials do not manage to produce solutions to the opposite logic gate. A third treatment was conducted using NS to replace the objective function. As previously, each trial was run for 500 generations, with 20 replicates performed. As is shown in Figure 3.9, when optimising with NS, the evolution manages to optimise solutions to *all* of the logic gates simultaneously.

3.3.6 Discussion

The logic gates experiment serves as a simple illustration that the optimisation of diversity in a population may lead to solutions to multiple independent objectives. Due to the absence of any objective fitness, solutions are not penalised for being *different* to any specific objective, rather they are promoted as being highly novel. The results presented here, are insufficient to draw any quantifiable conclusions, but are suitable to indicate the possibility for an application of NS for the optimisation of populations which contain solutions to multiple independent objectives.

3.4 Implications

The preliminary experiments presented in this chapter were undertaken to further assess the characteristics of the NS algorithm. In Section 3.2 a MOO was applied to the optimisation of controllers in the video game *Asteroids* using a linear combination of objective fitness and NS. The results from this experiment showed that, even in complex domains, if the fitness function does not suffer from deception, the addition of NS does not significantly increase objective fitness. Secondly, the results showed that optimising without objective fitness, i. e., using NS alone produces suboptimal controllers. However, the combination of objective fitness and NS, dependent upon an ideal mixing ratio (in this domain $\frac{1}{6}$ was optimal), produces more diverse controllers without significant effect upon objective performance. Therefore, when the aim is produce diverse sets of high performing controllers, such as for NPCs in video games, the MOO of objective fitness and NS is shown to be a viable method.

This experiment highlights the potential application of the combination of NS and objective fitness in domains where a diversity of behaviours are required alongside high objective fitness. A Video Game was specifically chosen in this experiment to highlight one potential domain where such a technique may be of use. However, in the presented experiment, the agents were optimsed to act as the player of the video game, due to the ease in which an objective fitness function (i. e., points scored) could be applied. The optimisation of NPCs, such as enemies or companion characters for the player would be an extremely interesting application of this approach, with the potential to produce diverse sets of enemies with a range of behaviours However, the design of an appropriate objective fitness function to combine with NS may be a difficult task.

Next, a simple logic gate experiment was undertaken in Section 3.3. In this experiment, two objective fitness functions were applied to optimise XOR and XNOR logic gates. As expected, both fitness functions successfully managed to optimise their complimentary logic gate. Also as expected, the conflicting logic gate was not located when optimising with objective fitness. However, it was shown that NS manages to optimise the *full* spectrum of logic gates simultaneously. This is perhaps unsurprising, due to the nature of NS, however it serves to illustrate the effectiveness of NS to simultaneously optimise objectives.

Therefore, it was decided to assess whether this property of the NS algorithm could be suitably applied to a more traditionally applied domain in ER, a maze navigation task. In the following chapter, NS is applied to a series of maze navigation domains, each with multiple potential exit points. The aim of the experiment is to simultaneously optimise controllers for each of the exits independently. Although it is shown that NS is capable of the simultaneous optimisation of multiple independent objectives, as the domain size increases, NS struggles. In light of this, a novel extension to NS was developed to assist with the simultaneous optimisation of multiple independent objectives. In Chapter 4 we present MADNS, an extension to the NS algorithm developed to assist NS with the simultaneous optimisation of solutions to multiple objectives. Variants for both NS and MCNS are presented.

4 Multiple Assessment Directed Novelty Search

4.1 Introduction

In Chapter 3 an experiment was undertaken in which a MOO method was applied for the evolution of diverse sets of strategies to a single objective, through the linear combination of NS and objective fitness. It was shown that optimising for pure novelty without any regard for objective functionality will likely result in a large amount of solutions without any use. However, stating that a solution is without use to a particular objective is not the same as stating it is without use to *any* objective. As was shown in the preliminary logic gates experiment, presented in Section 3.3, the solutions produced may, in fact, be useful to unrelated objectives which have gone unobserved. With this thought in mind, this chapter focusses on utilising the diversity of solutions produced in an evolutionary trajectory optimised with NS to simultaneously locate solutions to multiple independent objectives.

Evolution through NS is a directed exploratory search process, much different to random search (Lehman and Stanley, 2008). When optimising with NS, the evolution will aim to constantly produce *novel* behaviours; phenotypes previously not encountered will be highly ranked in the population. NS performs search within the phenotypic landscape. Although this search space does not directly map to the solution space, dependent upon the phenotypic descriptor used, a novel phenotype may indicate a potentially *useful* behaviour for *some* objective.

This is more directly the case when the phenotypic descriptor used is *aligned* with the solution space. Pugh et al. (2016b) shows that *alignment* of the phenotypic descriptor to the fitness landscape is important to producing solutions with high objective functionality when applying NS. A phenotypic descriptor is said to be aligned with the solution space if the information that the behaviour provides is easily mapped to objective fitness. This can be highlighted through the maze domain example, widely used in the application of NS.

In a traditional maze domain, the fitness is assessed by the Euclidean distance from the ending point of an agent to the exit. The most commonly used phenotypic descriptor in assessing NS is the MEP_{μ} descriptor, i. e., the ending position of the agent. Irrespective of the exit, the MEP_{μ} descriptor gives useful information about the performance of an agent. All of the possible paths that an agent may take to navigate towards a single point in the maze are conflated into the same phenotype; this allows a multitude of behaviours to be expressed as being identical. For example, all controllers which are unable to navigate, either via spinning on the spot, or for any other potential reason, will be given an identical phenotypic value (the starting x, y coordinates of the agent). It would be expected that the starting position of the agent will be a highly located phenotype early on, due to the high probability of evolution producing an agent unable to navigate. Therefore, all controllers without the ability to navigate will be given a low novelty score throughout the evolution. NS will instead focus search upon controllers which can navigate as far from the starting position as possible. In this sense, a novel solution is one which navigates far from the starting position, in the most sparsely visited areas of the maze. As evolution progresses with NS, it would be expected to locate controllers with the ability to navigate to *all* possible areas of the maze, and thus the exit, wherever that exit may happen to be placed. It may be the case that, due to this focussed divergent nature, evolution with NS will produce solutions to multiple independent objectives.

In this chapter, the hypothesis that a population optimised with NS will locate solutions to a number of independent objectives, is tested. Alongside this, to assist with the simultaneous optimisation of multiple independent objectives, Multiple Assessment Directed Novelty Search (MADNS) is introduced, a novel extension to NS, which exploits this potentially useful characteristic. NS alone may be sufficient to locate a range of solutions to independent objectives, however, it has been shown that NS struggles to locate solutions in large phenotypic landscapes. MCNS was introduced in Lehman and Stanley (2010b) to reduce this problem somewhat. However, it is shown that, MCNS still lacks the necessary pressure of exploitation to focus search towards areas of interest in large state spaces. Therefore, the MADNS algorithm is also applied as an extension to MCNS, as MCMADNS.

4.2 Approach

The approach taken to develop the MADNS algorithm, and the MCMADNS variant, is described in the following sections.

4.2.1 Optimising Independent Objectives

If a population is optimised to maximise phenotypic diversity, and the phenotypic descriptor is aligned to provide enough information about the type of behaviour which is useful to the domain, then it should be expected that the evolution will uncover solutions to multiple, independent objectives within this domain. Given the maze domain, for example, there is a high likelihood of an agent locating an exit placed at any location within the maze (although some locations may be more difficult to navigate towards). Therefore, if the aim is to produce an agent controller which finds the exit at the top left of the maze, NS may be successful. However, it is also likely for NS to produce an agent which navigates to an exit at the bottom right. Figure 4.1 illustrates all of the ending points located by controllers optimised through NS in the HARD maze domain. As can be seen by the solutions produced, there are potentially many independent objectives optimised when evolving through novelty alone.

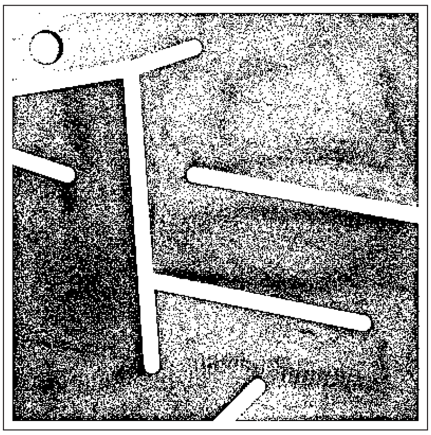


Figure 4.1: *Ending positions of agents evolved with NS in the HARD maze domain.*

4.2.2 Observation rather than Optimisation

The point illustrated here is that *it is through observation, rather than optimisation* that solutions are located in a population evolved with NS. This is a potentially advantageous characteristic. For example, if we were interested in discovering the possible locations that a particular robot controller was capable of navigating towards, applying NS and observing the solutions to multiple different objectives may save restarting countless evolutionary trials, each with a different objective function.

As was shown in the preliminary logic gates experiment, presented in Section 3.3, NS is capable of optimising towards multiple independent objectives unassisted in domains with small phenotypic landscapes. However, as the size of the search space, and thus the potential for exploration increases, NS creates an evolutionary trajectory which is spread sparsely across the landscape. Although this is hardly surprising, the result is that a particular objective that we would wish to locate may never be reached when optimising through novelty alone. Therefore, a method is proposed for directing the divergent search towards areas of the landscape which are of interest, without penalising the overall divergent nature of the search.

It is this characteristic of NS which we exploit to develop an extension to the NS algorithm, MADNS. MADNS utilises the divergent search procedure of NS to direct the population towards multiple independent objectives. The method applied in the MADNS algorithm allows for independent objectives to be optimised simultaneously, without one particular objective effecting another. A series of maze domain experiments with multiple exits and with varying levels of exploration potential are presented to determine the effectiveness of the proposed MADNS extension.

As the formal definitions of the algorithms presented within this section are all based upon the NS and MCNS algorithms, a brief reintroduction to both is provided below.

4.2.3 Novelty Search

As previously introduced in Section 2.8.1, the novelty of an individual is defined as the average distance between a measured phenotype of the individual μ and its k-NNs in phenotypic space. The set of phenotypic descriptors measured include both the behaviours of the current population and an archive of previously discovered novel behaviours. The assignment of novelty to an individual is thus given as in Equation [4.1].

$$f_{nov}(\rho) = \frac{1}{k}\sum_{i=1}^{k} dist(\mu, x_i) \tag{4.1}$$

4.2.4 Minimal Criteria Novelty Search

As previously defined in Section 2.8.3.1, MCNS is identical to NS, with the addition of a minimal criteria that each individual must meet, $f_{mc}(\rho) : \mathcal{P} \mapsto \mathbb{B}$. Any individual which does not meet this criteria is assigned a novelty score of 0. Therefore, MCNS is defined as in Equation [4.2]:

$$f_{mcns}(p) = \begin{cases} f_{nov}(\rho) & \text{if } f_{mc}(\rho) \; is \; true \\ 0 & \text{if } f_{mc}(\rho) \; is \; false \end{cases} \tag{4.2}$$

4.2.5 Maximum Objective Value

A simple linear combination of fitness values is unable to be extended to multiple independent objectives. However, assigning the solutions' maximum fitness value for all of the objectives may be a viable method to direct novelty towards objectives of interest. Given a particular solution in a population, $\rho \in \mathcal{P}$, and a domain with k objectives of interest defined by the functions $f_1(\rho), f_2(\rho), \ldots, f_k(\rho)$, where $f_k : \mathcal{P} \mapsto \mathbb{R}$, the maximum objective function is defined as $f_{max}(\rho)$ (Equation [4.3]).

$$f_{max}(\rho) = \max_{1 \le i \le k} \{f_i(\rho)\}. \tag{4.3}$$

The linear combination of $f_{max}(\rho)$ with NS is defined as $f_{ns-max}(\rho)$:

$$f_{ns-max}(\rho) = \frac{f_{nov}(\rho) + f_{max}(\rho)}{2}. \tag{4.4}$$

Both $f_{max}(\rho)$ (OBJ-MAX) and $f_{ns-max}(\rho)$ (NS-MAX) are assessed in Sections 4.4 and 4.5.

4.3 The Algorithm

MADNS is an extension to the NS algorithm that rewards high performing solutions over a number of predefined objectives. MADNS is identical to traditional NS, with the addition of a set of *assessments* — traditional objective fitness functions. The mechanism for fitness assignment in MADNS does not involve comparison between any of the assessments, which allows the functions to be partially or directly conflicting without negative effect on any particular assessment.

Specifically, the MADNS fitness assignment, defined in Equation [4.5], retains the novelty score for all members of the population other than the current highest ranking solutions for each of the assessments. These highest ranking solutions are assigned a score equal to the most novel solution in the current population.

Formally, let the set P denote the current population, with an individual solution defined as $\rho \in \mathcal{P}$. Next, for a given domain with k objectives, defined by the functions $a_1(\rho), a_2(\rho), \ldots, a_k(\rho)$, where $a_k : \mathcal{P} \mapsto \mathbb{R}$, let the set $A = \{a_1(\rho),\, a_2(\rho),\, \ldots,\, a_k(\rho)\}$. Let the subset $Q \subset \mathcal{P}$ contain the maximal solutions for each objective, where:

$$Q = \arg\max_{\rho\in\mathcal{P}} a_1(\rho) \bigcup \arg\max_{\rho\in\mathcal{P}} a_2(\rho) \bigcup \ldots \bigcup \arg\max_{\rho\in\mathcal{P}} a_k(\rho).$$

Let $f_{nov}(\rho)$, as defined in Equation [4.1], be the novelty of a solution and the maximal novelty value for the current population be defined as $\underset{\rho\in\mathcal{P}}{max}\ f_{nov}(\rho)$. Finally, let $f_{mad}(\rho)$ be the fitness of an individual solution, calculated as in Equation [4.5]:

$$f_{mad}(\rho) = \begin{cases} f_{nov}(\rho) & \text{if } \rho \notin Q, \\ \underset{\rho\in\mathcal{P}}{max}\ f_{nov}(\rho) & \text{if } \rho \in Q. \end{cases} \qquad [4.5]$$

A minimal criteria variant of this algorithm, MCMADNS, may be defined through the replacement of NS with MCNS (Equation [4.2]):

$$f_{mc-mad}(\rho) = \begin{cases} f_{mcns}(\rho) & \text{if } \rho \notin Q, \\ \underset{\rho\in\mathcal{P}}{max}\ f_{nov}(\rho) & \text{if } \rho \in Q. \end{cases} \qquad [4.6]$$

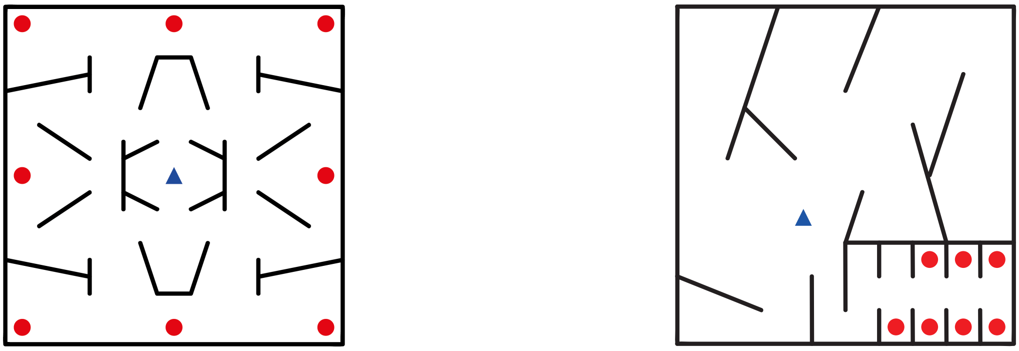
4.4 Experiment 1: Multiple Exit Maze Domains

An experiment was designed to assess the ability of NS to simultaneously optimise solutions to multiple independent objectives. The methods proposed in Sections 4.2 and 4.3 for directing NS towards independent objectives were tested in a range of maze domains with varying levels of *exploration potential.* An initial series of maze domains are presented in Section 4.4.1. Further testing was conducted in a maze domain with an increasing size, alongside an unbounded variant. The results of this secondary experiment are described in Section 4.5. The work from both experiments was presented at the *Genetic and Evolutionary Computation Conference* (GECCO) 2016, in the papers Smith et al. (2016a,b).

4.4.1 Domain

The experimental domain is based upon previous studies which have assessed NS and variants of the algorithm (Gomes et al., 2015, Lehman and Stanley, 2008, 2011a). The task domain is a simulated maze, in which an agent controller must navigate from an initial starting-point to one of a possible number of exit points within a fixed time limit. Both bounded and unbounded variants of the maze domains were tested.

Bounded Domains



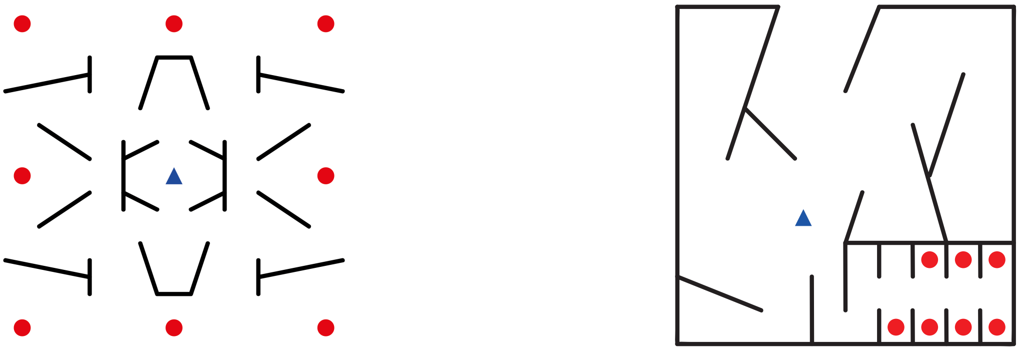
(a) *DIVERGE domain* **(b)** *SUBSET domain*

Figure 4.2: *Bounded maze domains*

DIVERGE Divergent maze with 8 exits in opposite directions. Providing solutions to all exits requires large levels of diversity within the evolutionary trajectory (Figure 4.2a).

SUBSET A maze with 7 exits in a small region of the possible space ($\approx 25\%$). As the starting direction of the agent is is the opposite direction to all of the exits, this maze requires phenotypic search to explore away from the initial direction of exploration (Figure 4.2b).

Unbounded Domains



(a) *DIVERGE-U domain* **(b)** *SUBSET-U domain*

Figure 4.3: *Unbounded maze domains*

DIVERGE-U The same maze as DIVERGE with unbounded exploration potential (Figure 4.3a).

SUBSET-U Identical to SUBSET with a small open section leading to an area with unbounded exploration potential (Figure 4.3b).

4.4.2 Agent Model

The agent controllers in the maze experiments are neural networks, evolved using the NEAT algorithm (Stanley and Miikkulainen, 2002). In a similar manner to previous maze navigation experiments (Gomes et al., 2015, Lehman and Stanley, 2008, 2011a), the agent's perception (Figure 4.4) utilises six range-finding sensors, which return the distance to the nearest obstacle. The range-finding sensors are mapped to $[0 : 1]$, where 1 is equivalent to the diagonal distance of the full maze (Figure 4.5).

In addition to this, the agent has four inputs which act as pie-slice sensors. The pie-slice which contains the line between the agent and the exit is activated with a value of 1. For multiple exit mazes, the closest exit from the agent at any particular

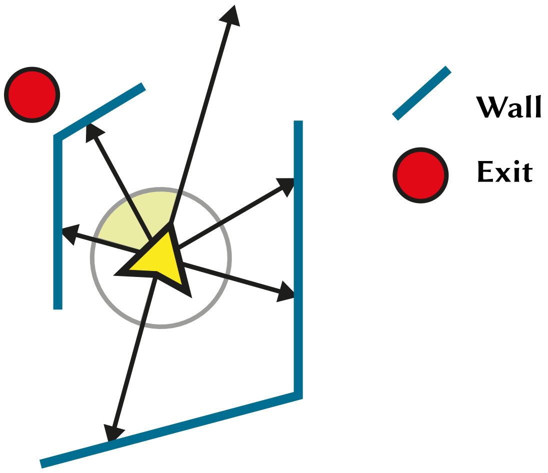


Figure 4.4: *Agent vision*

time step is used. The agent network has three outputs which control rotation left and right, and thrust respectively (Figure 4.5).

4.4.3 Objective Fitness and Phenotypic Descriptor

The objective fitness function in maze domain tasks typically reward candidate solutions for a smaller euclidean distance between the ending position of the agent and the exit (Gomes et al., 2015, Lehman and Stanley, 2008, 2011a). In line with previous studies, each of our objective fitness functions assesses the ending position of an agent to a particular exit. As the NS-MAX algorithm is computed using a linear combination, all values must fall within the same range to avoid one value overpowering another. Therefore we scale the maximum distance of a maze with width w and height h to $\sqrt{w^2 + h^2} \mapsto 1$.

A single objective fitness of a particular agent ρ is given as $f_i(\rho) = 1 - dist(\rho, e_i)$, where e_i is a single exit. The fitness for each of the exits is calculated and assigned to each agent in a manner dependent upon the particular algorithm used. In the unbounded domains the end position of an agent may fall outside of the maze boundary, resulting in a distance greater than the defined size. Therefore, in a similar manner to Gomes and Christensen (2013), to avoid negative fitness values the maximal distance recorded by an agent from the centre of the maze (c) is used to determine the fitness of an individual. $f_i(\rho) = d - dist(\rho, e_i)$, where $d = \max dist(\rho, c)$.

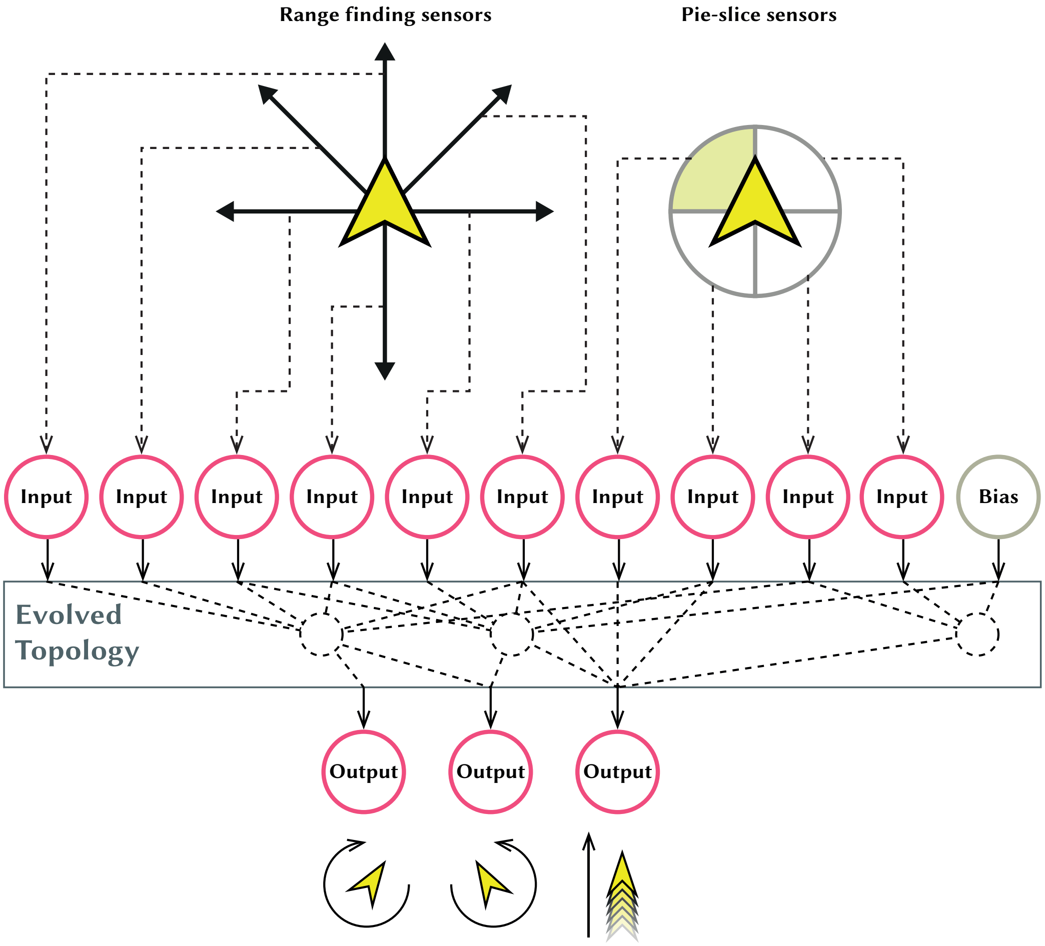


Figure 4.5: *Agent network topology*

Again, in line with previous maze navigation tasks, the phenotypic descriptor is the ending positions of the agent, $\text{MEP}_\mu = (\rho_x, \rho_y)$.

4.4.4 Evolutionary Criteria

In each evolutionary run, populations of 100 neural controllers were optimised for 1000 generations. Each algorithm was run in each domain 50 times and were given a different random seed value for each run. In order to ensure consistency between algorithms, identical random seed values were given to each of the algorithms in each run.

The simulation was performed using a bespoke domain written in the C++ programming language, developed to be similar to the original maze domain experiments in (Lehman and Stanley, 2008, 2011a). The implementation of the

NEAT algorithm used was developed as an extension to the MultiNEAT software in the C++ language.

4.4.5 Algorithms Tested

The algorithms tested in each of the maze domains are listed in Table 4.1.

Abbreviation	Description	Definition	Equation ref.
NS	Standard novelty search	$f_{nov}(\rho)$	Equation [4.1]
MCNS	Minimal criteria novelty search	$f_{mcns}(\rho)$	Equation [4.2]
OBJ-MAX	The maximum fitness value for each of the exits, i.e. the fitness to the closest exit at the end of the trial	$f_{max}(\rho)$	Equation [4.3]
NS-MAX	A linear combination of NS and OBJ-MAX	$f_{ns-max}(\rho)$	Equation [4.4]
MADNS	Multiple assessment directed novelty search	$f_{mad}(\rho)$	Equation [4.5]
MC-MADNS	Multiple assessment directed minimal criteria novelty search	$f_{mc-mad}(\rho)$	Equation [4.6]

Table 4.1: *Algorithms tested in maze domain experiment*

4.4.6 Calculating Performance

The performance of each algorithm was determined by the probability of it to locate all solutions in the domain within a predetermined number of generations. If solutions to all of the exits were not found after 1000 generations, the trial was deemed unsuccessful. The probability of success for each algorithm in each of the domains was calculated by measuring the cumulative probability to discover all solutions to the exits within a single evolutionary run after n generations. In a similar manner to Lehman (2012), cumulative probability (probability of success) is calculated as the number of trials which have located solutions for all exits after the given number of generations. A probability of success equal to 1.0 therefore indicates that all 50 trials have located solution to all exits.

