A Computational Study of the Role of Genetic Reuse in Evolvability

Chad W. Seys

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Department of Electrical Engineering and Computer Science

Case Western Reserve University

August 2012
We hereby approve the thesis/dissertation of

  Chad William Seys

candidate for the Doctor of Philosophy degree *

(signed)  Michael S. Branicky
          (chair of the committee)

  Randall D. Beer

  Ken A. Loparo

  Mehmet Koyuturk

  Hillel J. Chiel

(date)   April 15, 2011

*We also certify that written approval has been obtained for any proprietary material contained therein.
Contents

List of tables 6
List of figures 7
Acknowledgments 9
Abstract 10

1 Introduction 12

2 Background and literature review 19
   Genetic algorithms ........................................... 19
   Types of encodings ........................................... 20
   What is evolvability? ......................................... 23
       Conception of evolvability relevant to this thesis .......... 24
   Measuring evolvability ....................................... 25
       Individual evolvability .................................... 25
       Population evolvability – discoverability and adaptability 25
   Mechanisms of evolvability ................................... 30
       Mechanism of evolvability relevant to this thesis .......... 32
   Evolvability experiments ..................................... 32
       Experiments demonstrating that indirect encodings often have higher
       evolvability than direct encodings .......................... 32
       Experiments exploring the relationship between problem regularity and
       genotype reuse in the encoding .............................. 35
       Experiments demonstrating that changing environment selects for adapt-
       ability ......................................................... 37
       Segue to thesis experiments ................................ 39

3 Genotype reuse can be more important than genotype size to evolvabil-
   ity: hexapod experiments 41
   Methods ......................................................... 42
       Neural network .............................................. 42
       Hexapod body ................................................ 42
       Genetic algorithm .......................................... 46
       Neural network topology .................................... 46
       Encodings ..................................................... 49
       Naming the neural network ................................... 51
       Calculating the number of genotype parameters ............ 52
       Encodings and neural network topologies combinations of interest . 54
   Experiment types .............................................. 57
       Degree of discoverability experiments ....................... 57
       Equilibrium search .......................................... 59
       Adaptability experiments .................................... 60
4 Genotype reuse can increase evolvability on a problem with no geometric structural regularity: pseudoplant experiments

Methods .................................................. 77
  Fitness landscape .................................. 77
  Body models ....................................... 80
  L-system encoding ................................ 83
  Genetic algorithm ................................ 85
  Direct and indirect encodings implemented with L-systems .... 87
  Measuring genotype reuse ........................ 88

Experiment types ........................................ 89
  Rate of discoverability experiments .............. 89
  Adaptability experiments ........................ 91
  Statistical significance .......................... 93

Results .................................................. 96
  Indirect and direct encodings have similar discoverability using TOM .... 96
  Indirect encoding has higher adaptability than direct using TOM .... 98
  Genotype reuse allows adaptability by amplifying use of a body segment in TOM ........................................... 100
  Visualizing the reason why 'd' amplification changes evolvability .... 106
  Reducing the benefit of repetition – the EQ4 body model ............ 108
  Indirect and direct encodings have similar discoverability and adaptability in EQ4 .......................................... 109
  Fixed genotype reuse can have higher discoverability in EQ4 ....... 111
  Genotype reuse selected for by a changing environment in TOM .... 113
  Genotype reuse selected for by a changing environment in EQ4 .... 116

Discussion ............................................... 123

5 Conclusion ............................................ 126
Summary .................................................. 126
  More genotype reuse can be more important than a smaller genotype with respect to increasing evolvability ................. 128
  Genotype reuse can increase evolvability on problems with no geometric structural regularity .............................. 129
List of tables

1  A summary of the combinations of encoding, architecture, and number of neurons per subnet .......................................................... 52
2  The two pairs of directly and symmetrically encoded neural networks 56
3  TOM body model assignment of body segment names to surf/vol ratios 81
4  The different types of mutation and their relative probability ........ 86
5  Example effects of mutation operators ........................................... 87
### List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The genetic algorithm used in this thesis</td>
</tr>
<tr>
<td>2</td>
<td>Types of encodings</td>
</tr>
<tr>
<td>3</td>
<td>The “evolvability” conceptual spectrum</td>
</tr>
<tr>
<td>4</td>
<td>Indirect encodings are often found to have higher discoverability than direct encodings</td>
</tr>
<tr>
<td>5</td>
<td>Does an indirect encoding often have higher evolvability because it more easily exploits problem regularities?</td>
</tr>
<tr>
<td>6</td>
<td>Does an indirect encoding often have higher evolvability because it more easily exploits problem regularities?</td>
</tr>
<tr>
<td>7</td>
<td>Hexapod agent modeling a tripod gait</td>
</tr>
<tr>
<td>8</td>
<td>Example locally connected network architecture and one of its subnetworks</td>
</tr>
<tr>
<td>9</td>
<td>Example fully connected network architecture</td>
</tr>
<tr>
<td>10</td>
<td>Example symmetrically encoded neural network</td>
</tr>
<tr>
<td>11</td>
<td>Example directly encoded neural network</td>
</tr>
<tr>
<td>12</td>
<td>Typical direct versus indirect comparison</td>
</tr>
<tr>
<td>13</td>
<td>An illustration depicting the comparison between a directly encoded neural network with fewer genotype parameters than a symmetrically encoded neural network</td>
</tr>
<tr>
<td>14</td>
<td>Degree of discoverability for hexapod experiments is the fraction of searches whose highest fitness individual is greater than a target fitness of 0.75 after 500 generations</td>
</tr>
<tr>
<td>15</td>
<td>Hexapod neural network controllers must adapt to having their legs shortened</td>
</tr>
<tr>
<td>16</td>
<td>Adaptability is measured as the total amount of leg shortening in 5000 generations</td>
</tr>
<tr>
<td>17</td>
<td>Stages of adaptability experiments. Dashed vertical lines are filters described at their bottom</td>
</tr>
<tr>
<td>18</td>
<td>Stages of competition experiments</td>
</tr>
<tr>
<td>19</td>
<td>Symmetric encodings with more parameters are more discoverable and have higher adaptability than direct encodings with fewer parameters</td>
</tr>
<tr>
<td>20</td>
<td>Changing the encoding is sufficient to change the adaptability</td>
</tr>
<tr>
<td>21</td>
<td>Median change in leg length by generation during adaptability experiments performed using the indicated encoding/architecture type</td>
</tr>
<tr>
<td>22</td>
<td>Symmetric encodings win competitions against direct encodings when leg length changes</td>
</tr>
<tr>
<td>23</td>
<td>Fitness landscape showing an interaction between humidity and surf/vol ratio</td>
</tr>
<tr>
<td>24</td>
<td>The original body model (TOM) has surf/vol ratios assigned to four body segments in an unevenly distributed way</td>
</tr>
<tr>
<td>25</td>
<td>The EQ4 body model has four body segments whose surf/vol ratios evenly divide the range of possible surf/vol ratios</td>
</tr>
</tbody>
</table>
Rate of discoverability for pseudoplant experiments is the median generation at which search median fitness becomes greater than a target fitness .................................................. 90

Adaptability of a single search is measured as the total amount of humidity level change in 5000 generations ................................. 92

Flow of populations including filtering for determining median adaptability for a type of genotype ................................................. 93

Illustration of the confidence interval of the difference ............... 95

Genotypes with potential genotype reuse (PoGR) were not more discoverable than genotypes with genotype reuse not allowed at any mutation frequency (NoGR) ........................................... 97

Grouping searches by humidity level yields no difference when comparing discoverability of PoGR and NoGR .......................... 98

In TOM, the adaptability of populations in which any rule is allowed in the genotype is significantly higher than when no rule is allowed 100

'\(d\)' in rule are necessary to increase adaptability of PoGR .......... 102

Populations with a fixed rule of \(\{(z+\text{dd})\}\) have higher discoverability than PoGR populations with an evolvable rule in TOM ........ 104

Populations with a fixed rule of \(\{(z+\text{dd})\}\) have higher adaptability than PoGR populations with an evolvable rule in TOM .................. 105

Number of '\(d\)' body segments in the minimum length high fitness phenotype and total number of all other body segments at a variety of humidity levels for the TOM body model ................................. 107

In EQ4 there is no difference in evolvability between populations which allow genotype reuse and those that do not ................. 110

Evolvability of a fixed rule \(\{(z+\text{ab})\}\) compared to NoGR in the EQ4 body model ................................................................. 112

Environmental change selects for \(GR_{\text{avg\_num\_match}}\) in the TOM body model ................................................................. 115

Environmental change selects for \(GR_{\text{phen\_gen\_ratio}}\) in the TOM body model 116

Environmental change does not select for genotype reuse \(GR_{\text{avg\_num\_match}}\) 117

Environmental change does selects for genotype reuse \(GR_{\text{phen\_gen\_ratio}}\) 118

Environmental change selects for less genotype reuse \(GR_{\text{phen\_gen\_ratio}}\) as more body segments are added to the body model ........ 120

Environmental change selects for more negative difference in \(GR_{\text{avg\_num\_match}}\) as the number of body segments increases .............. 122

More complicated G-P maps than used in the experiments of this thesis 135
Acknowledgments

There are many people who made my graduate career possible and enjoyable.

First, many thanks to all the people who wrote the grant proposal which resulted in NSF funding NSF-9972747. I especially would like to recognize Professors Roy Ritzmann and Patrick Crago. Your hard work made this adventure possible.

Thanks also to Professor Randall Beer. You taught me (technical) writing from the ground up (but do not blame him for any remaining deficiencies), You also allowed me to explore my interests within the wide confines of evolutionary computing. Finally, you (and graduate school) readjusted my threshold of what is “too hard” to attempt.

Next, I miss you Beer Nuts! Especially my cohort Barry Drennan, Sean Psujek, Jeff Ames, Eldan Goldenberg, and Alan Calvitti. You provided many interesting conversations, insights, tutoring, and internet delivered distractions. We laughed, and we may have cried when we found an especially big haul of free food.

Finally, thank you Amanda, my girlfriend, fiancée, financier, wife, and best friend. Thank you for allowing me to ramble temporally, physically, and vocally, on until my ideas made more sense. Without your support my graduate career would have been very different.

---

\[1\] This material is based upon work supported by the National Science Foundation through an NSF/IGERT Training Program in Neuro-mechanical Systems under Grant No. 9972747. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.
A Computational Study of the Role of Genetic Reuse in Evolvability

by

Chad W. Seys

Abstract

When creating a genetic algorithm one must decide how the phenotype is encoded in the genotype. This decision can change the “evolvability” of the system, that is, how quickly high fitness solutions are found. A change in encoding can cause high fitness solutions to be found more quickly or result in no high fitness solutions found at all.

Direct and indirect encodings are two general types of encodings. Direct encodings are characterized by one genotype parameter mapping to one phenotype parameter. Indirect encodings are characterized by a one-to-many genotype to phenotype parameter mapping.

Researchers have often demonstrated that indirect encodings have higher evolvability than direct encodings on a variety of problems. The indirect encodings were hypothesized to be more evolvable for two main reasons, both a consequence of the one-to-many property of the encoding. First, when parts of the genotype are reused, a given phenotype can often be described with a smaller genotype. A smaller genotype space should be faster to search than a larger genotype space. Second, the one-to-many property often leads to regularity in the phenotype produced by indirect encodings. This may be useful in exploiting regularities in the problems used for demonstrating that indirect encodings have higher evolvabilities than direct encodings.

This thesis explores whether reusing parts of the genotype increases evolvability more than size of the genotype for an example system. It is found that genotype reuse is more important than genotype size.

The thesis also explores whether an indirect encoding can increase evolvability and be selected for on a problem on which high fitness can be achieved with
irregular solutions. It is found that it is very difficult to construct a problem which does not contain some form of regularity which can be exploited by the indirect encoding to increase evolvability. A problem which appears not to have regularity is found to have non-structural regularity. Based on this understanding, the problem is modified and the indirect encoding is found to have no or a slightly negative impact on evolvability on the modified problem.
1 Introduction

Evolutionary Algorithms (EAs) such as genetic algorithms (GAs) are computer programs which simulate an abstraction of natural evolution \[63\]. Each individual in a population is assigned a fitness. The individuals have a number of children in proportion to their assigned fitness. The most fit individuals have more children than the less fit. The children are created based upon their parents, but may have slight changes. The population of children then becomes the parent population, and the process repeats until a predetermined stopping point. The stopping point may be when a certain fitness is reached, a certain number of iterations (called generations) have elapsed, or some other criteria.

GAs are used for a wide range of purposes. In engineering GAs can be used to find solutions to problems in an automated, novel, or less biased way.

Another reason for using GAs is that they are an abstract version of natural evolution with which it may be possible to demonstrate fundamental principles of natural evolution. All life appears to have descended from a common ancestor \[90\][93\]. Thus, it is currently impossible for biologists to look at organisms from different origins to draw general conclusions. For example, one question which might be answerable by studying life evolved from multiple origins is whether or not complex organisms inevitably arise from simple, primitive ones \[2\]. GAs allow for multiple origins of life and subsequent evolution to be simulated and observed on a computer. From these experiments hypotheses about general principles of life anywhere can be generated.

As a first approximation, natural organisms can be conceptually divided into genotype and phenotype. The genotype is the inheritable information found in the genetic material while the phenotype is any part of the structure, function, or behavior of an organism \[15\][9\].

The genotype to phenotype map (G-P map) is the process by which the phenotype is created from the information found in the genotype \[1\].
The G-P map in natural organisms is the result of natural selection over billions of years [47]. It includes such structures and processes as the transcription and translation of proteins [31][64][91][48], regulation of genes [79], multicellularity and cell specialization [61][35], and development [25][16]. The G-P map is also influenced by environmental conditions such that the same genotype may create a different phenotype in different environments.

The G-P map in a genetic algorithm, often called an encoding or a representation, must be created by the designer of the GA. It can be implemented in a variety of ways. For example, aspects of the encoding can be designed to evolve as in nature, or they can be fully determined and fixed. For every problem there are many possible ways to encode an individual.

Interestingly, the encoding of an individual often changes the ease with which high fitness individuals can be found from low fitness starting populations. This is known as the “evolvability” of the encoding. (See Section 2 on page 23 for more definitions of evolvability.) Improving the evolvability of encodings has been a topic of research since GAs were first created.

Recently researchers have sought to increase the evolvability of encodings by designing them to more closely resemble the G-P map of natural organisms. Researchers wish to solve increasingly complex problems or create more complex artifacts. However, as the size of the genotype grows linearly, the possible combinations of values for that genotype grows exponentially [103]. This exponential growth in the size of the search space can lead to a scalability problem. Researchers reasoned that since natural organisms are more complex than the solutions created by GAs, the natural G-P map has properties which would be beneficial to incorporate into GA encodings [57].

These more naturalistic encodings, known by some as “indirect encodings” [54][89]-[72], emulate the natural G-P map with a varying degree of realism. Some emulate
development in natural organisms [36][7][54][11] while others create more abstract generative encodings [42][88].

The many indirect encodings and the natural G-P map which inspired them share an important property. They all have a one-to-many mapping from one element in the genotype to many elements in the phenotype. This means that each element in the genotype may create many elements of the phenotype. This one-to-many property will be referred to as \textit{genotype reuse} because parts of the genotype may be reused while creating the phenotype. Examples of the one-to-many property in the natural G-P map include the following. Cells contain protein fibers created by connecting many protein subunits. For example, microfilaments in eukaryotic cells are a polymer of the protein actin, created from a single gene [15]. On a larger scale, activation of the same genes at different locations during the development of multicellular organisms causes the growth of the same structure (e.g., an eye) [17].

The evolvability of indirect encodings are often compared with that of direct encodings [36][7][54][42][40][41][30][82][83][20]. Direct encodings have a one-to-one mapping between one element in the genotype to one element in the phenotype [42][89][20]. Direct encodings are the gold standard to improve upon because they are simpler to implement. If more complex indirect encodings can increase evolvability then they may be worth the extra effort of implementing them.

Indirect encodings have been shown to find high fitness individuals more quickly than direct encodings on a diversity of problems. Some examples include creating the shape of tessellating tiles [7], creating tables [40][41], locomotion [36][54][42][82][83], pattern matching [20], and playing games [73].

Demonstrating that indirect encodings can increase evolvability is an important first step showing that emulating the natural G-P map is worthwhile.

However, a number of questions about how indirect encodings increase evolvability are just beginning to be answered. What differences between indirect and direct
encodings are important? Will indirect encodings increase evolvability on all types of problems? Are there types of problems for which indirect encodings should not be used? Under what conditions can genotype reuse, such as found in the natural G-P map, be selected for?

Answers to these questions are important not only for knowledge of best practices in genetic algorithms but also, to the extent that indirect encodings share attributes with natural G-P maps, important to biology.

This thesis examines some details about how indirect encodings increase evolvability and the types of problems they are useful for.

First, an increase in the evolvability can be attributed to a number of differences between the direct and indirect encodings. One possibility is that the genotype reuse found in indirect encodings often leads to structural regularity (such as symmetry, repetition, and reuse [57][88]) in the phenotype. This may be useful for exploiting the regularities of certain problems [7][89][40]. Alternatively, when parts of the genotype are reused, a given phenotype can often be described with a smaller genotype [53][89][40][76][74][20][19]. A smaller genotype size means that the space of possible solutions is smaller and therefore faster to search.

It is difficult to know whether genotype reuse, smaller genotype, or both deserves credit for the improved performance of indirect encodings. Often both are credited [36][7][54][89]. It has been shown that an indirect encoding performs better as problem regularity increases [20]. However, the relative importance of the size of genotype space versus genotype reuse with respect to the evolvability of encodings is widely applicable yet remains unexplored.

The first experiments of this thesis (Section 3 on page 41) disentangle for an example system whether more genotype reuse or a smaller genotype size is more important for increasing evolvability. This is accomplished by comparing the evolvability of populations containing indirectly encoded individuals with large amounts of genotype
reuse and large genotypes to populations containing directly encoded individuals with no genotype reuse and small genotypes. The indirectly encoded genotypes are larger than the directly encoded genotypes, so if they have higher evolvability a smaller genotype size can not be credited. These experiments demonstrate that genotype size is not as important a factor as genotype reuse in determining evolvability. The indirect encoding in these experiments has higher evolvability than the direct encoding even though its genotypes are larger. This demonstrates that the size of the search space is not the sole or even most important factor in determining evolvability. The density of high fitness individuals in the search space and their arrangement are often more important.

Another question the first experiments of the thesis examines is whether or not a changing environment can select for an encoding with higher evolvability than another. Directly and indirectly encoded individuals were placed in the same population. When the GA search is performed the two types of encodings compete for population share. The indirectly encoded individuals win the competitions more often than directly encoded individuals if the environment is changed. Interestingly, initially lower fitness indirectly encoded individuals were able to overcome a fitness deficit to outcompete initially higher fitness directly encoded individuals. Though in these experiments the encodings were fixed (not changeable by mutations) this result suggests that selection pressure may be able to choose forms of encodings which have higher evolvability than other forms. This is also interesting given that environmental change is ubiquitous and that indirect encodings were inspired by natural G-P maps. Environmental change in the natural world may have played an important role in selecting for the genotype reuse property of natural G-P maps.

Many of the above comparisons of indirect encodings and direct encodings are performed on problems which are solvable by and perhaps favor phenotypes with structural regularity. For example, consider “tessellating tiles” where the goal is to
cover a surface with no gaps and no overlaps using one repeated shape [7]. Regular, symmetrical shapes can solve this problem. Another example is the creation of tables which are high fitness if they have a solid surface with height, area, and stability [41]. For the table to be stable, the forces holding the surface upright need to be balanced. This can be achieved by a symmetrical surface with equally spaced forces. A final example, a 3-D agent whose body was evolved and tasked with maximizing velocity on a 2-D plane would do well with two symmetrically applied forces [54].

Can indirect encodings also have higher evolvability than direct on problems which do not have geometric structural regularity? This is one of the questions explored by the final experiments of this thesis (Section 4 on page 75). A second experimental system is created in which fitness does not depend on geometric structural regularity. The system’s problem lacks spatial dimensions thereby avoiding adding a spatial bias as in many previous experiments. Consequently, geometric structural regularity does not play a role in determining fitness. In addition, the genotype is an evolvable encoding which has both direct and indirect forms. The evolvable encoding is designed to smoothly transition by mutations between direct and indirect forms allowing for variants of both forms exist in the population. This allows forms to be selected for or against according to their fitness and avoids any imposition of regularities which might occur if the encoding were hand designed and non-evolvable.