In Sections 4.4.8.1 and 4.5.3.1 we present the probability of success for the algorithms in each of the domains tested. Comparisons of significant difference between two algorithms were calculated throughout using a two-tailed Mann-Whitney U test (Gomes et al., 2015).

4.4.7 Calculating Diversity

In order to calculate the diversity of a particular algorithm in a given domain, the current maze with size $w \times h$ is divided into a 2-dimensional matrix M, of size $n \times n$ where for $M_{x,y} : x, y \in \mathbb{N}_{<n}$. After each trial the agent's MEP$_\mu$ phenotype, (ρ_x, ρ_y), is mapped to the corresponding matrix position $x = \lfloor n \cdot \frac{\rho_x}{w} \rfloor$, $y = \lfloor n \cdot \frac{\rho_y}{h} \rfloor$, $M_{x,y} \mapsto 1$. This process is repeated for each trial in the whole evolutionary run, resulting in a discretised representation of the domain, where for each (x, y), $M_{x,y} \in \{0, 1\}$. Solutions which fall outside of the maze, given by $0 > x > n \vee 0 > y > n$, are discarded.

At the end of the full evolutionary run, represented by α, the diversity score, $f_d(\alpha)$, is calculated as defined in Equation [4.7].

$$f_d(\alpha) = \frac{1}{n^2} \cdot \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} M_{x,y} \qquad [4.7]$$

The values for $f_d(\alpha)$ fall within the range $[0 : 1]$. The diversity calculations are presented in Section 4.4.8.2. As in the performance calculations, comparisons of significant difference between two algorithms were calculated throughout using a two-tailed Mann-Whitney U test.

4.4.8 Results

In the following section we present the probability of success (Section 4.4.8.1) as calculated in Section 4.4.6 and the diversity (Section 4.4.8.2) as calculated in Section 4.4.7 for each of the algorithms.

4.4.8.1 Performance

Figures 4.6 to 4.9 show the probability of success for each algorithm in each of the mazes for the 50 performed replicates, calculated as outlined in Section 4.4.6. The experiments were conducted in order to ascertain the differences in performance between NS and its variants (NS, MADNS, MCNS, MC-MADNS, NS-MAX) and in all cases, their probability of success outperforms OBJ-MAX. Therefore, the OBJ-MAX algorithm is excluded from further discussion in this section.

Bounded Domains

In the bounded domains, the probability of success between MADNS and NS and their minimal criteria variants, MC-MADNS and MCNS, is identical due to the impossibility of a solution to fail the minimal criteria. Therefore reference to NS and MADNS applies to MCNS and MC-MADNS.

Diverge Domain

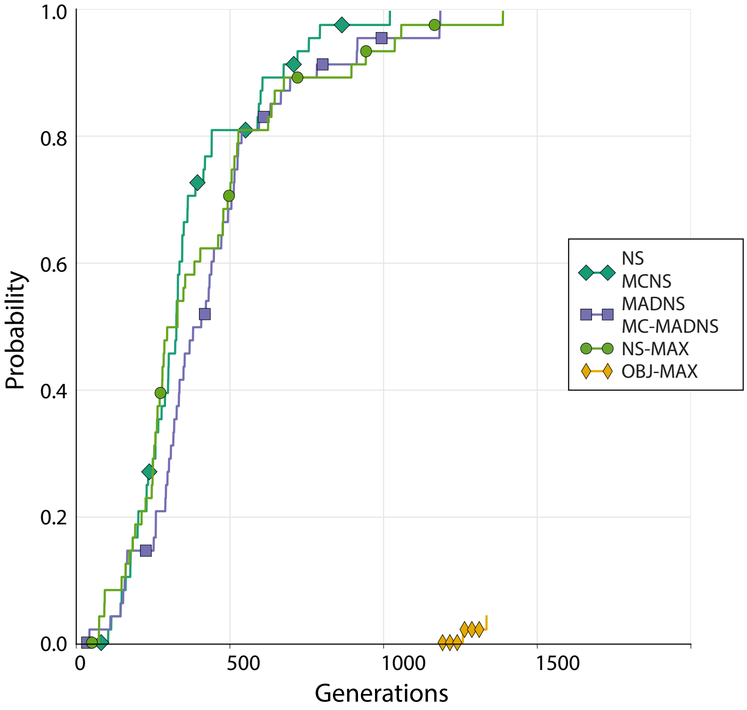


Figure 4.6: *Probability of success for DIVERGE domain*

	MADNS	MC-MADNS	MCNS	NS	NS-MAX
count	50	50	50	50	50
min	35	35	81	81	51
mean	429.163265	429.163265	355.204082	355.204082	402.326531
median	380.5	380.5	326.5	326.5	296.5
max	1184	1184	1032	1032	1388

Table 4.2: *Statistics for DIVERGE domain*

Table 4.2 gives an overview of the results for the DIVERGE domain. In the DIVERGE domain, NS slightly outperforms MADNS, reaching maximal probability of success in fewer generations (NS = 1032, MADNS = 1184). MADNS produced the fastest trial for all algorithms (MADNS = 35, NS = 81).

Subset Domain

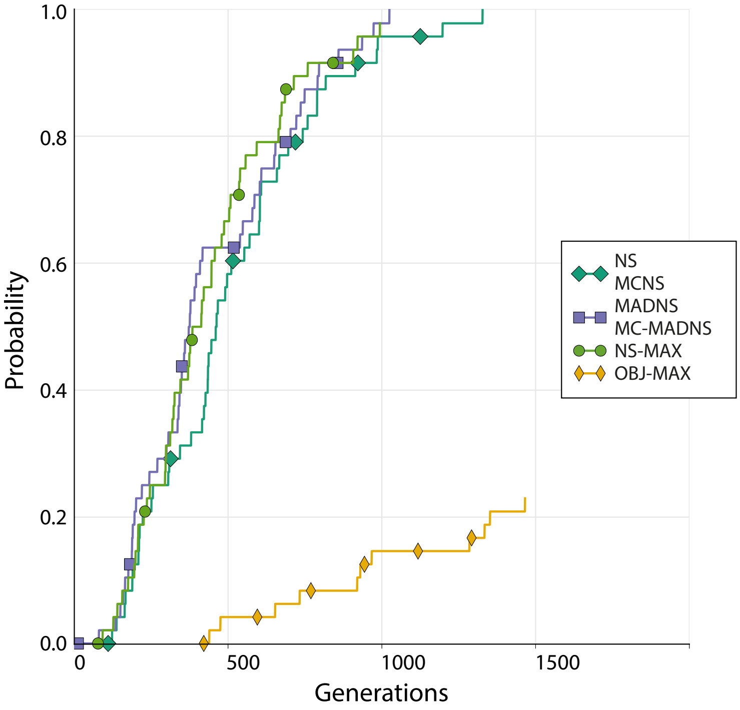


Figure 4.7: *Probability of success for SUBSET domain*

	MADNS	MC-MADNS	MCNS	NS	NS-MAX
count	50	50	50	50	49
min	12	12	110	110	77
mean	436.836735	436.836735	501.265306	501.265306	418.583333
median	372.5	372.5	460.5	460.5	380
max	1025	1025	1328	1328	994

Table 4.3: *Statistics for SUBSET domain*

Conversely, in the SUBSET domain, MADNS reaches maximal probability of success in fewer generations than NS (NS = 1328, MADNS = 1025) (Table 4.3). Again, MADNS produces the fastest trial, locating all solutions after 12 generations. NS-MAX produces comparable levels of performance to NS and MADNS in both the SUBSET and DIVERGE domains (Figures 4.6 and 4.7), however it fails to achieve maximal proability of success in the SUBSET domain, with 1 trial being unsuccessful after the 1500 generations.

Unbounded Domains

Diverge-U Domain

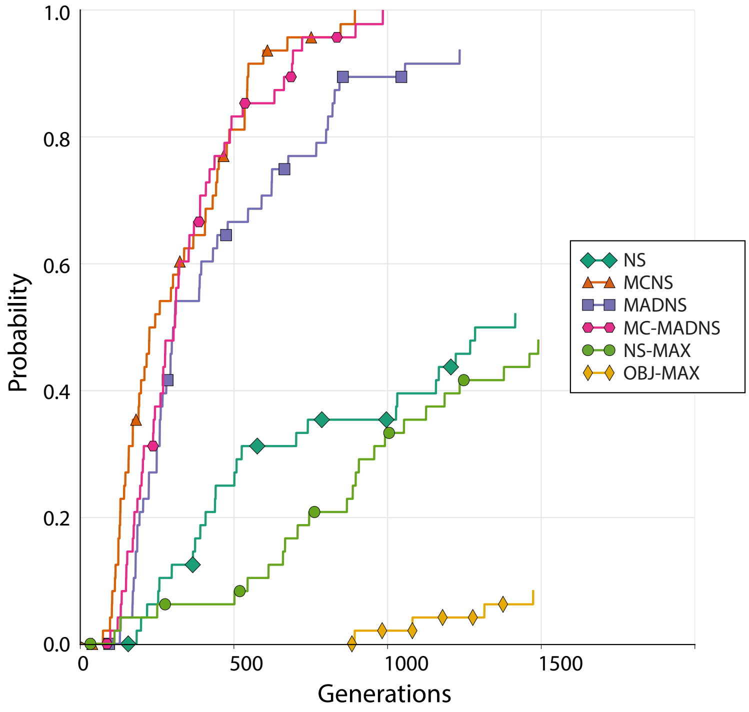


Figure 4.8: *Probability of success for DIVERGE-U domain*

	MADNS	MC-MADNS	MCNS	NS	NS-MAX
count	46	50	50	26	24
min	94	87	39	156	33
mean	409.260870	344.306122	307.530612	636.307692	808.666667
median	296.5	304.5	225.5	471.5	877.5
max	1235	985	894	1416	1491

Table 4.4: *Statistics for DIVERGE-U domain*

NS and NS-MAX are most affected by the unbounded domains. Only the minimal criteria algorithms are successful in all trials, with MCNS outperforming MC-MADNS (MCNS = 894, MC-MADNS = 985). The probability of success achieved by MCNS and MC-MADNS in the DIVERGE-U domain is of a level comparable to the DIVERGE domain (Figure 4.8).

Subset-U Domain

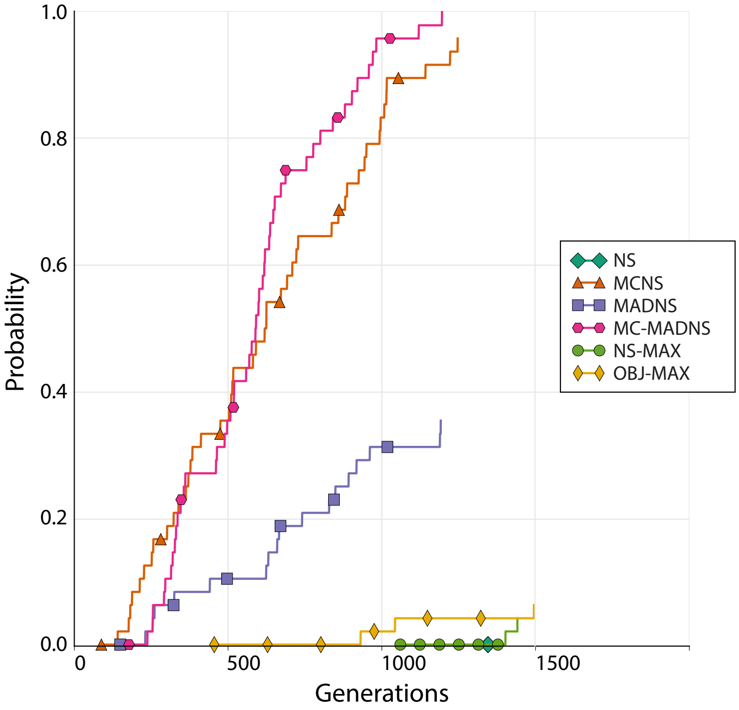


Figure 4.9: *Probability of success for SUBSET-U domain*

	MADNS	MC-MADNS	MCNS	NS	NS-MAX
count	18	50	47	1	3
min	149	178	88	1345	1060
mean	656.333333	584.367347	608.553191	1345	1301
median	663.5	589.5	591	1345	1402
max	1192	1196	1247	1345	1441

Table 4.5: *Statistics for SUBSET-U domain*

In the SUBSET-U domain, MC-MADNS and MCNS achieve higher probability of success than in the SUBSET domain (Figures 4.7 and 4.9), with MC-MADNS outperforming MCNS in this domain (Figure 4.9).

Although MADNS performs relatively sub-optimally in the SUBSET-U domain, it significantly outperforms both NS and NS-MAX $(p < 0.001)$ (Figure 4.9). NS-MAX performs extremely sub-optimally in the SUBSET-U domain

(Figure 4.9). NS achieves the worst performance of all algorithms in the SUBSET-U domain, locating solutions to all of the exits in only 1 of the trials (Figure 4.9).

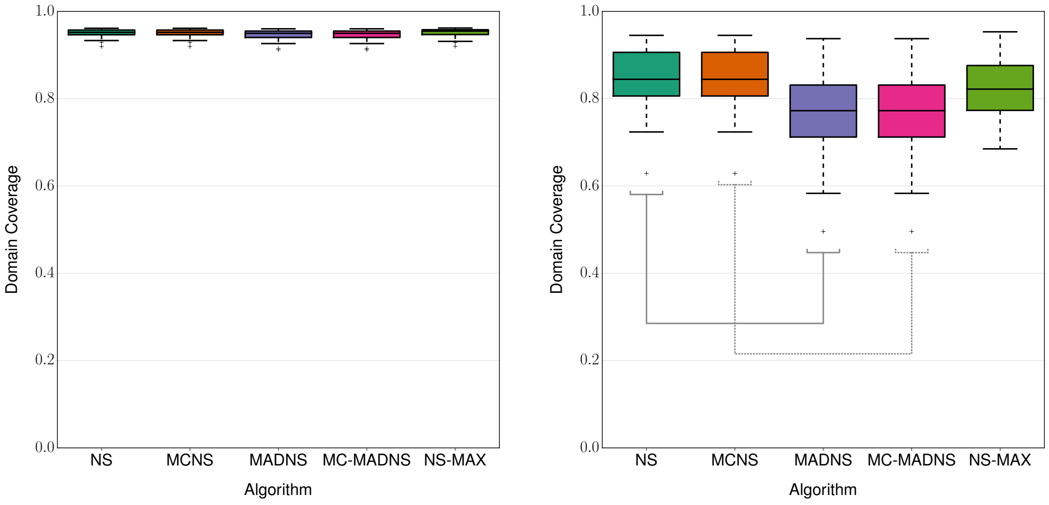
4.4.8.2 Diversity

The domain coverage for each of the algorithms, as calculated in Section 4.4.7, is presented for the bounded domains in Figures 4.10a and 4.10b and the unbounded domains in Figures 4.11a and 4.11b.

As in Section 4.4.8.1, the levels of diversity achieved by NS, MADNS, MCNS, MC-MADNS and NS-MAX outperform OBJ-MAX in all cases. Therefore, this algorithm is excluded from further discussion.

Bounded Domains

Similarly to Section 4.4.8.1, the levels of diversity between MADNS and NS and their minimal criteria variants, MC-MADNS and MCNS, are identical due to the impossibility of a solution to fail the minimal criteria.



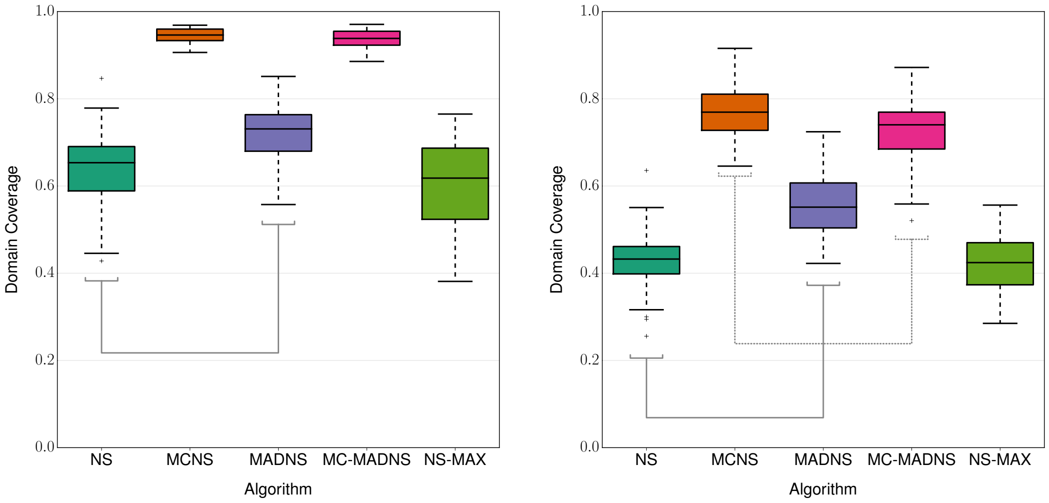
(a) *SUBSET domain* **(b)** *DIVERGE domain*

Figure 4.10: *Domain coverage for bounded domains*

In the SUBSET domain, all of the algorithms achieve extremely high levels of diversity ($median > 0.9$) with no significant difference between algorithms (Figure 4.10a).

In the DIVERGE domain, NS and MCNS achieve significantly higher levels of diversity than MADNS and MC-MADNS ($p < 0.0001$) (Figure 4.10b).

Unbounded Domains



(a) *SUBSET-U domain*

(b) *DIVERGE-U domain*

Figure 4.11: *Domain coverage for unbounded domains*

NS is most affected in the unbounded domains (Figures 4.11a and 4.11b). Unlike NS, there is no significant difference in diversity between the bounded and unbounded domains for MCNS, MADNS and MC-MADNS. MADNS achieves significantly higher levels of diversity than NS in both the DIVERGE-U and SUBSET-U domains ($p < 0.001$) (Figures 4.11a and 4.11b).

4.4.9 Discussion

The results from the multiple exit maze experiment show that NS is capable of locating multiple objectives unaided in small domains. However, as domain size increases, NS may struggle to locate particular objectives of interest. MCNS may be suitable to assist NS through reducing the potential for exploration in the phenotypic landscape, however, it requires the minimal criteria, in this case the bounds of the domain to be determined in advance of the evolutionary run. The results also show that MADNS significantly outperforms NS in unbounded domains, indicating that in large phenotypic landscapes, directing NS towards objectives of interest is beneficial. Therefore, in large search spaces, if multiple independent objectives are required, optimisation through MADNS is a suitable approach.

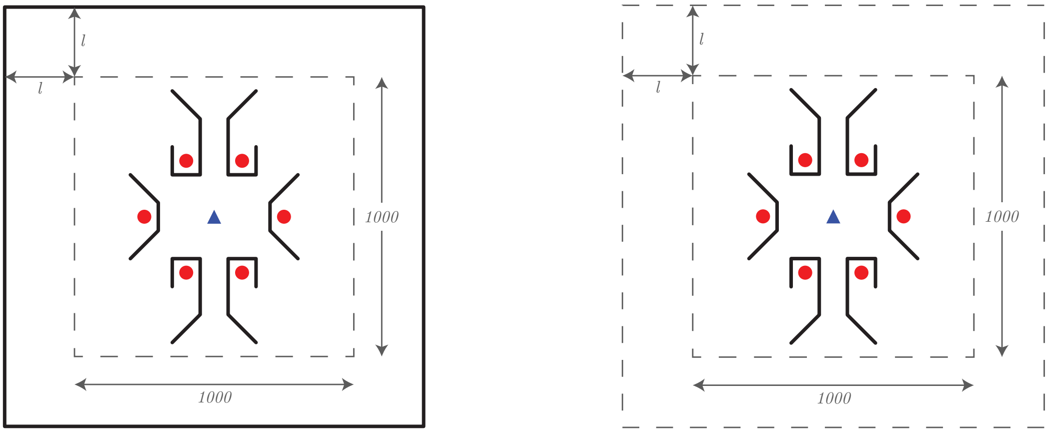
4.5 Experiment 2: Scaled Maze Domains

A further experiment was designed to assess the relative gain in objective performance through directing novelty in relation to increasing the potential for exploration in the feature space.

The scaled maze experiment was performed under the same conditions as the multiple exit maze experiment, with the agent model, objective fitness assessment, phenotypic descriptor and evolutionary criteria remaining identical (see Sections 4.4.2 to 4.4.4).

4.5.1 Domain

As in Section 4.4.1, the task domain is a simulated maze, in which an agent controller must navigate from an initial starting-point to one of a possible number of exit points within a fixed time limit. The maze domain utilised in this experiment is tested over a range of sizes. Both bounded and unbounded variants of the domain were tested.



(a) *SCALED domain* **(b)** *SCALED-U domain*

Figure 4.12: *SCALED and SCALED-U domains*

SCALED A divergent maze with 6 exits. The distance between the exits and the edges of the maze (Figure 4.12a, $w = 1000 + l$) is altered in each evolutionary trial. Triangle indicates start position. Circles represent exits. The values tested are $l = \{0, 4000, 9000\}$. Values were tested in 50 evolutionary trials for each algorithm.

SCALED-U A varying scale maze identical to SCALED with unbounded exploration potential, (Figure 4.12b). This unbounded domain is a much more difficult task for divergent evolution, as the potential space for exploration is potentially infinite.

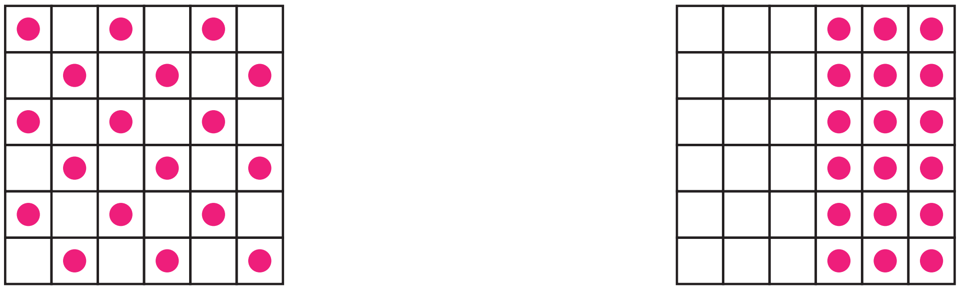
The scaled maze was designed in order to be a difficult task for divergent search. Navigating to an exit requires the agent to move initially outwards from the centre of the maze towards the edges (which will be expected to have a higher initial novelty value) before rotating and navigating back towards the centre of the maze (which will be expected to have a low initial novelty value). As the size of the domain increases ($w = 1000, 5000, 10000$), the area for exploration around the exits will become larger, making navigating towards the exits increasingly more difficult for novelty search.

Each of the mazes range from 1000-10,000 units, dependent upon the trial. The agent has a size of 20 units and the exits require the agent to be within 20 units of them. Each agent is given 4000 time steps to complete the maze.

4.5.2 Calculating Exploration Uniformity

In the previous experiment, diversity was calculated using the method presented in Section 4.4.7. This calculation measures the amount of domain coverage produced in an evolutionary trajectory. In small domain sizes, where coverage is of reasonably high levels, this method produces a suitable indication of population diversity. However, the domains presented in this experiment are larger in size, with the population being spread more sparsely across the phenotypic landscape. Therefore, in the domains presented in this experiment we utilise the *exploration uniformity* method to calculate diversity.

Figures 4.13a and 4.13b illustrate two example population resting positions (pink circles). Although it is apparent that the spread of the population is more evenly distrbuted in Figure 4.13a, both domains would achieve identical values when calculated using the domain coverage method presented in Section 4.4.7. Unlike the domain coverage calculation, the exploration uniformity method would



(a) *Example of an evenly spread population with 50% domain coverage*

(b) *Example of an unevenly spread population with 50% domain coverage*

Figure 4.13: *Example domains with 50% coverage*

return a higher value for the domain illustrated by Figure 4.13a than the domain in Figure 4.13b.

The exploration uniformity of a particular algorithm is calculated, as in Section 4.4.7, by initially dividing into a 2-dimensional matrix M. The final position of an individual, (ρ_x, ρ_y), is mapped to the corresponding region of M.

In a similar manner to Gomes et al. (2015), the exploration uniformity is calculated by measuring the similarity of the population distribution to the uniform distribution However, we calculate diversity of the current population at each generation rather than the cumulative exploration uniformity over the whole evolutionary run. Let $\mathcal{P}_t$ be the set of individuals in the population at generation t and let Ψ_t be the distribution of $\mathcal{P}_t$ over M. The exploration uniformity of the population, $D(\mathcal{P}_t)$, is calculated as the similarity between Ψ_t and the uniform distribution U. As in Gomes et al. (2015) the distance metric used is the Jensen-Shannon Distance (JSD). The exploration uniformity for a single generation is therefore defined as in Equation [4.8]:

$$\begin{aligned} D(\mathcal{P}_t) &= 1 - JSD(\Psi_t, U),\ where: \\ \Psi_t &= \left(\frac{|I_1|}{|\mathcal{P}_t|}, \dots, \frac{|I_{|\mathcal{P}_t|}|}{|\mathcal{P}_t|}\right), I_r = \{i \in \mathcal{P}_t | region(i) = r\} \\ U &= \left(\overbrace{\frac{1}{|M|} \times \dots \times \frac{1}{|M|}}^{n^2 \text{ times}}\right) \end{aligned} \qquad [4.8]$$

4.5.3 Results

In the following section we present the probability of success (Section 4.5.3.1) as calculated in Section 4.4.6 and the exploration uniformity (Section 4.5.3.2) as calculated in Section 4.5.2 for each of the algorithms.