These experiments demonstrate that the indirect form of the encoding can have higher adaptability than directly encoded initial populations. This indicates that, under some circumstances (explained in next paragraph), systems without spatial dimensions and spatial regularity also benefit from genotype reuse.

The mechanism by which the indirect form has higher evolvability than the direct form is found to be related to temporal, rather than spatial, regularity. The indirect form of the encoding increases evolvability by allowing repetition (through genotype reuse), a non-geometrical form of structural regularity. Repetition enables amplifica-
tion of changes to the phenotype, which results in the ability to adapt to a non-spatial environmental change more quickly. This points to the possibility that indirect encodings and genotype reuse may increase evolvability on a wide set of problems, whether they have spatial dimensions or not.

***

This thesis is organized as follows. Section 2 gives background and literature review of the main ideas relevant to this thesis. Section 3 gives evidence that genotype size is not as important a factor as genotype reuse in determining evolvability using a hexapod agent. Section 4 provides a demonstration that an indirect encoding can have higher evolvability than a direct encoding on a problem with a non-geometric form of regularity. Conclusions follow in Section 5.
2 Background and literature review

Genetic algorithms

A genetic algorithm (GA) is a computer simulation of natural evolution. The GA used in this thesis will be described below and is given as a flowchart in Figure 1. However, there are many variations on how GAs are implemented. For more details and variations one introductory textbook is [63].

An outline of a GA is the following:

A population of individuals is created. These individuals are any object which for which a way can be found to describe it in computer code. The description of these objects are called a representation [98][89], an encoding [89], or a genotype to phenotype map (G-P map) [11][55][94].

Each individual is assigned a fitness. The fitnesses of the individuals are evaluated to determine how well they satisfy predetermined criteria. This evaluation is performed by what is called a fitness function. The better the individual satisfies the predetermined criteria, the higher its fitness.

The individuals are selected for reproduction. The higher fitness an individual is, the higher probability the individual will have children. Not all individuals have children, while some have more than one.

A child is created by mutating a parent. One major departure from natural evolution in these experiments is that there is no crossover. Each child has one parent and may differ from that parent due to mutation.

Next, the population is assigned a fitness and the cycle either repeats or the search ends. Each cycle is one generation. The search may end after a predetermined number of generations elapse, a specific fitness is found, or other stopping criteria.
Figure 1: The genetic algorithm used in this thesis. A genetic algorithm models natural evolution. Each cycle through the procedure is one generation.

Types of encodings

There are many different encodings used in evolutionary computation. For every problem there are many possible ways of encoding a solution in a way that can be manipulated by the GA. Some of these encodings will find high fitness individuals more easily than others. These encodings are said to have higher evolvability. (The meaning of “evolvability” is discussed further in Section 2 on page 23.)

To answer the question “What causes some encodings to have higher evolvability than others?”, researchers have created categories based on shared attributes [7][89][41].

Though the categories created are slightly different in each classification, there is agreement on two broad categories: Direct encodings and everything else, sometimes known as indirect encodings.

**direct encoding** is characterized by a one-to-one mapping between an element of the genotype and an element of the phenotype [42][89][20]. An example of elements of genotype mapping to elements of phenotype is given in Figure 2(TOP LEFT).
**indirect encoding** is characterized by the possibility of one genotype parameter creating multiple elements of the phenotype \([54][89][72]\). An example is given in Figure 2(TOP RIGHT).

Researchers often further classify indirect encodings in multiple different ways, but the indirect and direct distinction is sufficient for this thesis \([7][89]\).

The possibility of genotype reuse is the attribute which distinguishes direct and indirect encodings. Hence, many genotype elements mapping to one phenotype element is another type of direct encoding (Figure 2(BOTTOM LEFT)) because genotype reuse is not possible. Another type of indirect encoding uses a many-to-many mapping between elements of the genotype and elements of the phenotype.

**genotype reuse** One genotype element being used to create multiple phenotype elements. Only possible with an indirect encoding.
Figure 2: Types of encodings. A direct encoding has a one-to-one (or many-to-one) mapping between genotype elements and phenotype elements. The distinguishing feature of a direct encoding is that genotype elements are used once and only once to create a phenotype element. The left column shows a one-to-one example at top and a many-to-one example on the bottom. Indirect encodings are distinguished by the possibility that a genotype element may be used more than once when creating phenotype elements. The right column shows a one-to-many example at top and a many-to-many example on the bottom. The top row encodings (black) are examined in this thesis, while the bottom row (gray) are left for future experiments.

Finally, indirect and direct encodings can be either fixed or evolvable. In a fixed encoding, elements of the genotype are determined by the researcher and not changed during the course of a GA search. In an evolvable encoding, elements of the genotype can be added, removed, or otherwise modified during the course of that GA.
search [94][62][33]. In the articles reviewed below, indirect encodings are usually also evolvable and the mapping from genotype to phenotype is under selection pressure.

**What is evolvability?**

“Evolvability” is a suggestive word with many conceptions in both biology and evolutionary computation. Dawkins is said [1] to have first used the word at the first Artificial Life conference [24]. It has since been used both in biology and evolutionary computation. Below some definitions of evolvability are given.

Pigliucci [69] (among others [60][43]) gives a review of ideas of evolvability in a biological context. He points out that in biology the idea of evolvability has been applied to a wide range of groupings as described in Figure 3: populations, species, and clades.

<table>
<thead>
<tr>
<th>Suggested term</th>
<th>Scale</th>
<th>Description</th>
<th>Effects</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heritability (sensu Houle)</td>
<td>Within populations</td>
<td>Standing pool of genetic variation and covariation</td>
<td>Determines the response to natural selection within populations</td>
<td></td>
</tr>
<tr>
<td>Evolvability (sensu Wagner &amp; Altenberg)</td>
<td>Within species</td>
<td>Includes variability (sensu Wagner &amp; Altenberg), depends on genetic architecture and developmental constraints</td>
<td>Affects long-term adaptation, channels evolution along non-random trajectories, allows mid-term exploration of phenotypic space</td>
<td></td>
</tr>
<tr>
<td>Innovation (sensu Maynard-Smith &amp; Szathmary)</td>
<td>Within clades</td>
<td>As for within species, but includes the capacity to overcome standing genetic and developmental constraints, opening new areas of phenotypic space for further evolution</td>
<td>Generates major phenotypic (morphological, behavioural or physiological) breakthroughs (novelties)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: The “evolvability” conceptual spectrum. Scale increases from top to bottom. From Pigliucci [69].

According to Pigliucci, evolvability at one end of the conceptual spectrum is related to the existing genetic variation in a population. This standing genetic variation is
the potential for the population to respond to selection. Selection changes the relative frequencies or distribution of a trait. The example of this type evolvability given in the top row of Figure 3 appears to be the classic story of peppered moths adapting pollution darkened trees [34]. Both light speckled moths and dark moths exist in the original population, but the dark moths are much less common. As the lichen on trees dies off and the trunks of the trees darken due to pollution, the light speckled moths become easier for birds to see. The birds eat the light speckled moths more often than the dark moths. The dark moths become a much larger part of the population.

On the other end of the conceptual spectrum evolvability is the propensity to evolve novel structures. The ability to overcome genetic and developmental constraints. Pigliucci gives as examples the origin of multicellularity and the origin of sexual reproduction. This is also the way Dawkins’ defines evolvability [24]. This conception of evolvability is not well defined in terms of how it could be measured.

**Conception of evolvability relevant to this thesis**

Finally, there is the conception of evolvability at the middle region of the spectrum. This conception was made well known by the biologists Wagner and Altenberg in an article written for both biologists and computer scientists [98]. A key idea is that traits have both variation and variability. Variation is the observable difference in traits in an existing population. Variability is the variance of a trait which occurs after new mutations. The example given in the second row of Figure 3 is that of a variety of bird species. Each species has beaks adapted to different types of foods. This illustrates that the bird genetic architecture can create beaks of a wider variety of shapes than the variation within one species.

Wagner and Altenberg later go on to specify that “evolvability is the genome’s ability to produce adaptive variants when acted upon by the genetic system”. This definition of evolvability constrains the mechanism of evolvability to be located in
the genotype to phenotype map (G-P map). The G-P map determines possible phenotypes. When mutation and crossover (and transpositions, etc.) occur, some G-P maps produce more neutral or adaptive phenotypes than others, and are therefore more evolvable.

The idea that some G-P maps have more evolvability than others is important to evolutionary algorithms (EAs). In EAs, the evolvabilities of alternative G-P maps can be compared. Some attributes will lead to higher evolvabilities and others lower. This design decision impacts the quality of the solutions found to the problem of interest.

**Measuring evolvability**

One must measure evolvability to be able to compare evolvabilities of different G-P maps.

**Individual evolvability**

One class of methods to measure evolvability is focused on the fitness of an individual’s children. One method is the find the probability that an individual’s child will be more fit [86]. This probability can be calculated by applying the genetic operators to an individual to generate a number of children, then evaluating the children and finding the fraction which are more fit than the parent. Another method measures the expected (e.g., average) fitness of an individual’s child [86][85]. This class of measurement methods does not seem to be widely used, but see as examples [86][44].

**Population evolvability – discoverability and adaptability**

More commonly, evolvability is measured in two ways both of which in previous papers I have called the **discoverability** of the search [82][83]. These two ways are described further in the following paragraphs. The outline is that these two ways are related by the attribute that the GA search begins with a naively initialized
population. The difference between these two measurements is that one ends after a predetermined number of generations and the other after a target fitness is reached.

For “degree of discoverability” the search is ended after a predetermined number of generations \([7][54][42][74][41]\). This measure of evolvability has been also been called “quality” \([74]\). The experiments are performed as follows: The population is initialized to some naive state. Then the GA search is run for a fixed number of generations and performed multiple times. The average (or best) fitness computed for each search is recorded and the average or best fitness for all searches is computed. The evolvability is reported as a (best or average) fitness value. High evolvabilities correspond to high fitnesses. This measure of evolvability is appropriate for situations in which the range of fitness and what corresponds to “high fitness” is not known ahead of time.

Alternatively, for “rate of discoverability” the search is ended after a target fitness is achieved \([87][74][70][82][83]\). This measure of evolvability has also been named “efficiency” \([74]\). The experiments are performed by first initializing the population to some naive state. Then the GA is run until a target fitness is reached (or some maximum number of generations is reached, whichever comes first). The number of generations to achieve the target fitness are recorded for multiple GA searches, and the average or median number of generations is calculated. The evolvability is reported as the (average or median) number of generations needed to achieve the target fitness. High evolvabilities correspond to fewer numbers of generations. This measure is most appropriate in situations when a “high fitness” can be assigned based on knowledge of what a given fitness score indicates.

Often these two measures will indicate the same relative evolvability of encodings \([74]\). Searches which reach a target fitness in a few generations also often have a high ending fitness after a predetermined number of generations. But consider the scenario in which an encoding more quickly increases the population’s fitness initially, but plateaus in fewer generations than a different encoding. In this scenario, the two
different forms of discoverability may indicate different relative evolvabilities.

Using either “degree of discoverability” or “rate of discoverability” one must decide on either an ending generation or a “high fitness” target, respectively, at which to end the search.

Changing this decision will likely change the measured discoverability. The difference in discoverabilities might be greater, less, or no longer significant. In the worst case, the relative discoverability of encodings can reverse depending on the choice of ending fitness or generation. This can happen, for example, if one encoding takes very few generations to reach a low fitness plateau while another encoding takes many generations to reach a high fitness plateau. Choosing a low ending generation would show the first encoding to be more discoverable while choosing a high ending generation would show the second encoding to be more discoverable. Similarly, choosing a low target fitness would show one encoding to be more discoverable, while choosing a high target fitness would indicate that the other encoding is more discoverable.

As a consequence, discoverabilities have as their context the ending generation or target fitness chosen. Ideally one should check a variety of ending generations or target fitnesses to more fully understand the discoverability of an encoding.

A less common way of measuring evolvability is what could be called **adaptability**.

In such experiments the common attribute is that the “environment” is changed and the impact on some measure of fitness is observed.

The definition of “environment” as used by the genetic algorithm literature needs clarification. The environment is any part of the computational model which is not encoded by the genotype. Changing the environment may result in the same genotype being assigned a different fitness value. The fitness function itself or the requirements which cause a high fitness to be assigned by a fitness function can be changed throughout a GA search [26][49][75][72][44][92]. For example, a target network topology which achieves high fitness can be changed during the GA search [44]. Alternatively, if part
of the agent is encoded by the genotype and part of the agent is not, the part of the agent not encoded by the genotype can be changed, possibly changing the fitness of the agent [87][82][83]. For example, [87] evolves a controller for a simulated two wheeled robot (similar to Khepera). The controller is encoded by the genotype but the rest of the robot is not. The controller is initially evolved in a robot which has a motor speed of \( X \). Later the evolved controllers are placed in robots with motor speeds of \( \frac{1}{2}X \) and \( \frac{1}{4}X \). These robots with slower motor speeds are referred to by the authors as “modified environments”. This can be confusing if one thinks of the whole robot as the phenotype rather than only the controller (which is the part encoded by the genotype). Finally, there are experiments in which the computational model in which a more commonly defined environment (non-agent and non-fitness function) is changed [58][80]. All these forms of environmental change are equivalent because they are all a non-genotype change to the model which may result in a change to the fitness value assigned to a given genotype.

Adaptability is measured in a wide variety of ways. One way is to change the environment at multiple constant rates and observe the population’s average fitness [58]. Higher adaptability populations can adapt to more of the rates and have higher average fitnesses. Alternatively, one can present a number of distinct environments for a fixed number of generations and observe the fraction of searches which are able to achieve high fitness in all of them [44]. Another measure is to first perform a search until a high fitness individual is found, then change the environment once and measure the number of generations needed to again find a high fitness individual [87].

Adaptability in this thesis is measured as follows: The environment is changed in very small increments until the fitness of a originally high fitness population becomes lower than a threshold (but is still high fitness). The incremental environmental change and fitness evaluation is performed without mutating the individuals in the population. Then the GA search is resumed with mutation and selection occurring,
but no environmental change. Once the population’s fitness is higher than the threshold, the GA search is paused and the environment is changed in small increments until the fitness of the population is lower than the threshold again. This process repeats for a predetermined number of generations. The total amount of environmental change is recorded as the “adaptability”.

This measurement of adaptability was used instead of the above because it avoids the need to calibrate the size or rate of environmental change. The experiments cited above [58][44] change the environment in a way that ignores the population’s fitness. This is more realistic with respect to environmental change found in nature. However, the method used in this thesis has the advantage of requiring fewer GA searches because a GA search is not performed on the same initial population at different constant rates of change. This is needed with the experiments cited above because too high of a rate will not allow the population to maintain high fitness, too low of a rate will not allow the population to adapt to the most environmental change possible, and these “too high” and “too low” rates will differ between initial populations. Additionally, a given population can regain fitness at a different rate than another population in the same environment. Continuously adjusting the amount of change according to the ability of the population to regain fitness achieves the largest total amount of environmental change while maintaining a high fitness population.

Adaptability is of interest for a number of reasons. First, it is a more biologically realistic measure of evolvability than discoverability. Measuring discoverability requires finding a random or otherwise naive population, which does not occur in nature. Additionally, a changing environment has been shown to select for adaptability [58][44]. This is of interest for those researchers trying to create evolvable encodings [94][62][33].

A third way to measure evolvability is by competition between individuals encoded in different ways [83][26]. A competition is a way to measure relative adapt-
Discoverability, adaptability, and competitions are mechanism-independent measures of evolvability. They do not depend on how the genotype creates the phenotype. Instead, they are a measure of how easily a population can change to increase or maintain fitness.

**Mechanisms of evolvability**

What causes some G-P maps to have higher evolvability than others? Wagner and Altenberg suggest that biological G-P maps have high evolvability because they have “modularity” [98]:

“By modularity we mean a genotype-phenotype map in which there are few pleiotropic effects among characters serving different functions, with pleiotropic effects falling mainly among characters that are part of a single functional complex. Such a design is expected to improve evolvability by limiting the interference between the adaptation of different functions.”

In other words, a G-P map with high evolvability is one that creates phenotypes with “complexes” which can be changed (to a high degree) without changing other complexes. Furthermore, different complexes have different functions which interact to determine the phenotype and fitness. This idea of modularity seems to align with what we observe in the natural world: Organisms are not homogenous collections of molecules but instead have complexes which are separated by binding strength differences between molecules, membranes separating cells and creating compartments in cells, and groupings of a specific cell type.
Many evolutionary computation researchers have focused on G-P maps creating modular phenotypes to increase evolvability. These “modules” correspond to the “complexes” idea of Wagner and Altenberg. Some examples of modules include genetic regulatory networks which are active at different times during genotype to phenotype mapping [11], body structures [54], pre-adapted strings which can be combined (a model of symbiosis) [102], electrical circuits [49], various types of neural networks [36][13][74][49][78], and metabolic networks [38].

However, the concept of modularity, even more so than the concept of evolvability, is loaded with definitions. There are definitions of modularity relating not only to biology and computer science, but also to engineering, mathematics, cognitive science, and art [99][14].

Even within those articles which cite Wagner and Altenberg [98], the concept of modules could be said to be inspired by, rather than based on, the concept put forward by Wagner and Altenberg. Only in one article [101] were the modules shown to be non-interacting units of the genotype aligned to functional complexes in the phenotype. In other articles, the assignment of modularity is made based on network topological measures [74][49] or repeated structures in the phenotype [11][54]. Whether these structural modules are also functionally independent of one another has not been shown [100].

There are several other mechanisms which, though not claimed to be modular, have been shown to increase evolvability. Some examples include the ability to select for the probability that different elements of the genotype will be mutated or crossed-over together [37][68][75][72], mutation rates which can be influenced by selection [12][5], large scale genetic moves (similar to transposons) [27], and reduction in the dimensions of the genetic search space (often as a side-effect of using some type of module) [53][89][40][76][74][20][19].

A less biologically motivated way to increase evolvability over standard GAs is to
use Estimation of Distribution Algorithms. In EDAs populations are created using a probability distribution [67]. The probability distributions are generated by examining where high fitness individuals are located in parameter space. The next generation is created using the probability distribution as a guide. This can lead to exploitation of problem regularities when creating candidate solutions. Because EDAs are not based on what is known in biology, they are less motivating to biologists or computer scientists creating models of biology.

**Mechanism of evolvability relevant to this thesis**

Ultimately what influences evolvability is how changes to the genotype cause changes in the phenotype [98][95]. In the evolutionary computation literature inspired by Wagner and Altenberg [98], the mechanism of evolvability is found in the encoding of the individual. The encoding determines how the genotype is mapped to the phenotype. It determines which phenotypes are probable and possible.

**Evolvability experiments**

**Experiments demonstrating that indirect encodings often have higher evolvability than direct encodings**

Researchers have performed numerous comparisons of the evolvability of direct and indirect encodings in many problem domains: creating the shape of tessellating tiles [7], creating tables [40][41], locomotion of agents [36][54][42][82][83][23], pattern matching [20], and playing games [73].

In general, researchers report that indirect encodings have higher evolvability than direct encodings. One example comparison is by Gruau [36] illustrated in Figure 4. These experiments compare a direct encoding with an indirect encoding for creating a neural network controller (including both its topology and connection weights) for locomotion of a hexapod agent. Gruau finds that the indirect encoding has higher
discoverability than the direct encoding.

**Indirect requires fewer evaluations than direct**

Figure 4: Indirect encodings are often found to have higher discoverability than direct encodings. In this case from [36] the indirect encoding requires fewer fitness evaluations than the direct encoding to find equivalently high fitness individuals. The left hexapod contains a diagram of one of the high fitness neural networks created by the direct encoding and the right diagram one of the high fitness neural networks created by the indirect encoding. The direct encoding required 3.6 times more evaluations than the indirect encoding.

There are at least three hypothesized mechanisms by which the genotype reuse of an indirect encoding increases evolvability.

First, phenotypes created by indirect encodings will often have regularities (such as symmetry, segmentation, repeating patterns, and other self-similarity) which may be useful in exploiting the regularities of some problems [7][89][40]. These regularities are a result of the one-to-many genotype to phenotype mapping that characterizes indirect encodings. In this case the one-to-many mapping is a bias or a constraint which causes an indirect encoding to produce phenotypes with regularities more often
than a direct encoding. Continuing the example from [36] in Figure 5, the direct encoding produces a monolithic neural network with no discernible regularity while the indirect encoding produces a network composed of three identical subnetworks.

Figure 5: Does an indirect encoding often have higher evolvability because it more easily exploits problem regularities? From [36], the neural network produced by the direct encoding has no obvious regularity (LEFT), whereas the neural network produced by the indirect encoding is composed of three repeated subnetworks (RIGHT).

Second, because parts of the genotype are reused, an indirect encoding of a phenotype is often smaller than a direct encoding. Fewer genotype parameters result in fewer possible genotypes and a reduction in the dimensionality of the space searched by the GA. The GA can search a smaller space more quickly than a larger space, possibly finding high fitness individuals more easily and resulting in more evolvability [53][89][40][76][74][20][19]. In Figure 6, the direct encoding has a genotype which is four times larger than the indirect encoding and a corresponding exponential increase in the search space size.
Figure 6: Does an indirect encoding often have higher evolvability because it more easily exploits problem regularities? From [36], the neural network produced by the direct encoding has a larger number of nodes (a.k.a. neurons) and uses a larger genotype to describe them (LEFT), whereas the neural network produced by the indirect encoding has four times fewer nodes and can therefore describe it with a smaller genotype (RIGHT).

A third mechanism is specific to the adaptability definition of evolvability. An indirect encoding will change many parts of the phenotype when one part of the genotype is changed. This coordinated change can be useful for adapting to a changing environment. Additionally, it is also hypothesized [98] and shown [58][72][26] that a changing environment can select for adaptability. Thus it may be that indirect encodings bootstrap their evolvability in a changing environment in a feedback loop

Experiments exploring the relationship between problem regularity and genotype reuse in the encoding

Only a few articles test the hypothesized mechanisms responsible for increased
discoverability and adaptability. (Instead the goal is a proof of concept that indirect encodings can have higher evolvability than direct encodings.)

Two articles explore the benefit to discoverability of a match between the regularities in the phenotype created by indirect encodings and regularities in the problem to be solved. Hornby [41] and Clune [20] use complementary approaches. Hornby [41] holds problem regularity constant while additional types of genotype reuse are enabled, while Clune [20] uses the same indirect encoding while problem regularity is adjusted.

Hornby [41] shows for one model system that as more types of genotype reuse are enabled, higher fitness individuals are easier to discover. The indirect encoding used is a grammar rewriting system\(^2\). It can reuse its genotype by iteration, recursion, and control what is reused with conditional statements. The indirect encoding becomes a direct encoding if iteration and recursion are not allowed. The goal is to create simulated physical tables. The fitness function includes height, surface structure, stability, and a penalty for excess material. As more types of genotype reuse were enabled, higher fitness tables became easier and easier to discover. This is partially due to increased stability which results when multiple legs of the table are created with the same element of the genotype [39]. This implies that table designs of high fitness are more easily achieved with phenotypes with structural regularity.

Clune [20] shows for another model system that as problem regularity decreases, the discoverability of an indirect encoding also decreases. The system consists of a grid of input nodes connected to a same size grid of output nodes. Each input node may connect to a number of output nodes ranging from none to all. The task is to connect an input node to one specified output node. Regularity is varied by changing the pattern by which input nodes are assigned to output nodes. Initially regularity is high and the input node and output node are in the same row and column. Next, the

\(^2\)specifically, an extended parametric L-System
input and output node are within the same row, but the fraction which are within the same column decreases until no input and output node are within the same column. Finally, the fraction of input and output nodes within the same column is also decreased to zero. The indirect encoding is a function which takes as input the coordinates of the input and output node and outputs whether the two nodes are connected. This function is composed of multiple functions whose identities are under selective pressure and can be evolved. It was found that the indirect encoding is more discoverable when there was a relatively high amount of regularity in the connection pattern between input and output nodes. This implies that indirect encodings are more suited for more regular problems.

Experiments demonstrating that changing environment selects for adaptability

Other articles investigate the role of changing environment to select for increased adaptability [58][75][49][72][26][66].

These experiments usually compare the adaptability of a single evolvable indirect encoding when evolved in different amounts of environmental change. For example, an unchanging environment and a changing environment. The differences in adaptability are then attributed to differences in the form of the indirect encoding brought about by selection pressures present in the changing environment but not present in the unchanging environment. (In contrast, the majority of evolvability experiments in EA, such as those described in the previous section, focus on discoverability of one encoding (e.g. direct) versus another (e.g. indirect) in an unchanging environment.)

Two known articles include both a direct and indirect encoding when investigating the ability of changing environment to select for increased adaptability [75][44]. Both use genetic regulatory networks as an indirect encoding.

Reisinger’s [75] task was to create a bilaterally symmetric binary pattern. Three encodings were used; a direct encoding which was a binary string representing the
pattern, a modified-direct encoding which included parameters to link mutation between parts of the binary string, and a genetic regulatory network (GRN) in which the pattern was represented by the network’s state after a fixed number of iterations. First, the populations were evolved in a changing environment for 150 generations. The target pattern was changed every generation a varying amount depending on the experiment. The probability that a bit in the pattern would change each generation ranged from 0 to 0.5. This resulted in a pattern whose change ranged from never to randomly generated from generation to generation. Then the populations were tested on a final pattern and allowed to evolve for 100 generations. The “efficiency” was measured as the average fitness of the best individual in each of the final 100 generations. Because efficiency measures how well a population created through GA search adapts to a new fitness requirement, it will be referred to as “adaptability” to keep nomenclature consistent with the rest of this thesis.

For the modified-direct and the GRN encodings, the adaptability was lower in experiments where the pattern had low or high rates of change and bulged higher at intermediate rates of change. Reisinger interpreted this to mean that the intermediate rates of pattern change contained enough information about the symmetrical bias of the solution space to select for that bias in the modified direct and GRN encodings. The rates of pattern change at the extremes of never changing and randomly changing did not provide enough information about the symmetry of the problem to select for bias in the modified direct and GRN encodings.

In contrast, the direct encoding did not show an upwards bulge in adaptability at intermediate rates of change but was instead flat across all rates of change. This is because the mapping of genotype to phenotype could not be changed by selection (it was not an evolvable encoding) and could not be biased.

Interestingly, the direct encoding had higher adaptability at every rate of pattern change than the modified-direct and GRN encodings. This means that the direct
encoding found higher average fitness best individuals when adapting to new target patterns. For this problem, the direct encoding is a better choice if one were simply trying to find the most fit solutions.

A few articles investigate the role of a changing environment to select for adaptability using a direct encoding with additional parameters that change how genotype is mapped to phenotype [58][72][26]. Although the adaptability changes are a result of selection on the encoding, because these encodings are not indirect, the mechanism is not genotype reuse. Therefore these experiments are an important reminder that a difference in evolvability can originate for reasons other than those associated with genotype reuse. For example, epistasis is the interaction between parts of the genotype when creating the phenotype. Two articles show that a changing environment selects for less epistasis and more adaptability [58][26]. The third article [72] uses linkages to coordinate mutations between parts of the genotype in order to create a bilaterally symmetric phenotype. The number of mutation linkages could be manipulated to be few or many. It was found that the fewer the number of mutation linkages (and the larger the linked parts of the genotype) the more easily a changing environment could select for mutation linkage values which biased the phenotypes towards bilateral symmetry. From this it was concluded that selection for adaptability depends on the number of mutations needed in order to change the genotype to phenotype map to have a bias that is appropriate to the problem being solved.

Segue to thesis experiments

The articles reviewed above clarify two circumstances which allow indirect encodings to have more evolvability than direct encodings.

First, those articles show that problems from many domains have inherent regularities which indirect encodings can often exploit because of their one-to-many G-P mapping. One can conclude that evolvability is highest when the encoding generates phenotypes whose structure matches that of the problem domain.
Second, the articles show that a changing environment can perform selection on G-P maps such that they become more adaptable. An indirect encoding evolved in a changing environment is often shown to be more adaptable than when evolved in an unchanging (or randomly changing) environment on a highly regular problem.

However, there are other aspects of indirect encodings which have not been investigated. This thesis investigates two additional circumstances.

First, the reviewed articles do not investigate the contribution of parameter space size in conjunction with genotype reuse on evolvability. A smaller parameter space has been hypothesized to increase evolvability \[53\][89][40][76][74][20][19]\ because it could be explored in fewer generations. However a decrease in parameter space size usually comes with an increase in genotype reuse. Which is more important? Section 3 investigates whether genotype size or genotype reuse increases evolvability more in one model system.

Second, these papers do not investigate whether an indirect encoding can have higher evolvability than a direct encoding on a problem in which fitness does not depend on geometric structural regularity. In most of the above papers the problem used had some form of geometric regularity. This might lead to the conclusion that indirect encodings only increase evolvability in problems with geometric regularity. In support of this conclusion one paper [20] gave an example of the discoverability of an indirect encoding increasing as problem regularity increases. However, there are other forms of regularity in addition to geometric regularity. Section 4 explores the possibility that an indirect encoding can have higher evolvability on a problem in which fitness does not depend on geometric structural regularity.
3 Genotype reuse can be more important than genotype size to evolvability: hexapod experiments

As described previously, the increase in evolvability of indirect encodings over direct encodings can be attributed to multiple reasons.

This uncertainty exists because when a directly encoded solution is indirectly encoded, both genotype reuse increases and genotype size decreases. Either change could result in more evolvability. Increased genotype reuse causes more structural reuse in the phenotype which may match regularities in the problem domain. Fewer genotype parameters could also result in faster search because the dimensionality of search space is smaller.

This section compares the relative importance of genotype reuse and number of genotype parameters on evolvability. The system consists of a neural network controlling a simulated hexapod agent in a simple walking task. The amount of genotype reuse can be changed by changing the encoding of the neural network. The direct encoding of the neural network has no genotype reuse while the indirect encoding has a large amount of genotype reuse. The number of genotype parameters can be changed by changing the number of interneurons and how the neurons are connected to each other, also known as the neural network’s architecture.

By switching between neural network architectures and changing the number of interneurons, it is possible to compare the evolvability of an indirectly encoded neural network with a larger genotype to a directly encoded neural network with a smaller genotype. Additional experiments are performed to control for changes in the number of interneurons and the architecture.

The evolvability of these encoding/architecture combinations are examined from several different perspectives described on page 25.

The results show that, for these experiments, genotype reuse is more important
than genotype size in the relative evolvability of encodings. The indirect encoding has higher evolvability than the direct encoding, even if the indirect encoding’s genotype is larger than the direct encoding’s genotype. The indirect encoding also wins competitions for population share against direct encodings, even if the indirect encoding has a lower fitness at the beginning of the competition.

Methods

Neural network

The hexapod is controlled by the output of a continuous-time recurrent neural network (CTRNN):

$$\tau_c \dot{y}_c = -y_c + \sum_{j \in \text{IN}} w_{jc} \sigma(y_c + \theta_c) ,$$

where $y_c$ is the state of the “current” neuron $c$, $\dot{y}_c$ denotes the time rate of change of this state, $\tau_c$ is the neuron’s membrane time constant, $w_{jc}$ is the strength of the connection from neuron $j$ to the current neuron $c$, $\theta_c$ is a bias term, $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic function, and $\text{IN}$ is the set of all neurons with connections to the current neuron, including the current neuron because of a self-connection. Connection weights and biases were constrained to the range $\pm 16$ and time constants to the range $[0.5, 10]$. The state of each neuron was initialized randomly to a value between $\pm 0.1$. This random noise helped break symmetry in what otherwise could be a completely symmetrical neural network architecture and initial body arrangement. The body and neural network were integrated using the forward Euler method with a step size of 0.1.

Hexapod body

The body (Fig. 7) is a hexapod agent based on prior work by Beer & Gallagher [32] and similar to [53]. Three pairs of legs are attached to the 50 unit long body.
One pair of legs were located at each end of the body’s length. The other leg pair was attached to the middle of the length. The length of the leg can be adjusted (and lengths between 5 and 15 were used in the experiments).

The output of the neural network determines the forces applied to the body. The architecture of the neural network will be described more fully below. There are at least three neurons associated with each leg. MaxLegForce (0.05) is scaled by output two neurons, BackSwing (BS) or ForwardSwing (FS) (output range between 0 and 1) to become ForwardForce and BackwardForce. These two forces are analogous to the forces generated by two opposing muscle groups on a joint: they can sum to cancel with no resultant force. The leg can move through an angle of $\pm \frac{6\pi}{24}$ radians, with 0 radians being perpendicular to the body’s long axis. This range is drawn as two solid lines on either side of the legs in Figure 7. At the extreme $\frac{\pi}{24}$ regions the MaxLegForce is scaled down by WeakerFactor (0.1). These extreme regions are enclosed by solid lines and the dashed lines on either side of the legs in Figure 7. These extreme “weak regions” are designed to degrade fitness gradually compared to the alternative of the leg suddenly not being able to apply force when outside its operating range.

ForwardForce and BackwardForce act on the body in different ways depending on whether the foot is raised or lowered and whether the body is “supported” or “unsupported”.

The foot can be raised or lowered. The Foot (FT) neuron’s output raises and lowers the foot. Half the FT output range causes the foot to be lowered and the other half causes the foot to be raised. If the foot is raised, then the ForwardForce and BackwardForce sum to create a torque which changes the angular velocity of the leg. Mass is not modeled as distributed along the length of the leg. As a result, torque changes the leg’s angular velocity independent of the leg’s length. The maximum torque (0.5) is great enough to move the leg through its entire range faster than the maximum velocity MaxVelocity (1.0) of the hexapod. This allows legs swinging
forward to be equal to or faster than legs swinging backward, which is necessary for a successful tripod gait.

The body can be supported or unsupported. If the lowered feet form a polygon which contains the center of the body, the agent is considered “supported”. The ForwardForce and BackwardForce from all legs with feet on the ground are summed and the resultant applied to the body. The resultant changes the velocity of the body. As the body moves the leg angle is computed from their attachment point on the body and the point where the foot touches the ground. When legs move past the outer angle limits their feet are disengaged from the ground and the leg angle reset to just inside the outer angle limit. The length of the leg does not change the amount of force applied to the body. As a consequence, the body can achieve the same velocity no matter the leg length. If the body is unsupported, then a large amount of friction (25) is applied in a direction opposite its velocity.

The MaxVelocity prevents the body from “outrunning” the legs. Without a maximum velocity the hexapod body accelerates to a point where the velocity of the body is greater than the ability of the legs to swing forward to continue a tripod gait. This causes a hexapod with a good tripod gait to eventually fall as the forward pushing tripod of legs is not replaced by the other tripod of legs before reaching the angle limit of the legs. One way to solve this problem would be to give the hexapod sensory feedback so that the neural network’s output can be modified based on leg angle. Instead, the body’s linear maximum velocity and the leg’s angular maximum angular velocity were chosen to allow the leg to sweep through its full range fast enough to maintain a tripod gait. Additionally, a small amount of friction opposes velocity even when the agent is supported so that the legs must continue to accelerate the agent rather than merely have a supporting arrangement.

The body is initialized with all legs at the forward-most limit of the non-weak region and all feet down so that the body is supported.
Fitness ranges from 0 to 1. The fitness is the average velocity if the average velocity is positive (forward) and zero if the average velocity is negative.

“High fitness” individuals are defined as any individual with a fitness $\geq 0.75$. This threshold was determined by observing the hexapod walking behavior. “High fitness” individuals begin walking by first pushing backwards with all six legs, decoupling one set of three legs from the other three, reaching forward, and beginning a tripod gait.

A change in leg length does not change the angular limits that a leg can swing through. However, the linear distance between the angular limits changes. The foot of a longer leg can travel farther in linear distance between its angular limits than a shorter leg can travel between the same angular limits.

Fitness degrades as a leg is shortened because the foot encounters the angular limits sooner with a short leg than when the leg was longer. A neural network described
below, previously evolved to control a longer leg will swing for a too long a time relative to a shorter leg. This forces the foot off the ground, possibly causing the hexapod to be unsupported and thereby decreasing its velocity.

As mentioned above, the same fitness can be achieved by legs of any length. When the foot is up angular acceleration and velocity depends on only ForwardForce and BackwardForce (determined by the output of FS and BS). When the foot is down the force applied to the body again depends only on ForwardForce and BackwardForce (determined by the output of FS and BS). Therefore, the leg and body can move with the same accelerations and velocities at every length.

**Genetic algorithm**

A real-valued genetic algorithm (GA) is used to evolve the CTRNN parameters. For every CTRNN parameter there is a GA parameter linearly rescaled to lie between [-1, 1]. A population of 100 individuals is maintained and parents are selected for reproduction using a linear rank-based method with the most fit individual producing an average of 1.1 offspring. Children are generated by mutation of selected parents by adding the parent’s vector to a random displacement vector with uniformly distributed direction and normally distributed magnitude [3]. The mutation magnitude has a zero mean. Mutation variance is set to different values depending on the stage of the experiment. Use of elitism (preserving the best individual found by the search through all generations) and the number of generations of search also vary by stage of experiment. The mutation variance, elitism setting, and number of generations used is explained on page 57.

**Neural network topology**

Every neural network is composed of six subnetworks. The six subnetworks are associated with each of the six legs. They contain three motor neurons (FT, BS,
FS) and 0 to 3 interneurons. (An interneuron is a neuron which does not have a connection with an effector.) The number of interneurons is adjusted along with the topology of connections (“architecture”) to increase or decrease the number genotype parameters.

The subnetworks are connected to each other in two different topologies, referred to in this thesis as “architectures”.

One architecture is “locally connected”. Each leg (and its associated subnetwork) has a neighboring leg (and subnetwork) on its counterclockwise and clockwise as one moves around the outside of the hexapod. In the locally connected architecture each type of neuron in each subnetwork is connected to the same type of neuron in the neighboring two subnetworks. For example, each FT neuron is connected to a FT in each of the two neighboring subnetworks. Similarly each interneuron is connected to an interneuron in each of the two neighboring subnetworks. Finally, every neuron within the subnetwork is connected to every other neuron within the subnetwork. Figure 8 shows a locally connected neural network and a detail of one of its subnetworks.
Figure 8: Example locally connected network architecture (lower right) and one of its subnetworks (upper left). The neurons of the subnetwork are represented as circles and the connections between neurons as arrows. (Thickness of connection indicates the weight of the connection.) There is one subnetwork per leg, for a total of six subnetworks. Neurons within a subnetwork are connected to every other neuron within the subnetwork. Outward pointing arrows show connections to analogous neurons in neighboring subnetworks. Motor neurons are named Foot (FT), BackSwing (BS), ForwardSwing (FS), and in this example, one interneuron (.).

The other architecture is “fully connected”. This architecture has the same subnetworks as the locally connected architecture: FT, BS, FS, and 0 to 3 interneurons as described below. Every neuron is connected to every other neuron in the network. A fully connected architecture is illustrated in Figure 9.
Figure 9: Example fully connected network architecture. Each black dot is a neuron and the lines are connections between them. This neural network has the required three motor neurons (FT, BS, FS) and three optional interneurons per subnetwork for a total of six neurons per leg. Every neuron is connected to every other neuron.

Encodings

Two encodings, a fixed direct and a fixed indirect named “symmetric encoding”, are used to store the neural network parameters in the genotype.

The direct encoding is a one-to-one mapping of genotype parameters to phenotype parameters. I.e. every parameter of every neuron is present in the genotype.

In the symmetric encoding, the parameters for only one subnetwork are stored in the genotype. When the neural network is created, the parameters for the stored subnetwork are duplicated six times (once for each leg) and connections are made between the subnetwork (described on page 46) to create the whole network.

The differences phenotype parameters stored in the genotype for the two encodings can be seen by comparing Figure 10 and 11. These two Figures both show the same
locally connected neural network with four neurons per subnetwork. Figure 10 draws a symmetrically encoded neural network’s subnetworks in gray when the subnetwork’s parameters are *not* stored in the genotype and black when the subnetwork is stored in the genotype. The gray subnetworks are duplicates of the black subnetwork. Figure 11 draws a directly encoded neural network’s subnetworks all in black to illustrate that all of their parameters are stored in the genotype.