4.5.3.1 Performance

The probability of success for the SCALED domain with sizes $w = \{1000, 5000, 10000\}$ is presented in Figure 4.16. As the algorithms are unable to reliably locate solutions to all exits in the SCALED-U domains, we present the number of exits each algorithm managed to locate in Figures 4.17 and 4.18

Bounded Domains

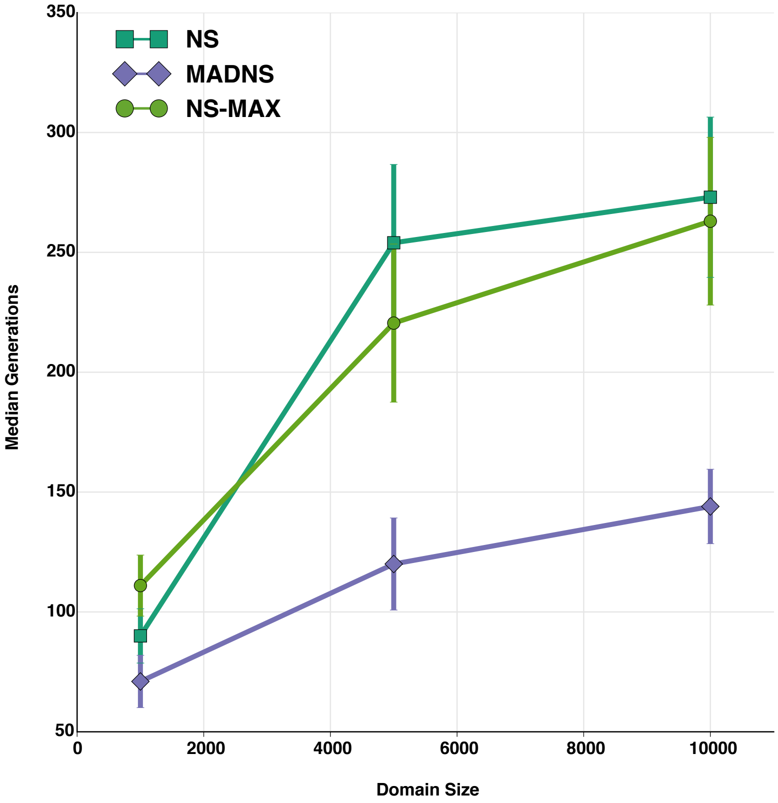
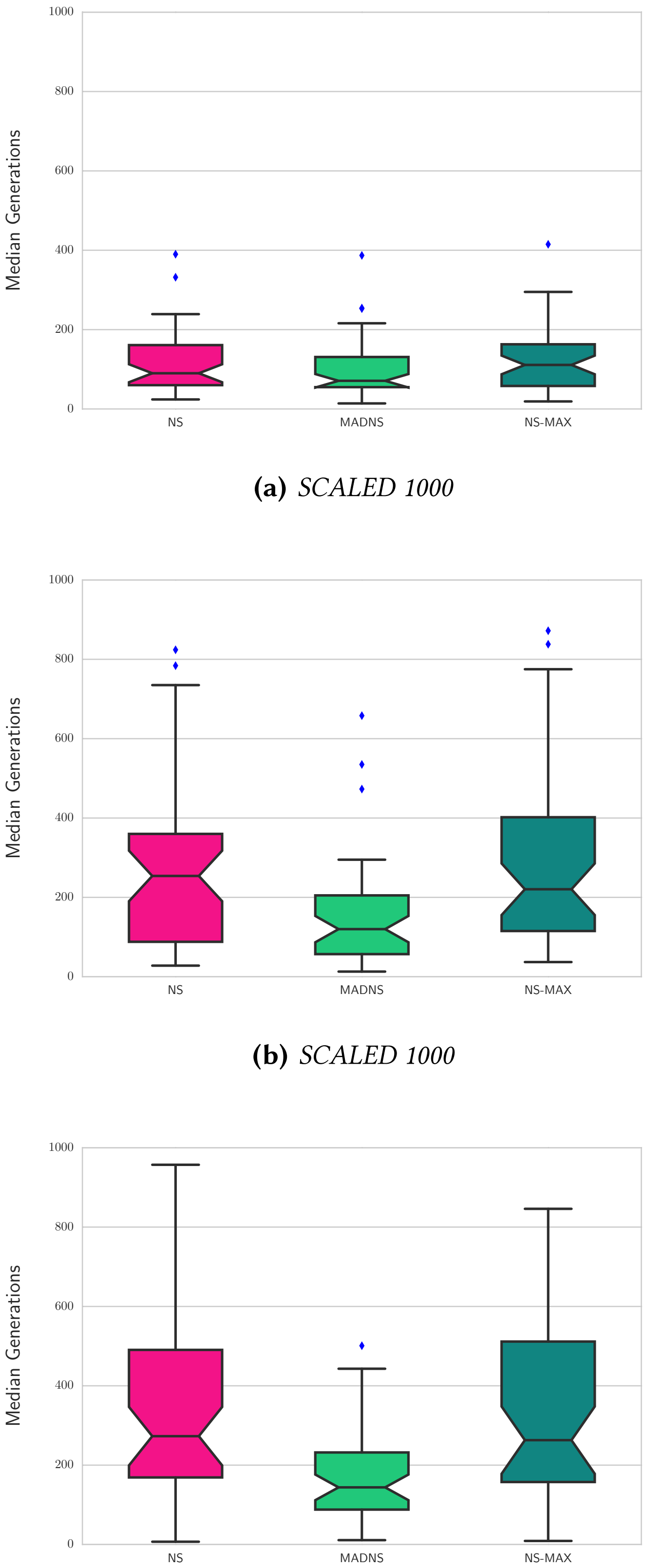


Figure 4.14: *Median number of generations taken to locate all exits as domain size increases in SCALED domain*

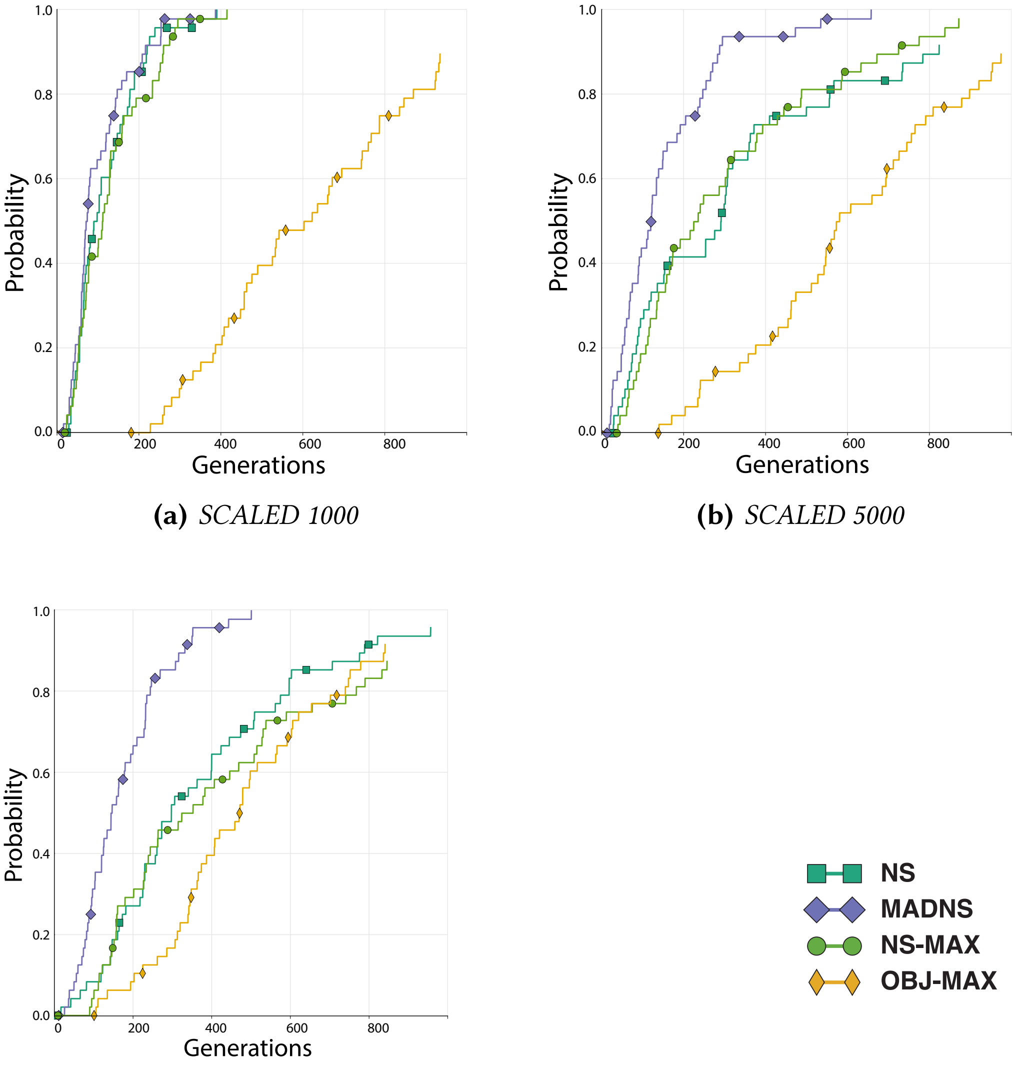


(a) *SCALED 1000*

(b) *SCALED 1000*

(c) *SCALED 1000*

Figure 4.15: *Median number of generations taken to locate all exits as domain size increases in SCALED domain*



(a) *SCALED 1000*

(b) *SCALED 5000*

(c) *SCALED 10000*

Figure 4.16: *Probability of Success in SCALED domains*

As the SCALED maze is a bounded domain, the results for NS and MADNS are identical to their respective minimal criteria variants, MCNS and MC-MADNS.

Figure 4.16 shows the probability of success for each algorithm in the SCALED domain over 50 replicates.

There is no significant difference between NS and MADNS in the smallest size domain ($w = 1000$), with both algorithms locating all exits in all trials within 400 generations. MADNS slightly outperforms NS on both the maximum number of generations taken to locate all exits, (MADNS = 387, NS = 390) and the median value (MADNS = 71.0, NS = 90.0).

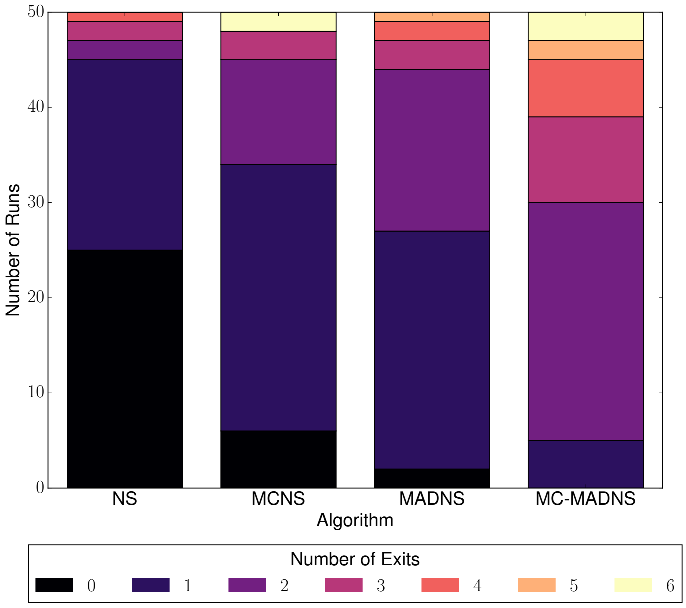


Figure 4.17: *Number of exits located in SCALED-U domain ($w = 1000$)*

However, as shown in Figures 4.14 and 4.15, MADNS significantly outperforms NS in both the larger SCALED domains ($w = \{5000, 10000\}$). The results for NS in the largest domain are similar in performance to OBJ-MAX. The difference in performance between MADNS and NS can be seen to increase as the domain size increases, as shown in Figure 4.14.

NS-MAX performs similarly to NS, with no significant difference between the algorithms in any of the domain sizes (Figures 4.14 to 4.16).

Unbounded Domains

Tables 4.6 and 4.7 and Figures 4.17 and 4.18 present the number of exits the algorithms managed to locate within the allocated 1000 generations. In the smaller domain ($w = 1000$, Figure 4.17) only MADNS and MC-MADNS manage to locate all exits in any of the trials. MADNS significantly outperforms NS ($p \leq 0.01$) and MC-MADNS significantly outperforms MCNS ($p \leq 0.01$). MC-MADNS performs most optimally, locating at least 1 exit in all trials.

In the larger domain ($w = 10000$, Figure 4.18), all algorithms perform sub-optimally. However, NS and MCNS are more severely affected. Surprisingly, MCNS has the worst performance out of all of the algorithms, locating no exits in 41 out of 50 trials. Similarly, NS locates no exits in 34 out of 50 trials.

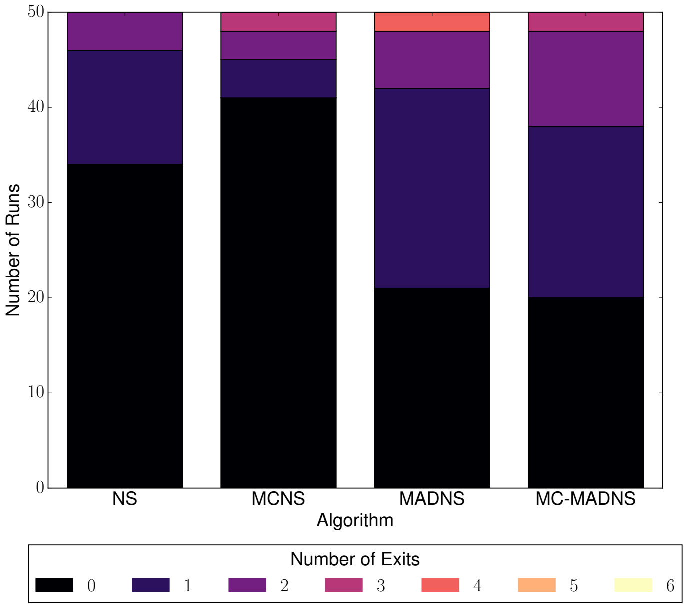


Figure 4.18: *Number of exits located in SCALED-U domain ($w = 10000$)*

	number of exits located						
Algorithm	0	1	2	3	4	5	6
NS	25	20	2	2	1	0	0
MADNS	6	28	11	3	0	0	2
MCNS	2	25	17	3	2	1	0
MC-MADNS	0	5	25	9	6	2	3

Table 4.6: *Frequencies of the number of exits located within 1000 generations in the SCALED-U domain ($w = 1000$)*

	number of exits located						
Algorithm	0	1	2	3	4	5	6
NS	34	12	4	0	0	0	0
MADNS	21	21	6	0	2	0	0
MCNS	41	4	3	2	0	0	0
MC-MADNS	20	18	10	2	0	0	0

Table 4.7: *Frequencies of the number of exits located within 1000 generations in the SCALED-U domain ($w = 10000$)*

4.5.3.2 Diversity

In this section we present the exploration uniformity for each of the algorithms, as calculated in Section 4.5.2.

Bounded Domain

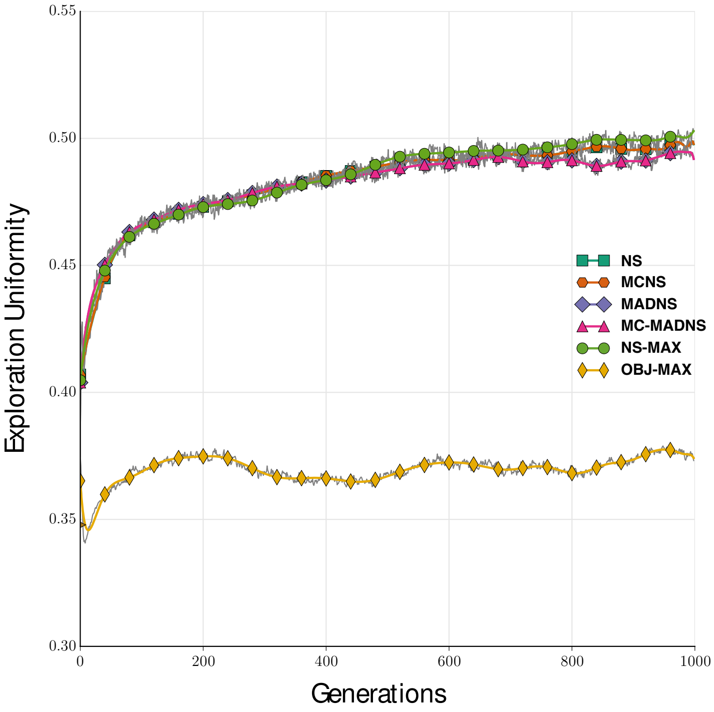
Figures 4.19a and 4.19b show the exploration uniformity for the SCALED domains for $w = 1000$ and $w = 10000$. As expected, in a similar manner to Section 4.5.3.1, results for the minimal criteria variants are identical for this domain, therefore have been excluded from discussion. In all cases, populations optimised through NS exhibit greater diversity in the population than MADNS.

As in Section 4.5.3.1, NS-MAX performs similarly to NS, with no significant difference between the algorithms in any of the domain sizes. The relative difference between the levels of exploration uniformity for NS and MADNS increases alongside domain size (Figures 4.19a and 4.19b).

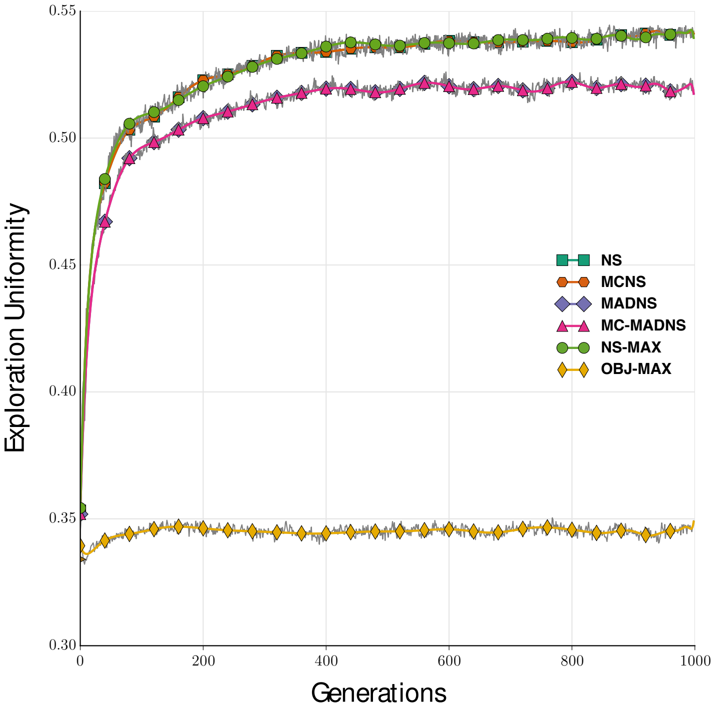
Unbounded Domain

The diversity is extremely unstable in the SCALED-U domain for all sizes of w (Figures 4.20a and 4.20b). However, due to solutions falling outside of the domain being penalised in the minimal criteria variants, the effect upon these algorithms is less severe.

MCNS and MC-MADNS perform relatively similarly throughout all domain sizes. However, as the domain size increases, the benefit gained by the minimal criteria decreases (Figures 4.20a and 4.20b). This is expected behaviour, as the minimal criteria will have less effect as domain size increases, due to fewer solutions falling outside of the domain. Therefore, although the levels of exploration uniformity are still increasing after 1000 generations, it is expected that there would be no relative increase in distance between the Minimal criteria and standard variants in subsequent generations, due to the reduced effect of the minimal criteria.

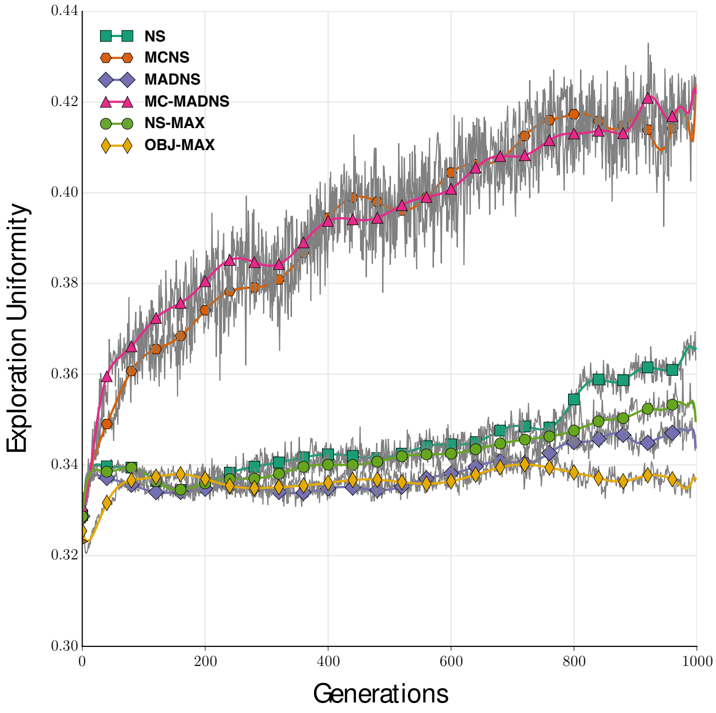


(a) *Exploration uniformity in SCALED domain, $w = 1000$*

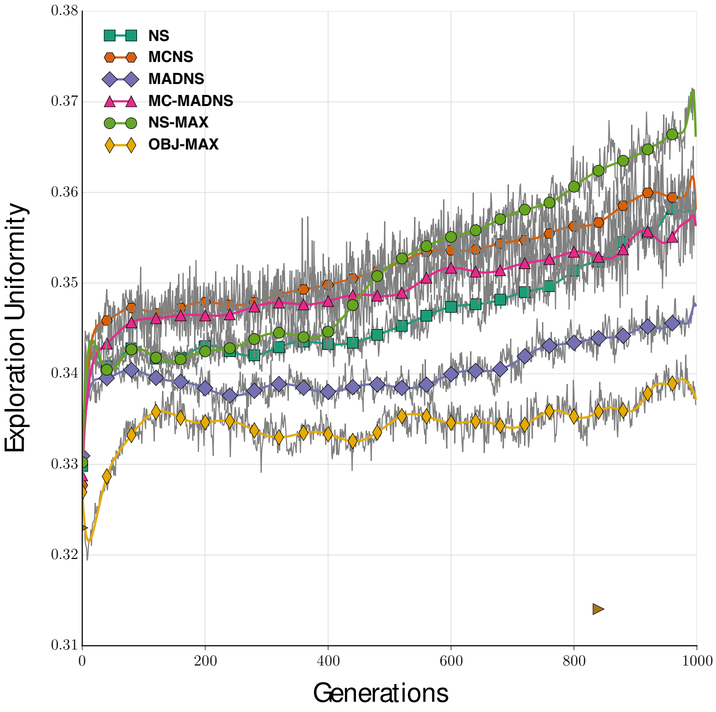


(b) *Exploration uniformity in SCALED domain, $w = 10000$*

Figure 4.19: *Exploration uniformity in SCALED domain, $w = 1000, 10000$*



(a) *Exploration uniformity in SCALED-U domain, $w = 1000$*

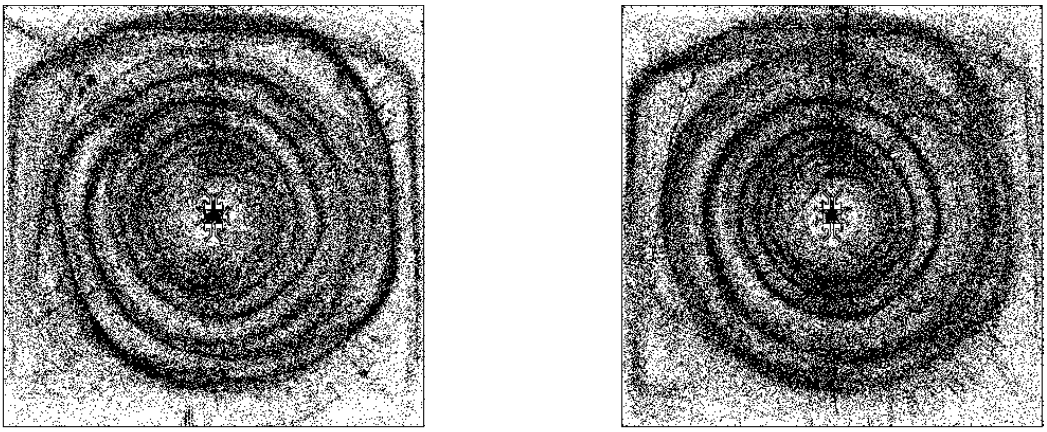


(b) *Exploration uniformity in SCALED-U domain, $w = 10000$*

Figure 4.20: *Exploration uniformity in SCALED-U domain, $w = 1000, 10000$*

4.5.4 Discussion

Figures 4.21 and 4.22 illustrate a series of full evolutionary runs in the SCALED (Figures 4.21a and 4.21b) and SCALED-U (Figures 4.22a and 4.22b) domains. The figures indicate the final resting positions of every agent over the course of the full trial. As is shown, in the bounded domain both NS and MADNS explore the full area of the maze. However, in the unbounded domain, SUBSET-U, the divergent nature of NS pushes the search to focus upon areas of the landscape outside of the domain. MADNS resists this pressure for exploration (and high novelty), by constantly attracting the search towards the areas of interest.

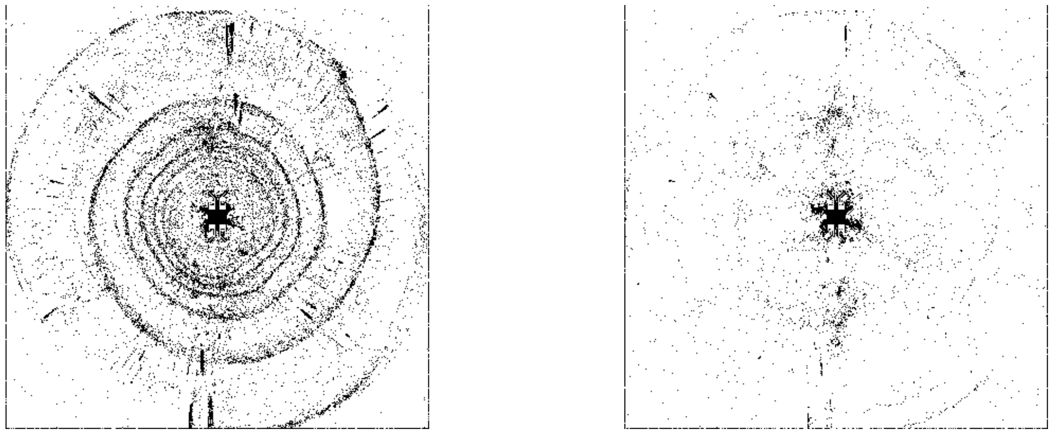


(a) *MADNS in SCALED domain* **(b)** *NS in SCALED domain*

Figure 4.21: *Example exploration of domain over a single evolutionary run for MADNS and NS in SCALED domain (identical random seeds, w = 5000)*

The algorithms struggle to optimise all exits in the unbounded domain. A potential reason for this behaviour may be that, in the bounded domain, the borders of the maze act as not only barriers, but also as reference points to the agent, aiding the controller to position itself in novel areas of the maze. Without this frame of reference and with the added potential for a controller to leave the edges of the maze in all directions, the divergent evolution struggles.