Note that the symmetrically encoded neural networks are a subset of the directly encoded neural networks. Every symmetrically encoded neural network could also be directly encoded. This is important because when symmetric encodings are shown to have higher evolvability than direct encodings in the experiments described below, it is not because the direct encoding cannot in principle create the same individuals as the symmetric. Instead, there must be other reasons, as described below.

**Figure 10:** Example symmetrically encoded neural network. The parameters of one subnetwork is stored in the genotype. When the phenotype is created the subnetwork is duplicated six Example symmetrically encoded neural network times and connected to form the full network. This is depicted by coloring the subnetwork stored in the genotype black and the duplicate subnetworks gray.
Figure 11: Example directly encoded neural network. The parameters of all neurons are stored in the genotype. This is depicted in the figure by coloring all neurons black.

**Naming the neural network**

A neural network in these experiments can be fully specified using three attributes: encoding, architecture, and number of neurons per subnetwork.

- Encoding can either be symmetric (S) or direct (D).
- Architecture is either locally- (L) or fully- (F) connected.
- Number of neurons per subnetwork \(n\). \(n\) can be any number equal to or greater than 3 (for the FT, BS, and FS neurons). \(n\) is \(\frac{1}{6}\)th the total number of neurons in the network.

Only a few choices of \(n\) are needed to test whether or not genotype encoding is more important than genotype size. These choices of \(n\) will be described on page 54.
Calculating the number of genotype parameters

The number of parameters in the genotype depends on the architecture (locally- or fully-connected), the number of interneurons, and the encoding (symmetric or direct). The number of parameters in the symmetrically encoded genotypes is simply \( \frac{1}{6} \) that of the direct encoding.

Symmetrically encoded (S), locally-connected (L) genotypes (SL\(n\)) are of size:

\[
\text{NumParamsSymmetric}_{\text{local}}(n) = n^2 + 4n
\]

where \( n \) is the number of neurons per subnetwork (three motor neurons plus number of interneurons (if any)). Each subnetwork is fully-connected, with every neuron connected to every other neuron. This has the same topology as a complete digraph, with neurons corresponding to vertices and connections corresponding to edges. A complete digraph has twice as many edges as a complete graph \( \left( \frac{1}{2}n(n-1) \right) \) (which has only one edge connecting every vertex): \( n(n-1) = n^2 - n \). So the number of genotype parameters to store connection weights between neurons in the subnetwork is also \( n^2 - n \). Each neuron also has three other parameters, a membrane time constant \( \tau_c \), and a self-connection weight \( w_{cc} \), and a bias \( \theta_c \) for a total of \( 3n \) more genotype parameters for the subnetwork: \( n^2 - n + 3n = n^2 + 2n \). Finally, every neuron in the subnetwork is connected to the equivalent neuron in the two subnetworks associated with the clockwise and counterclockwise neighboring legs for an additional
2n connection weight parameters: \( n^2 + 2n + 2n = n^2 + 4n \). In a symmetrically encoded neural network only a single subnetwork’s parameters are stored in the genotype. The phenotype is created by unpacking the single subnetwork’s parameters six times to create and join six subnetworks.

Symmetrically encoded (S), fully-connected (F) genotypes (SF\( n \)) are of size:

\[
NumParamsSymmetric_{full}(n) = n^2 + 2n + 5n^2
\]

Here there are \( n^2 + 2n \) genotype parameters associated with the connections and properties of the neurons within each subnetwork the same as in the SL\( n \) neural networks. The neurons in each subnetwork are also fully- and bi-directionally connected with every other neuron in the other subnetworks. Consider first a single neuron in the subnetwork which is stored in the genotype. It is connected to \( n \) neurons in 5 other subnetworks (associated with the other five legs) for \( 5n \) connection weight parameters. The remaining \( n - 1 \) neurons in the subnetwork also have the same number of connections for an additional \( (n - 1)5n \) connection weight parameters: \( (n - 1)5n + 5n = 5n^2 - 5n + 5n = 5n^2 \) connections between subnetworks. The total number of genotype parameters for a symmetrically encoded, fully-connected genotype with \( n \) neurons per subnetwork is \( n^2 + 2n + 5n^2 \).

The number of parameters for the directly encoded genotypes is simply six times more than the symmetrically encoded genotypes (DL\( n \) and DF\( n \)).

\[
NumParamsDirect_{local}(n) = 6NumParamsSymmetric_{local}(n)
\]

\[
NumParamsDirect_{full}(n) = 6NumParamsSymmetric_{full}(n)
\]
Encodings and neural network topologies combinations of interest

As discussed previously (on page 32) there are multiple reasons why indirect encodings are found to have higher evolvability than direct encodings. The main two put forward are that indirect encodings have smaller genotypes than direct encodings and that indirect encodings create phenotypes with regularities which match the regularities of the problem (due to genotype reuse).

The example from Gruau [36] used on page 32 is a good reminder of the typical experiment comparing direct and indirect encodings. These results are summarized in Figure 12 on the following page. The direct encoding (left) is encoded by more genotype parameters, has less genotype reuse, and has lower evolvability than the indirect encoding (right).
Figure 12: Typical direct versus indirect comparison. This example from [36] shows two neural networks evolved to control a hexapod whose task was to maximize velocity. The left neural network is created by a direct encoding. The right is created by indirect encoding. The indirect encoding has fewer genotype parameters, more genotype reuse, and more evolvability than the direct encoding.

Other researchers’ experiments find similar results [36][7][54][42][40][41][30][82][83]-[20].

One can then ask the question “Which is more important to evolvability, smaller genotype size or more genotype reuse?”

To answer this question for the hexapod model system, the evolvability of directly encoded neural networks of a given number of genotype parameters should be com-
pared to indirectly encoded neural networks with more genotype parameters. DL3 is the smallest directly encoded neural network possible: Locally connected neural networks have fewer parameters than fully connected and $n = 3$ is the smallest $n$ because each subnetwork needs at least three motor neurons. The formulas on page 52 calculate that DL3 has 126 genotype parameters. Next, we desired to find a symmetric encoding which has slightly more genotype parameters. This neural network is SF5 with 160 genotype parameters. To begin to form a trend, another pair of neural networks were selected to compare: DL4 with 192 genotype parameters and SF6 with 228 genotype parameters. These pairs along with the size of their genotypes are listed in Table 2. If parameter space size alone were the sole determinant of evolvability and a larger parameter space meant less evolvability, then one would expect DL3 (126 genotype parameters) to have higher evolvability than SF5 (160) and DL4 (192) to have higher evolvability than SF6 (228).

\[
\begin{array}{|c|c|}
\hline
\text{DL3 (126)} & \text{vs.} & \text{SF5 (160)} \\
\text{DL4 (192)} & \text{vs.} & \text{SF6 (228)} \\
\hline
\end{array}
\]

Table 2: The two pairs of directly and symmetrically encoded neural networks. The size of their genotypes is in parenthesis. The symmetric encoding in each pair has slightly more genotype parameters than the direct encoding.

Figure 13 is a graphical representation of one of these pairs: DL4 versus SF6. Notice that the neural network with more genotype reuse also has a larger genotype size. This is a reversal of the relative size of genotype size as compared to typical comparisons of direct and indirect encodings.
Figure 13: An illustration depicting the comparison between a directly encoded neural network with fewer genotype parameters (LEFT) than a symmetrically encoded neural network (RIGHT). (In this illustration, DL4 versus SF6.) The “versus” arrow symbolizes the evolvability comparisons performed. If the symmetrically encoded neural network on the right is more evolvable, then structural regularity is more important than genotype size. The neural network connections shown in black are stored in the genotype, while those in gray are present only in the phenotype and are duplicates of the black. All neural network parameters are stored in the genotype for when directly encoded, so its connections are all drawn in black (LEFT). The symmetrically encoded neural network (RIGHT) has only one subnetwork stored in the genotype, drawn in black. The other five subnetworks are duplicates, drawn in gray.

**Experiment types**

The evolvability experiments were of three types: (degree of) discoverability experiments, adaptability experiments, and competitions.

**Degree of discoverability experiments**

High fitness individuals were found through GA search beginning with a naively initialized population. The genotype parameters for each of the 100 individuals in
the population were initialized by choosing a uniformly distributed random number between [-1,1]. A GA search was performed for 500 generations with a mutation variance of 0.05 and with elitism. The search was said to have “discovered” a high fitness individual if the best individual in the population had a fitness of greater than 0.75.

The fitness of 0.75 was chosen as a cutoff by subjective observation. The behavior of a hexapod with fitness greater than or equal to 0.75 was a fairly coordinated tripod gait. The legs swung through the full range of the angles except the extreme “weak” angles of the leg’s range of motion. This behavior was judged to indicate the neural network has a qualitatively “good” output relative to its body.

The discoverability of each encoding/architecture was determined as follows. GA searches initialized from different random seeds were run until 20 high fitness individuals were found or 1000 attempts were made without any high fitness individual found, whichever came first. Discoverability is reported as number of searches in which a high fitness individual was found divided by the total number of attempts. Discoverability for an example three searches is shown in Figure 14.
Figure 14: Degree of discoverability for hexapod experiments is the fraction of searches whose highest fitness individual is greater than a target fitness of 0.75 after 500 generations. This illustration shows the median fitness of three imaginary searches by generation. Two of three reach the target fitness by generation 500, so discoverability for this grouping of searches is $\frac{2}{3} = 0.66$.

**Equilibrium search**

An equilibrium search is a situation where the population’s location in parameter space and GA settings (such as mutation variance and selection pressure) are balanced to allow high fitness populations to exist for an indefinite number of generations without the use of elitism.

An attempt to create an equilibrium search is performed in the following way: A high fitness individual is copied to fill a population. A GA search is performed for a predetermined number of generations. The GA search is an equilibrium search if the population is above a fitness threshold in the last generation.
Multiple attempts to find an equilibrium searches were performed for each discoverability experiment in which a high fitness individual (fitness greater than or equal to 0.75) was found. Each high fitness individual was used to seed GA searches run at mutation variances from $1 \times 10^{-6}$ to $1 \times 10^{-1}$ for each power of 10. A range of mutation variances was used because the layout of parameter/fitness space is unknown and the mutation variances which will lead to an equilibrium GA search could not be determined a priori. The initial population of each attempt consisted of 100 copies of the high fitness individual. If the population was able to maintain a median fitness $\geq 0.75$ in the last generation (500) of GA search without elitism it is deemed to be an equilibrium GA search.

One caveat arises as a result of GA searches using direct encodings not finding any high fitness individuals. Attempts to find equilibrium GA searches for indirect encodings use indirectly encoded high fitness individuals as seeds, as expected. But because no high fitness individuals were found in the discoverability experiments for the direct encoding the same indirectly encoded high fitness seed individual were used instead. First, the indirectly encoded individual genotype was mapped to a phenotype. Then the phenotype was inverse mapped back into a directly encoded genotype. This inverse mapping is always possible without loss of information because the direct encoding is a superset of the indirect encoding. Finally, the newly created directly encoded individuals are used to seed attempts to find equilibrium populations for the direct encoding. As the GA search proceeds mutations cause the directly encoded individuals to diverge from their indirectly encoded ancestor.

This caveat is also a control. Both direct and indirect equilibrium searches found originate from identical high fitness seed individuals.

**Adaptability experiments**

Next, the adaptability was measured using the equilibrium searches found as described in previous section. For each population, the median fitness is recorded as
the target fitness. Then each equilibrium search is duplicated 5 times. The population, mutation variance, and other GA settings are duplicated, but the initial neural network states $y$ are initialized to random values between $\pm 0.1$. Next, the length of the legs (starting at 15) of the individuals in the population are shortened (by 0.1) until the population fitness is less than the target fitness (Figure 15). The GA search starts again and continues until the median population fitness is greater than or equal to the target fitness. Then the GA search is paused and the legs are incrementally shortened again. This is repeated until either 5000 generations has passed, or rarely, a leg length of 10 was reached. An imaginary adaptability search illustrating the effects of alternating leg length shortening and GA search on fitness is shown in Figure 16.

The total change in leg length for each search is recorded and the median for each set of 5 duplicates run at the same mutation variance is calculated. Finally, the medians of each mutation variance are compared and the largest becomes the adaptability of that seed individual. Later, the adaptabilities of different types of seed individuals are grouped and their medians compared. An illustration of the adaptability experiments from high fitness individuals from discoverability experiments, to equilibrium populations, to reruns with changing leg length is shown in Figure 17.

Figure 15: Hexapod neural network controllers must adapt to having their legs shortened. The legs are not specified by the genotype. Shortening the leg causes the amount of time needed for a full swing of the leg to decrease. Neural networks whose oscillations are tuned to swing for one length leg lose fitness when the leg length is shortened.
Figure 16: Adaptability is measured as the total amount of leg shortening in 5000 generations. This figure shows an imaginary population’s median fitness at each generation of the GA search. Leg shortening is indicated by dashed vertical lines. Arrows indicate two of the eight times leg shortening occurs. This results in a loss of fitness. Between each leg shortening GA search occurs and an eventual gain in fitness. One of the eight interludes of GA search is indicated by a bracket. Leg shortening occurs any time the fitness becomes greater than the threshold fitness (which is the population’s median fitness at the beginning of the adaptability GA search).
Discoverability experiments

Perform adaptability experiment on equilibrium searches

Generate equilibrium searches from seed.

Largest median adaptability of 5 reruns is the adaptability of the seed individual.

Figure 17: Stages of adaptability experiments. Dashed vertical lines are filters described at their bottom. High fitness individuals from the discoverability experiments are used to generate equilibrium populations at many different mutation variances. Equilibrium searches with median fitness $\geq 0.75$ proceed to be used in adaptability experiments. The largest amount of median shortening from the set of equilibrium searches created from a seed individual becomes the adaptability of the seed individual. Adaptabilities of seed individuals are grouped by different encodings/architectures and compared.

Competitions

Competitions were performed in a similar manner as adaptability experiments. Recall that each high fitness individual from the discoverability experiments potentially created equilibrium searches at multiple mutation variances. From the set of equilibrium searches created by the same seed, the one with the highest median population fitness was chosen to compete. Then the most fit halves of two equilibrium search from the encodings/architectures of interest are placed in the same population. Each individual and its descendents are mutated with a mutation variance matching the equilibrium search it came from. In some experiments the GA search was then continued as in the adaptability experiments: Leg shortening alternated with GA search in a cycle as described above. In other experiments using the same combinations of equilibrium searches the leg length was not altered. Each pairing of subpopulations
is repeated five times using different initial neural network states. A subpopulation’s initial median fitness varies depending on the initial neural network states. If either subpopulation does not have an initial median fitness of greater than 0.75 for any competition, all competitions with that pair of subpopulations is thrown out. The competition is terminated when all individuals are of the same encoding or after 5000 generations, whichever comes first. A win is assigned to the encoding which makes up the majority of the population at the end of the search. Ties are assigned as a win for the direct encoding to avoid the results favoring the indirect encoding.

Figure 18: Stages of competition experiments. Dashed vertical lines are filters described at their bottom. Two halves of equilibrium searches of different encodings are placed in the same population and five different times with different initial neural network states. A GA search is run on the population. The encoding which has the greatest number of individuals in the population wins the competition after 5000 generations.

Statistical significance

The significance of the differences of the various measurements was determined by comparing the confidence intervals of the measurement created by bootstrapping
Bootstrapping is a method for generating distributions of a statistic such as the mean or median by repeatedly sampling the data to create a distribution of the statistic of interest. The confidence interval for the statistic can then be determined from the distribution of that statistic.

The 95% confidence intervals are compared in these experiments. If the 95% confidence intervals do not overlap, then there is a 5% or less chance that the measures are the same, i.e. $p < 0.05$.

**Results**

**Symmetric encoding is more discoverable and has higher adaptability than direct encoding**

It was found that “discoverability” mostly depends on the encoding type rather than the genotype parameter space size. Discoverability is the fraction of searches finding high fitness individuals from a random initial population. High fitness individuals were discovered for all symmetrically encoded neural networks regardless of parameter space size or neural network architecture. No high fitness individuals were discovered for the directly encoded neural networks (after at least 1000 attempts) even though they were of the same neural network architecture or smaller parameter space size than the symmetrically encoded neural networks.

Figure 19(left) shows the discoverability results as the fraction of searches performed which discovered a high fitness neural network. The symmetrically encoded fractions are marked with a dark gray dot, while the directly encoded fractions are marked with a light gray dot. The fraction of high fitness individuals found for SF5 is $20/2019=0.0099$, for SF6 is $20/5589=0.0035$, and for DL3 and DL4 is $0/1000=0.0$.

Degree of adaptability mostly depends on the encoding type rather than genotype size. In these experiments, degree of adaptability is the final leg length after 5000 generations of incremental shortening. The adaptability results for DL3, SF5, DL4,
and SF6 are shown in Figure 19(right). The $x$-axis is the number of genotype parameters and the $y$-axis is the change in leg length after 5000 generations. It was found that the symmetrically encoded neural networks are more adaptable even when the symmetric encoding has more genotype parameters than the direct encoding. SF6 (228 genotype parameters) is significantly more adaptable than DL4 (192) and the median adaptability for SF5 (160) is higher than DL3 (126) median adaptability, though not significantly.

Figure 19: Symmetric encodings with more parameters are more discoverable and have higher adaptability than direct encodings with fewer parameters. **Left:** Discoverability of the key encoding/architecture comparisons. Fraction of searches beginning with random initial population resulting in discovery of high fitness best individual plotted against size of genotype (dots) and 0.95 confidence intervals of the fraction found (solid vertical lines). High fitness individuals were discovered for all symmetrically encoded neural network architectures. No high fitness individuals were found if directly encoded. (DL3 and DL4 $n=1000$, SF5 $n=2019$, SF6 $n=5589$) See Section 14 on page 59 for methodology. **Right:** Median change in leg length after 5000 generations (dots) and 0.95 confidence intervals of the median (solid vertical lines). Symmetrically encoded neural network were able to adapt to a greater leg length change than directly encoded neural networks, and significantly so for the DL4/SF5 pair. (DL3 $n=405$, DL4 $n=505$, SF5 $n=485$, SF6 $n=570$)
Encoding is more important to adaptability than architecture or number of neurons

The previous section showed that two symmetrically encoded fully connected neural networks with more genotype parameters were more adaptable than a pairing of directly encoded locally connected neural networks with fewer genotype parameters.

This section reports control experiments that demonstrate it is not the architecture or the number of interneurons that led to the above results. In other words, the architecture or number of interneurons did not limit or enhance adaptability, so it must be the encoding which did.

First, experiments were performed to demonstrate that the symmetrically encoded neural networks can do worse if the same topologies are used as in the key comparisons, but are instead directly encoded. SF5 and SF6 are re-encoded to be DF5 and DF6. The architecture (F—fully connected) and number of neurons per leg (5, 6) remain the same, but the encoding has changed (from S—symmetric to D—direct). Figure 20(left) plots the results showing that adaptability decreases for both topologies when directly encoded. This decrease is significant when re-encoding SF6 to DF6.

Second, experiments were performed to demonstrate that the directly encoded neural networks can do better if the same topologies are used as in the key comparisons, but are instead symmetrically encoded. DL3 and DL4 are re-encoded to be SL3 and SL4. The architecture (L—locally connected) and number of neurons per leg (3, 4) remain the same, but the encoding has changed (from direct to symmetric) has changed. Figure 20(right) plots the results showing that adaptability increases for both topologies when symmetrically encoded. This increase is significant for the re-encoding of DL4 to SL4.

From this it can be concluded that the encoding of the neural network is responsible for the increased adaptability in the key comparisons in Figure 19. The increased adaptability of SF5 and SF6 relative to DL3 and DL4 is not conferred by a fully con-
nected (rather than locally connected) architecture or more neurons: this topology has lower adaptability if directly encoded (Figure 20(LEFT)). The decreased adaptability of DL3 and DL4 is not due to a locally connected (rather than fully connected) architecture or fewer neurons: this topology has higher adaptability if symmetrically encoded (Figure 20(RIGHT)).

![Graph showing change in leg length vs. number of parameters in genome for different neural network topologies.](image)

Figure 20: Changing the encoding is sufficient to change the adaptability. Arrows indicate a comparisons of neural networks with the same architecture and number of neural networks, but different encodings. LEFT: The SF5 and SF6 neural network topologies can have lower adaptability, specifically if directly encoded. (DF5 $n=540$, DF6 $n=495$, SF5 $n=485$, SF6 $n=570$) RIGHT: The DL3 and DL4 neural networks can have higher adaptability, specifically if symmetrically encoded. (DL3 $n=405$, DL4 $n=505$, SL3 $n=400$, SL4 $n=505$)

**Symmetric encoding has a higher adaptation rate**

There are two aspects of adaptability: degree and rate. The previous sections dealt with the degree of adaptability. Adaptation rate may be more important than degree when, for example, two populations compete for the same resources while their environments change. If one population is less adaptable in degree, but has a higher adaptation rate, it may outcompete a population with greater degree of adaptability but lower adaptation rate.
Overall, symmetrically encoded neural networks adapt more quickly than the directly encoded neural networks. The adaptation rate during the adaptability experiments can be displayed by graphing median leg length change per generation. The median leg length change for the first 300 generations of the adaptability experiments are shown in Figure 21(LEFT). SF6 already appears to be more adaptable than DL4 in the first 25 generations, but SF5 and DL3 appear to have approximately the same median adaptability until generation \(\sim 225\). Figure 21(RIGHT) shows that the difference in the median leg length change grows as the search continues until generation 5000.

Figure 21: Median change in leg length by generation during adaptability experiments performed using the indicated encoding/architecture type. LEFT: SF5 and DL3 adapt at approximately the same rate for the first 225 generations, but later (RIGHT) SF5 appears to adapt to a greater degree, though the difference at 5000 generations is not significant at \(p<0.05\) (Fig. 19). (DL3 \(n=405\), DL4 \(n=505\), SF5 \(n=485\), SF6 \(n=570\))

What would happen if the two encoding types were competing for population share during the adaptability experiments? SF6 is significantly more adaptable than DL4 at generation 5000 (as described on page 65). One might therefore expect SF6 to outcompete DL4. In contrast, SF5 is not significantly more adaptable than DL3 at generation 5000. In addition, SF5 and DL3 have approximately the same adaptability until generation 225 (Figure 21(LEFT)). Will either of these encodings outcompete
the other? These experiments are performed below.

**Larger symmetric encodings outcompete smaller direct encodings when leg length changes**

Would the difference in adaptation rates seen in Figure 21 be enough to allow the symmetric encoding with larger parameter space to outcompete the direct encoding with a smaller parameter space?

To test this, competitions between the symmetrically and directly encoded neural networks were set up as described on page 63. Once again, the comparisons and competitions of interest were between a directly encoded neural network and a symmetrically encoded neural network of slightly larger genotype. Specifically, all pairs of equilibrium populations of DL3 (126 parameters) and SF5 (160) competed against each other, as did all equilibrium population pairs of DL4 (192) and SF6 (228).

If, after a number of generations, individuals of one encoding type replaced every individual of the other encoding type, then that remaining encoding type “won” the competition. As a control, the competitions were also run without changing the leg length. In this control, it is expected that the initially most fit equilibrium search will outcompete the less fit regardless of encoding.

First, it was found that if the leg length was not changed the initially most fit equilibrium search won the competition. The type of encoding gave no advantage. This is shown in Figure 22(bar pair left pane) where the most initially fit population wins the vast majority of competitions.

The symmetric encoding has higher adaptability when the leg length is repeatedly shortened (in the same way as the adaptability experiments). Figure 22(right pane, 1st bar pair) shows that symmetric encodings outcompete direct encodings when adaptation to leg shortening is required to maintain high fitness. This includes competitions where the symmetrically encoded equilibrium search population is higher.
and lower fitness than the directly encoded equilibrium search population. A stronger test of the ability of the symmetric encoding to outcompete the direct encoding would be to compare the subset of competitions where the symmetrically encoded equilibrium population is of lower fitness than the directly encoded equilibrium population.

For this subset of competitions Figure 22(right pane, 2nd bar pair) shows the symmetrical encoding still wins a significant majority of the competitions. This is noteworthy given that, as reported above, the higher fitness equilibrium search wins the vast majority of competitions when leg length does not change (Figure 22(left pane)).

![Bar chart](image)

**Figure 22**: Symmetric encodings win competitions against direct encodings when leg length changes. **BAR PAIR LEFT PANE**: When the leg length is not changed, the initially higher fitness subpopulation wins most of the competitions (0.97 +0.02/-0.01, \( n=2320 \)). **RIGHT PANE**: When leg length is changed during a competition, the symmetrically encoded subpopulation wins the majority of competitions (1ST BAR PAIR) (0.78 +0.02/-0.02, \( n=2360 \)), even when comparing the subset of competitions in which the symmetrically encoded subpopulation is initially less fit than the directly encoded subpopulation. (2ND BAR PAIR) (0.65 +0.03/-0.04, \( n=750 \)).
Competitions may be a more sensitive indicator of adaptability than the measure of “adaptability” used above (the median total amount of leg length change). This is exemplified by comparisons of the adaptability of SF5 and DL3 above (Figure 19). In this comparison, SF5 is more adaptable than DL3, but not significantly more. Additionally, SF5 and DL3 have approximately the same adaptation rate for the first 225 generations. Yet, even in competitions where SF5 populations are less fit than DL3, SF5 wins a significant majority (0.64±0.08). This is the same fraction of wins as for SF6 vs. DL4, where the measured adaptability of SF6 is significantly greater at 5000 generations than DL4 (Figure 19). Interestingly, while the fraction of wins is the same for SF5 vs. DL3 and SF6 vs. DL4, there is a difference in the median duration of the competition. SF5 vs. DL3 competitions duration is more than one hundred generations longer than SF6 vs. DL4 competitions. This difference in competition duration may be due to the relative difference in adaptation rates between the different encodings. SF6 has a much higher adaptation rate than DL4, while SF5 and DL3 have about the same adaptation rate in the first 225 generations (see Fig. 21).

These results indicate that the difference in the adaptation rate observed in Figure 21 is enough to give the symmetric encoding the competitive advantage over the direct encoding. This advantage is detectable even when (in the case of SF5 vs. DL3) there is no significant difference between their adaptability at 5000 generations (based on the common definition of significance described on page 64). This advantage occurs even though the symmetric encoding has a larger parameter space than the the direct encoding.

Discussion

It was shown that for this particular model and two pairs of neural network types
that the number of genotype parameters is less important to evolvability than genotype reuse. Symmetrically encoded neural networks containing more genotype parameters were more discoverable, had higher adaptability, and outcompeted directly encoded neural networks.

To increase confidence in the results and these conclusions one would want increase the number of direct and indirect pairs compared. Using the formulas for genotype size on page 52 one could find the next largest neural networks pairs to test.

| DL5 (270) vs. SF7 (308) |
| DL6 (360) vs. SF8 (400) |
| ... vs. ... |

Looking at discoverability trends from the two pairs of comparison performed shown in Figure 19 on page 66, one could be led to hypothesize that directly encoded neural networks will continue to find 0 high fitness individuals in 1000 attempts. Additionally, it appears as though as the number of genotype parameters increase, discovering high fitness symmetrically encoded individuals becomes less likely. More than 1000 attempts may need to be made. Looking at adaptability trends in Figure 19 on page 66 one might predict that as the neural networks increase in size adaptability will continue to increase. Part of the observed increase in adaptability might come from additional interneurons. Experiments with a single-legged stepper [6] showed that higher fitness individuals were found when the neural network had an interneuron compared to no interneuron. There was more freedom to fine tune the gait pattern of the agent. However, a combination of increasing genotype size and decreasing benefit from adding interneurons is likely to cause adaptability to level out or even decrease with increasing neural network size.

Another limitation which should be explored is the effect of population size on the results. Only one population size (100 individuals) was used for all experiments. One might hypothesize that a larger population would increase evolvability because more
individuals allows more chances to find high fitness individuals (and visa versa).

There are many possibilities underlying the increased evolvability of a symmetrically encoded neural network. One possibility is that though the genotype space of the symmetrical encoding is larger, the density of high fitness solutions is also higher. This would affect discoverability by allowing the initial search to find a high fitness region more quickly. The symmetric encoding also coordinates changes of neural network parameters whereas the direct encoding does not. The six subnetworks are changed in the same way at the same time. This would have the effect of making regions of the direct encoding’s parameter space unreachable, but if these regions are lower fitness, not being able to spend time searching them would increase evolvability.

The increased evolvability of the symmetric encodings is not due to extra-dimensional bypass. Extra-dimensional bypass is the escape from local fitness maxima by the addition of search dimensions [22]. The extra search dimensions cause what was a local fitness maxima to become shallower to the extent that mutation can move the population out of the local fitness maxima and eventually to higher fitness regions of search space. In these experiments the direct encoding is a superset of the symmetric encoding. I.e. all individuals in the symmetric encoding are also directly encodable. If evolvability increased by extra-dimensional bypass in these experiments, the direct encoding should have higher evolvability because it has the same dimensions as the symmetric encoding plus more. This is the opposite of the results of these experiments.

Competition between encodings may be more sensitive to evolvability differences than the other measures of evolvability. Competitions showed encodings to have significantly higher evolvability even when the adaptability measured for each encoding separately was not significantly different. Thus competition may be a powerful way to select for forms of the encoding with higher evolvability.
4 Genotype reuse can increase evolvability on a problem with no geometric structural regularity: pseudoplant experiments

In the hexapod experiments above, indirect encodings are shown to have higher discoverability and adaptability than direct encodings on a problem where structural regularity of the solution is a known benefit. There are at least two symmetries in the hexapod problem. First, high fitness was achieved by the body moving as far as possible in the time allotted in a bilaterally symmetric environment. This requires bilaterally balanced forces for highest fitness. Second, the symmetric encoding created the neural network by connecting six identical subnetworks. The indirect encoding was well suited for this task because the subnetworks could function in identical ways and achieve high fitness. While the direct encoding could create the same phenotypes as the indirect encoding each parameter would need to be adjusted independently. The indirect encoding produced higher fitness phenotypes more frequently and therefore had higher evolvability.

Forms of structural regularity are found in the other comparisons of direct and indirect encodings. Covering a surface with no gaps and no overlaps using one repeated shape is an example of self-similarity [7]. Creating a level, raised, solid surface (a.k.a. “a table”) is symmetric because the normal forces must counter gravity to prevent tipping [41]. Finally a 3-D agent tasked with walking in a straight line would achieve high fitness if forces were applied symmetrically [54].

A question that remains to be answered is “Could an indirect encoding have higher evolvability than direct on a problem which does not have geometric structural regularity?” As discussed earlier (Section 2), Clune [20] gives an example of an system in which the indirect encoding has less evolvability as the geometric regularity of the problem decreases. However, those experiments did not examine the effect of a changing environment which has been shown to select for evolvability [58][72][26]. It may
be that environmental change would play a role in whether or not indirect encodings can also be useful with respect to problems without geometric regularity.

To answer this question, evolvability experiments below are performed using a problem which does not have a geometric structural regularity. Originally the problem was intended to not have any structural regularity. However during the course of the experiments the problem was found to have a non-geometric form of structural regularity.

The task consists of a “pseudoplant” which must adjust its surface to volume ratio (surf/vol ratio) to match the environment’s humidity level to achieve high fitness. The environment is changed by changing the humidity level. The pseudoplant’s body is modeled as any number of body segments. These body segments are not connected and have no structural relationship to each other. The plant’s surf/vol ratio is calculated as the average surf/vol ratio of its body segments. Fitness is determined by an interaction between the plant’s surf/vol ratio and the environment’s humidity level. Consequently, structural regularity would not seem to play a role in determining fitness.

The pseudoplant’s genotype encoding has a number of desirable properties. (The details are on page 83.) First, genotype reuse is explicitly implemented and adjustable in small increments. This allows easy control over the amount of genotype reuse for different experimental conditions. Second, the only difference between indirect and direct forms of the encoding is whether or not there is genotype reuse. Third, because genotype reuse is adjustable in small increments, a mutation operator can be used to convert between the different forms in an incremental way. Finally, the phenotypes that can be created with genotype reuse can also be created with no genotype reuse. Thus any difference in evolvability is therefore due to the amount of genotype reuse and not limitations in the possible phenotypes.

Experiments were performed to compare the evolvability of the direct and indirect
forms of the pseudoplants. Both discoverability and adaptability experiments were performed. The direct form has no genotype reuse (by definition) and is not allowed to gain it by mutation. The indirect form begins with no genotype reuse but may gain it through mutation. (In other words, the indirect form populations begin as the direct form of the encoding and have the potential to become the indirect form through mutation.) No explicit competitions were performed as in the hexapod experiments. However, within each population of the indirect form there is competition among individuals with varying amounts of genotype reuse. The amount of genotype reuse is under selective pressure.

It will be shown that for this model the indirect encoding is not more or less discoverable than the direct encoding if allowed to freely mutate. I.e. beginning from a naively initialized population, the indirect encoding does not find high fitness solutions in fewer generations than the direct encoding. However, the indirect encoding does allow more adaptability to environmental change. One mechanism by which adaptability is enabled by the indirect encoding is explored. This mechanism takes advantage of a non-geometrical form of structural regularity. A body model which attempts to eliminate this mechanism succeeds in reducing its ability to increase evolvability greatly, but not entirely. From this it will be argued that systems in which genotype reuse does not improve evolvability are likely to be rare.

**Methods**

**Fitness landscape**

A fitness landscape which depends on two variables was created. One variable is controlled by the individual’s genotype and is evolvable (subject to mutation and selection). The other variable is part of the environment and is not evolvable. Fitness is determined by an interaction between the two variables.

While one can think of this fitness landscape in the abstract, a story about plants
makes it easier to understand. When plants gather light they must cope with (at least) two conflicting demands: gathering light and staying hydrated. A higher surface area to body volume ratio (surf/vol ratio) can gather more light but also allows more water loss through evaporation. A large surf/vol ratio in a dry environment results in dehydration or death, a low fitness. A small surf/vol ratio in a wet humid environment might not result in low fitness by itself. The plants would not die of dehydration. Instead these plants would have a low fitness relative to other plants in the same locale with a larger surf/vol ratio who were able to gather more energy, have more offspring, and outcompete the plants with a smaller surf/vol ratio.

This story guides the creation of a fitness landscape which will be known as “The Original Model” (TOM). An interaction of two variables determine the fitness of the plants: surf/vol ratio and humidity. The story specifies that high surf/vol ratio plants are most fit in high humidity environments and low surf/vol ratio plants are most fit in low humidity areas. Intermediate values of surf/vol ratios are most fit at intermediate levels of humidity.

Figure 23 shows the fitness landscape for the following experiments graphically. The more fit a surf/vol ratio is at a given humidity, the lighter the area in the fitness landscape graph. For every humidity value there is one surf/vol ratio which has perfect fitness and a continuum of surf/vol ratios with lesser fitnesses. Thus, a high fitness individual will gradually become low fitness if humidity level is gradually changed but the individual’s offsprings’ genotypes are not.
Figure 23: Fitness landscape showing an interaction between humidity and surf/vol ratio. Combinations achieving a high fitness are lighter than low fitness combinations. A humidity level change will shift an individual horizontally on the fitness landscape. An individual with a high fitness surf/vol ratio at a given humidity level will not remain high fitness if humidity level changes enough.

More specifically, the fitness landscape was created by a function of two variables, ratio and humidity:

$$fitness(ratio, humidity) = e^{-\left(\frac{(ratio - ratioMin) + (humidity - humMin)(ratioMax - ratioMin)}{\text{numMax}\text{numMin}}\right)^2}$$

where ratio is an individual pseudoplant’s average surf/vol ratio, humidity is the environment’s humidity level, ratioMin and ratioMax are constants setting the minimum
and maximum values for the average surf/vol ratio, $\text{humMin}$ and $\text{humMax}$ are constants setting the minimum and maximum values for the humidity level, and $\text{gaussian}$ is a constant controlling how fast fitness decreases as the surf/vol ratio changes from the ideal for a given humidity level. In these experiments $\text{gaussian}=10$. A larger $\text{gaussian}$ causes the white high fitness peak to be broader. This makes it possible for more surf/vol ratios to achieve high fitness at any given humidity level. During adaptability experiments (described on page 91) this resulted in less selection pressure for a change in the body’s average surf/vol ratio to adapt to new humidity levels. Humidity levels were arbitrarily chosen to range from $\text{humMin} = 0$ to $\text{humMax} = 1$. The average surf/vol ratio ranges from $\text{ratioMin} = 6$ to $\text{ratioMax} = 204$, as determined by the body model described next.

Substituting these constants into the fitness equation simplifies it to $\text{fitness} = e^{-0.005 \times (\text{ratio} - 198 \times \text{humidity} - 6)^2}$. With the constraints above, fitness ranges from 0 (least fit) to 1 (most fit). A fitness of greater than 0.95 is called “high fitness”.

**Body models**

Individuals to be evolved in the fitness landscape described above were created.

Each individual is composed of body segments. Body segments have the attributes of surface area and volume from which a surface area to volume ratio is calculated. The individual’s fitness is determined by the average surf/vol ratio of all its body segments and the humidity level. The number of body segments each individual is composed of is evolvable and was not restricted (i.e., $[0, \infty)$). (The genotype encoding is described on page 83.) The type of body segments available for use in an individual’s body were defined in advance and were not evolvable.

Two of different body models defined by the body segments possible were used.

In “The Original Model” (TOM) four different body segments types were defined. The four body segments were given a single letter “name” and assigned surf/vol ratios.
In TOM the body segments were defined in a way motivated by the plant story above. One segment was broad and flat, another long and thin, another a large cube, and the last a small cube. This gives a variety of shapes with which to adapt to humidity levels. Finally, a fifth special body segment was defined which does not affect the individual’s average surf/vol ratio or fitness. This special body segment exists so that rules in the initial population can be created which do not change the fitness of the individuals and therefore do not bias selection for or against genotype reuse\(^3\).

The body segments’ names, surf/vol ratio, and other attributes for TOM are listed in Table 3.

<table>
<thead>
<tr>
<th>name</th>
<th>surf/vol ratio</th>
<th>shape</th>
<th>dimensions</th>
<th>surface area</th>
<th>vol</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>6</td>
<td><img src="a.png" alt="Image" /></td>
<td>1 x 1 x 1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>20.2</td>
<td><img src="b.png" alt="Image" /></td>
<td>2 x 0.15 x 0.33</td>
<td>2.02</td>
<td>0.1</td>
</tr>
<tr>
<td>c</td>
<td>60</td>
<td><img src="c.png" alt="Image" /></td>
<td>0.1 x 0.1 x 0.1</td>
<td>0.06</td>
<td>0.001</td>
</tr>
<tr>
<td>d</td>
<td>204</td>
<td><img src="d.png" alt="Image" /></td>
<td>1 x 1 x 0.01</td>
<td>2.04</td>
<td>0.01</td>
</tr>
<tr>
<td>Z</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 3: TOM body model assignment of body segment names to surf/vol ratios.

The body segments’ surf/vol ratios are not evenly distributed over the surf/vol ratio range. Figure 24 shows this uneven distribution graphically. A line is drawn across the fitness landscape at the surf/vol ratio corresponding to the indicated body

\(^3\)Unfortunately the side effect of this is to create more “neutral networks” which might effect evolvability. This limitation is discussed on page 124.
segment’s surf/vol ratio. The body segments ‘a’, ‘b’, and ‘c’ are all grouped in one half of the surf/vol range while ‘d’ is at the extreme surf/vol ratio in the other half.

Figure 24: The original body model (TOM) has surf/vol ratios assigned to four body segments in an unevenly distributed way.

A second body model with four body segments equally spaced throughout the surf/vol ratio range was created and named EQ4. This body model will be compared with TOM for reasons which will be explained in the Results (on page 96). For now, notice that it has the same number of body segments as TOM (four) but they are evenly spaced throughout the surf/vol ratio range.
Figure 25: The EQ4 body model has four body segments whose surf/vol ratios evenly divide the range of possible surf/vol ratios.

Different combinations of body segments achieve high fitness at different humidity levels. At some humidity levels an individual can achieve high fitness with a single body segment. Referring to Figure 24 or 25, this is at those humidity levels where the body segment’s line crosses the light colored area. At other humidity levels, a combination of body segments is needed to create an average surf/vol ratio which achieves high fitness. These humidity levels are those where no body segment line crosses the light colored area.

**L-system encoding**

An L-system [56][46] is used to encode the pseudoplant. L-systems of various types
have been used extensively with EAs to encode neural networks [52][10], plants [45] [65][94], architectural structures [21], and tables [40].

An L-system is a grammar $G = (V, \omega, P)$, where $V$ is an alphabet, $\omega$ is an axiom, and $P$ is a set of production rules.