As shown in Section 4.5.3.1, as the phenotypic landscape increases, the divergent evolutionary trajectory pursued by NS requires more assistance to focus upon areas of interest. This is hardly surprising, as NS purposefully has no concept of an objective built within the algorithm. In fact, NS was introduced with the specific desire to *abandon objectives* (Lehman and Stanley, 2008), alleviating the tendency for search to be drawn towards local optima, particularly in deceptive domains. This divergent exploration technique has proved successful in domains with small phenotypic landscapes, however, as the landscape expands, the search for pure novelty often requires assistance to focus the optimisation. While MCNS goes somewhat towards restricting the exploration, as we have shown, specifically directing the search through MCMADNS assists further still, without fully removing the benefits of divergent exploration.



(a) *MADNS in SCALED-U domain* **(b)** *NS in SCALED-U domain*

Figure 4.22: *Example exploration of domain over a single evolutionary run for MADNS and NS in SCALED-U domain (identical random seeds, w = 5000)*

The experiments also indicate that, potentially any area of the phenotypic landscape could be used as an attractor, regardless of the objective assigned to it. However, further studies in domains where the alignment between the phenotypic landscape and the objective function are less direct would be needed to fully establish the relationship between phenotypic exploration and objective functionality.

4.6 Implications

In this chapter, MADNS was introduced, a novel extension to the NS algorithm. The evidence presented from the experimental findings in a range of maze domains shows that NS is capable of the simultaneous optimisation of multiple independent objectives, however, as the size of the phenotypic landscape increases, MADNS significantly outperforms standard NS.

Although the maze navigation domains utilise a phenotypic descriptor which directly maps to the objective landscape, solutions to multiple objectives should also be located when the feature space and objective space are less directly mapped. For example, a population of neural agent controllers evolved for diversity of evasion and pursuit behaviours would as likely contain examples of predatory behaviour as behaviours characteristic of prey if the controller was capable of exhibiting such behaviours.

The extension of NS through the application of the MADNS algorithm is a viable method for the simultaneous optimisation of multiple independent objectives. The experimental validation was performed in a domain in which the phenotypic landscape is strongly aligned to the objective landscape, NS has been shown to be successful in domains where the landscape is less directly mapped, for example biped walking (Lehman and Stanley, 2008). Therefore, we would expect the application of MADNS to less directly mapped domains to be successful, although further validation in different domains may be necessary to fully establish this.

However, it was also found that as phenotypic landscape size increases, the NS algorithm may be unable to focus the search procedure on relevant areas of the landscape. Although the introduction of MCNS somewhat alleviates this problem in unbounded domains, the results have shown that in large feature spaces, search through the utilisation of novelty still struggles.

The development of a divergent search procedure which utilises a hierarchical structure in order to focus upon relevant areas of the phenotypic landscape may be a more suitable approach. Therefore, in Chapter 5 we establish such a method. The introduction of an hierarchical search procedure for divergent phenotypic search is the focus of the next chapter — through the introduction of Spatial, Hierarchical, Illuminated NeuroEvolution (SHINE), an alternative novel method for rapid, hierarchical exploration of low-dimensional phenotypic landscapes.

5 Spatial, Hierarchical, Illuminated NeuroEvolution

5.1 Introduction

Novelty search, as previously introduced in Section 2.8 and discussed throughout this thesis, is an algorithm which removes the need for a traditional objective function through the assignment of high fitness values to novel behaviours in a population. The objective fitness function is replaced by a behavioural distance metric, which is used to determine the novelty of an individual in a population. High novelty is assigned to individuals which exhibit features with a large distance to both the rest of the population and an archive of previously encountered, highly novel phenotypic traits. Although NS may outperform objective fitness search, especially when applied to deceptive domains and with a phenotypic descriptor which is well aligned to the objective (Chapters 3 and 4), it has been shown that the assessment of behavioural novelty alone is insufficient as a generalisable evolutionary technique in many tasks, especially in domains with large feature spaces (Cuccu and Gomez, 2011, Mouret, 2011).

Since its introduction, there have been a number of MOEA variants which extend the NS algorithm, aiming to harness its divergent nature for a range of objective optimisation procedures (Section 2.8.5). In Chapter 4, a novel extension to the NS algorithm was introduced. The MADNS algorithm exploited the inherently divergent nature of NS for the optimisation of multiple independent objectives. It was shown, that MADNS was useful in directing NS towards multiple conflicting objectives, especially within large search spaces.

In this chapter, we focus upon a class of MOEAs which combine the optimisation of objective functionality with phenotypic diversity. This class of algorithms are increasingly termed as *illumination* algorithms. Novelty Search with Local Competition (NS-LC) was introduced by Lehman and Stanley (2011b) in order to alleviate the problems inherent to NS, which were explored in Chapters 3 and 4 of this thesis. NS-LC is a MOEA which combines local objective fitness optimisation with global phenotypic exploration. More recently, the MAP-Elites algorithm, as introduced in Mouret and Clune (2015), is an evolutionary procedure that aims to find the highest performing solution at each point in a low-dimensional behaviour space. Taking inspiration, in part, from NS-LC, MAP-Elites is a hybridization of objective driven and divergent search. In MAP-Elites, evolution proceeds through the maintenance of an archive of previously high performing individuals, with each individual being assigned to bin within a discrete, low dimensional representation of the feature space. Offspring for subsequent generations are randomly selected from the archive of high performing, yet behaviourally diverse individuals.

Due to the ability of MAP-Elites to highlight the highest performing solutions in a phenotypic landscape, Mouret and Clune introduce the term *illumination algorithm* to separate it from traditional optimisation algorithms (Mouret and Clune, 2015).

In this chapter, the Spatial, Hierarchical, Illuminated NeuroEvolution (SHINE) algorithm is introduced, a novel method for illuminating phenotypic landscapes, which is shown, through the presented experimental results, to explore low dimensional phenotypic landscapes more thoroughly and rapidly than two current state of the art illumination algorithms, MAP-Elites and NS-LC. Similarly to MAP-Elites, our proposed SHINE algorithm selects future populations from an archive of previous solutions. However, the archive in the SHINE algorithm is maintained within an hierarchical, spatially partitioned tree structure. Both the weighting of offspring selection and the number of representatives assigned to the archive are calculated from the depth of the vertices within which the solutions reside. Candidate solutions which exhibit phenotypic traits in more crowded areas of the landscape are assigned to vertices deeper within the tree, and are penalised accordingly. This allows the evolutionary trajectory to focus on larger, shallower areas of the landscape, producing a divergent, and iteratively more focused search procedure.

This chapter is organised as follows — in Section 5.2, the approach taken to develop the SHINE algorithm, utilising methods from computational geometry is outlined. In Section 5.3, the proposed SHINE algorithm is formally defined, highlighting the methods for archive management, spatial partitioning and selection of offspring in a 2-dimensional, quadtree implementation. An initial experimental domain, selected to assess the ability of the SHINE algorithm to explore the phenotypic landscape, is presented in Section 5.4. In this experiment, the algorithm is applied to the HARD maze domain, to highlight the ability of SHINE to overcome deception, much in the same manner as NS. The results from this experiment show that SHINE significantly outperforms both NS and MAP-Elites when configured using a *corner* sorting distance metric in a deceptive domain.

A further maze domain, previously applied to the assessment of MADNS in Section 4.5, the SCALED maze domain, is utilised to establish the ability of SHINE to explore large phenotypic landscapes. The results from this experiment show that SHINE significantly outperforms both NS and the MADNS extension.

A final experimental domain is presented in Section 5.5. In this domain, the phenotypic descriptor measures the modularity and connection weights of the underlying evolved ANNs. The corner sorting archive metric is replaced with an objective based metric for archive management, in order to establish the suitability of shine to perform as an illumination algorithm. The results from this experiment show that SHINE outperforms both MAP-Elites and NS-LC using the assessment criteria recently established for the evaluation of illumination algorithms (Mouret and Clune, 2015).

Finally, in Section 5.6, the findings from the presented experiments are summarised, highlighting limitations and further suggestions for the application and development of the SHINE algorithm.

5.2 Approach

Optimisation in the MAP-Elites algorithm progresses through the subdivision of a low dimensional phenotypic landscape into a discrete matrix of bins. Each bin contains the most highly performing individual currently located with the relevant phenotypic trait for that bin. However, the MAP-Elites algorithm does not specifically focus search upon bins which have not previously been visited, rather they are filled due to the underlying exploration of the EA, for example through mutation. This has previously been highlighted as a limitation to the MAP-Elites algorithm in Mouret and Clune (2015). However, in suggesting one potential method for overcoming the problem, Mouret and Clune indicate that preliminary trials showed no significant improvement over the standard MAP-Elites algorithm. In the SHINE algorithm, a different approach is taken to encourage search to focus upon unvisited areas of the landscape, through the application of a method from computational geometry often applied within the development of video games (Samet, 1984).

One of the major considerations in the development of video games is measurement of collisions between spatial entities. As video games are usually updated in real time, and often involve multiple collisions between spatial elements, efficient methods for collision detection are necessary. Many methods from computational geometry have been successfully applied to collision detection in video games and are still widely used in current game development. Parallels exist between the requirements of collision detection in video games and in efficiently locating densely populated areas of low dimensional phenotypic search spaces. One widely applied method is to superimpose a spatially partitioned tree structure over the space and to assign vertices within the tree to the entities which fall within it. When applied to collision detection, the tree may be searched, excluding vertices which relate to irrelevant areas of the landscape. Therefore, only entities which are contained within the relevant branches of the tree need to be checked for collisions.

Another characteristic of applying a hierarchical tree structure is that, as the depth of the vertices increases, the area covered by each vertex decreases. Alongside this, tree subdivision does not occur unless the area has previously been visited by a predefined number of entities; the deeper within the tree, the more densely populated the area of the landscape. Therefore, an accurate estimation of the density and area of the surrounding landscape to an individual entity can be easily established through the information intrinsically held within the entity's containing vertex.

This information is utilised through the selection operator of the SHINE algorithm to penalise phenotypes in more dense areas of the phenotypic landscape, forcing search to focus on sparse, less visited areas.

5.3 The Algorithm

SHINE is an illumination algorithm designed for rapid and thorough exploration of low-dimensional feature spaces. SHINE promotes divergent search through penalising solutions which are in more crowded areas of a predefined, low dimensional phenotypic landscape. The algorithm utilises a spatially partitioned tree for the maintenance of an archive of phenotypic representatives. The mechanisms applied to both the storage and selection of the representatives are designed specifically to weight subsequent generations towards more offspring in sparse areas of the landscape.

The SHINE algorithm shares similarities to both NS and MAP-Elites. As inMAP-Elites, SHINE maintains an archive of previous solutions which are selected for inclusion by low-dimensional discrete phenotypic traits. However, SHINE utilises an hierarchical, spatially partitioned tree structure for archive maintenance. MAP-Elites stores a single elite within each area of the feature space; the current best performing individual at an objective function. When applying the corner sorting method (Section 5.3.4), SHINE maintains multiple individuals within each vertex of the archive tree which are chosen by their distance to the boundaries of their particular phenotypic trait, in a manner more aligned with NS.

Therefore, the SHINE algorithm also differs from MAP-Elites in that it directly aims to optimise sparse areas of the feature space. In Section 5.3 we introduce the main SHINE procedure, outlining both the generalised, n-dimensional definition and an example 2-dimensional implementation which utilises a quadtree structure (Samet, 1984).

5.3.1 Main Procedure

The main procedure of the SHINE algorithm begins by initializing a random population $\mathcal{P}$ with n random individuals (Algorithm 3, lines 2-4). In each generation, every individual ρ is assessed in the domain and a phenotypic descriptor is measured and assigned to μ (lines 6-9). The tree, $\mathcal{V}$, is queried with μ (line 8). After all individuals in the current population have been assessed and the tree structure updated, $\mathcal{P}$ is added to the archive (line 10). A new archive is calculated and assigned to $\mathcal{X}$ (line 10). All individuals are removed from the population, which is then repopulated with mutated offspring from the updated archive $\mathcal{X}$ via weighted roulette selection (lines 12-16). This procedure is

Algorithm 3 SHINE ALGORITHM Main procedure.

```
Domain specific functions:
        RandomSolution,  MeasureBehaviour,  MutateSolution
Precondition:
        $\mathcal{X} \leftarrow \varnothing$                      ▷ initialise an empty set of representative solutions $\mathcal{X}$
        $\mathcal{P} \leftarrow \varnothing$                      ▷ initialise an empty set $\mathcal{P}$ to contain population
        $\mathcal{B} \leftarrow [l_1 : r_1] \times \ldots \times [l_n : r_n]$      ▷ $n$ dimensional bounding area $\mathcal{B}$
        $\mathcal{V} \leftarrow$ CreateTree($\mathcal{B}$)               ▷ initialise empty tree $\mathcal{V}$

 1 procedure SHINE
 2     while $|\mathcal{P}| < n$ do
 3         $\mathcal{P} \leftarrow$ RandomSolution()
 4     end while
 5     do
 6         $\forall\, \rho \in \mathcal{P}$ do
 7             $\mu \leftarrow$ MeasureBehaviour($\rho$)
 8             QueryTree($\mu$, $\mathcal{V}_r$)
 9         end for
10         $\mathcal{X} \leftarrow$ UpdateArchive($\mathcal{P}$, $\mathcal{V}$)
11         $\mathcal{P} \leftarrow \varnothing$
12         while $|\mathcal{P}| < n$ do
13             $x \leftarrow$ RouletteSelection($\mathcal{X}$)
14             $x' \leftarrow$ MutateSolution($x$)
15             $\mathcal{P} \leftarrow \mathcal{P} \cup x'$
16         end while
17     while $|\mathcal{X}| > 0$
18 end procedure
```

repeated until a terminating condition is met, or alternatively after a predefined number of generations (line 17).

5.3.2 Phenotypic Descriptor

As in the NS and MAP-Elites algorithms, SHINE assesses potential solutions through the measurement of low-dimensional phenotypic traits. The phenotypic trait of a solution, μ , is defined as an n-tuple, $\mu = (a_1, a_2, \ldots, a_n)$. SHINE also requires the bounding area of μ . The bounding area is defined as

$\mathcal{B} = [a_1 : b_1] \times [a_2 : b_2] \times \ldots \times [a_n : b_n]$, where a_n and b_n denote the minimum and maximum possible values for dimension n. $\mathcal{B}$ is used to determine the area of the root vertex in the phenotypic tree, $\mathcal{V}$.

5.3.3 Phenotypic Tree

SHINE maintains an archive of potential genomes in an hierarchical, spatially partitioned tree. The tree structure, $\mathcal{V}$, is created upon initialisation of the main procedure (Algorithm 3). The CreateTree($\mathcal{B}$) method initialises $\mathcal{V}$ with a single root vertex, $\mathcal{V}_r$. The number of dimensions and bounding area of μ are required to initialise $\mathcal{V}_r$.

5.3.3.1 Required Constants

The SHINE algorithm requires 2 pre-defined constants to control the subdivision of $\mathcal{V}$. We define constant α to be the maximum depth of the tree and constant β as the maximum number of points which may fall within a leaf vertex before it is divided. These constants are used to determine both the underlying phenotypic tree structure and the archive of representatives.

5.3.3.2 Tree Subdivision

At each generation, μ is measured for every individual ρ . $\mathcal{V}$ is then queried with μ (Algorithm 3, lines 6-9). The QueryTree(μ, $\mathcal{V}_r$) method controls the subdivision of $\mathcal{V}$ (Algorithm 3, line 8).

The relevant containing vertex for μ within $\mathcal{V}$ is determined as follows. Let v represent the relevant vertex of $\mathcal{V}$. Let the bounding area of v be defined by $\mathcal{B}$, where $\mathcal{B}_{a_1} < \mu_1 \leq \mathcal{B}_{b_1} \wedge \mathcal{B}_{a_n} < \mu_n \leq \mathcal{B}_{b_n}$. Let v_d be the depth within the tree and $|v|$ be the number of descriptors currently assigned to v . If the capacity of v has been exceeded and the maximum depth has not been reached, such that $|v| > \beta \wedge v_d < \alpha$, then v is subdivided into c equal sized regions. The number of regions, c, is determined by the number of dimensions in μ . Let $|\mu|$ represent the number of dimensions of a phenotypic descriptor and let $c = 2^{|\mu|}$. Each vertex of $\mathcal{V}$ will be subdivided into c child vertices (each dimension being split into 2 equal regions). Therefore, 3-dimensional traits ($|\mu| = 3$) would require an octree ($c = 2^3 = 8$) structure.

5.3.3.3 2-Dimensional Implementation

In this section, a 2-dimensional implementation of the algorithm is illustrated. We define a phenotypic descriptor as an ordered pair $\mu = (x, y)$. As $c = 4$, the resulting structure of $\mathcal{V}$, is given as a quadtree (Samet, 1984).

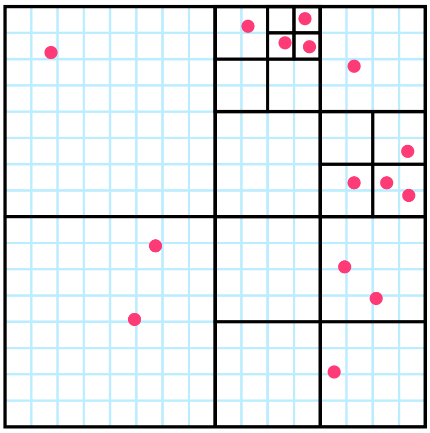


Figure 5.1: *Example quadtree structure*

Figures 5.1 and 5.2 illustrate an example quadtree structure with parameters $\alpha = 4$ and $\beta = 2$. During each generation, all individuals are assessed and $\mathcal{V}$ is queried with μ . Let v represent the relevant vertex of $\mathcal{V}$ for descriptor μ . Let the bounding area $\mathcal{B} = [v_{x1} : v_{x2}] \times [v_{y1} : v_{y2}]$, where $v_{x1} < \mu_x \leq v_{x2} \wedge v_{y1} < \mu_y \leq v_{y2}$. Let v_d be the depth within the tree and $|v|$ be the number of descriptors currently assigned to v . If $|v| > \beta \wedge v_d < \alpha$ then v is subdivided into 4 equal sized regions, i. e., top-left, top-right, bottom-left and bottom-right (TL, TR, BL, BR, Figure 5.2). All descriptors within v are then assigned to their relevant child vertices.

5.3.4 Archive Management

After the tree has been queried by the population, the resulting structure is utilised to determine the distribution of the archive of representatives from which subsequent populations are selected. Membership of the archive is weighted, dependent upon the depth of the representatives' containing vertex. Shallower vertices in the tree structure are assigned more representatives.

Representatives do not alter the structure of the tree, rather the relevant vertex for a potential representative's phenotypic descriptor determines whether it is added to the archive. Let $|\mu|$ represent the dimensions of a phenotypic descriptor and

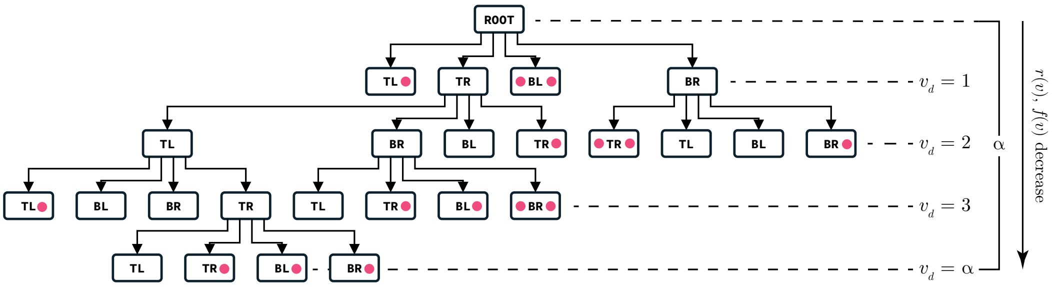


Figure 5.2: *Overview of spatial partitioning in the SHINE archive. ($\alpha = 4, \beta = 2$).*

let $c = 2^{|\mu|}$. Equation [5.1] defines the maximum number of representatives $r(v)$ which may be assigned to a particular vertex.

$$r(v) = (\alpha - v_d + 1)^c \qquad [5.1]$$

The number of representatives within a single vertex will therefore fall within the range $1 \leq r(v) \leq (\alpha + 1)^c$. Let $\mathcal{X}$ be the set of all representative within vertex v . If the capacity of v is reached, such that $|\mathcal{X}| = r(v)$, representatives from $\mathcal{X}$ are selected for addition or removal based upon a sorting function $d(x)$. This sorting function determines the distribution of representatives *within a single leaf vertex*. In alignment with this, let x be a potential representative for inclusion within the archive, where $x \notin \mathcal{X}$. Let $w \in \mathcal{X}$ be the weakest current representative $w = \arg\max_{\forall i \in \mathcal{X}} d(i)$. The updated archive of representatives, which we define as $\mathcal{X}'$, is determined as in Equation [5.2]:

$$\mathcal{X}' = \begin{cases} \mathcal{X} \bigcup x & \text{if } |\mathcal{X}| < r(v) \\ \mathcal{X} & \text{if } |\mathcal{X}| = r(v) \text{ and } d(x) > d(w) \\ \{\mathcal{X} \setminus w\} \bigcup x & \text{if } |\mathcal{X}| = r(v) \text{ and } d(x) \leq d(w) \end{cases} \qquad [5.2]$$

Dependent upon the particular type of search required, various metrics may be proposed. For example, defining $d(x)$ as an objective function would allow the archive to behave in a similar manner to the MAP-Elites algorithm (Mouret and Clune, 2015), selecting elite representatives for inclusion within the phenotypic tree. Additional metrics based upon NS (Lehman and Stanley, 2008) or hybrid novelty-objective measures (Smith et al., 2016a,b) may be of particular interest for further testing of the algorithm in different domains.

In the maze domain experiments, presented in Section 5.4, the *corner* sorting metric is used, a distance function which favours representatives in the outer corners of the containing vertex, encouraging representatives to focus on the areas closest to neighbouring vertices and increasing the chance of mutated offspring to acquire phenotypic traits in neighbouring cells.

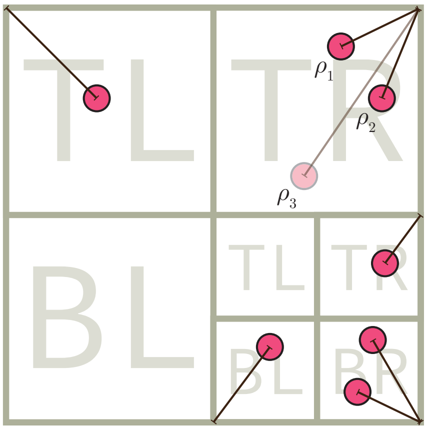


Figure 5.3: *An illustration of the* corner *sorting method for representative selection*

Figure 5.3 illustrates the *corner* method for representative selection. Representatives are sorted by distance from the outer corner of their assigned vertex's position in the quad tree structure (i.e. representatives in top-left vertices are sorted by their distance from top left corner of the vertex). Once the number of representatives exceeds the maximal threshold, as defined in Equation [5.1], the representative with the largest distance is removed.

In the retina experiment, presented in Section 5.5, an objective function is used in replacement of the corner sorting method. This function simply returns the objective fitness of the individual, $d(\rho) = \rho_{fitness}$. Therefore, the representatives within each leaf vertex of the archive are sorted by objective fitness.

5.3.4.1 Proportional Selection

SHINE utilises a traditional roulette wheel method for the selection of offspring. Potential solutions are selected from the complete set of current representatives within the tree $\mathcal{X} = \{\mathcal{X}_{v_1} \bigcup, ..., \bigcup \mathcal{X}_{v_{|\mathcal{V}|}}\}$. The fitness $f(x)$ of a representative x in vertex v is obtained by calculating the reciprocal of the sum of the vertices' depth v_d and its normalised population $\frac{v_p}{\beta}$. Defined as $\frac{1}{(v_d+\frac{v_p}{\beta})}$ and simplified in Equation [5.3].

$$f(x) = \frac{\beta}{\beta v_d + v_p} \qquad [5.3]$$

This fitness assignment results in a lower probability of selection of representatives within smaller (deeper within the tree) and more crowded (higher population) areas of the phenotypic landscape, allowing the search procedure to concentrate on larger and sparser vertices within the tree.

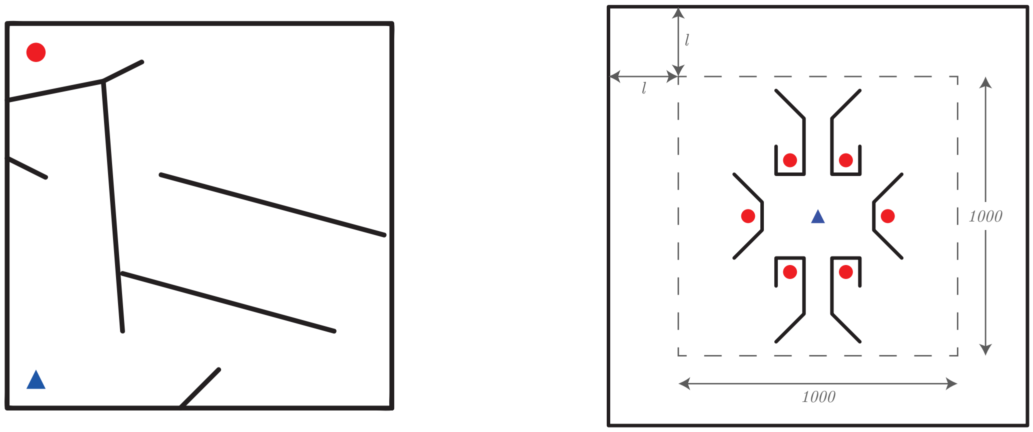
5.4 Experiment 1: Maze Domains

The SHINE algorithm and the HARD maze experiment Section 5.3 and Figure 5.4a were presented at the 14th International Conference on Parallel Problem Solving from Nature, Edinburgh, UK, 2016 in the paper Smith et al. (2016c).

5.4.1 Motivation

The aim of the first experiment is to assess the objective functionality and phenotypic exploration in an evolutionary trajectory optimised with the SHINE algorithm in comparison to NS, MADNS and MAP-Elites (dependent upon the algorithm's relevance to the domain). Therefore, we select domains with either a deceptive objective function or which require a high level of exploration to produce a successful solution.

5.4.2 Domain



(a) *HARD domain*

(b) *SCALED domain*

Figure 5.4: *The HARD & SCALED maze domains. (Triangle indicates agent start position, circle indicates exit)*

As in the experiments used to assess the MADNS algorithm (Sections 4.4 and 4.5), the domain utilised in this experiment is adapted from previous studies which have assessed NS and variants of the algorithm (Gomes et al., 2015, Lehman and Stanley, 2008). In this experiment, two maze domains were tested.

Hard Domain

Initially, the SHINE algorithm was tested within the HARD maze domain, which is classified as a deceptive domain, as shown in Figure 5.4a. The HARD maze is particularly difficult for objectively focussed EAs to reliably find solutions (as shown by the OBJECTIVE algorithm in Section 5.4.9.1).

As in Section 4.4, the HARD maze is of the size 1000 × 1000 units, the agent has a size of 20 units and successfully reaching the exit requires the agent to be within 20 units. Each agent is given 4000 time steps to complete the maze. Populations of 200 controllers were optimised for 1000 generations.

Scaled Domain

As in Section 4.5.1, the SCALED domain is a divergent maze with 6 exits. The distance between the exits and the edges of the maze (Figure 5.4b, $w = 1000 + l$) may be altered to produce a larger search space. Triangle indicates start position. Circles represent exits. In this experiment, only the largest of the sizes tested in Section 4.5 is used. The value tested are $w = 100000$. 50 replicates were performed for each algorithm.

5.4.3 Agent Model

As in Chapter 4, the agent controllers are ANNs which are evolved using the NEAT algorithm Stanley and Miikkulainen (2002). The input and output mappings of the ANNs to the sensors an actuators of the controller are identical to the previous maze domain experiments as previously illustrated in Section 4.4.2.

The simulation was performed using the same software framework as in Chapter 4, extended to include the SHINE algorithm. The implementation is a bespoke domain written in the C++ programming language, developed to be similar to the original maze domain experiments in Lehman and Stanley (2008). The implementation of the NEAT algorithm used was developed as an extension to the MultiNEAT software in the C++ language[1].

[1] © 2012 Peter Chervenski. http://multineat.com/index.html

5.4.4 Objective Fitness and Phenotypic Descriptor

As in previous maze experiments (Gomes et al., 2015, Lehman and Stanley, 2008), the objective fitness of a solution ρ is calculated as $f(\rho) = l - dist(\rho, e)$, where l is the diagonal length of the maze and e is the exit to the maze. Again, in line with previous studies, the MEP_μ phenotypic descriptor is used, calculated from the ending position of the agent, $\text{MEP}_\mu = (\rho_x, \rho_y)$.

5.4.5 α and β Parameters

A series of trial runs in the experimental domain were performed with a range of α and β values: $\alpha = (3, 4, 5, ..., 12, 13, 14)$, $\beta = (20, 40, 60, ..., 120, 140, 160)$. The values $\alpha = 7$ and $\beta = 80$ produced the most reliable and optimal results and are therefore used in our experimental setup (please refer to Appendix B.2). Testing in further domains and with differing population sizes would be required to ascertain whether these values are universally optimal.

5.4.6 Algorithms Tested

Due to the differing nature of the presented domains, a different set of algorithms were used for each domain. Th algorithms which are assessed in each domain are outlined in Table 5.1.

Abbreviation	Full name	HARD	SCALED
NOVELTY	Novelty Search (NS)	✓	✓
MADNS	Multiple Assessment Directed Novelty Search (MADNS)		✓
MAP-ELITES	Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)	✓	
SHINE	Spatial, Hierarchical, Illuminated NeuroEvolution (SHINE)	✓	✓
OBJECTIVE	Traditional single objective fitness	✓	

Table 5.1: *Algorithms tested in HARD and SCALED maze domain experiments*

The HARD maze domain contains a single deceptive solution. The aim of this experiment was to assess the ability of SHINE to overcome deception as effectively as the NS algorithm. The MAP-Elites algorithm was also assessed in this domain to enable comparisons between SHINE. It is unexpected that

MAP-Elites would outperform NS in this domain due to the combination of the deceptive fitness function and aligned phenotypic descriptor; as MAP-Elites stores an elite for each cell within the phenotypic landscape, the alignment of the descriptor to the objective fitness is such that the fitness of the elites will almost directly map to the cell they are placed within. It is, however, of interest to ascertain what effect a deceptive fitness function has upon the performance of the MAP-Elites algorithm.

In the SCALED domain, MAP-Elites could not be included without extending the algorithm to multiple independent objectives, or running the algorithm for each objective individually. However, the aim of this domain was to gain a comparison of the performance of SHINE to the MADNS algorithm presented in Chapter 4. Therefore MAP-Elites is not tested in this domain. A more suitable domain for comparing SHINE and MAP-Elites is presented in Section 5.5. Similarly, traditional single objective fitness was removed from assessment, as the previous results from Section 4.5 show that it performs significantly sub optimally in this domain.

Each algorithm was run for 50 replicates, with a different random seed in each replicate. In order to ensure consistency between algorithms, identical random seed values were given to each of the algorithms in each replicate.

5.4.7 Calculating Performance

The performance of each algorithm was determined by the probability of it to locate all solutions in the domain within a predetermined number of generations. If solutions to all of the exits were not found after 1000 generations, the trial was deemed unsuccessful. The probability of success for each algorithm in each of the domains was calculated by measuring the cumulative probability to discover all solutions to the exits within a single evolutionary run after n generations. The probability of success and the number of generations taken to locate (all of) the exit(s) are presented for the HARD domain in Figures 5.5 and 5.6 and for the SCALED domain in Figures 5.7 and 5.8.

5.4.8 Calculating Diversity

As in Sections 4.4 and 4.5, the diversity was calculated through the measurement of the domain coverage and exploration uniformity (Section 5.4.8). These calculations are briefly reintroduced as follows:

Domain Coverage

The cumulative coverage of the domain is calculated at each generation in the trial over 1000 generations. The domain is divided into a 2-dimensional matrix M, where $|M| = n \times n$. In our presented results, $n = 30$. The final position of an individual (ρ_x, ρ_y) is mapped to the corresponding region of M. Let M' be the set of the regions of M which contain individuals: $M' = \{x : x \in M \wedge |x| > 0\}$. Domain coverage is then calculated as $\frac{|M'|}{|M|}$.

Exploration Uniformity

Let $\mathcal{P}_t$ be the set of individuals in the population at generation t and let Ψ_t be the distribution of $\mathcal{P}_t$ over M. The exploration uniformity of the population, $D(\mathcal{P}_t)$, is calculated as the similarity between Ψ_t and the uniform distribution U. The distance metric used is the JSD. The exploration uniformity for a single generation is therefore defined as in Equation [5.4]:

$$\begin{aligned} D(\mathcal{P}_t) &= 1 - JSD(\Psi_t, U),\ where: \\ \Psi_t &= \left(\frac{|I_1|}{|\mathcal{P}_t|}, \ldots, \frac{|I_{|\mathcal{P}_t|}|}{|\mathcal{P}_t|}\right), I_r = \{i \in \mathcal{P}_t | region(i) = r\} \\ U &= \left(\overbrace{\frac{1}{|M|} \times \cdots \times \frac{1}{|M|}}^{n^2 \text{ times}}\right) \end{aligned} \qquad [5.4]$$

5.4.9 Results

The performance of each of the algorithms are presented for the HARD domain in Figures 5.5 and 5.6 and for the SCALED domain in Figures 5.7 and 5.8. The domain coverage and exploration uniformity are presented in Figures 5.9 and 5.10. As the OBJECTIVE algorithm performs significantly sub-optimally to all other algorithms in both the performance (finding solutions in only 3 of the 50 trials) and diversity measures, and was included solely for illustrative purposes, it is excluded from further discussion.

[1] SHINE, NOVELTY and MAP-ELITES were successful in all trials.

5.4.9.1 Performance

The objective performance of the tested algorithms is presented for the HARD domain in Figures 5.5 and 5.6 and for the SCALED domain in Figures 5.7 and 5.8.

Hard Domain

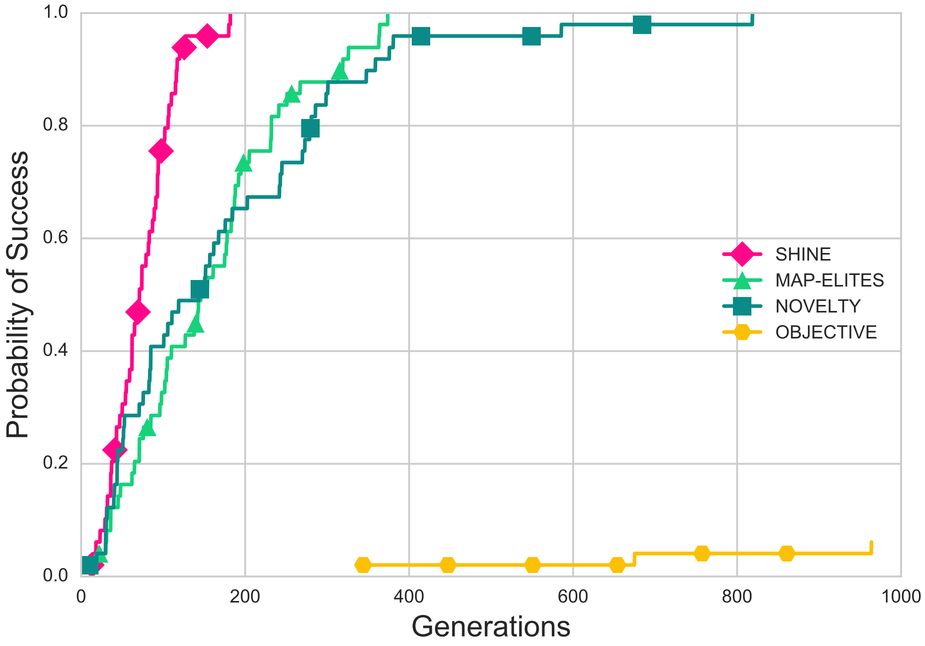


Figure 5.5: *Probability of success in HARD maze domain*

As illustrated in Figure 5.5, all 3 algorithms located solutions to the maze in all 50 trials, resulting in a probability of success of 1.0. The SHINE algorithm locates solutions to all exits in significantly less generations ($p < 0.001$), with the slowest trial locating all solutions after 182 generations, compared with 374 generations for MAP-ELITES and 819 generations for NOVELTY. Both NOVELTY and MAP-ELITES follow a similar gradient of ascent, however NOVELTY requires a higher number of generations to locate a solution in 3 of the trials.

Figure 5.6 shows the number of generations taken to find a successful solution. The SHINE algorithm requires a significantly fewer number of generations, with a median value of 71. MAP-ELITES and NOVELTY achieve similar results, with median values of 146 and 141 generations respectively.

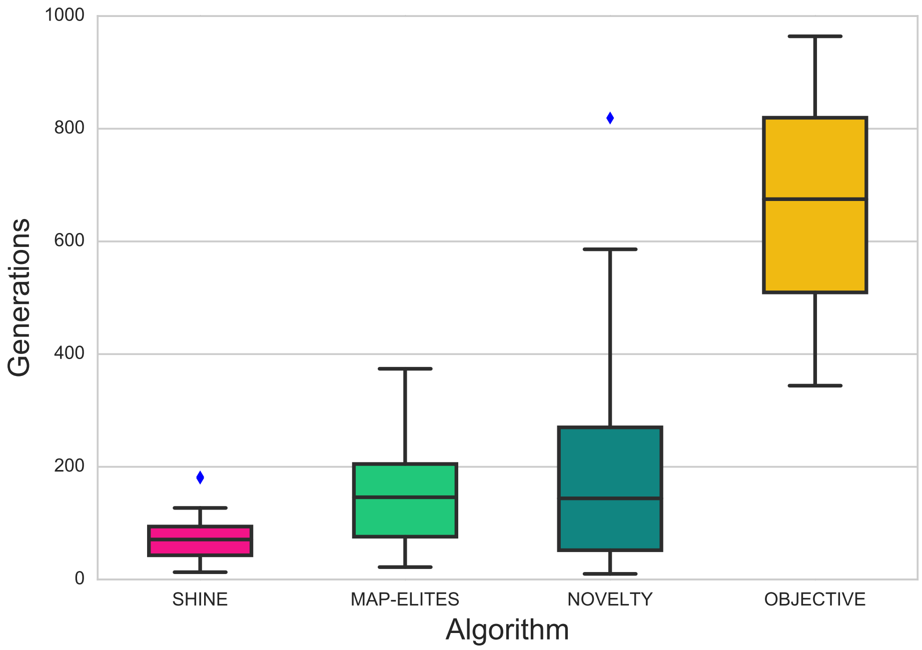


Figure 5.6: *Number of Generations taken to locate the exit in the HARD maze domain in successful trials*

Scaled Domain

As shown in Figure 5.7, both SHINE and MADNS located solutions to all exits in the maze in all 50 replicates, resulting in a probability of success of 1.0. NOVELTY managed to locate all exits in 47 of the 50 replicates. As is expected, MADNS significantly outperforms NOVELTY in this domain, as previously shown in Section 4.5. However, SHINE significantly outperforms both MADNS and NOVELTY. This is perhaps surprising in this domain, as there is no objective pressure applied to the configuration of the archive maintenance of the SHINE algorithm. Maximum probability of success is reached significantly faster ($p < 0.001$) by the SHINE algorithm, after 226 generations, compared with 483 generations for MADNS.

Figure 5.8 shows the number of generation taken to locate all solutions within a single evolutionary run. The median number of generations taken to locate all exits is significantly lower for SHINE ($median = 74$) compared to MADNS ($median = 145$) and for MADNS compared to NOVELTY ($median = 311$).

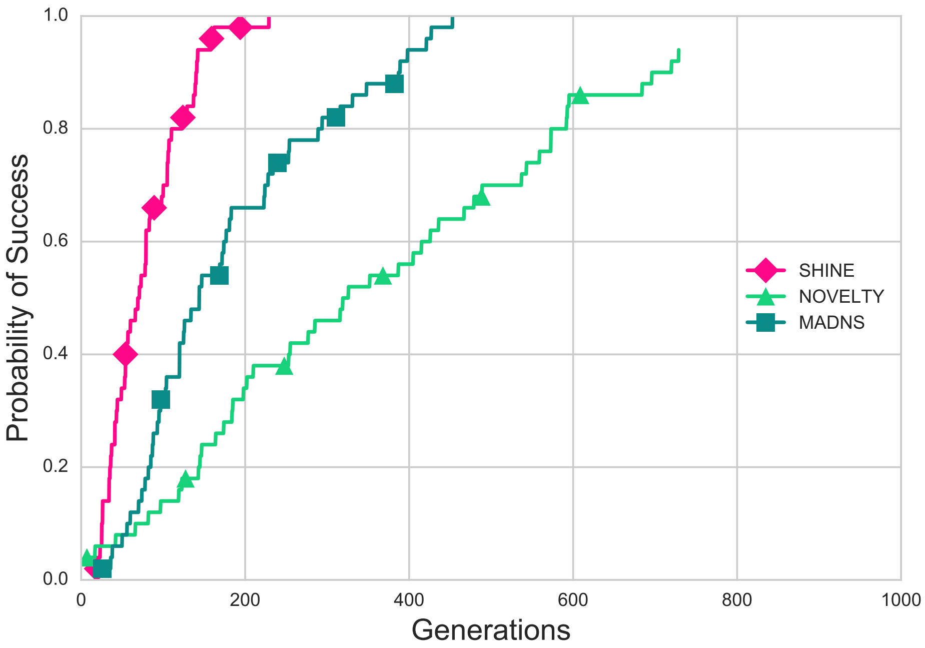


Figure 5.7: *Probability of success in SCALED maze domain*

5.4.9.2 Diversity

The domain coverage and exploration uniformity are presented for the HARD domain in Figures 5.9 and 5.10.

Hard Domain

Figure 5.9 shows the exploration uniformity for each of the algorithms over 1000 generations. The maximum mean level of exploration uniformity is achieved by the SHINE algorithm, 0.51912 after 772 generations. However, it achieves comparably high levels after 232 generations, remaining relatively stable throughout the evolution. Both MAP-ELITES and NOVELTY fail to achieve this maximal level within 1000 generations, however the exploration uniformity is still increasing for both algorithms at the end of the trial. The maximum mean level achieved by MAP-ELITES is 0.50584 after 984 generations. NOVELTY achieves a maximal value of 0.51408 after 988 generations. Therefore an evolutionary run with a higher number of generations may allow MAP-ELITES and NOVELTY to achieve a level of exploration uniformity similar to SHINE.

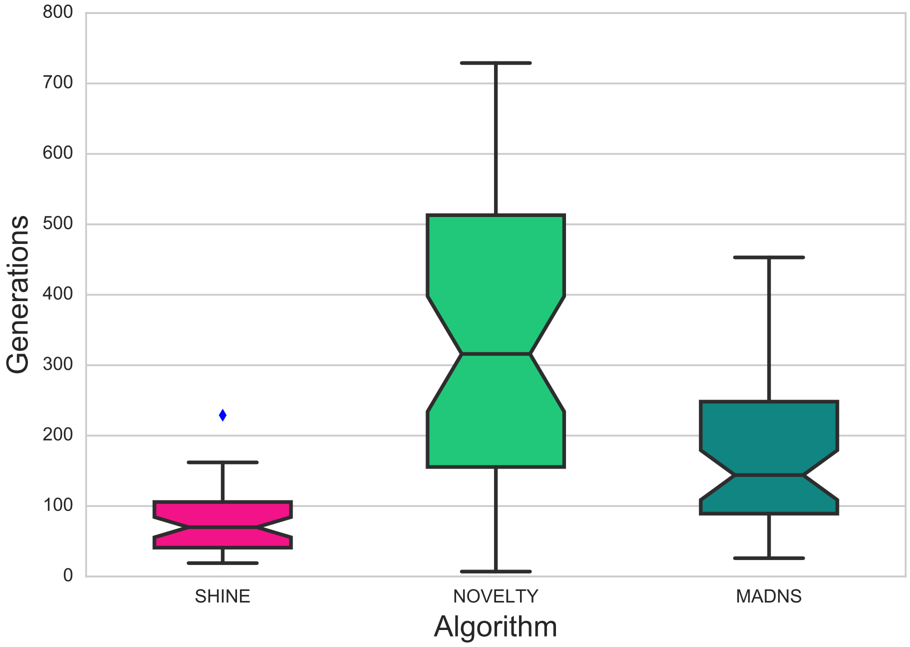


Figure 5.8: *Number of Generations taken to locate all exits in the SCALED maze domain*

Figure 5.10 shows the proportion of the domain covered by the population. All three algorithms produce similar levels of domain coverage for the initial 400 generations. However, beyond this SHINE covers significantly more of the domain than both NOVELTY and MAP-ELITES.

5.4.10 Discussion

The simulated maze domain experiment was selected to assess two aspects of the SHINE algorithm The first domain, the HARD domain was selected to test the ability of SHINE to overcome a deception. Secondly, the SCALED domain was chosen to evaluate the divergent nature of SHINE in large search spaces. The results from the deceptive, HARD maze domain show that the SHINE algorithm outperforms both NS and MAP-Elites, two state of the art algorithms for divergent phenotypic search.

The results from the SCALED maze domain experiment show that SHINE is less effected by increasing domain size than both NS and MADNS. The results show that the hierarchical tree structure and approach taken for archive maintenance and offspring selection in the SHINE algorithm are viable methods for rapid phenotypic exploration.

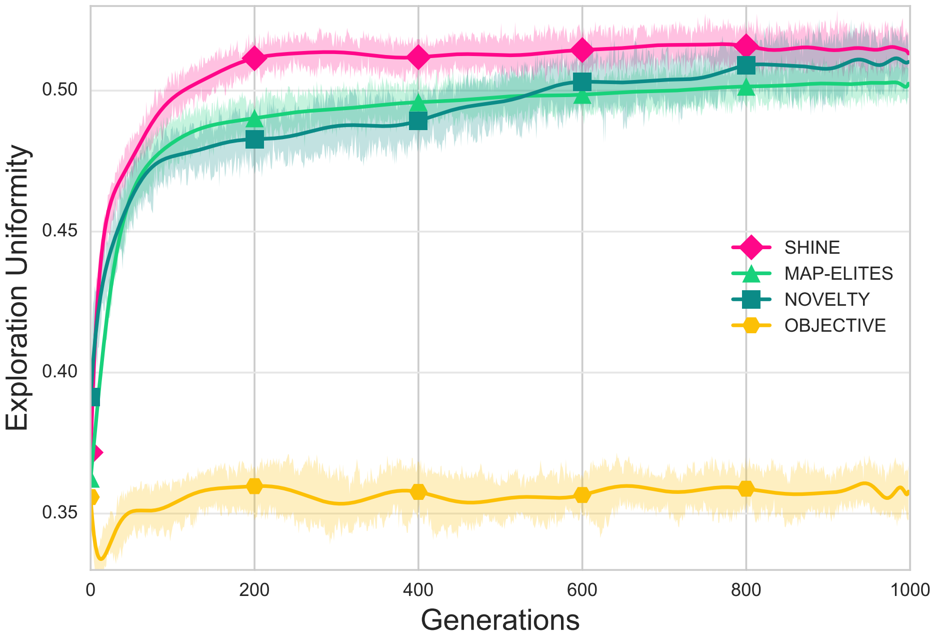


Figure 5.9: *Exploration uniformity of the current population at each generation*

However, further experimental validation is required in order to establish the performance of the SHINE algorithm in domains with a less aligned mapping between the phenotypic descriptor and the objective landscape. Also, the performance of SHINE as an illumination algorithm, in a domain directly comparable to the performance of MAP-Elites would further establish the suitability of the SHINE method for the rapid optimisation of *useful* phenotypes. As suggested in Section 5.3.4 a replacement of the *corner* method applied for archive maintenance in this experiment to an objective function would allow SHINE to be compared more directly with MAP-Elites in a more directly objective-fitness focussed domain. In the following experiment (Section 5.5), the objective fitness archive maintenance method is applied to SHINE to assess its performance as an illumination algorithm, in a domain previously used for the assessment of MAP-Elites (Mouret and Clune, 2015).

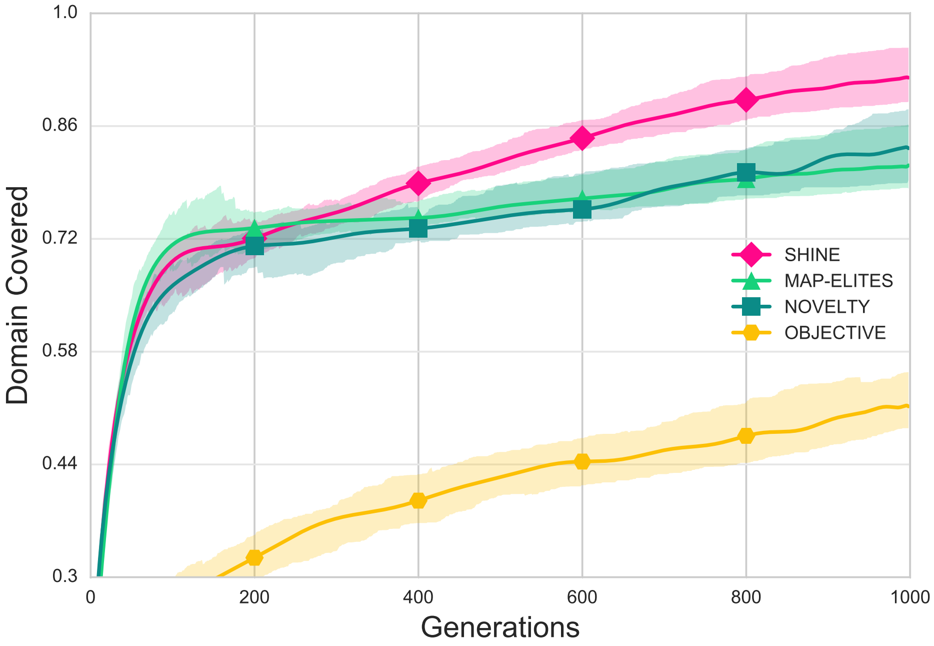


Figure 5.10: *Cumulative proportion of domain coverage by the population after each generation. (Shaded area indicates* 25^{th} *to* 75^{th} *percentiles.)*

5.5 Experiment 2: Network Modularity

5.5.1 Motivation

The archive maintenance method in applied to the SHINE algorithm in the maze domain experiment (Section 5.4) was selected for its ability to perform rapid divergent search within the phenotypic landscape. However, SHINE was developed for application as an illumination algorithm, which aim to highlight areas of the phenotypic landscape that contain solutions with high objective fitness. It is unlikely that, without direct selection pressure upon objective fitness, SHINE would be able to perform well as an illumination algorithm. Therefore, the *corner* distance metric was replaced with an objective fitness assessment for the maintenance of the archive within each leaf vertex of the tree.

In order to assess the suitability of SHINE as an illumination algorithm, an experimental domain was selected that has been previously applied for the assessment of the MAP-Elites algorithm (Mouret and Clune, 2015). Alongside this, the criteria used for the evaluation of the algorithms within this experiment are suggested by Mouret and Clune as giving a good indication of the overall performance of an illumination algorithm (Mouret and Clune, 2015).

The experimental domain selected for this experiment is the *artificial retina* domain, in which an ANN representing two retinas is optimised to activate when both of the retinas are presented with a number of predefined patterns. Due to the nature of the abstraction within this domain, the phenotypic descriptor used is the MCC_μ phenotypic descriptor, which measures two dimensions of the underlying ANN structure, a less aligned phenotype than the MEP_μ descriptor used within the maze domains.

In the following sections, the retina experiment is presented. In Section 5.5 a description of the domain is given, with the underlying ANN topology outlined in Section 5.5.3.1. The assessment criteria, *global performance*, *global reliability*, *precision* and *coverage*, as suggested by Mouret and Clune (2015), are formally defined in Section 5.5.4.