The alphabet is a set of all elements which can be used in the axiom and production rules. In these experiments, the alphabet contains the names of the body segments: $V = \{a, b, c, d, z\}$.

The axiom is an initial string containing letters from the alphabet. An example axiom for these experiments: $\omega = zzz$.

Production rules are used to rewrite the axiom a specified number of iterations. A rule is written as A→B. This means any ‘A’ in the axiom is replaced by a ‘B’. ‘A’ is the ‘predecessor’ and ‘B’ is the ‘successor’. An iteration produces an intermediate production to which the rules are applied the next iteration. An example rule for these experiments: rules = {z+abb}.

When the rules are applied for the specified number of iterations the result is the final production. This string is interpreted as the individual’s phenotype.

These experiments use a restricted form of a non-parametric context-free deterministic L-system. Context-free means that a match of the predecessor does not depend on the surrounding letters. The L-system is non-parametric because if a predecessor matches, the successor is substituted unconditionally. Deterministic means no two rules can share the same predecessor. These experiments further restrict the number of rules to 0 or 1. Finally, the number of iterations will be 1 for all experiments. One rule and one iteration is the simplest way to implement genotype reuse.

The following L-system is of the restricted type found in these experiments.

\[
\text{axiom} = zzz
\]
\[
\text{alphabet} = \{a, b, c, d, z\}
\]
\[
\text{rules} = \{z+abb\} 
\]
The number of iterations for these experiments is 1. Rewriting the above L-system for one iteration:

<table>
<thead>
<tr>
<th>iteration</th>
<th>production</th>
<th>explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>zzz</td>
<td>this is the axiom</td>
</tr>
<tr>
<td>1</td>
<td>abbabbabb</td>
<td>applying rule {z-&gt;abb} to axiom</td>
</tr>
</tbody>
</table>

A shorter convention of writing L-systems is used in this paper. The alphabet is always \{a,b,c,d,z\}, so it is not written. The axiom and rule are written in compact form \{axiom,\{rules\}\}. The L-system directly above is written \{zzz,\{z->abb\}\}.

Below are a few more example L-systems and their productions. Whether or not the genotype has no genotype reuse, potential for genotype reuse, or genotype reuse is also stated. The first example shows a genotype which will never have genotype reuse because the mutation operators used in these experiments do not allow addition of rules. (The mutation operators are explained more fully on the current page). The second two genotypes have no genotype reuse currently, but could be mutated such that the predecessor matches a body segment in the axiom. The last two genotypes have genotype reuse.

<table>
<thead>
<tr>
<th>L-system</th>
<th>final production</th>
<th>type of genotype reuse</th>
<th>explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>{abdda,{}}</td>
<td>abdda</td>
<td>no genotype reuse</td>
<td>no rule, no matches</td>
</tr>
<tr>
<td>{abdda,{c+zzz}}</td>
<td>abdda</td>
<td>potential genotype reuse</td>
<td>non-matching rule</td>
</tr>
<tr>
<td>{abdda,{z+zz}}</td>
<td>abdda</td>
<td>potential genotype reuse</td>
<td>non-matching rule</td>
</tr>
<tr>
<td>{cbbc,{c+cccc}}</td>
<td>cccccbbcccc</td>
<td>genotype reuse</td>
<td>rule matches twice</td>
</tr>
<tr>
<td>{ddd,{d+abcd}}</td>
<td>abcdabcdabcd</td>
<td>genotype reuse</td>
<td>rule matches thrice</td>
</tr>
</tbody>
</table>

**Genetic algorithm**

Mutations occur at a mutation frequency abbreviated “mf”. \(mf\) is the expected frequency of change per letter per generation. The expected number of mutations per
“string” (either axiom or rule) is \( mf \times \text{length}(\text{string}) \). The actual number of mutations is determined by randomly choosing an number from a Poisson distribution (which consists of integers) whose mean is \( mf \times \text{length}(\text{string}) \) [94].

The genotype can be mutated in four ways. First, a randomly chosen letter can be *deleted* from the string. Second, a randomly chosen letter in the string can be *substituted* for by a randomly chosen letter from the alphabet. Third, a randomly chosen letter from the alphabet can be *inserted* into the string at a randomly chosen location. Fourth, a randomly chosen letter in the string can be *duplicated* (once) and placed next to the original letter. Examples of these four types are given in Table 5.

The relative probability with which each type of mutation occurs is listed in Table 4. The relative probability is the probability that if there is a mutation it will be that specific type. Notice that the probability of a letter being added to a string (duplicate plus insert) is equal to the probability of a letter being removed (delete). The probability of letter addition or removal are equal so that there is not a mutational bias towards shrinking or growing axiom and rule string lengths.

<table>
<thead>
<tr>
<th>mutation type</th>
<th>prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>substitute</td>
<td>0.5</td>
</tr>
<tr>
<td>duplicate</td>
<td>0.125</td>
</tr>
<tr>
<td>insert</td>
<td>0.125</td>
</tr>
<tr>
<td>delete</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 4: The different types of mutation and their relative probability. The total probability of the sets of operators which increase or decrease the length of the strings are equal.

Below an example L-system \{abc, {z+d}\} is mutated in the four ways. Change to the string are underlined. The axiom and rule are treated as plain strings for these mutation operations. The rule’s successor and predecessor are concatenated into one string.
Table 5: Example effects of mutation operators. Each row gives an example L-system in the “before” column which is mutated by the mutation type. The resulting L-system appears in the “after” column.

<table>
<thead>
<tr>
<th>before</th>
<th>mutation</th>
<th>after</th>
</tr>
</thead>
<tbody>
<tr>
<td>{abc, {d+dd}}</td>
<td>substitute</td>
<td>{aac, {d+dd}}</td>
</tr>
<tr>
<td>{abc, {z+d}}</td>
<td>duplicate</td>
<td>{abc, {z+zd}}</td>
</tr>
<tr>
<td>{abc, {a+d}}</td>
<td>insert</td>
<td>{abzc, {a+d}}</td>
</tr>
<tr>
<td>{abc, {c+b\bb}}</td>
<td>delete</td>
<td>{_bc, {c+b\bb}}</td>
</tr>
</tbody>
</table>

A special situation arises when a rule’s string contains a single letter. How to interpret this as a rule? It was decided to remove this type of rule from the individual’s genotype. Furthermore, a rule cannot be added to a genotype through mutation. These two decisions cause a bias towards rule removal (and a decrease in genotype reuse). This bias towards loss of rules is important for the validity of these experiments. If genotype reuse increases during GA search it is not due to a mutational bias adding rules to genotypes, but instead some selection pressure for rules.

Crossover was not allowed in these experiments.

**Direct and indirect encodings implemented with L-systems**

In the introduction to this Section (page 75), it was stated that a comparison between direct and indirect forms of an encoding was to be made. Now that the L-system encoding has been described, the direct and indirect forms can be described in terms of the L-system. The direct encoding is implemented with the use of an empty rule: \{\}. There is no rule rewriting possible and no genotype reuse. The indirect encoding is implemented by non-empty rules. These rules allow the potential for genotype reuse and the possibility that the successor will be used multiple times when creating the phenotype. This is a one-to-many mapping and is therefore an indirect encoding.
Measuring genotype reuse

The L-system allows genotype reuse to be easily visible. The L-system implements genotype reuse by rewriting the axiom using the rule. For example, the genotype \{cdc, {c+acc}\} becomes the phenotype “accdacc”. Each ‘c’ in the axiom is rewritten by the rule to be an ‘acc’ in the phenotype. The same phenotype can be created with no genotype reuse using the genotype \{accdacc, {}\}. Since there is no rule rewriting this genotype has no genotype reuse.

More quantifiably, one definition of genotype reuse is to find the average number of rule matches for the genotype. This can be calculated by counting the number of times the predecessor of the rule matches elements of the axiom, then divide by the length of the axiom. The averaging allows comparison of genotypes with axioms of different length. An important modification is to only count matches if the rule’s successor is greater than 1 in length. A rule whose successor has only one element may be the identity rule (such as \{d\rightarrow d\}). In this case there is no change to the sequence of letters and one would not want a non-zero amount of genotype reuse. How to count matches of rules whose successor is length one but is not the identity rule (such as \{d\rightarrow c\}) is less clear. If there is a match the number of body segments does not increase. However, the resulting phenotype has more of the successor body segments than otherwise. This is a type of reuse, so this type of match is counted.

The formula for the first type of genotype reuse is:

\[ \text{GR}_{\text{avg\_num\_match}} = \frac{\text{num\_non\_idem\_rule\_matches(geno\_type)}}{\text{length(axiom)}} \]

The example \{cdc, {c+acc}\} has \( \text{GR}_{\text{avg\_num\_match}} = \frac{\text{num\_non\_idem\_rule\_matches(geno\_type)}}{\text{length(axiom)}} = \frac{2}{3} \).

Another way to measure genotype reuse is to compare the size of the genotype to the size of the phenotype. Any increase in size of the phenotype when compared to the axiom is due to substitution(s) of the successor for the predecessor. The size of the phenotype can be measured as the length of the string listing the body segments.
in the phenotype. The size of the genotype can be measured as the number of body segments in the axiom plus the number in the successor.

The formula for the second type of genotype reuse is:

\[ GR_{\text{phen\_gen\_ratio}} = \frac{\text{length(phenotype)}}{\text{length(axiom)} + \text{length(successor)}} \]

The example \{cdc, \{c-acc\}\} has \( GR_{\text{phen\_gen\_ratio}} = \frac{\text{length(accdacc)}}{\text{length(cdc)} + \text{length(acc)}} = \frac{7}{6} \).

**Experiment types**

Experiments were performed to determine the effect of genotype reuse on (rate of) discoverability and adaptability. Discoverability experiments were performed first. The resulting populations are used as the seeds for the adaptability experiments. The flow of experiments is shown graphically in Figure 28. The experiments are described in detail in the next sections.

**Rate of discoverability experiments**

“Discoverability” in these experiments is defined as the number of generations which elapse between a naive initial population and a population with a median fitness ≥ 0.95 or 5000 generations, whichever comes first. Genotypes were initialized with a single randomly chosen letter from the alphabet. The rule was initialized in a number of ways depending on the comparison being made. E.g. with potential for genotype reuse or no genotype reuse allowed. The rule initialization will be described with the results. Measurement of the median discoverability of three searches is illustrated in Figure 26.
Figure 26: Rate of discoverability for pseudoplant experiments is the median generation at which search median fitness becomes greater than a target fitness. This illustration shows three imaginary searches whose median number of generations to reaching the target fitness is 3695. The discoverability of this grouping of searches is therefore 3695.

Discoverability may depend on the mutation frequency or the humidity level at which a search is run. Therefore for all comparisons, searches were run at a variety of humidity levels and mutation frequencies. Humidity levels ranged from [0,1] in 0.01 increments. Mutation frequencies were not regularly spaced because it was found that higher mutation frequencies perturbed genotypes so much that high fitness individuals were rare. To reduce the number of searches, the mutation frequencies chosen were more concentrated in the range of 0.000624 to 0.05 and had larger spacings from 0.05 to 1. (Mutation frequencies used: 0.000624 0.00125 0.0025 0.00375 0.005 0.006 0.007 0.008 0.009 0.01 0.014 0.018 0.022 0.026 0.03 0.034 0.038 0.042 0.046 0.05 0.06 0.07 0.08 0.09 0.1 0.125 0.15 0.175 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.)
Five attempts to discover a high fitness population were performed at each (humidity level and mf) combination. The 95% confidence interval around the median discoverability was calculated using bootstrapping described on page 93.

**Adaptability experiments**

Experiments were performed to explore the effect of genotype reuse on adaptability. In these experiments “adaptability” is defined as the total amount of humidity level change while maintaining a high fitness population over 5000 generations.

An adaptability experiment was performed as follows: A GA search is initialized with a high fitness population (median fitness $\geq 0.95$) from the discoverability experiments. The humidity level and mutation frequency are also transferred with the population. The humidity level is then changed until the population’s median fitness is below a “high fitness threshold” $\geq 0.95$. The initial direction of humidity level change is chosen randomly. Next, a GA search is performed until the population’s median fitness rises above the high fitness threshold. The GA search is then halted and the humidity level is changed until the median fitness again drops below the high fitness threshold. This is repeated for 5000 generations. If a search reaches one of the extremes in humidity level (0 or 1) the direction of humidity change reverses. In this way, the humidity level may repeatedly oscillate from one extreme to the other. The total amount of humidity level change that has occurred during the search is the search’s “adaptability”. An adaptability search is illustrated in Figure 27.
Figure 27: Adaptability of a single search is measured as the total amount of humidity level change in 5000 generations. This figure shows an imaginary population’s median fitness at each generation of the GA search. Humidity level change occurs any time the fitness becomes greater than a threshold. Vertically dashed lines indicated when humidity level changes. Arrows indicate two of the eight humidity level changes. This results in a loss of fitness. Between each humidity level change GA search occurs and an eventual gain in fitness. One such GA search is indicated with a bracket.

Searches with only transient amounts of adaptability were excluded when calculating the median adaptability of groups of searches. Filtering was performed by grouping searches in the two experimental conditions of interest by the mutation frequency at which they were run. Then the total amount of humidity level change of each search was checked. If at least one search from both experimental conditions had a total humidity level change of 2 or greater, then all searches at that humidity level were kept for comparison. Otherwise all searches at that mutation frequency were excluded. An adaptability of 2 or greater means that the population was able to adapt to every humidity level at least once. (E.g. humidity level 0 to 1 then back to 0 would be an adaptability of 2.) A population with an adaptability lower than 2 failed to adapt to every humidity level once within 5000 generations. Filtering focuses
attention on those mutation frequencies where at least one search was able to adapt to all humidity levels (once).

Figure 28: Flow of populations including filtering for determining median adaptability for a type of genotype. The two types of genotypes are potential for genotype reuse or no potential for genotype reuse. First, discoverability experiments are performed. Those populations which achieve high fitness go on to be used in adaptability experiments. Next, adaptability experiments are performed. Adaptability is then calculated as the median of total humidity level change after excluding those populations which have a total environmental change of less than 2.

Statistical significance

The significance of differences between means or medians reported in the results section are determined by examining the confidence intervals created by bootstrapping [29][97]. The confidence interval for the mean or median difference is found. The difference in the measurements and confidence intervals is $(measure_1 - measure_2) \pm \{top_1 - bottom_2, bottom_1 - top_2\}$, where $measure$ is the median or mean, the subscript indicates either the first or second measure, $top$ is the most positive value of the confidence
interval, and bottom is the most negative value of the confidence interval for the first or second measure as indicated by the subscript.

If the confidence interval of the mean or median difference includes zero, then the two means or medians are not significantly different. If the confidence interval does not include zero, the two means or medians are significantly different. In these experiments the confidence interval contains 95% of the statistic of interest’s distribution, corresponding to a \( p < 0.05 \).

Figure 29 shows three differences of confidence intervals. The first column shows a positive significant difference with a red highlight. Second column a negative significant difference with a blue highlight. The third column shows a non-significant difference and no highlight. This highlighting convention will be used in the following results graphs to make significant differences easier to see.
Figure 29: Illustration of the confidence interval of the difference. In the top row are three pairs of measurements with confidence intervals. In the bottom row are the confidence intervals of the difference for those pairs. The two axes are in the same scale. Measurements whose confidence intervals do not overlap are significantly different and the confidence interval of the difference does not include zero.


Results

Indirect and direct encodings have similar discoverability using TOM

Discoverability of populations with no genotype reuse and with the the potential for genotype reuse were compared. Searches with no genotype reuse (NoGR) were initialized with genotypes of the form \{random body segment,\{\}. The genotype has no rule, as indicated by the empty braces, \{\}. A rule cannot be gained by mutation. Searches with potential for genotype reuse (PoGR) were initialized with genotypes of the form \{random body segment,\{z\rightarrow z\}\}. These genotypes have a rule: \{z\rightarrow z\}. Initially it is an identity rule and does not affect the final production. Through mutation the rule can be changed to rewrite the axiom or be lost entirely. The following table summarizes the way the populations were initialized in these comparisons:

<table>
<thead>
<tr>
<th>potential genotype reuse (PoGR)</th>
<th>no genotype reuse (NoGR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>{random body segment,{z\rightarrow z}}</td>
<td>{random body segment,{}</td>
</tr>
</tbody>
</table>

The Figure labels follow a convention which quickly summarizes the experimental comparison. The convention is <type of experiment><forms of encoding><body model>. The type of experiment can be discoverability or adaptability. The form of encoding is labeled with the rule for each encoding, because this is what is different about the form of encoding. For example, instead of writing “\{random body segment,\{z\rightarrow z\}\} vs. \{random body segment,\{\}” the label would be \{z\rightarrow z\} vs. \{\}. The rule in the label is the rule with which the individuals are initialized with at the beginning of the experiment. The order of form of encoding indicates that \{z\rightarrow z\} measurements are subtracted from \{\} measurements. Finally, body model can be TOM or EQ4.

There was no significant difference in the median discoverability of the PoGR and NoGR searches. This was verified for a number of different groupings of the searches. First, comparing all PoGR versus all NoGR (and ignoring mutation frequency and humidity level at which the search was performed) yields no significant difference
(Figure 30(left pane)). There was also no significant difference in the median if the searches were grouped by mutation frequency at which they were performed (Figure 30(right pane)). Finally, there is reason to suspect that genotype reuse may provide more of an advantage in some humidity levels than others. (These reasons are purposefully withheld here, but are explained on page 106.) However, grouping searches by humidity level also yields no significant discoverability difference between NoGR and PoGR (Figure 31).

Figure 30: Genotypes with potential genotype reuse (PoGR) were not more discoverable than genotypes with genotype reuse not allowed at any mutation frequency (NoGR). (n=18685 for both \{z\rightarrow z\} and {} in the “all mf” (all mutation frequencies grouped together) difference)
Figure 31: Grouping searches by humidity level yields no difference when comparing discoverability of PoGR and NoGR. \( n=18685 \) for both \{z→z\} and {} in the “all hum” (all humidities grouped together) difference.

**Indirect encoding has higher adaptability than direct using TOM**

Genotype reuse does lead to an increase of adaptability in the TOM body model. The adaptability of NoGR (genotype reuse not allowed) populations are compared to that of PoGR (potential genotype reuse) populations:

\[
\text{PoGR vs. NoGR}
\]

Both NoGR and PoGR populations are seeded from the same NoGR discoverability experiments. PoGR genotypes are created by modifying the NoGR genotypes with the addition of an identity rule \{z→z\} (which does not change the phenotype). This
rule may mutate throughout the search and begin creating genotype reuse. PoGR
genotypes have no rule and are not allowed to gain one.

The adaptability is measured by grouping the searches in two ways. First, the
difference in the median total humidity change between PoGR and NoGR searches
with the same mutation frequency is found. Second, all mutation frequencies are
combined and the difference in median total humidity change between PoGR and
NoGR is found.

The results of these comparisons are shown in Figure 32. It shows that potential
genotype reuse searches to adapt to more humidity change than searches where geno-
type reuse is not allowed. In the Figure, the $x$-axis indicates the mutation frequency
at which the searches were run. The $y$-axis gives the difference in the median total
humidity level change. Significant differences are highlighted with red if positive and
blue-green if negative.

Overall, genotype reuse allows more adaptability. The farthest left dot in Figure 32
is the difference in medians for searches of all mutation frequencies. This difference
is significant, with PoGR adapting to more humidity level change than NoGR. The
other dots plot the differences in medians with searches grouped by the mutation
frequency at which they were performed. At every mutation frequency PoGR is
either significantly more adaptable or no different than NoGR.
Figure 32: In TOM, the adaptability of populations in which any rule is allowed in the genotype is significantly higher than when no rule is allowed. This shows that genotype reuse and the indirect encoding allow more adaptability. ($n=4547$ for both $\{z\rightarrow z\}$ and $\{}$ in the “all mf” difference)

Genotype reuse allows adaptability by amplifying use of a body segment in TOM

Genotype reuse enables more adaptability by a mechanism which could be called “d-amplification”, or more generically, repetition. If the rule contains multiple ‘d’ s in the successor, one mutation will insert or remove multiple ‘d’ s from the phenotype. E.g. If the genotype is $\{aaad, \{d+dd\}\}$ the phenotype is “aaadd”. If a ‘d’ is inserted into the axiom the genotype could be $\{aaadd, \{d+dd\}\}$ and the phenotype will be
“aaadddd”, an addition of two ‘d’. (Note that order of the letters does not change fitness, so inserting a ‘d’ into the axiom anywhere would have the same effect.)