The results for each of these measurements are presented in Section 5.5.5. Finally, the findings from this experiment are discussed, with further suggestions given, in Section 5.5.6.

5.5.2 Algorithms tested

This domain was selected to evaluate the performance of SHINE as an illumination algorithm; in this domain, the SHINE algorithm is configured to use the fitness sorting method for archive maintenance (Section 5.3.4). The algorithms tested within this domain are identical to the experiment as presented in Mouret and Clune (2015) with the addition of the proposed SHINE algorithm. However, in this experiment random sampling was removed due to its suboptimal performance (Mouret and Clune, 2015). Therefore, the algorithms tested in this domain are MAP-Elites (MAP-ELITES), NS-LC (NS-LC), SHINE (SHINE) and objective fitness (OBJECTIVE). As in Mouret and Clune (2015), Novelty Search with Local Competition (NS-LC) is tested in this domain due to its ability to perform as an illumination algorithm. the NS-LC algorithm, previously described in Section 2.8.3.3, is briefly reintroduced as follows.

Novelty Search with Local Competition

The aim of NS-LC is to optimise individuals with high objective fitness within local phenotypic niches. The local competitiveness of an individual, $f_{lc}(\rho)$ is calculated as the number of individuals with a lower objective fitness than ρ, i. e., the cardinality of the dominated set $\mathcal{D}$, as given in Equation [5.5]:

$$\begin{aligned} \mathcal{D} &= \{x \in \mathcal{X}_{knn} \,|\, f_{obj}(\rho) > f_{obj}(x)\}, \\ f_{lc}(\rho) &= |\mathcal{D}|\,. \end{aligned} \tag{5.5}$$

NS-LC is then applied through the MOO of $max\ (f_{nov}(\rho), f_{lc}(\rho))$.

5.5.3 Domain

The artificial retina domain is identical to one used in Mouret and Clune (2015), which, in turn, is adapted from Clune et al. (2013) and Kashtan and Alon (2005). The following description of the experimental domain is adapted from Clune et al. (2013). The problem in the retina experiment involves an ANN that receives stimuli from eight inputs (Figure 5.12).

The left four inputs can be thought of as the left retina, and the final four inputs as the right retina (Figure 5.12). Patterns given the left and right retina may contain an *object*, meaning a pattern of interest. The full set of *objects* are

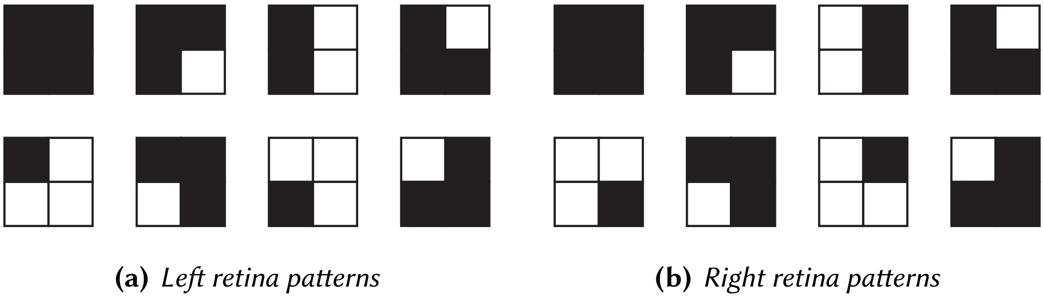


Figure 5.11: *Left & right retina patterns*

illustrated in Figure 5.11. The set of patterns for the left (Figure 5.11a) and right (Figure 5.11b) retinas contain both identical and different objects. The ANNs are evolved to activate when both retinas are presented with an object of interest.

5.5.3.1 Network Structure

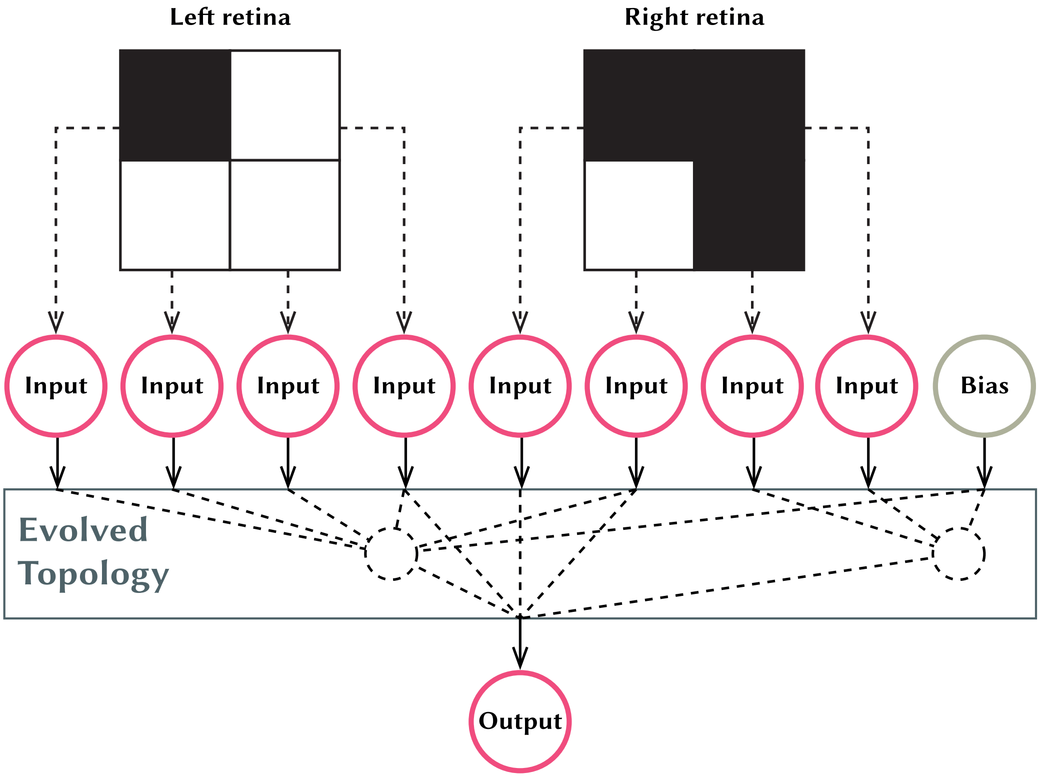


Figure 5.12: *Retina neural network topology*

The ANNs are optimised using the NEAT algorithm (Stanley and Miikkulainen, 2002). Each ANN has eight inputs, the first four represent the left retina, and the last four the right retina. There is one output, for which the activation level is used to determine whether the ANN has classified both retinas as containing objects.

5.5.3.2 Objective Fitness and Phenotypic Descriptor

In the experiment, each ANN is given all 256 possible input patterns. The ANN must answer true (> 0) or false (< 0) as to whether both retinas contain objects of interest. The objective fitness is measured as the percentage of correct answers for all of the 256 inputs.

The phenotypic descriptor used in this domain is the same as in Mouret and Clune (2015), the MCC_μ phenotypic descriptor. This is a 2-dimensional descriptor in which the first dimension is calculated as the squared sum of the connection weights in the ANN. The second dimension is the modularity of the ANN. The modularity of an ANN is measured using Mouret's implementation[2], in the C++ programming language, of an efficient approximation of Newman's modularity score (Leicht and Newman, 2008).

5.5.4 Measurement Criteria

The criteria used for measuring the performance of the algorithms in this experiment are selected as suggest by Mouret and Clune (2015). These measurements were developed to specifically assess the overall performance of an illumination algorithm. A successful illumination algorithm should be capable of not only optimising high performing global solutions, but also high performing solutions throughout the phenotypic landscape. Alongside this, an illumination algorithm should have the ability to explore as much of the phenotypic landscape as is theoretically possible. Therefore, the measurements used are *global performance*, *global reliability*, *precision* and *coverage*, which are defined in the following sections.

[2] `https://github.com/jbmouret/network_toolbox`

5.5.4.1 Global Performance

Global performance is measured as the maximum fitness located at any cell in a single run. The map m represents a map from a single run. Therefore, the global performance of a single run, represented as $P(m)$, is defined as in Equation [5.6]:

$$P(m) = \max\{m_{x,y} \mid [x_{min} : x_{max}], [y_{min} : y_{max}]\} \qquad [5.6]$$

5.5.4.2 Global Reliability

Global reliability is calculated as follows. We define the optimal map as $\mathcal{O}$, which represents the matrix containing the highest performing solutions for all trials at each cell of the phenotypic landscape. Therefore, $\mathcal{O}_{x,y}$ is the highest performing solution located at cell (x, y). Let $\mathcal{M}$ represent the complete set of final maps from all runs of the experiment, where m is a map from a single run. Thus global reliability of a map, defined as $R(m)$, is calculated as in Equation [5.7]:

$$\begin{aligned} \mathcal{O}_{x,y} &= \max_{m \in \mathcal{M}} m_{x,y}, \\ R(m) &= \frac{1}{|\mathcal{O}|} \cdot \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} \frac{m_{x,y}}{\mathcal{O}_{x,y}}. \end{aligned} \qquad [5.7]$$

5.5.4.3 Precision

Precision is measured in the same manner as global reliability, however only cells for which a phenotype was located in that run are averaged against. As stated by Mouret and Clune (2015): “This measure addresses the following question: when a cell is filled, how high-performing is the solution relative to what is possible for that cell?”.

Let $\mathcal{N}$ represent the map containing solutions which were located in the run m, where $\mathcal{N} = \{\forall m_{x,y} \mid [x_{min} : x_{max}], [y_{min} : y_{max}], filled(m_{x,y}) = 1\}$. The function $filled(m_{x,y}) \mapsto \{0, 1\}$ returns the value 1 if the cell $m_{x,y}$ has a solution within it, and a 0 otherwise. The cardinality of the filled map, $|\mathcal{N}|$ is used to represent the number of cells for which solutions were located in that run.

Formally, the precision of a map m is represented by $Pr(m)$, and defined as in Equation [5.8]:

$$Pr(m) = \frac{1}{|\mathcal{N}|} \cdot \sum_{x,y} \frac{m_{x,y}}{\mathcal{O}_{x,y}}, \qquad [5.8]$$

where $x,\, y = \{[x_{min} : x_{max}], [y_{min} : y_{max}] \,|\, filled(m_{x,y}) = 1\}$.

5.5.4.4 Coverage

The domain coverage is calculated by dividing the number of cell located in a single run by the total number of cells which are theoretically possible to locate. As noted by, Mouret and Clune (2015), the total number of possible cells is impossible to determine prior to the experiment, therefore an approximation is made by counting the number of cells located by any run in any treatment. Using the notation from the previous sections, for a single run m, the filled map is represented as $\mathcal{N}$. Therefore, coverage represented as $C(m)$, is defined as in Equation [5.9]:

$$C(m) = \frac{|\mathcal{N}|}{|\mathcal{O}|} \qquad [5.9]$$

	MAP-ELITES	NS-LC	OBJECTIVE	SHINE
mean	96.250000	92.363281	95.483569	94.882812
std	2.871615	3.042906	2.380780	2.773452
min	90.625000	87.500000	92.015598	90.234375
25%	94.531250	90.527344	93.799162	92.968750
50% (median)	95.703125	91.796875	95.038887	94.335938
75%	98.437500	95.312500	97.099207	96.875000
max	100.000000	96.875000	100.000000	100.000000

Table 5.2: *Global performance in artificial retina experiment*

5.5.5 Results

In the following sections, the results for each of the criteria defined in Section 5.5.4 are presented. In Section 5.5.5.1 the results for global performance are given. The results for global reliability, precision and coverage are presented in Sections 5.5.5.2 to 5.5.5.4 receptively. Alongside this, heat-maps which illustrate the objective performance over the full phenotypic landscape are presented in Section 5.5.5.5.

5.5.5.1 Global Performance

As shown in Figure 5.13 and Table 5.2 There is no significant difference between the final global performance of MAP-ELITES and SHINE $(p = 0.0574)$. The median global performance achieved for both SHINE and MAP-ELITES is near optimal, SHINE $=$ 96.875 and MAP-ELITES $=$ 95.703. As is shown, both algorithms perform near optimally in all runs. Out of the 20 replicates performed for each algorithm, both SHINE and MAP-ELITES achieved the maximum objective fitness score of 100 in 6 replicates. The minimum objective fitness achieved by SHINE $=$ 92.185, and slightly higher by MAP-ELITES $=$ 93.750. NS-LCis the only algorithm which does not manage to achieve maximal performance in any replicate $(max = 96.875)$.

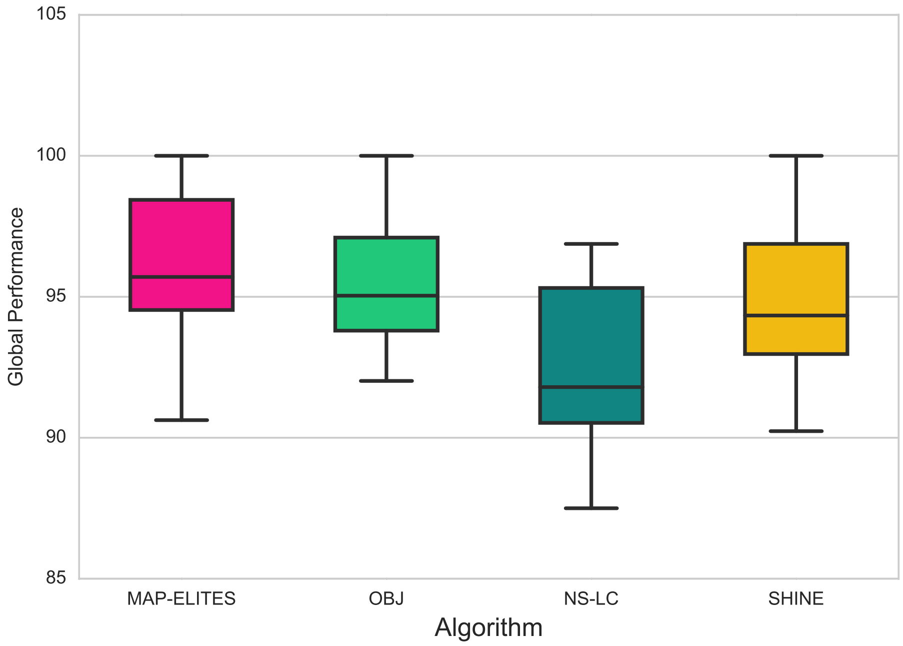


Figure 5.13: *Global performance in RETINA domain*

	MAP-ELITES	NS-LC	OBJECTIVE	SHINE
mean	0.288109	0.143508	0.085059	0.698124
std	0.008099	0.020115	0.013504	0.115281
min	0.271564	0.106040	0.061722	0.491806
25%	0.282107	0.126392	0.079380	0.613639
50% (median)	0.289589	0.146805	0.084479	0.678436
75%	0.293963	0.155233	0.091542	0.793096
max	0.299620	0.185350	0.112341	0.941037

Table 5.3: *Reliability in artificial retina experiment*

5.5.5.2 Global Reliability

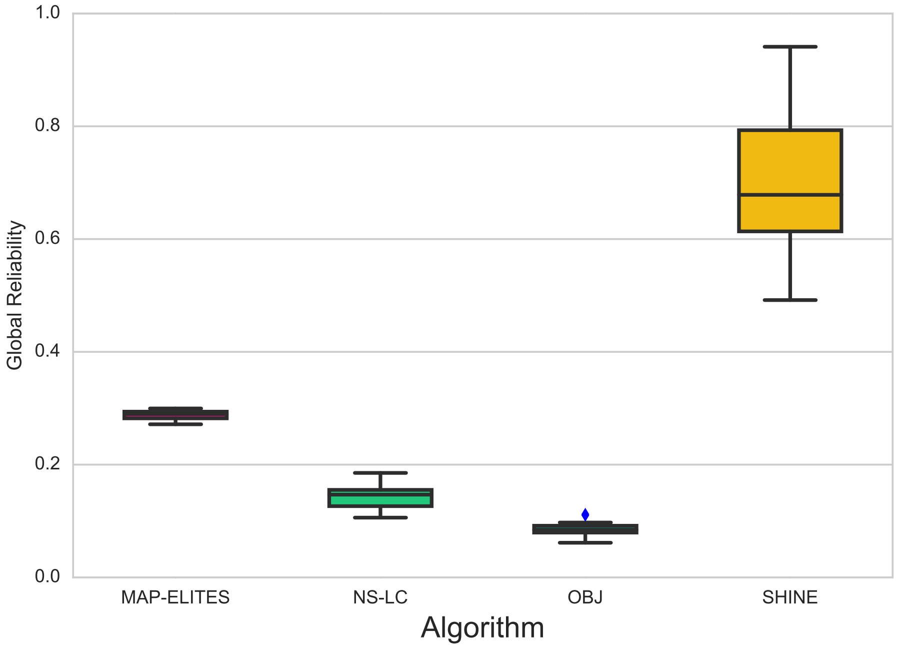


Figure 5.14: *Reliability in RETINA domain*

As shown in Figure 5.14, SHINE achieves a significantly higher level of global reliability than MAP-ELITES ($p = 3.0199 \times 10^{-11}$). The median global reliability achieved for SHINE $=$ 0.7331. The median value achieved by MAP-ELITES $=$ 0.2850. MAP-ELITES significantly outperforms NS-LC ($p < 0.001$). All algorithms significantly outperform OBJECTIVE. A breakdown of the results for reliability are given in Table 5.3.

	MAP-ELITES	NS-LC	OBJECTIVE	SHINE
mean	0.906338	0.889557	0.598279	0.929396
std	0.009027	0.021969	0.046864	0.020716
min	0.884232	0.848441	0.551382	0.895551
25%	0.901324	0.875866	0.555955	0.914301
50% (median)	0.907659	0.886315	0.589989	0.926891
75%	0.911991	0.907601	0.633650	0.939185
max	0.921680	0.928508	0.683617	0.970434

Table 5.4: *Precision in artificial retina experiment*

5.5.5.3 Precision

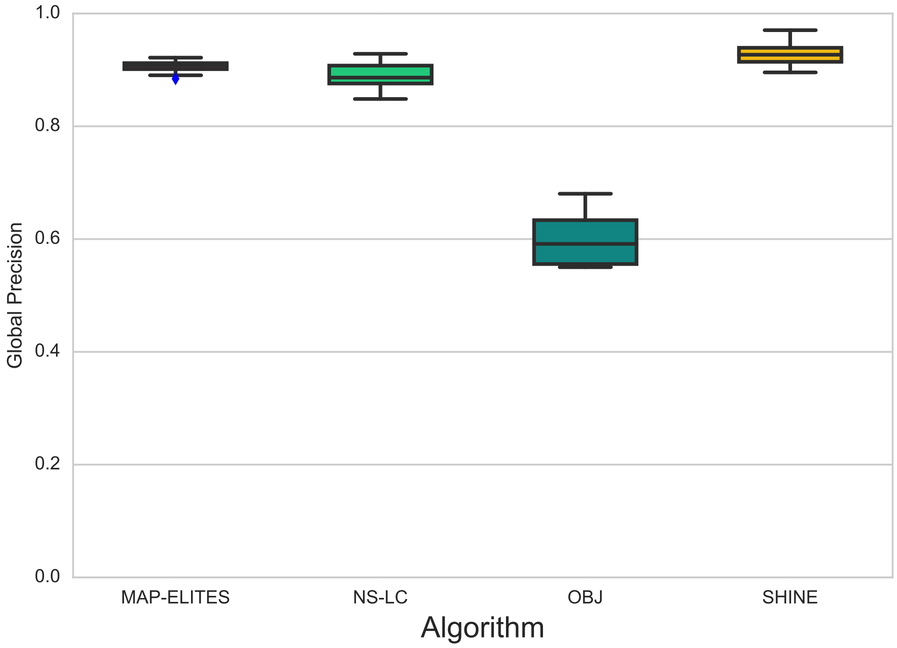


Figure 5.15: *Precision in RETINA domain*

As shown in Figure 5.15, SHINE achieves a significantly higher level of precision than both MAP-ELITES and NS-LC ($p = 2.6098 \times 10^{-10}$). The median precision value achieved for SHINE = 0.9253. The median value achieved by MAP-ELITES = 0.8886. There is no significant difference between MAP-ELITES and NS-LC. MAP-ELITES, NS-LC and SHINE all significantly outperform OBJECTIVE. A breakdown of the results for precision are given in Table 5.4.

	MAP-ELITES	NS-LC	OBJECTIVE	SHINE
mean	0.326182	0.161430	0.142040	0.750233
std	0.038613	0.023075	0.019362	0.116220
min	0.298652	0.120657	0.110499	0.538067
25%	0.308881	0.144722	0.127492	0.666366
50% (median)	0.321151	0.164373	0.143250	0.745111
75%	0.326395	0.177853	0.155069	0.853783
max	0.485381	0.210272	0.174767	0.984526

Table 5.5: *Coverage in artificial retina experiment*

5.5.5.4 Coverage

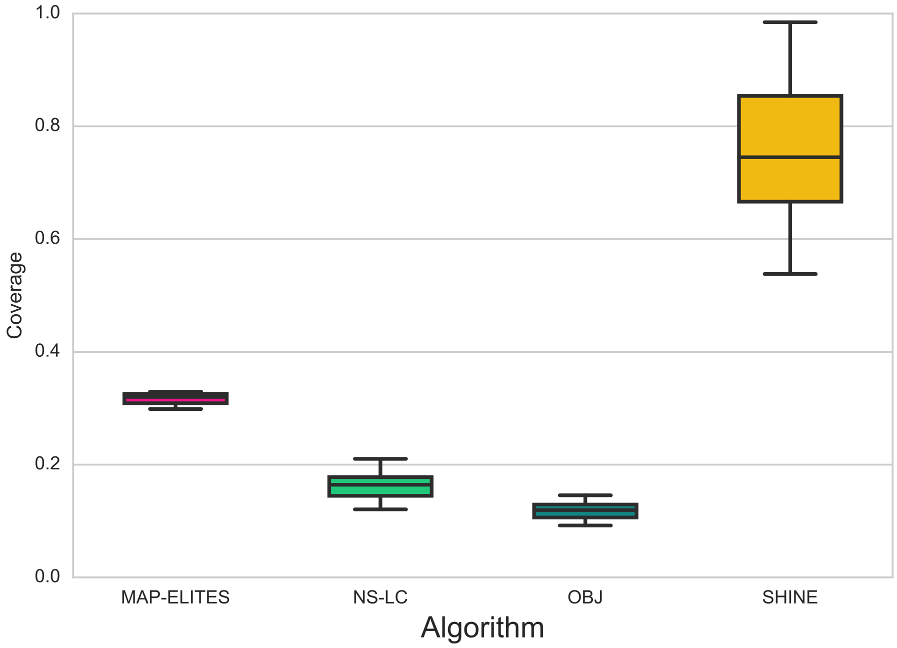


Figure 5.16: *Coverage in RETINA domain*

As shown in Figure 5.16, SHINE achieves a significantly higher level of coverage than MAP-ELITES ($p = 3.0198 \times 10^{-11}$). MAP-ELITES significantly outperforms both NS-LC and OBJECTIVE ($p < 0.001$). The median precision value achieved for SHINE $= 0.7909$. The median value achieved by MAP-ELITES $= 0.3213$. Both NS-LC and OBJECTIVE achieve significantly lower levels of coverage than MAP-ELITES and SHINE. A breakdown of the results for precision are given in Table 5.4.

5.5.5.5 Phenotypic Landscape Coverage

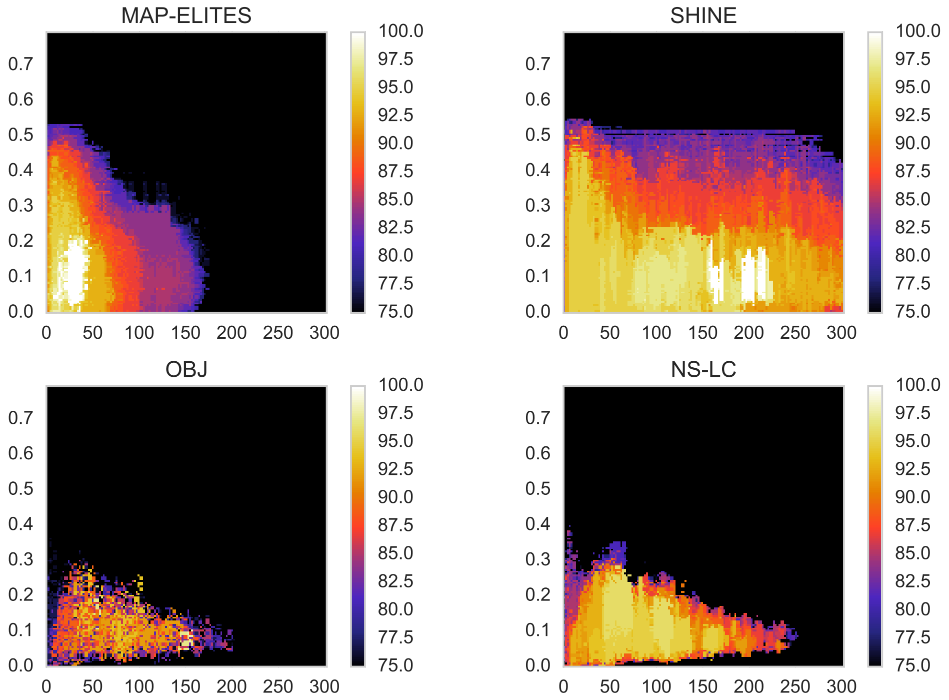


Figure 5.17: *Phenotypic landscape coverage in RETINA domain*

As shown by the heatmaps presented in Figure 5.17, SHINE covers significantly more of the phenotypic landscape than MAP-ELITES and NS-LC. MAP-ELITES has a higher level of coverage than NS-LC, although NS-LC manages to locate high performing individuals in areas which are not reached by MAP-ELITES. NS-LC and OBJECTIVE achieve similar levels of coverage, however the level of coverage by NS-LC is more complete and with more high performing individuals. Surprisingly, SHINE manages to optimise areas of the landscape outside the bounding area of the archive for the connection cost dimension. The maximum connection cost achieved by SHINE is 587, compared with 162 for MAP-Elites. The bounding area for connection cost is the range $[0:300]$.

5.5.6 Discussion

As shown by the results in Section 5.5.5, when configured with and objective fitness sorting metric, the SHINE algorithm may be suitably applied as an illumination algorithm. The objective performance of both SHINE and MAP-Elites is similar, however, SHINE manages to search a larger area of the phenotypic landscape. In doing so, SHINE *illuminates* areas of the phenotypic landscape which contain high performing individuals, undiscovered by MAP-Elites. SHINE also optimises phenotypes in areas of the landscape discovered by MAP-Elites more successfully.

Interestingly, SHINE manages to uncover a high performing area of the phenotypic landscape undiscovered by either MAP-Elites or NS-LC. As shown in Figure 5.17, the area where connection cost ($ccost$) is approximately $(210 < ccost < 230)$ shows optimal performing individuals. Alongside this, there are areas of the landscape which MAP-Elites discovers, but does not optimise fully $(40 < ccost < 110)$. Therefore, SHINE illuminates the areas of the phenotypic landscape which are potentially of high objective fitness more successfully than MAP-Elites

5.6 Implications

In this chapter, a novel method for rapid exploration of low dimensional feature spaces was introduced. The SHINE algorithm applies an hierarchical tree structure to explore sparsely visited areas of the phenotypic landscape. In order to asses the performance of SHINE, a number of experimental domains were selected.

Initially, in Section 5.4, the ability of SHINE to overcome deception was evaluated. As shown by the results in Section 5.4.9.1, SHINE overcomes deception, optimising solutions in significantly less time than both NS and MAP-Elites.

A second maze domain with multiple possible exits was tested within, in order to establish the divergent nature of search performed by SHINE in domains with large phenotypic landscapes. Surprisingly, the SHINE algorithm located solutions to all exits significantly faster than both NS and the MADNS extension introduced within this thesis.

Finally, in Section 5.5, the corner sorting method for archive maintenance within SHINE was replaced with an objective fitness function, in order to establish the suitability of SHINE as an illumination algorithm. In the retina domain, previously used to asses illumination algorithms (Clune et al., 2013, Mouret and Clune, 2015), the SHINE algorithm was shown to outperform both MAP-Elites and NS-LC in a number of the criteria established for the measurement of illumination algorithms.

As is shown in all of the experimental work presented in this chapter, the SHINE algorithm shows great promise for potential application as an illumination algorithm. However, there is still further validation required to fully establish the algorithm's performance. MAP-Elites has been shown to be extendible to the real world application of robot controllers Cully et al. (2015). As the results from the experiments presented in this chapter show, there is no reason why this should not also be the case for the SHINE algorithm. Therefore, the assessment of SHINE beyond simulation, in real world domains, would be an interesting future assessment of the algorithm.

6 Conclusions

6.1 Illuminations

Producing sets of solutions with both high quality and diversity has been show to be a useful technique for the optimisation of controllers in deceptive and complex domains (Mouret and Clune, 2015). The introduction of NS allowed researchers within the field of ER to begin to explore EAs inspired by IML, which has shown great promise in its application to ML. However, the complete abandonment of objectives proposed in NS has been shown to be potentially problematic — too extreme in its removal of objective fitness. The phenotypic descriptor used when measuring novelty may somewhat assist in producing useful solutions; when NS is applied using a phenotypic descriptor that is reasonably well aligned to determining the usefulness of a behaviour, it has shown to be capable of avoiding deception and producing multiple diverse strategies to objectives (Lehman and Stanley, 2008).

A further, previously unexplored, characteristic of the NS algorithm was exploited within this thesis. An extension to NS to assist the algorithm in the simultaneous optimisation of multiple independent objectives was introduced. The MADNS algorithm was shown to remedy the problems faced by optimisation through NS in large landscapes, allowing search to focus upon multiple independent areas of interest without causing a significant restriction to the divergent nature of NS. The MADNS extension also does not suffer from negatively effecting opposing objectives, even if the individual objectives are directly conflicting. Further assistance to objective search in unbounded domains was also introduced, through the development of the MCMADNS algorithm, a variant of MADNS applied to MCNS.

The development of NS-LC reintroduced the concept of objective fitness to NS, however the algorithm focusses within local phenotypic niches, and still suffers from the computationally expensive measurement metric utilised by NS. Developing from the promising nature of NS-LC, the term *illumination* algorithm was introduced to focus the area of IML in EC upon the optimisation of pheontypicaly diverse solutions with high objective functionality.

The MAP-Elites algorithm was the first such dedicated illumination algorithm, and however much promise the algorithm shows, by admission of its authors, there are a number of limitations in its initial definition (Mouret and Clune, 2015). In this thesis, some of these limitations were explored and improved upon through the development of a novel illumination algorithm, SHINE. Through the application of a commonly used method in computational geometry for collision detection, hierarchical spatial partitioning, a novel algorithm was developed to allow the search procedure to focus upon sparsely visited areas of the phenotypic landscape, while applying a fine grain, objectively focused search within these areas. The SHINE algorithm, introduced in Chapter 5 was shown to outperform MAP-Elites, NS and NS-LC in domains with phenotypic descriptors that measured either the behaviour of the controller, or the underlying topology of the ANN.

In the remainder of this chapter, the thesis is concluded as follows. Initially, the scope of the research presented in this thesis with relation to the initial research questions is presented in Section 6.2 followed by implications for the application of the presented research in Section 6.3. Next, the contributions made within this thesis are summarised in Section 6.4. Potential avenues for future developments of the work presented within this thesis are suggested in Section 6.5. Finally, in Section 6.6, the closing remarks are given.

6.2 Scope of Research

This thesis aimed to provide a development into the application of IML to EC. Through the study of two state of the art algorithms utilised for the promotion of phenotypic diversity in EC, the unique characteristics of divergent phenotypic search were explored. Initially, in Section 3.2, a MOO of NS and objective fitness was applied to a video game domain in order to assess the suitability of such an approach to produce diverse sets of high performing agents. An experiment was designed in order to answer the following two research questions:

> Can NS be combined with objective fitness in order to produce a range of high performing solutions which also exhibit high phenotypic diversity?

> Does the addition of NS have a negative effect upon the objective fitness of the solutions?

It was shown that, although NS produces highly diverse behaviours, without the addition of objective fitness, performance is severely negatively affected. However, dependent upon an ideal mixing ratio, the linear combination of NS and objective fitness was shown to be a successful method for producing diverse sets of controllers without negative effect to objective fitness.

Following this, a simple logic gates experiment was designed in order to assess the ability of NS to simultaneously optimise multiple independent objectives. The following research question was tested by aiming to observe if a population optimised through NS would produce controllers for a complete set of logic gates.

> Does a population optimised with NS produce solutions to multiple independent objectives?

The experiment showed that NS was capable of optimising multiple independent objectives. However, due to its divergent exploratory nature, NS has been previously shown to struggle in optimising solutions as the size of a search space increases, (Cuccu and Gomez, 2011). It would also be expected that, as the search space increases, the ability of NS to locate multiple objectives would be negatively affected. Therefore, in Chapter 4, a number of proposed methods for the MOO of NS and objective fitness for optimising multiple independent objectives were assessed in a range of maze domains in order to answer the following question:

> Can the combination of novelty and objective search assist with the optimisation of multiple independent objectives as the size of the search space increases?

It was shown, that as the size of the domain increased, NS became increasingly unable to optimise multiple independent objectives. The MADNS algorithm, introduced in Section 4.3, was shown to become more effective than NS as the size of the domain increased. A further extension, the MCMADNS algorithm was introduced for unbounded domains, and was shown to outperform MCNS.

Following this, the promotion of phenotypic diversity was applied to a more recently introduced class of algorithms, the *illumination* algorithm. A series of maze navigation experiments were undertaken in Section 5.4 and an artificial retina experiment in Section 5.5, in order to address the following:

> Can the direct promotion of phenotypic exploration be beneficial to the intended functionality of an illumination algorithm?

A state of the art algorithm, MAP-Elites was extended in order to directly promote phenotypic diversity. The SHINE algorithm, introduced in Section 5.3. SHINE was shown to outperform both MAP-Elites and NS in the criteria expressed in Mouret and Clune (2015) as being effective for determining a successful illumination algorithm.

6.3 Applications

The research presented in this thesis has shown the potential for the application of IML to EC. Most specifically, it has been shown that in domains where the overall goal is to produce diverse sets of high performing solutions, the methods presented within this research are highly suitable. There are numerous domains in which such an approach is of interest. Perhaps the most obvious is the domain of video games, in which producing agents which have diverse strategies of play is of paramount importance for player enjoyment. Beyond this domain, there are also numerous, but perhaps less obvious applications of such methods An application to route finding could be easily imagined, in which diverse routes were optimised to provide a new morning walk to work each day, making it more enjoyable and less repetitive.

There is also the application of such approaches to robotics, in which the fragility of engineered controllers is of concern when applied to complex domains. It has also been shown that the promotion of phenotypic diversity is a beneficial method for overcoming deception. In complex domains, when the design of an objective fitness function is difficult, the application of the algorithms presented in this thesis may be of benefit. Illumination algorithms such as MAP-Elites have been shown to perform successfully in robotic tasks such as damage control (Mouret and Clune, 2015), therefore the SHINE algorithm could be potentially applied to such tasks.

The presented algorithms in this thesis may be applicable to any domain in which there is an uncertain, deceptive and large search spaces, or when phenotypic diversity is required alongside objective functionality.

6.4 Summary of Contributions

In summary, the contributions presented in this thesis are as follows. Initially, an experiment was presented in which a linearly combined MOO of NS and objective fitness was applied to a video game domain. The experiment demonstrated the applicability of such a method for the optimisation of diverse and useful game playing agent controllers. Following results from a simple logic gate domain, the capacity for NS to simultaneously optimise multiple independent, conflicting objectives was highlighted as a potentially useful characteristic of the algorithm previously unexplored in the literature. Therefore, in Chapter 4, the MADNS algorithm, a novel extension to NS for the simultaneous optimisation of multiple independent and potentially conflicting objectives in large phenotypic landscapes, was introduced. MADNS was tested in a number of simulated maze navigation tasks with multiple exits, both partially and directly conflicting in objective assessment. The MADNS algorithm was shown to outperform NS as the size of the maze increases. Alongside this, an extension to MCNS, the MCMADNS algorithm was presented for unbounded domains.

A novel *illumination algorithm* for rapid exploration of low dimensional phenotypic landscapes was introduced in Chapter 5. The SHINE algorithm was shown to outperform two current state-of-the-art illumination algorithms, MAP-Elites and NS-LC in a series of maze navigation domains, and an artificial retina experiment. The SHINE algorithm was shown to explore large phenotypic landscapes in fewer generations, and with higher *quality-diversity* measures, than MAP-Elites.

6.5 Further Work

The SHINE algorithm has been shown to b e promising addition to the class of illumination algorithms. The algorithm, as introduced in this thesis, applies widely applied methods from computational geometry to develop upon the work of Mouret and Clune (2015). The SHINE algorithm has been shown to outperform the current state of the art illumination algorithms in its current form, however, there are a number of suggestions which could be made for further potential improvements and developments to the algorithm. The focussed hierarchical nature of the algorithm opens up new possibilities for a distributed population approach; as each leaf vertex within the tree in SHINE contains an archive of multiple locally similar phenotypes, these vertices could be used to spawn separate populations, each being a new root node within a local SHINE tree. Although this would be computationally expensive, the parallel nature of the procedure could allow each local SHINE tree to be optimised in isolation of the others. This parallelistion, would however may require further consideration for phenotypes produced within a local niche that express behaviours outside of their niche. Therefore, to avoid redundancy and replication in the production of solutions, a communication system between local instances may need to be established.

Another limitation of the SHINE tree structure is the explosive nature of the archive with the increase of dimensions in the phenotypic descriptor. Although it has been shown that a high dimensional phenotypic descriptor is unnecessary for the divergent search to optimise useful behaviour, in more complex domains, higher dimensional phenotypic descriptors may be required. This problem of extending to higher dimensional phenotypic landscapes is also faced by the MAP-Elites algorithm There has already been some attempt to remedy this problem, the recent extension to the MAP-Elites algorithm, currently in pre-publication in , applies Voroni to allow subdivision of the archive to higher dimensions without the explosive characteristic of the spatial partitioning method applied in SHINE and MAP-Elites.

IN this thesis, two different archive management procedures were suggested, the *corner distance* metric, and objective fitness. Although these metrics were specifically selected for the relevant domain within the experiments, they are not domain specific metrics. However, a metric which combines the the exploration based corner sorting method and the exploitation based objective fitness assessment may be more domain agnostic.

The domains presented in the experimental work within Chapter 5 were selected for their wide use in the assessment of similar algorithms. However, one of the most important measures for illumination algorithms, within ER, is their suitability for application beyond simulation. The MAP-Elites algorithm has been shown to be successful when applied to physical robots, and due to the performance of the SHINE algorithm in the simulated domains presented, it would be expected for SHINE to be similarly successful. However, testing within physical robots would need to done in order for this to be fully established.

6.6 Closing Remarks

The power of IML to assist with the capabilities of AI beyond directly objective policies is still embryonic in development, however, even at this early phase, has shown large potential for the future development of autonomous systems and ML. The parallel research within the field of ER into the divergent search for potentially useful phenotypic traits shows similar promise.

Within this thesis, this recent area of research has been furthered through both the exploitation of a useful, previously unexplored, characteristic of NS and the introduction of a novel illumination algorithm, capable of utilising a rapid and fine-grained exploration of low dimensional phenotypic landscapes for the optimisation of high quality solutions.

Although to fully *abandon objectives* may have been an extreme position to adopt, it certainly uncovered many of the potential stepping stones towards overcoming deception and open-ended evolutionary processes. The considered reintroduction of objective function as a parameter, through the illumination algorithm, has further shown the potential for this area of ER. Once such algorithms have reached the maturity to be applied in real-world situations, which is increasingly the case, their full potential may begin to be explored. As this thesis has shown, the application of IML to EC is a promising area of research, which could potentially extend the capabilities of future autonomous systems far beyond the current approaches.

References

Hussein A. Abbass and Kalyanmoy Deb. Searching Under Multi-evolutionary Pressures. In *Proceedings of the 2Nd International Conference on Evolutionary Multi-criterion Optimization*, EMO'03, pages 391–404, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 978-3-540-01869-8. URL http://dl.acm.org/citation.cfm?id=1760102.1760135.

Gianluca Baldassarre and Marco Mirolli, editors. *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013a. ISBN 978-3-642-32374-4 978-3-642-32375-1. URL http://link.springer.com/10.1007/978-3-642-32375-1.

Gianluca Baldassarre and Marco Mirolli. Intrinsically Motivated Learning Systems: An Overview. In Gianluca Baldassarre and Marco Mirolli, editors, *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 1–14. Springer Berlin Heidelberg, 2013b. ISBN 978-3-642-32374-4 978-3-642-32375-1. URL http://link.springer.com/chapter/10.1007/978-3-642-32375-1_1. DOI: 10.1007/978-3-642-32375-1_1.

Mark A. Bedau, Emile Snyder, and Norman H. Packard. A Classification of Long-term Evolutionary Dynamics. In *Proceedings of the Sixth International Conference on Artificial Life*, ALIFE, pages 228–237, Cambridge, MA, USA, 1998. MIT Press. ISBN 978-0-262-51099-8. URL http://dl.acm.org/citation.cfm?id=286139.286165.

D. E. Berlyne. Novelty and Curiosity as Determinants of Exploratory Behaviour1. *British Journal of Psychology. General Section*, 41(1-2):68–80, September 1950. ISSN 2044-8295. doi: 10.1111/j.2044-8295.1950.tb00262.x. URL http://onlinelibrary.wiley.com/doi/10.1111/j.2044-8295.1950.tb00262.x/abstract.

D. E. Berlyne. Novelty, complexity, and hedonic value. *Perception & Psychophysics*, 8(5):279–286, 1970. ISSN 0031-5117, 1532-5962. doi: 10.3758/BF03212593. URL http://link.springer.com/article/10.3758/BF03212593.

D. E Berlyne. *Aesthetics and psychobiology*. Appleton-Century-Crofts, New York, 1971. ISBN 978-0-390-08670-9. OCLC: 239942.

Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies – a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002. ISSN 1572-9796. doi: 10.1023/A:1015059928466. URL http://dx.doi.org/10.1023/A:1015059928466.

Lam T. Bui, Jürgen Branke, and Hussein A. Abbass. Diversity As a Selection Pressure in Dynamic Environments. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, GECCO '05, pages 1557–1558, New York, NY, USA, 2005. ACM. ISBN 978-1-59593-010-1. doi: 10.1145/1068009.1068257. URL http://doi.acm.org/10.1145/1068009.1068257.

Luigi Cardamone, Daniele Loiacono, and Pier Luca Lanzi. Evolving Competitive Car Controllers for Racing Games with Neuroevolution. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pages 1179–1186, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-325-9. doi: 10.1145/1569901.1570060. URL http://doi.acm.org/10.1145/1569901.1570060.

Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson. The evolutionary origins of modularity. In *Proc. R. Soc. B*, volume 280, page 20122863. The Royal Society, 2013.

Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 45(3): 35, 2013.

Mihaly Csikszentmihalyi. Flow and the psychology of discovery and invention. *New Yprk: Harper Collins*, 1996.

Giuseppe Cuccu and Faustino Gomez. When Novelty is Not Enough. In *Proceedings of the 2011 International Conference on Applications of Evolutionary Computation - Volume Part I*, EvoApplications'11, pages 234–243, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-20524-8. URL http://dl.acm.org/citation.cfm?id=2008402.2008430.

Antoine Cully and Jean-Baptiste Mouret. Behavioral Repertoire Learning in Robotics. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 175–182, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1963-8. doi: 10.1145/2463372.2463399. URL http://doi.acm.org/10.1145/2463372.2463399.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, May 2015. ISSN 0028-0836. doi: 10.1038/nature14422. URL http://www.nature.com/nature/journal/v521/n7553/full/nature14422.html.

Dipankar Dasgupta and Douglas R McGregor. Designing application-specific neural networks using the structured genetic algorithm. In *Combinations of Genetic Algorithms and Neural Networks, 1992., COGANN-92. International Workshop on*, pages 87–96. IEEE, 1992.

Kenneth Alan De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, Ann Arbor, MI, USA, 1975. AAI7609381.

Kalyanmoy Deb. Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evol. Comput.*, 7(3):205–230, September 1999. ISSN 1063-6560. doi: 10.1162/evco.1999.7.3.205. URL http://dx.doi.org/10.1162/evco.1999.7.3.205.

Kalyanmoy Deb and David E Goldberg. An investigation of niche and species formation in genetic function optimization. In *Proceedings of the 3rd international conference on genetic algorithms*, pages 42–50. Morgan Kaufmann Publishers Inc., 1989.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

Stéphane Doncieux and Jean-Baptiste Mouret. Dynamic Behavioral Diversity. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '12, pages 1453–1454, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1178-6. doi: 10.1145/2330784.2330986. URL http://doi.acm.org/10.1145/2330784.2330986.

Stéphane Doncieux, Jean-Baptiste Mouret, Nicolas Bredeche, and Vincent Padois. Evolutionary Robotics: Exploring New Horizons. In Stéphane Doncieux, Nicolas Bredèche, and Jean-Baptiste Mouret, editors, *New Horizons in Evolutionary Robotics*, number 341 in Studies in Computational Intelligence, pages 3–25. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-18271-6 978-3-642-18272-3. URL http://link.springer.com/chapter/10.1007/978-3-642-18272-3_1. DOI: 10.1007/978-3-642-18272-3_1.

Agoston E Eiben and Cornelis A Schippers. On evolutionary exploration and exploitation. *Fundamenta Informaticae*, 35(1-4):35–50, 1998.

Dario Floreano and Francesco Mondada. Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-network Driven Robot. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior : From Animals to Animats 3: From Animals to Animats 3*, SAB94, pages 421–430, Cambridge, MA, USA, 1994. MIT Press. ISBN 978-0-262-53122-1. URL http://dl.acm.org/citation.cfm?id=189829.189962.

L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial Intelligence Through Simulated Evolution.* John Wiley & Sons, 1966. URL https://books.google.co.uk/books?id=QMLaAAAAMAAJ.

Tobias Friedrich, Pietro S. Oliveto, Dirk Sudholt, and Carsten Witt. Theoretical Analysis of Diversity Mechanisms for Global Exploration. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, GECCO '08, pages 945–952, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-130-9. doi: 10.1145/1389095.1389276. URL http://doi.acm.org/10.1145/1389095.1389276.

David Goldberg, Kalyanmoy Deb, and Bradley Korb. Messy genetic algorithms: Motivation, analysis, and first results. *Complex systems*, (3):493–530, 1989.

David E. Goldberg and Jon Richardson. Genetic Algorithms with Sharing for Multimodal Function Optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pages 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc. ISBN 978-0-8058-0158-3. URL http://dl.acm.org/citation.cfm?id=42512.42519.

Jorge Gomes and Anders L. Christensen. Generic Behaviour Similarity Measures for Evolutionary Swarm Robotics. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 199–206, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1963-8. doi: 10.1145/2463372.2463398. URL http://doi.acm.org/10.1145/2463372.2463398.

Jorge Gomes, Paulo Urbano, and Anders Lyhne Christensen. Progressive Minimal Criteria Novelty Search. In Juan Pavón, Néstor D. Duque-Méndez, and Rubén Fuentes-Fernández, editors, *Advances in Artificial Intelligence – IBERAMIA 2012*, number 7637 in Lecture Notes in Computer Science, pages 281–290. Springer Berlin Heidelberg, November 2012. ISBN 978-3-642-34653-8 978-3-642-34654-5. URL http://link.springer.com/chapter/10.1007/978-3-642-34654-5_29. DOI: 10.1007/978-3-642-34654-5_29.

Jorge Gomes, Pedro Mariano, and Anders Lyhne Christensen. Avoiding Convergence in Cooperative Coevolution with Novelty Search. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 1149–1156, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-2738-1. URL http://dl.acm.org/citation.cfm?id=2617388.2617428.

Jorge Gomes, Pedro Mariano, and Anders Lyhne Christensen. Devising Effective Novelty Search Algorithms: A Comprehensive Empirical Study. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 943–950, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3472-3. doi: 10.1145/2739480.2754736. URL http://doi.acm.org/10.1145/2739480.2754736.

Faustino J. Gomez. Sustaining Diversity Using Behavioral Information Distance. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pages 113–120, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-325-9. doi: 10.1145/1569901.1569918. URL http://doi.acm.org/10.1145/1569901.1569918.

Daniele Gravina, Antonios Liapis, and Georgios N Yannakakis. Surprise search: Beyond objectives and novelty. In *Proceedings of the Genetic and Evolutionary Computation Conference. ACM*, 2016.

Georges Harik. Finding multimodal solutions using restricted tournament selection. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, 1995.

Inman Harvey. *The artificial evolution of adaptive behaviour.* PhD thesis, University of Sussex, 1993.

E. J. Hastings, R. K. Guha, and K. O. Stanley. Evolving content in the Galactic Arms Race video game. In *2009 IEEE Symposium on Computational Intelligence and Games*, pages 241–248, September 2009. doi: 10.1109/CIG.2009.5286468.

Matthew Hausknecht, Piyush Khandelwal, Risto Miikkulainen, and Peter Stone. Hyperneat-ggp: A hyperneat-based atari general game player. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 217–224. ACM, 2012.

Matthew Hausknecht, Joel Lehman, Risto Miikkulainen, and Peter Stone. A neuroevolution approach to general atari game playing. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(4):355–366, 2014.

R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on*, pages 593–605 vol.1, 1989. doi: 10.1109/IJCNN.1989.118638.

F. Herrera, M. Lozano, and J. L. Verdegay. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artif. Intell. Rev.*, 12(4):265–319, August 1998. ISSN 0269-2821. doi: 10.1023/A:1006504901164. URL http://dx.doi.org/10.1023/A:1006504901164.

John H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975. URL http://doi.apa.org/psycinfo/1975-26618-000.

Clark Leonard Hull. *Principles of Behavior: An Introduction to Behavior Theory.* D. Appleton-Century Company, Incorporated, 1943. Google-Books-ID: 6WB9AAAAMAAJ.

W Johannsen. The Genotype Conception Of Heredity. *The American Naturalist*, 45(531):129–159, 1911.

Nadav Kashtan and Uri Alon. Spontaneous evolution of modularity and network motifs. *Proceedings of the National Academy of Sciences of the United States of America*, 102(39):13773–13778, 2005.

Duško Katić and Miomir Vukobratović. Survey of Intelligent Control Techniques for Humanoid Robots. *J. Intell. Robotics Syst.*, 37(2):117–141, June 2003. ISSN 0921-0296. doi: 10.1023/A:1024172417914. URL http://dx.doi.org/10.1023/A:1024172417914.

Maryam Mahsal Khan, Arbab Masood Ahmad, Gul Muhammad Khan, and Julian F Miller. Fast learning neural networks using cartesian genetic programming. *Neurocomputing*, 121:274–289, 2013.

Joshua D. Knowles, Richard A. Watson, and David Corne. Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, EMO '01, pages 269–283, London, UK, UK, 2001. Springer-Verlag. ISBN 978-3-540-41745-3. URL http://dl.acm.org/citation.cfm?id=647889.736521.

John R Koza. Gene duplication to enable genetic programming to concurrently evolve both the architecture and work-performing steps of a computer program. In *Proc. of the 14th International Joint Conference on Artificial Intelligence*. Citeseer, 1995.

Joel Lehman. *Evolution through the Search for Novelty*. PhD thesis, University of Central Florida, 2012. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.348.7148&rep=rep1&type=pdf.

Joel Lehman and Kenneth O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the Eleventh International Conference on Artificial Life (Alife XI*. MIT Press, 2008.

Joel Lehman and Kenneth O. Stanley. Efficiently Evolving Programs Through the Search for Novelty. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 837–844, New York, NY, USA, 2010a. ACM. ISBN 978-1-4503-0072-8. doi: 10.1145/1830483.1830638. URL http://doi.acm.org/10.1145/1830483.1830638.

Joel Lehman and Kenneth O. Stanley. Revising the Evolutionary Computation Abstraction: Minimal Criteria Novelty Search. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 103–110, New York, NY, USA, 2010b. ACM. ISBN 978-1-4503-0072-8. doi: 10.1145/1830483.1830503. URL http://doi.acm.org/10.1145/1830483.1830503.

Joel Lehman and Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011a.