This mechanism was verified in four ways.

First, the existence of ‘d’s in the rule are necessary for PoGR experiments to be more adaptable than NoGR experiments. This can be demonstrated by running experiments very similar to the above in which PoGR and NoGR were compared, but disallowing the use of ‘d’ in the PoGR rules:

\[
\text{PoGR, no ‘d’ in rule} \quad \text{vs.} \quad \text{NoGR}
\]

Disallowing ‘d’ in the rule does not prohibit ‘d’ in the axiom, so the possible phenotypes does not change.

Figure 33 shows that disallowing ‘d’ results in an adaptability which is not significant different than NoGR where no rule and no genotype reuse is allowed.
Figure 33: ‘d’ in rule are necessary to increase adaptability of PoGR. Not allowing ‘d’ in PoGR (\texttt{rule = \{z→z, no 'd'\}}) decreases its adaptability such that it is not significantly different from NoGR (\texttt{rule = \{\}}). (\(n=3151\) for both \{z→z, no ‘d’\} and \{\} in the “all mf” difference)

Second, disallowing body segments other than ‘d’ from being in the rule results in adaptabilities which are not significantly different than PoGR. More explicitly, PoGR \{z→z, no ‘a’\}, PoGR \{z→z, no ‘b’\}, and PoGR \{z→z, no ‘c’\} all have adaptabilities which are not significantly different than PoGR. (The data from these experiments are not plotted.)

\begin{align*}
\text{PoGR, no ‘a’ in rule} & \quad \text{vs.} \quad \text{PoGR} \\
\text{PoGR, no ‘b’ in rule} & \quad \text{vs.} \quad \text{PoGR}
\end{align*}
PoGR, no ‘c’ in rule vs. PoGR

This points to the letter ‘d’ being the only letter which must be in the rule for genotype reuse to allow more adaptability.

Third, populations with a fixed rule of \{(z\cdot dd)\} have increased discoverability and adaptability when compared to PoGR populations (where the rule is allowed to mutate freely).

fixed rule \{(z\cdot dd)\} vs. PoGR

The parenthesis in the rule indicate that it is not allowed to mutate. When \{(z\cdot dd)\} is a fixed part of the genotype adding or removing one ‘z’ to the axiom causes two ‘d’ to be added or removed from the phenotype. This increase in discoverability is in contrast to the initial discoverability experiment in which the rule was allowed to mutated freely (PoGR) was no different than when no rule was allowed (NoGR) (result reported on page 96). Thus it is possible to fix the rule and use the “d-amplification” mechanism to increase discoverability over the discoverability of a search which the rule is determined by mutation and selection. Figure 34 plots discoverability results and Figure 35 plots adaptability results of the genotypes with the fixed rule \{(z\cdot dd)\} compared to PoGR searches. The fixed rule \{(z\cdot dd)\} searches have higher discoverability (fewer generations until a high fitness individual is found) and adaptability (more environmental change).
Figure 34: Populations with a fixed rule of \{(z\rightarrow dd)\} have higher discoverability than PoGR populations with an evolvable rule in TOM. Since faster discovery results in lower discoverability numbers, a negative difference indicates a higher discoverability. ($n=18685$ for both \{(z\rightarrow dd)\} and \{z\rightarrow z\} in the “all mf” difference)
Figure 35: Populations with a fixed rule of \{(z\rightarrow dd)\} have higher adaptability than PoGR populations with an evolvable rule in TOM. \(n=4547\) for both \{(z\rightarrow dd)\} and \{z\rightarrow z\} in the “all mf” difference.

That a fixed rule can be designed which allows higher evolvability than an evolved rule in PoGR populations shows that evolved rules do not have optimal evolvability. This is not meant to imply that a fixed rule \{(z\rightarrow dd)\} is the optimum. Instead it shows that the rules PoGR populations do possess increase evolvability less than \{(z\rightarrow dd)\}. Evolved rules are constantly being acted upon by mutation and selection. Even if a population were initialized to some optimum it would quickly mutate to less than optimal forms.

Fourth and finally, fixing the rule to rewrite ‘z’s found in the axiom to any single body segment other than ‘d’ causes adaptability to be not significantly different than
PoGR. Specifically, populations with fixed rule \((z\rightarrow aa)\), fixed rule \((z\rightarrow bb)\), and fixed rule \((z\rightarrow cc)\) all have adaptabilities which are not significantly different than PoGR. The summary of comparisons are:

- fixed rule \((z\rightarrow aa)\) vs. PoGR
- fixed rule \((z\rightarrow bb)\) vs. PoGR
- fixed rule \((z\rightarrow cc)\) vs. PoGR

This again shows that it is the letter ‘d’ which must be in the rule for genotype reuse to allow more adaptability. (The data from these experiments are not plotted.)

**Visualizing the reason why ‘d’ amplification changes evolvability**

The distribution of body segments in the TOM body model requires a large number of ‘d’ body segments to achieve high fitness at high humidity levels. This can be demonstrated by finding the minimum length string of body segments which achieves a high fitness at a variety of humidity levels. The number of ‘d’ needed can then be plotted by humidity level. Note that a high fitness population created by GA search will probably contain longer than minimum length strings. Since fitness does not depend on length, longer than minimum length strings will also be high fitness and will exist in populations produced by the GA. But to change the average surf/vol ratio, even more ‘d’ will be needed to counteract the effects of other body segments.

Figure 36 shows the number of ‘d’ body segments in the minimum length string which achieves high fitness as black dots plotted by humidity level. The gray dots indicate the total number of all other body segments in a minimum length string. The vertical gray bars mark where a single body segment will achieve perfect fitness.
Figure 36: Number of ‘d’ body segments in the minimum length high fitness phenotype (black dots) and total number of all other body segments (gray dots) at a variety of humidity levels for the TOM body model. Humidity levels at which single body segments achieve high fitness marked with vertical gray lines.

High fitness can only be achieved at higher humidity levels by using some ‘d’. This is because three out of four of the body segments have surf/vol ratios which achieve perfect fitness at humidity levels less than 0.3, while ‘d’ must be used at humidity levels greater than 0.3. As a consequence, any average surf/vol ratio above 0.3 must be created using at least one ‘d’ body segment.

Additionally, the number of ‘d’ needed at higher humidity levels is more than twice that of other body segments at any humidity level. The number of ‘d’ in the minimum length string at high humidity levels (>0.3) is 10. The longest minimum length string
at low humidity levels (<0.3) is 4.

The plant analogy motivates a body model with asymmetries which lead to genotype reuse increasing evolvability. The plant analogy dictates that body segment surf/vol ratios be assigned in a manner in which three out of four body segments are clustered in one half of the range of surf/vol ratios, while the other body segment (‘d’) is at the other extreme end of the range. This clustering appears to cause genotype reuse to be useful in allowing the number of the outlying body segment (‘d’) to be changed quickly. This facilitates quickly adjusting the average surf/vol ratio and increases evolvability.

Evolvability experiments are performed using a body model with four body segments whose surf/vol ratios assigned evenly throughout the range (and so called EQ4) in the next section. Comparison of EQ4 with TOM gives an idea of how much the asymmetry of TOM allowed genotype reuse to facilitate an increase in evolvability.

Reducing the benefit of repetition – the EQ4 body model

The ability to add or remove multiple ‘d’ body segments from the phenotype in one mutation likely increases evolvability due to the structure of the TOM body model. Referring to Figure 24 on page 82 one can see that most of the vertical red lines indicating that a single body segment can achieve perfect fitness are at humidity levels less than 0.3, while ‘d’ is the only body segment which achieves perfect fitness at humidity level 1. The need for multiple ‘d’ at one end of the humidity range in order to achieve a high fitness average surf/vol ratio probably leads to the mechanism of “d-amplification”.

A body model in which the body segments are equally spaced was used to test this hypothesis. The body model EQ4 has four body segments, the same as TOM, but they are spaced equally over the humidity level range. See the Methods Section on page 82 for details.
Two questions were asked about EQ4. Does a genotype with mutable rules have higher evolvability than a genotype with no rule? This compares PoGR and NoGR in the EQ4 body model. Second, would a genotype with a fixed rule (such as \{(z\rightarrow dd)\}) have a higher evolvability than a genotype with no rule as was found for TOM?

**Indirect and direct encodings have similar discoverability and adaptability in EQ4**

Both the discoverability and adaptability of populations with evolvable genotype reuse (indirect encoding form) were not significantly different than populations with no genotype reuse (direct encoding form). Discoverability and adaptability experiments were performed on the PoGR and NoGR populations as in TOM:

PoGR (EQ4) vs. NoGR (EQ4)

If genotype reuse in any form gives a fitness advantage in EQ4 it is not enough to be selectable in a detectable way in these experiments. There is no significant difference in the median discoverability or adaptability at a variety of mutation frequencies and if grouping all mutation frequency as shown in Figure 37. This is different than TOM, where the evolvable genotype reuse had no difference for discoverability but did have higher adaptability than populations with no genotype reuse.
Figure 37: In EQ4 there is no difference in evolvability between populations which allow genotype reuse and those that do not. Discoverability difference overall (TOP LEFT PANE) and at various mutation frequency (TOP RIGHT PANE). ($n=18685$ for both \{z→z\} and \{} in the “all mf” differences) Adaptability difference overall (BOTTOM LEFT PANE) and at various mutation frequency (BOTTOM RIGHT PANE). ($n=4510$ for both \{z→z\} and \} in the “all mf” differences)
**Fixed genotype reuse can have higher discoverability in EQ4**

A fixed rule amplifying a single letter (\{(z\rightarrow aa)\}, \{(z\rightarrow bb)\}, \{(z\rightarrow cc)\}, \{(z\rightarrow dd)\}) does not increase evolvability relative to no rule. (Data not shown.) Thus “d-amplification”, the ability of genotype reuse to amplify ‘d’ to increase evolvability been successfully eliminated from EQ4. In addition, just as in TOM, amplifying other single body segments by genotype reuse does not increase evolvability.

In summary, there is no significant difference in evolvability between the following genotypes:

- fixed rule \{(z\rightarrow aa)\} (EQ4) vs. NoGR (EQ4)
- fixed rule \{(z\rightarrow bb)\} (EQ4) vs. NoGR (EQ4)
- fixed rule \{(z\rightarrow cc)\} (EQ4) vs. NoGR (EQ4)
- fixed rule \{(z\rightarrow dd)\} (EQ4) vs. NoGR (EQ4)

However, a fixed rule which amplifies two different body segments does increase discoverability, but not adaptability. Fixing the rule to be \{(z\rightarrow ab)\} allows higher discoverability overall (Figure 38(TOP)) and adaptability (Figure 38(BOTTOM)) at one mutation frequency. Given the equal distribution of body segments in EQ4, other rules which amplify nearby pairs of body segments such as \{(z\rightarrow bc)\} and \{(z\rightarrow cd)\} may also increase adaptability a small amount at certain mutation frequencies. (This was not tested.)

- fixed rule \{(z\rightarrow ab)\} (EQ4) vs. NoGR (EQ4)
Figure 38: Evolvability of a fixed rule \(\{z\rightarrow ab\}\) compared to NoGR in the EQ4 body model. Populations with the fixed rule \(\{z\rightarrow ab\}\) are more discoverable than populations with no rule (TOP) \((n=18685\) for both \(\{z\rightarrow ab\}\) and {} in the “all mf” differences). They are less adaptable overall, but there is one mutation frequency which is more adaptable (BOTTOM) \((n=4510\) for both \(\{z\rightarrow ab\}\) and {} in the “all mf” differences).

\[
\text{diff median gen of high fitness pop}
\]

\[
\text{diff median total env change}
\]
The previous result indicates a possibility that genotype reuse increases some types of evolvability in a majority of systems. The EQ4 body model was designed to eliminate the possibility that amplifying body segments through genotype reuse would increase evolvability. However, even in this revised model a fixed rule can increase discoverability.

Most systems which occur naturally will probably not have (the analogy to) equally distributed body segments in an environment which does not favor some kind of symmetry. Instead it seems more probable that an evolved natural system will have some forms of nonuniformity either in the genotype to phenotype mapping, the environment in which the organism lives, or both. This will lead to genotype reuse allowing increased evolvability through either structural regularity or repetition.

**Genotype reuse selected for by a changing environment in TOM**

Next comparisons of the measurements of genotype reuse, $GR_{avg\_num\_match}$ and $GR_{phen\_gen\_ratio}$ were performed. Genotype reuse is hypothesized to be selected for by a changing environment and this leads to increased evolvability. Until now the results only reported the differences in evolvability between genotypes with differing types of genotype reuse. Those results showed that in TOM genotypes with genotype reuse *happened* to have increased evolvability and in EQ4 did not *happen* to change evolvability. The following result report on whether genotype reuse is increased by a changing environment.

Experiments were performed to determine whether genotype reuse was selected for by changing environment in the adaptability experiments. Since the “d-amplification” mechanism depends on genotype reuse, one would expect genotype reuse to be selected for in searches if this mechanism accounts for the increase in evolvability.
GA searches were performed on PoGR populations in both changing and unchanging environments:

\[
\text{PoGR, changing environment} \quad \text{vs.} \quad \text{PoGR, unchanging environment}
\]

Genotype reuse was measured in the two ways described on page 88 in the final populations.

It was found that \( GR_{\text{avg_num_match}} \) (Figure 39) and \( GR_{\text{phen_gen_ratio}} \) (Figure 40) were both significantly higher in populations which evolved in a changing environment. This is consistent with the “d-amplification” mechanism allowing adaptation to environmental change. A larger \( GR_{\text{avg_num_match}} \) in the changing environment indicates that there are more matches and therefore more predecessors being replaced with successors in a changing environment compared to unchanging environment. A larger \( GR_{\text{phen_gen_ratio}} \) in the changing environment indicates that the phenotype/genotype ratio is larger in the changing environment compared to unchanging environment.
Figure 39: Environmental change selects for $\text{GR}_{\text{avg_num_match}}$ in the TOM body model. ($n=9094$)
Figure 40: Environmental change selects for $\text{GR}_{\text{phen_gen_ratio}}$ in the TOM body model. ($n=9094$)

**Genotype reuse selected for by a changing environment in EQ4**

The same experiments to determine whether genotype reuse is selected for by a changing environment were also performed for the EQ4 body model.

<table>
<thead>
<tr>
<th>PoGR, changing environment</th>
<th>PoGR, unchanging environment</th>
</tr>
</thead>
</table>

Remember that in EQ4 populations with no genotype reuse (NoGR) had the same adaptability as populations with a potential for genotype reuse (PoGR) (Figure 37 on page 110). One might expect from this that genotype reuse would not be selected for by changing environments.
In EQ4 \( GR_{avg\_num\_match} \) was not significantly different overall between populations evolved in changing and unchanging environments (Figure 41). This is as expected: If genotype reuse does not provide (enough of) a fitness advantage in a changing environment then genotype reuse should not be selected for.

Interestingly there is one mutation frequency at which \( GR_{avg\_num\_match} \) is higher in a changing environment in EQ4. This mutation frequency is also the one at which populations with the fixed rule \( \{z\rightarrow ab\} \) has higher adaptability than populations with no rule (Figure 38 on page 112 (BOTTOM)). This might be a sampling error as a result of the in the particular experiments performed. But because this feature is consistent in other experiments (related on the next page) it may point to interesting an interesting balance between environmental change, mutation frequency, and selection which may be interesting to disentangle.

![EQ4 GR](image)

Figure 41: Environmental change does not select for genotype reuse \( GR_{avg\_num\_match} \).

\( (n=9020) \)

The results for \( GR_{phen\_gen\_ratio} \) in EQ4 are not so easily interpreted. The overall
trend is that $GR_{\text{phen_gen_ratio}}$ is higher in a changing environment than an unchanging environment. This is true for comparison of adaptability experiments run at four individual mutation frequencies and all mutation frequencies grouped together. A higher $GR_{\text{phen_gen_ratio}}$ in a changing environment indicates that a changing environment selects for longer successors than an unchanging environment. However, because $GR_{\text{avg_num_match}}$ is not significantly different (as described above) there is apparently no enrichment of the number of matching body segments in the axiom. This might mean that the rule successors are lengthening mostly due to genetic drift encouraged a bit by the changing environment.

![EQ4 GR phen_gen_ratio](image)

Figure 42: Environmental change does select for genotype reuse $GR_{\text{phen_gen_ratio}}$. ($n=9020$)

Increasing the number of possible body segments by expanding the L-system alphabet in the model decreases the number of mutation frequencies at which genotype reuse is greater in the changing environment for both $GR_{\text{phen_gen_ratio}}$ and $GR_{\text{avg_num_match}}$. Adding a body segment halfway between each existing body segment in EQ4 creates
EQ7. Adding a body segment halfway between each existing body segment in EQ7 creates EQ13.

In Figure 43(left), one can see that the number of mutation frequencies at which $GR_{\text{phen\_gen\_ratio}}$ is significantly greater in a changing environment decreases by one going from EQ4 to EQ7. $GR_{\text{phen\_gen\_ratio}}$ is still significantly higher in the changing environment at three individual mutation frequencies and all mutation frequencies grouped together. When one moves from EQ7 to EQ13 one finds that $GR_{\text{phen\_gen\_ratio}}$ is not significantly different when all mutation frequencies are grouped together (Figure 43(right)).
Figure 43: Environmental change selects for less genotype reuse $GR_{\text{phen} \_ \text{gen}_\text{ratio}}$ as more body segments are added to the body model. $GR_{\text{phen} \_ \text{gen}_\text{ratio}}$ decreases monotonically when adding body segments to EQ4 to create EQ7 and then EQ13. (EQ7 $n=11146$, EQ13 $n=13910$)

The difference between $GR_{\text{avg} \_ \text{num}_\text{match}}$ in a changing and an unchanging environment shows a similar pattern, also decreasing with increasing numbers of body
segments. In EQ4 the difference between changing and unchanging environments for all mutation frequencies grouped together is not significantly different, but there are a number of mutation frequencies at which the amount of genotype reuse is actually less in a changing environment (Figure 44(LEFT)). This trend continues in EQ13 where the difference in $GR_{avg\_num\_match}$ is negative for all mutation frequencies grouped together (Figure 44(RIGHT)).
Figure 44: Environmental change selects for more negative difference in GR_{avg\_num\_match} as the number of body segments increases. The difference in GR_{avg\_num\_match} decreases monotonically becoming negative as body segments are added to EQ4 to create EQ7 and then EQ13. (EQ7 n=11146) (EQ13 n=13910)

These trends may indicate that genotype reuse actually makes adaptation to environmental change more difficult as more body segments are added to eventually
create EQ13. The genotypes in the unchanging environment may gain genotype reuse through genetic drift, while genotype reuse of populations in a changing environment is selected against. There is no difference in the adaptability of NoGR and PoGR populations in EQ13 (data not shown), possibly indicating that the PoGR populations are able to shed genotype reuse introduced by genetic drift fast enough for it not to hinder adaptation. The trends here are mostly not explainable with current knowledge and indicate directions for future work.

Discussion

The above experiments found that genotype reuse increases adaptability of populations on a problem where fitness does not depend on geometric structural regularity. Instead, genotype reuse was found to enable repetition, a non-geometric form of structural regularity. Repetition may be another case in which indirect encodings are well suited to increase evolvability.

One difference between these experiments and the hexapod experiments [82][83] is that genotype reuse only increased adaptability, not both discoverability and adaptability. One explanation is that in the hexapod experiments a large degree of genotype reuse was manually specified. The selective advantage conferred by genotype reuse was correspondingly large. In the current experiments, the individuals begin with a rule that does not lead to genotype reuse (the identity rule). Because of this genotype reuse was not present until specific mutations occurred. The initial degree of genotype reuse is zero and then after specific mutations genotype reuse is small (relative to hexapod experiments). This small degree of genotype reuse may not provide enough fitness advantage upon which selection can act in the number of generations needed to discover a high fitness pseudoplant. Alternatively, whatever rules were generated during the search did not confer enough fitness advantage to be selected
for at a rate high enough to overcome destruction of those rules by mutation. Man-
ually specifying genotype reuse by fixing the rule to be \((z \rightarrow \text{dd})\) to take advantage of “d-amplification” (Figure 34) did increase discoverability of the indirect encoding much like the manually specified genotype reuse in the hexapod experiments.

Performing a genetic algorithm analysis similar to [77] might provide a way to quan-
tify the conditions under which genotype reuse is selected for. Whether environment is changing or not was shown to be important by these experiments. A more in depth analysis could reveal the rate of environmental change which is important as well as how other potential factors such as mutation rate, selection pressure, and population size influence the results.

Deciding to use a fitness neutral ‘z’ body segment when creating the model may have had unintended consequences. This body segment was created to provide a mechanism for creating rules in the initial population which would not bias that population towards genotype reuse. However, it also creates a larger “neutral network” [4]. The nodes of a neutral network are genotypes of the same fitness. The nodes are connected by an edge when a genetic operator such as mutation can transform one genotype into the other. Neutral networks can increase the interconnectivity of search space and change evolvability [28]. In these experiments, one way this could increase adaptability, is by being “pre-adapted” to past environments [66]. Pre-adaptation could be implemented with a rule whose predecessor is a ‘z’ and whose successor specifies body segments which achieve a high fitness in some environment. As the environment changes towards the one for which the rule is pre-adapted, insertion of a ‘z’ into the axiom would eventually cause an individual to be higher fitness than the others in the population. It and its children would gain population share and increase the population median fitness rapidly leading to faster environmental change and an increase in the adaptability measure. One drawback of this strategy is that when far from the pre-adapted environment insertion of ‘z’ into the axiom would cause the
individual to be low fitness. If the fitness is low enough this could cause the individual to not reproduce and eventually no more individuals which are pre-adapted to the past environment would exist in the population.

However, initial indications are that using a ‘z’ body segment may not change the overall results. First, preliminary adaptability experiments in which the ‘z’ body segment was not used shows adaptability to be not significantly different. The full range of experiments and comparisons have not been performed. Additionally, examining the body segments in experiments in which the successor was not fixed (such as \{(z→da)\}) shows that the body segments roughly track the current environment. This is contrary to what is expected if increased adaptability occurs by the mechanism of the successor being pre-adapted to certain past environments.

Finally, some exploration of the effect on population size and fitness thresholds is warranted to determine the robustness of results to these perturbations.

I discuss how the pseudoplant experiments might be applicable to the evolution of natural G-P maps in the Final Discussion on page 132.
5 Conclusion

Summary

This thesis examines genotype reuse and its effect on evolvability.

Genotype reuse is the one-to-many property found in both natural genotype to phenotype mappings (G-P maps) and in indirect encodings used to describe artifacts in a form computers can manipulate. Genotype reuse exists if one element of the genotype is used to create many elements in the phenotype.

In contrast, direct encodings have a one-to-one mapping between elements of the genotype and elements of the phenotype.

Evolvability is a term used to describe the ease with which changes to a genotype can result in production of high fitness individuals. Often evolvability is a relative term. Some encodings of solutions to the same problem have higher evolvability than others. In genetic algorithms, two major types of evolvability exist. One form, “discoverability,” is related to how difficult it is to go from a naively initialized population to one with high fitness individuals. A second form, “adaptability,” is related to how fast a population recovers fitness after an environmental change.

Genotype reuse has several effects which may enable increased evolvability. Genotype reuse, because of the one-to-many genotype to phenotype property, produces regular phenotypes. This facilitates various types of symmetry, self-similarity, and other repetition of phenotype elements which may often achieve high fitness for problems related to the physical world [57]. Genotype reuse enables coordination of changes in the phenotype. For example, while water striders have multiple legs, their length is partially determined by a single gene and mutations to that gene changes leg length in a coordinated way [51]. Coordinated changes in the phenotype may be useful for adapting to environmental changes of various types, such as when the climate changes or when a potential niche is encountered. While symmetry and other forms of regular-
ity are possible without genotype reuse, they are more difficult because they are not created by a one-to-many G-P map. Each element must be adjusted to match other elements independently. Finally, genotype reuse allows a given size phenotype to be described with a smaller genotype and therefore can be considered a form of compression [96][30]. A smaller genotype space can be explored for high fitness individuals in fewer generations.

One disadvantage of genotype reuse which may decrease evolvability is that the one-to-many property that links and coordinates phenotype change also imposes limits on how a phenotype can change. For example, in a variety of insects one of the functions of a gene Ultrabithorax is to increase the size of appendages [51]. This is an example of genotype reuse acting in a coordinated way. However, in water striders the middle pair of legs are long relative to the rear pair of legs. In water spiders Ultrabithorax expression and function has changed from their ancestors to allow elongation of the middle leg pair and relative shortening of the rear leg pair [51]. Thus an evolutionary change was necessary before a more adaptive pattern of relative leg length could occur in the water spider [51]. Among other possibilities, the evolutionary changes that allowed a shortening of the rear pair of legs relative to the middle pair of legs are the addition of new elements to the water strider genome [51]. Before this change to the water strider genome, genotype reuse may have been limiting how the water strider’s phenotype could change by lengthening and shortening rear and middle pairs of legs simultaneously.

Natural G-P maps have inspired researchers to create indirect encodings. The limits that genotype reuse places on how a phenotype may change appears to be less important than its advantages, at least for natural organisms. Natural G-P maps create phenotypes which have evolved from simpler beginnings to more complex modern organisms. This scalability shows a possible way for GAs to also create complex phenotypes to solve more complicated problems.
Indirect encodings are researchers’ attempts to abstract the essential properties of the natural G-P map such as genotype reuse for use in GAs. Indirect encodings often require more time to implement and more computational resources than direct encodings [94]. For these reasons direct encodings were developed for use in GAs first. Often a researcher will compare an indirect encoding against a direct encoding to show that the indirect encoding’s additional complexity is justified by increases in evolvability.

Indirect encodings have been demonstrated to have higher evolvability than direct encodings on a variety of problems [36][7][54][42][40][41][30][82][83][20].

This thesis explored some of the hypothesized mechanisms and conditions under which indirect encodings increase evolvability.

More genotype reuse can be more important than a smaller genotype with respect to increasing evolvability

As mentioned earlier, two effects of genotype reuse are biasing of the phenotype towards regular structures and the ability to describe a given phenotype with a smaller genotype. Both of these effects are the result of a one-to-many G-P map and often occur together. This makes it difficult to determine whether regularity of the phenotype or decreased genotype size is more important to increasing evolvability.

Section 3 demonstrates that more genotype reuse is more important than a smaller genotype size for one experimental system. The system consists of a hexapod agent controlled by a neural network in a simple walking task. The most important experimental attribute of the system was the ability to reverse the normal relationship between genotype reuse and genotype size. Normally, the more genotype reuse, the smaller the genotype needed to describe a given phenotype. In this experimental system, individuals with a large amount of genotype reuse and a large genotype size could be created. Once this was reversal was accomplished, the evolvability of popu-
lations of individuals with large amounts of genotype reuse and large genotypes were compared to populations with no genotype reuse and small genotypes.

The hexapod experiments show that genotype reuse can be more important than genotype size with respect to evolvability. High fitness hexapods were discovered more quickly in the larger genotype search space of the indirect encoding. The indirect encoding was also able to adapt to environmental change more quickly than the direct encoding. This shows that the size of the search space is not the sole or even most important factor in determining evolvability. The density of high fitness individuals in the search space and their arrangement can be more important.

The hexapod experiments also show that encodings with higher adaptability can be selected for over encodings with lower adaptability. Directly and indirectly encoded hexapods were placed in the same population. The indirectly encoded hexapods had been shown to have higher adaptability in previous experiments. When the GA search is performed the two types of encodings compete for population share. The indirect encoding wins the competitions more often than the direct encodings if the hexapods’ environment is changed. This is true even if the indirectly encoded hexapods are initially lower fitness than the directly encoded hexapods. This shows genotype to phenotype mappings with higher adaptability can be selected for in changing environments.

Discussion of these experiments, including limitations, can be found on page 72 and below.

**Genotype reuse can increase evolvability on problems with no geometric structural regularity**

In the above examples the indirect encodings perform well on problems which favor (or at least are solvable by) phenotypes with structural regularity. For example, the hexapod agents can achieve high fitness with neural networks with a large degree of structural regularity. The indirect encoding creates the hexapods’ neural network
by replicating one subnetwork six times (once for each leg). This replication of the subnetwork is a form of structural regularity.

Can indirect encodings also have higher evolvability than direct on problems which do not have geometric structural regularity?

The final part of this thesis (Section 4 on page 75) explored this possibility. The system consists of a pseudoplant agent which achieves high fitness when its surface to volume ratio (surf/vol ratio) is appropriate for the humidity level. While surf/vol ratio depends on the shape of the pseudoplant’s body as a whole, it does not depend on any specifics of the body’s geometry. Consequently, geometric structural regularity would not seem to play a role in determining fitness.

The pseudoplant’s genotype is an *evolvable encoding* which has both direct and indirect forms. The evolvable encoding is designed to allow a smooth transition between direct and indirect forms through mutations. This allows variants of both forms to exist in the population and be selected for or against according to their fitness. Because the encoding can be changed by mutation, one can create a simple initial population and allow it to evolve to avoid accidental imposition of regularities which might occur if the encoding were hand designed and non-evolvable.

The pseudoplant experiments show that genotype reuse can be useful in a system which was designed to avoid bias towards geometrically regular phenotypes. It was found that genotype reuse increased *adaptability* to environmental change, but had no detectable effect on *discoverability*. The “mechanism” by which genotype reuse increases adaptability is what could be called repetition. This repetition did not increase fitness due to spatial regularity. Instead, the indirect form of the encoding allowed faster change in the surf/vol ratio of the pseudoplant and thereby faster adaptation to changed environment than the direct form.

Discussion of the pseudoplant experiments, including limitations, can be found on page 123 and below.
Final Discussion

Understanding how G-P maps affect evolvability is of relevance to both to both artificial and natural evolution.

When using a GA one would like to choose an encoding which will find high fitness solutions in a tolerable number of generations while not being complicated to implement. Which encoding, direct or indirect, is suitable for which problem? Is the additional difficulty of implementing an indirect encoding justified by the generated phenotypes?

The results of this thesis suggest that indirect encodings are suited to a large variety of problems. Indirect encodings have been repeatedly demonstrated [36][7][54][42]-[40][41][30][82][83][20] to increase evolvability over direct encodings on problems with geometric regularities. In addition, the pseudoplant experiments were designed to have no geometric regularities. Despite this lack of geometric regularities, the indirect encoding performed at least as well and sometimes better than the direct encoding. A problem can have unexpected regularities, such as temporal regularity, that may be exploited by an indirect encoding. The size of the genotype search space is not as important to finding high fitness solutions as the density of the solutions.

One hypothesis is that a performant general purpose encoding has at least the following two properties. First, the encoding should have both direct and indirect forms which can be transitioned between by mutation. Selection can then determine the amount of genotype reuse which that is present in the G-P mapping. This would allow both regular and irregular phenotypes to be created by different forms of the same encoding. Second, transitioning between direct (no genotype reuse) and indirect (genotype reuse) forms of the encoding should occur with as few mutations as possible. The GA search is a competition among individuals with different characteristics, including the form of their encoding. Each mutation is a barrier which increases the number of generations needed to explore the possible forms of the encoding. If it takes
many mutations and many generations to explore different forms of the encoding, then the population may converge to a local optima at which the encoding does not exploit regularities which would confer greater fitness.

Understanding the origin of G-P maps, their evolution, and their effect on evolvability is also relevant to biology. Knowledge of how natural G-P maps have evolved would account for the evolution of phenotypic novelty in the history of life on this planet [69]. An understanding of how G-P maps can evolve would allow prediction of the possible shapes of life on other planets. For example, do complex organisms inevitably arise from simple ones [2]?

In particular, the findings of the pseudoplant experiments suggest that genotype reuse could increase evolvability and be selected for under widespread conditions. The pseudoplant model is simple and can be broken down and examined for its generality.

First, the organism’s fitness depends on two variables.

One variable depends on the environment, that is, not influenced by the organism’s genotype. It is a real number which can take on any value within a range. It could represent humidity, temperature, pressure, or other continuously valued natural phenomena.

The other variable is internal to the organism and is determined by its genotype. The internal variable is determined by averaging a list of numbers drawn from a set of four constant real numbers. The constants could be considered an abstraction of anything which exists as a quantum such as atoms or molecules. Their interaction is a simple average. This could be considered an abstraction of any phenotypic property whose average is relevant to determining fitness (to a first approximation). Surface area (as used in this thesis) or density are physical traits that would likely be of importance to fitness everywhere.

The internal and external variables interact to create a fitness landscape. The landscape in this thesis was smooth with neighboring points having nearly the same
fitness value [50]. Also, for any given value of the environmental variable there is one
global fitness peak. Whether or not a more complex fitness landscape with multiple
peaks, internal, and external variables would provide similar results has yet to be
tested.

The environmental variable changes in an unnatural way in these experiments.
The environmental variable changes only when the population’s fitness is above some
high fitness threshold. This was done so that the environmental variable changed
as much as possible without the fitness landscape shifting under the population so
fast that the population would no longer be tracking a fitness maximum. Changing
the environmental variable in a more natural way, e.g. sometimes rapidly, sometimes
slowly within some bounds, could be tested. One hypothesis is that so long as the
population is on a fitness gradient which selects for the same type of change for a “long
enough” number of generations, then genotype reuse will also be selected for. (“Long
enough” will depend on the specific fitness landscape.) But if the environmental
change is so rapid that the net effect is no fitness gradient then genotype reuse would
not be selected for, just as it was not when there was no environmental change. The
smooth fitness landscape used in the pseudoplant model, with its widespread fitness
gradient and single global fitness maximum (for a given value of the environmental
variable), would probably allow a wide range of environmental rates of change. A
rugged fitness landscape would be less tolerant of fast environmental change because
the fitness gradient would change its orientation more quickly as the environment
changed. No environmental change was shown to not select for genotype reuse in these
experiments. But how fast the environment does have to change before a significant
amount of genotype reuse is selected for was not explored. Environmental change
is almost ubiquitous, but the way in which it changed in this thesis was unrealistic.
Further experiments with different types of environmental change could be performed
to discover how this affects the results.
The only genetic operator used in the pseudoplant model is mutation. Mutations are changes to the genotype and presuppose some kind of information storage molecules. Mutations (of DNA) can be caused by sources external to the organism, such chemicals and radiation in the environment [59]. It is hard to imagine that these sources of mutation would not exist everywhere.

Crossover is not used in these experiments. It originated later in the history of life and is therefore not as general as mutation [84][8]. However, whether similar results are obtained with the addition of a crossover operator to the model would not be difficult to test.

The model used in this thesis connects genotype to phenotype by a mapping as though they were physically distinct objects. This is a common first approximation made in genetics [15][9]. A more accurate description of an organism recognizes that the genetic material and its organization is under selective pressure and part of phenotype. In the RNA world hypothesis of early life, the RNA is the entire organism [18]. The genotype and phenotype are the same. In the experiments of this thesis, genotype reuse depends on the mapping from genotype to phenotype. It is not clear how genotype reuse could be implemented in an RNA world.

In addition, the G-P mapping of natural organisms may be many-to-one or many-to-many (elements of genotype to elements of phenotype). In these experiments the G-P mappings were one-to-one or one-to-many. A more realistic G-P mapping could be implemented in an L-system by having more than one rule and more than one rewrite iteration. One rule would rewrite the productions of other rules on successive iterations. Multiple rewrites certainly makes following how the genotype maps to phenotype more confusing, but it does not necessarily take away the fitness advantage of genotype reuse. Consider first the case of an organism with a many-to-one G-P map. The many genotype elements which effect one phenotype element could be thought of as a single element for purposes of genotype reuse. This is another form of
a direct encoding and genotype reuse is not possible. A many-to-many G-P map is an indirect encoding and implements genotype reuse. (A many-to-one and a many-to-many mapping are illustrated in Figure 45) In a model in which many-to-many G-P maps are possible, the fitness advantage that genotype reuse was shown to have in these experiments would be balanced with influences from multiple genotype elements. These multiple influences might lower the ability of the genotype to create structured phenotypes which can be biased to the structure of the problem.

![Diagram of direct and indirect encoding](image)

**Figure 45:** More complicated G-P maps than used in the experiments of this thesis. Many genotype elements mapping to one phenotype element cannot implement genotype reuse and is a direct encoding. A many-to-many mapping can implement genotype reuse and is an indirect encoding.

Other simplifications of the pseudoplant model may also make the results more general. For example, is mutation alone with no environmental change enough to select for genotype reuse? Toussaint [94] gives an example of a plant-like agent encoded with an L-system in which genotype reuse increases over generations, but environmental change is not used. He argues that smaller genotypes creating large phenotypes through genotype reuse are less affected by mutation than larger genotypes creating similarly sized phenotypes. Further investigations of these results is of interest because this line of research would further broaden the known conditions which select for genotype reuse.
In summary, genotype reuse has been shown to be selected for by environmental change and increase evolvability in a simple pseudoplant model. This model has some known inaccuracies, but correcting them would not be difficult and not expected to significantly change the results. If this is the case, this model predicts that organisms with a G-P mapping capable of genotype reuse would increase in evolvability if subjected to moderate amounts of environmental change.

**Future work**

In addition to the improvements suggested above, there are several other directions for further exploration.

Is there, in general or in “interesting” special cases, a relationship between discoverability and adaptability? Whether or not discoverability and adaptability are positively correlated is important to “evolution of evolvability” [98][69].

For example, in the hexapod experiments there was a positive correlation between increased discoverability, adaptation rate, and degree of adaptability. The symmetric encoding was able to discover high fitness individuals and had a high adaptation rate and adaptability degree. The direct encoding discovered no high fitness individuals and had a lower adaptation rate and adaptability degree. A high discoverability is correlated with a high adaptation rate and high adaptability degree.

In some fitness landscapes there will be no relationship between discoverability and adaptability. For example, a landscape could be completely flat with the exception of a localized fitness gradient. If no individual of a naively initialized population happened to be in the localized fitness gradient the population would move about randomly due to random mutations and random selection. The discovery of the localized fitness gradient will be a chance occurrence related to the size of the fitness gradient relative to the entire search space, the number of individuals in the popu-
lation, and the amount of mutation. The adaptability will depend at least on the problem and the encoding of the solution. Discoverability and adaptability on this landscape do not appear to be related. Arguing for the opposite conclusion, the indirect encodings in both the hexapod and pseudoplant systems increased discoverability and adaptability over the direct encoding. (Discoverability only increased using the fixed rule \(\{z \rightarrow d d\}\) in the pseudoplant experiments.) Not only did using a different encoding increase the density of high fitness gradients, but those gradients were also more easily moved upon. This may be true of an "interesting" subset of problems such as that found in the origin and evolution of life.

One could expect interesting consequences in the history of life if discoverability, adaptation rate, and degree of adaptability of an encoding are not always positively correlated. One example is during the origin and evolution of "body plans" [71]. It is possible that the adaptation rate of the winning body plans were high, but the adaptability degree were low. The result would be that extant life is not as adaptable as it could be. The losing body plans may have been able to better adapt to future environmental change, exploit new niches, or coevolve than the winning body plans.

Even if these forms of evolvability are generally correlated, the history of life might be more analogous to a single GA search than repeated GA searches (by which the evolvability measurements were calculated in the experiments above). For any single competition between the encodings in the hexapod model there is a slightly more than 20% chance that the encoding with a lower median adaptability will win. In natural history there may have been instances where chance caused the G-P maps to evolve along paths which lead to less evolvability in the long term.

Finally, also with relevance to biology, the relationship between adaptability, adaptive radiation, and organization of the G-P map would be interesting to explore. Adaptive radiation is the emergence of new forms of an organism after encountering a new environment [15]. This is similar to adaptability in that an existing high fitness
population changes to become higher fitness due to a changing environment. In the adaptive radiation situation however, the organism's movement controls how fast the environment changes. This is similar to how environmental change was implemented in these experiments. Some G-P maps would be better suited for exploiting new niches than others. One hypothesis is that encountering a new environment would have similar effects on a G-P map as being exposed to a changing environment. A G-P map created by exposure to a changing environment would also have more of an ability to exploit new environments than a G-P map evolved in a more static environment.

By studying encodings, we improve our ability to evolve artificial systems and understand the evolution of natural systems.
References


K. H. ten Tusscher and P. Hogeweg. Evolution of networks for body plan