Joel Lehman and Kenneth O. Stanley. Evolving a Diversity of Virtual Creatures Through Novelty Search and Local Competition. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 211–218, New York, NY, USA, 2011b. ACM. ISBN 978-1-4503-0557-0. doi: 10.1145/2001576.2001606. URL http://doi.acm.org/10.1145/2001576.2001606.

Joel Lehman and Kenneth O. Stanley. Novelty Search and the Problem with Objectives. In Rick Riolo, Ekaterina Vladislavleva, and Jason H. Moore, editors, *Genetic Programming Theory and Practice IX*, Genetic and Evolutionary Computation, pages 37–56. Springer New York, 2011c. ISBN 978-1-4614-1769-9 978-1-4614-1770-5. URL http://link.springer.com/chapter/10.1007/978-1-4614-1770-5_3. DOI: 10.1007/978-1-4614-1770-5_3.

Joel Lehman, Kenneth O. Stanley, and Risto Miikkulainen. Effective Diversity Maintenance in Deceptive Domains. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 215–222, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1963-8. doi: 10.1145/2463372.2463393. URL http://doi.acm.org/10.1145/2463372.2463393.

Elizabeth A Leicht and Mark EJ Newman. Community structure in directed networks. *Physical review letters*, 100(11):118703, 2008.

Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. Enhancements to Constrained Novelty Search: Two-population Novelty Search for Generating Game Content. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 343–350, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1963-8. doi: 10.1145/2463372.2463416. URL http://doi.acm.org/10.1145/2463372.2463416.

Samir W Mahfoud. *Niching methods for genetic algorithms.* PhD thesis, University of Illinois, 1995.

Andrew P Martin. Increasing genomic complexity by gene duplication and the origin of vertebrates. *The American Naturalist*, 154(2):111–128, 1999.

Paul Martin, Paul Patrick Gordon Bateson, and Patrick Bateson. *Measuring behaviour: an introductory guide.* Cambridge University Press, 1993. URL https://books.google.co.uk/books?hl=en&lr=&id=XFocGI3cgG4C&oi=fnd&pg=PR11&ots=5Vj5tiiY9L&sig=g5uQSsl9X3TinCi27_qrjG0ePyY.

Henok Mengistu, Joel Lehman, and Jeff Clune. Evolvability search: Directly selecting for evolvability in order to study and produce it. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, pages 141–148, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4206-3. doi: 10.1145/2908812.2908838. URL http://doi.acm.org/10.1145/2908812.2908838.

Jean-Arcady Meyer, Phil Husbands, and Inman Harvey. Evolutionary Robotics: A Survey of Applications and Problems. In *Proceedings of the First European Workshop on Evolutionary Robotics*, pages 1–21, London, UK, UK, 1998. Springer-Verlag. ISBN 978-3-540-64957-1. URL http://dl.acm.org/citation.cfm?id=645403.651831.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

David E. Moriarty and Risto Miikkulainen. Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, (AI94-224):11–32, 1996. URL http://nn.cs.utexas.edu/?moriarty:mlj96.

Hirotaka Moriguchi and Shinichi Honiden. Sustaining Behavioral Diversity in NEAT. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 611–618, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0072-8. doi: 10.1145/1830483.1830595. URL http://doi.acm.org/10.1145/1830483.1830595.

J. B. Mouret and S. Doncieux. Encouraging Behavioral Diversity in Evolutionary Robotics: An Empirical Study. *Evol. Comput.*, 20(1):91–133, March 2012. ISSN 1063-6560. doi: 10.1162/EVCO_a_00048. URL http://dx.doi.org/10.1162/EVCO_a_00048.

Jean-Baptiste Mouret. Novelty-Based Multiobjectivization. In Stéphane Doncieux, Nicolas Bredèche, and Jean-Baptiste Mouret, editors, *New Horizons in Evolutionary Robotics*, number 341 in Studies in Computational Intelligence, pages 139–154. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-18271-6 978-3-642-18272-3. URL http://link.springer.com/chapter/10.1007/978-3-642-18272-3_10. DOI: 10.1007/978-3-642-18272-3_10.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv:1504.04909 [cs, q-bio]*, April 2015. URL http://arxiv.org/abs/1504.04909. arXiv: 1504.04909.

Jean-Baptiste Mouret and Stéphane Doncieux. Overcoming the Bootstrap Problem in Evolutionary Robotics Using Behavioral Diversity. In *Proceedings of the Eleventh Conference on Congress on Evolutionary Computation*, CEC'09, pages 1161–1168, Piscataway, NJ, USA, 2009a. IEEE Press. ISBN 978-1-4244-2958-5. URL http://dl.acm.org/citation.cfm?id=1689599.1689754.

Jean-Baptiste Mouret and Stéphane Doncieux. Using Behavioral Exploration Objectives to Solve Deceptive Problems in Neuro-evolution. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pages 627–634, New York, NY, USA, 2009b. ACM. ISBN 978-1-60558-325-9. doi: 10.1145/1569901.1569988. URL http://doi.acm.org/10.1145/1569901.1569988.

Enrique Naredo and Leonardo Trujillo. Searching for Novel Clustering Programs. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 1093–1100, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1963-8. doi: 10.1145/2463372.2463505. URL http://doi.acm.org/10.1145/2463372.2463505.

Stefano Nolfi and Dario Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Bradford Company, Scituate, MA, USA, 2004. ISBN 978-0-262-64056-5.

Charles Ollion and Stéphane Doncieux. Why and How to Measure Exploration in Behavioral Space. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 267–274, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0557-0. doi: 10.1145/2001576.2001613. URL http://doi.acm.org/10.1145/2001576.2001613.

P. Y. Oudeyer, F. Kaplan, and V. V. Hafner. Intrinsic Motivation Systems for Autonomous Mental Development. *Trans. Evol. Comp*, 11(2):265–286, April 2007a. ISSN 1089-778X. doi: 10.1109/TEVC.2006.890271. URL http://dx.doi.org/10.1109/TEVC.2006.890271.

Pierre-Yves Oudeyer, Frederic Kaplan, Pierre-Yves Oudeyer, and Frederic Kaplan. What is intrinsic motivation? A typology of computational approaches. *Frontiers in Neurorobotics*, 1:6, 2007b. doi: 10.3389/neuro.12.006.2007. URL http://journal.frontiersin.org/article/10.3389/neuro.12.006.2007/full.

Alain Pétrowski. A clearing procedure as a niching method for genetic algorithms. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 798–803. IEEE, 1996.

M. Preuss, A. Liapis, and J. Togelius. Searching for good and diverse game levels. In *2014 IEEE Conference on Computational Intelligence and Games*, pages 1–8, August 2014. doi: 10.1109/CIG.2014.6932908.

Justin K. Pugh, L. B. Soros, Paul A. Szerlip, and Kenneth O. Stanley. Confronting the Challenge of Quality Diversity. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 967–974, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3472-3. doi: 10.1145/2739480.2754664. URL http://doi.acm.org/10.1145/2739480.2754664.

Justin K. Pugh, L. B. Soros, and Kenneth O. Stanley. An Extended Study of Quality Diversity Algorithms. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, GECCO '16 Companion, pages 19–20, New York, NY, USA, 2016a. ACM. ISBN 978-1-4503-4323-7. doi: 10.1145/2908961.2909000. URL http://doi.acm.org/10.1145/2908961.2909000.

Justin K. Pugh, L. B. Soros, and Kenneth O. Stanley. Searching for Quality Diversity When Diversity is Unaligned with Quality. In Julia Handl, Emma Hart, Peter R. Lewis, Manuel López-Ibáñez, Gabriela Ochoa, and Ben Paechter, editors, *Parallel Problem Solving from Nature – PPSN XIV*, number 9921 in Lecture Notes in Computer Science, pages 880–889. Springer International Publishing, September 2016b. ISBN 978-3-319-45822-9 978-3-319-45823-6. URL http://link.springer.com/chapter/10.1007/978-3-319-45823-6_82. DOI: 10.1007/978-3-319-45823-6_82.

Sebastian Risi, Sandy D. Vanderbleek, Charles E. Hughes, and Kenneth O. Stanley. How Novelty Search Escapes the Deceptive Trap of Learning to

Learn. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pages 153–160, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-325-9. doi: 10.1145/1569901.1569923. URL http://doi.acm.org/10.1145/1569901.1569923.

Sebastian Risi, Charles E Hughes, and Kenneth O Stanley. Evolving Plastic Neural Networks with Novelty Search. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 18(6):470–491, December 2010. ISSN 1059-7123. doi: 10.1177/1059712310379923. URL http://dx.doi.org/10.1177/1059712310379923.

Richard M. Ryan and Edward L. Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, pages 54–67, 2000a.

Richard M. Ryan and Edward L. Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1):68–78, 2000b. ISSN 1935-990X 0003-066X. doi: 10.1037/0003-066X.55.1.68.

Hanan Samet. The Quadtree and Related Hierarchical Data Structures. *ACM Comput. Surv.*, 16(2):187–260, June 1984. ISSN 0360-0300. doi: 10.1145/356924.356930. URL http://doi.acm.org/10.1145/356924.356930.

B. Sareni and L. Krahenbuhl. Fitness Sharing and Niching Methods Revisited. *Trans. Evol. Comp*, 2(3):97–106, September 1998. ISSN 1089-778X. doi: 10.1109/4235.735432. URL http://dx.doi.org/10.1109/4235.735432.

J. D. Schaffer, D. Whitley, and L. J. Eshelman. Combinations of genetic algorithms and neural networks: a survey of the state of the art. In *Combinations of Genetic Algorithms and Neural Networks, 1992., COGANN-92. International Workshop on*, pages 1–37, Jun 1992. doi: 10.1109/COGANN.1992.273950.

Jürgen Schmidhuber. Maximizing Fun by Creating Data with Easily Reducible Subjective Complexity. In Gianluca Baldassarre and Marco Mirolli, editors, *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 95–128. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-32374-4 978-3-642-32375-1. URL http://link.springer.com/chapter/10.1007/978-3-642-32375-1_5. DOI: 10.1007/978-3-642-32375-1_5.

Jimmy Secretan, Nicholas Beato, David B D'Ambrosio, Adelein Rodriguez, Adam Campbell, Jeremiah T Folsom-Kovarik, and Kenneth O Stanley. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403, 2011.

Fernando Silva, Miguel Duarte, Luís Correia, Sancho Moura Oliveira, and Anders Lyhne Christensen. Open issues in evolutionary robotics. *Evolutionary computation*, 24(2):205–236, 2016.

B. F. Skinner. *Science And Human Behavior.* Simon and Schuster, 1953. ISBN 978-0-02-929040-8. Google-Books-ID: Pjjknd1HREIC.

Davy Smith, Laurissa Tokarchuk, and Chrisantha Fernando. Evolving Diverse Strategies Through Combined Phenotypic Novelty and Objective Function Search. In Antonio M. Mora and Giovanni Squillero, editors, *Applications of Evolutionary Computation*, number 9028 in Lecture Notes in Computer Science, pages 344–354. Springer International Publishing, April 2015. ISBN 978-3-319-16548-6 978-3-319-16549-3. URL http://link.springer.com/chapter/10.1007/978-3-319-16549-3_28. DOI: 10.1007/978-3-319-16549-3_28.

Davy Smith, Laurissa Tokarchuk, and Geraint Wiggins. Exploring Conflicting Objectives with MADNS: Multiple Assessment Directed Novelty Search. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, GECCO '16 Companion, pages 23–24, New York, NY, USA, 2016a. ACM. ISBN 978-1-4503-4323-7. doi: 10.1145/2908961.2908975. URL http://doi.acm.org/10.1145/2908961.2908975.

Davy Smith, Laurissa Tokarchuk, and Geraint Wiggins. Harnessing Phenotypic Diversity Towards Multiple Independent Objectives. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, GECCO '16 Companion, pages 961–968, New York, NY, USA, 2016b. ACM. ISBN 978-1-4503-4323-7. doi: 10.1145/2908961.2931654. URL http://doi.acm.org/10.1145/2908961.2931654.

Davy Smith, Laurissa Tokarchuk, and Geraint Wiggins. Rapid Phenotypic Landscape Exploration Through Hierarchical Spatial Partitioning. In Julia Handl, Emma Hart, Peter R. Lewis, Manuel López-Ibáñez, Gabriela Ochoa, and Ben Paechter, editors, *Parallel Problem Solving from Nature – PPSN XIV*, number 9921 in Lecture Notes in Computer Science, pages 911–920. Springer International Publishing, September 2016c. ISBN 978-3-319-45822-9 978-3-319-45823-6. doi: 10.1007/978-3-319-45823-6. URL http://link.springer.com/chapter/10.1007/978-3-319-45823-6_85.

K. O. Stanley, B. D. Bryant, and R. Miikkulainen. Real-time Neuroevolution in the NERO Video Game. *Trans. Evol. Comp*, 9(6):653–668, December 2005. ISSN 1089-778X. doi: 10.1109/TEVC.2005.856210. URL http://dx.doi.org/10.1109/TEVC.2005.856210.

Kenneth O. Stanley. *Why Evolutionary Robotics Will Matter*, pages 37–41. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-18272-3. doi: 10.1007/978-3-642-18272-3_3. URL http://dx.doi.org/10.1007/978-3-642-18272-3_3.

Kenneth O Stanley and Joel Lehman. *Why Greatness Cannot Be Planned*. Springer International Publishing, 1 edition, 2015.

Kenneth O. Stanley and Risto Miikkulainen. Evolving Neural Networks Through Augmenting Topologies. *Evol. Comput.*, 10(2):99–127, June 2002. ISSN 1063-6560. doi: 10.1162/106365602320169811. URL http://dx.doi.org/10.1162/106365602320169811.

Christopher Stanton and Jeff Clune. Curiosity Search: Producing Generalists by Encouraging Individuals to Continually Explore and Acquire Skills throughout Their Lifetime. *PLOS ONE*, 11(9):e0162235, September 2016. ISSN 1932-6203. doi: 10.1371/journal.pone.0162235. URL http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0162235.

Niko Tinbergen. On aims and methods of ethology. *Zeitschrift für Tierpsychologie*, 20(4):410–433, 1963.

Andrea Toffolo and Ernesto Benini. Genetic Diversity As an Objective in Multi-objective Evolutionary Algorithms. *Evol. Comput.*, 11(2):151–167, May 2003. ISSN 1063-6560. doi: 10.1162/106365603766646816. URL http://dx.doi.org/10.1162/106365603766646816.

Leonardo Trujillo, Gustavo Olague, Evelyne Lutton, and Francisco Fernández de Vega. Behavior-based Speciation for Evolutionary Robotics. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, GECCO '08, pages 297–298, New York, NY, USA, 2008a. ACM. ISBN 978-1-60558-130-9. doi: 10.1145/1389095.1389147. URL http://doi.acm.org/10.1145/1389095.1389147.

Leonardo Trujillo, Gustavo Olague, Evelyne Lutton, and Francisco Fernéndez De Vega. Discovering Several Robot Behaviors Through Speciation. In *Proceedings of the 2008 Conference on Applications of Evolutionary Computing*, Evo'08, pages 164–174, Berlin, Heidelberg, 2008b. Springer-Verlag. ISBN 978-3-540-78760-0. URL http://dl.acm.org/citation.cfm?id=1787943.1787963.

Adriano Vinhas, Filipe Assunção, João Correia, Aniko Ekárt, and Penousal Machado. Fitness and Novelty in Evolutionary Art. In Colin Johnson, Vic Ciesielski, João Correia, and Penousal Machado, editors, *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, number 9596 in Lecture Notes in Computer Science, pages 225–240. Springer International Publishing, March 2016. ISBN 978-3-319-31007-7 978-3-319-31008-4. URL http://link.springer.com/chapter/10.1007/978-3-319-31008-4_16. DOI: 10.1007/978-3-319-31008-4_16.

Gunar E Liepins Michael D Vose. Deceptiveness and genetic algorithm dynamics'. *Foundations of Genetic Algorithms 1991 (FOGA 1)*, 1:36, 2014.

Shimon Whiteson and Peter Stone. Evolutionary Function Approximation for Reinforcement Learning. *J. Mach. Learn. Res.*, 7:877–917, December 2006. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1248547.1248578.

D Whitley. Fundamental principles of deception. *Foundations of Genetic Algorithms 1991 (FOGA 1)*, 1:221, 2014.

L. Darrell Whitley. The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of the 3rd International*

Conference on Genetic Algorithms, pages 116–123, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc. ISBN 1-55860-066-3. URL `http://dl.acm.org/citation.cfm?id=645512.657257`.

Georgios N Yannakakis and Julian Togelius. A panorama of artificial and computational intelligence in games. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(4):317–335, 2015.

A Software and Computational Facilities Used

The experimental work presented in this thesis was developed upon existing open-source and research based software platforms. The *Asteroids* game domain was developed by the author using the Processing[1] creative computing framework for the JAVA programming language The version of NEAT used in both the experiments presented in Chapter 3 was the ANJI[2] software library.

All of the remaining experimental work, presented in Chapters 4 and 5, was developed by the author as an extension to the MultiNEAT[3] software library for the C++ programming language. The algorithms developed within this thesis are available as an extension to the MultiNEAT library[4].

The experiments were all performed utilising the Queen Mary University of London MidPlus computational facilities, supported by QMUL Research-IT and funded by EPSRC grant EP/K000128/1.

[1] `https://processing.org/`

[2] `http://anji.sourceforge.net/`

[3] `http://www.multineat.com/index.html`

[4] Please contact author davy@paper-boy.org

B

Parameters For Experimental Studies

B.1 MultiNEAT Parameters

YoungAgeTreshold = 5

YoungAgeFitnessBoost = 1.1

SpeciesMaxStagnation = 50

StagnationDelta = 0.0

OldAgeTreshold = 30

OldAgePenalty = 1.0

KillWorstSpeciesEach = 15

KillWorstAge = 10

SurvivalRate = 0.2

CrossoverRate = 0.7

OverallMutationRate = 0.25

InterspeciesCrossoverRate = 0.0001

MultipointCrossoverRate = 0.75

RouletteWheelSelection = true

PhasedSearching = false

DeltaCoding = false

SimplifyingPhaseMPCTreshold = 20

SimplifyingPhaseStagnationTreshold = 30

ComplexityFloorGenerations = 40

NoveltySearch_K = 15

NoveltySearch_P_min = 0.5

NoveltySearch_Dynamic_Pmin = true

NoveltySearch_No_Archiving_Stagnation_Treshold = 150

NoveltySearch_Pmin_lowering_multiplier = 0.9

NoveltySearch_Pmin_min = 0.05

NoveltySearch_Quick_Archiving_Min_Evaluations = 8

NoveltySearch_Pmin_raising_multiplier = 1.1

NoveltySearch_Recompute_Sparseness_Each = 25

MutateAddNeuronProb = 0.01

SplitRecurrent = true

SplitLoopedRecurrent = true

MutateAddLinkProb = 0.03

MutateAddLinkFromBiasProb = 0.0

MutateRemLinkProb = 0.0

MutateRemSimpleNeuronProb = 0.0

```
LinkTries = 32
RecurrentProb = 0.25
RecurrentLoopProb = 0.25
MutateWeightsProb = 0.90
MutateWeightsSevereProb = 0.25
WeightMutationRate = 1.0
WeightMutationMaxPower = 1.0
WeightReplacementMaxPower = 1.0
MaxWeight = 8.0
MutateActivationAProb = 0.0
MutateActivationBProb = 0.0
ActivationAMutationMaxPower = 0.0
ActivationBMutationMaxPower = 0.0
MinActivationA = 1.0
MaxActivationA = 1.0
MinActivationB = 0.0
MaxActivationB = 0.0
TimeConstantMutationMaxPower = 0.0
BiasMutationMaxPower = WeightMutationMaxPower
MutateNeuronTimeConstantsProb = 0.0
MutateNeuronBiasesProb = 0.0
MinNeuronTimeConstant = 0.0
MaxNeuronTimeConstant = 0.0
MinNeuronBias = 0.0
MaxNeuronBias = 0.0
MutateNeuronActivationTypeProb = 0.0
ActivationFunction_SignedSigmoid_Prob = 0.0
ActivationFunction_UnsignedSigmoid_Prob = 1.0
ActivationFunction_Tanh_Prob = 0.0
ActivationFunction_TanhCubic_Prob = 0.0
ActivationFunction_SignedStep_Prob = 0.0
ActivationFunction_UnsignedStep_Prob = 0.0
ActivationFunction_SignedGauss_Prob = 0.0
ActivationFunction_UnsignedGauss_Prob = 0.0
ActivationFunction_Abs_Prob = 0.0
ActivationFunction_SignedSine_Prob = 0.0
ActivationFunction_UnsignedSine_Prob = 0.0
ActivationFunction_SignedSquare_Prob = 0.0
ActivationFunction_UnsignedSquare_Prob = 0.0
ActivationFunction_Linear_Prob = 0.0
DisjointCoeff = 1.0
ExcessCoeff = 1.0
WeightDiffCoeff = 0.5
ActivationADiffCoeff = 0.0
ActivationBDiffCoeff = 0.0
TimeConstantDiffCoeff = 0.0
ActivationFunctionDiffCoeff = 0.0
```

```
CompatTreshold = 5.0
MinCompatTreshold = 0.2
CompatTresholdModifier = 0.3
CompatTreshChangeInterval_Generations = 1
CompatTreshChangeInterval_Evaluations = 10
```

B.1.1 Maze Domain Parameters

Parameter Name	**Value**
Dynamic compatibility	*True*
Compatibility threshold	2.0
Young age threshold	15
Species maximum stagnation	100
Old age threshold	35
Minimum species	5
Maximum species	25
Roulette wheel selection	*True*
Mutate link probability	0.02
Recurrent probability	0.1
Overall mutation rate	0.15
Mutate add link probability	0.08
Mutate add neuron probability	0.01
Mutate weights probability	0.80
Maximum weight	8.0
Weight mutation maximum power	0.2
Weight replacement maximum power	1.0
Mutate activation A probability	0.02
Activation A mutation maximum power	0.5
Minimum activation A	0.01
Maximum activation A	1.0
Mutate neuron activation type probability	0.03

Table B.1: *NEAT parameters*

B.1.2 Neuron Activation Function Probabilities

Neuron Activation Probability	Value
Signed sigmoid	0.2
Unsigned sigmoid	0.3
Tanh	0.1
Tanh cubic	0.0
Signed step	0.0
Unsigned step	0.0
Signed Gaussian	0.1
Unsigned Gaussian	0.0
Abs	0.0
Signed sine	0.1
Unsigned sine	0.1
Linear	0.1

Table B.2: *Activation function probabilities for maze experiments*

B.2 Optimal α and β Values in Maze Domains

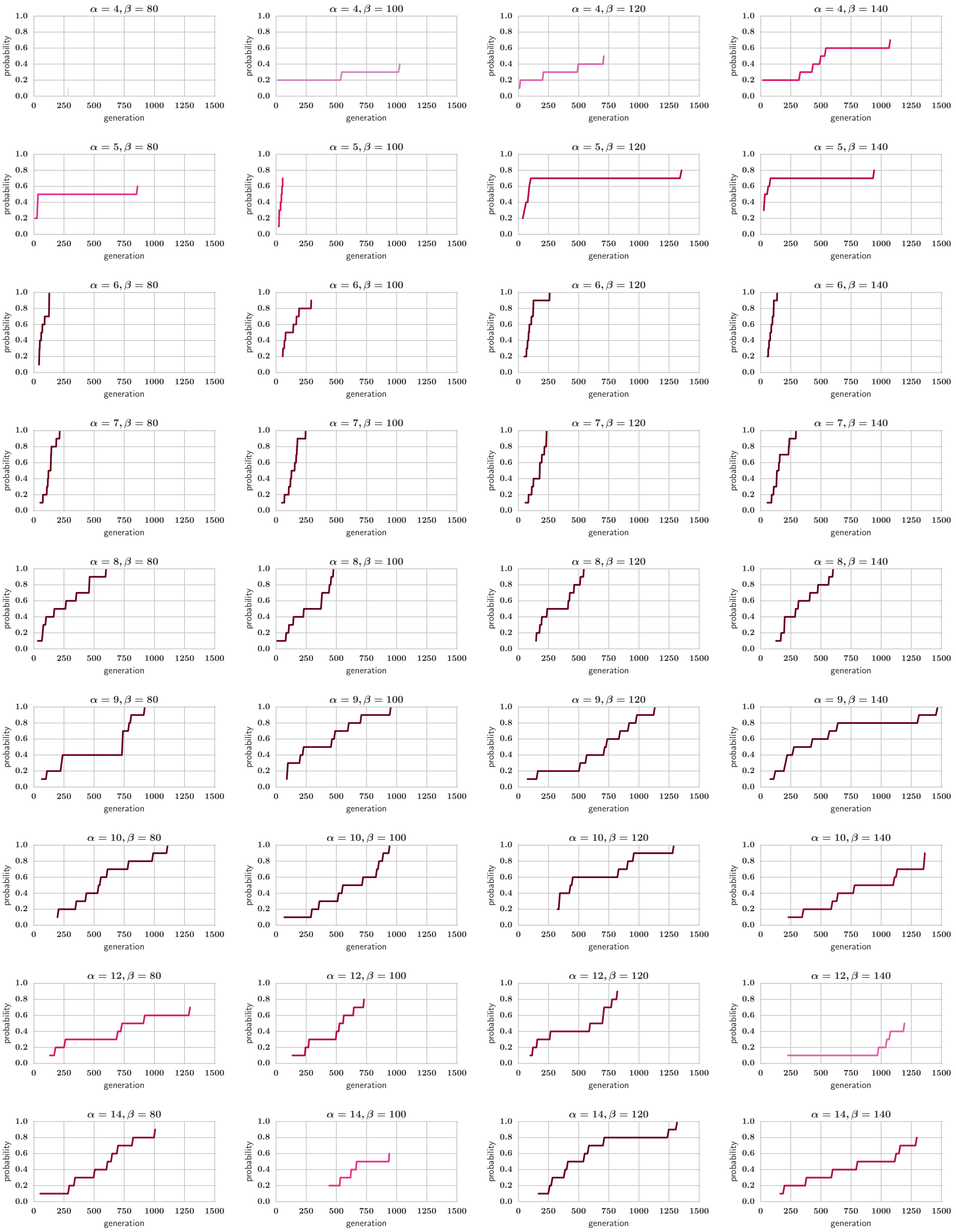


Figure B.1: *Determining optimal α and β values in HARD domain*