UNIVERSITY OF SOUTHAMPTON
FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
ELECTRONICS AND COMPUTER SCIENCE

THE EVOLUTION OF EVOLVABILITY

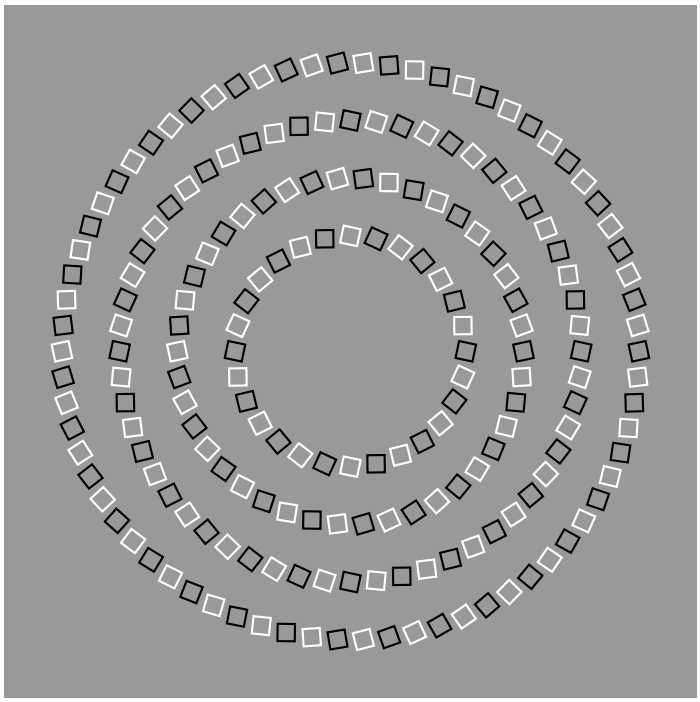
HOW EVOLUTION LEARNS TO EVOLVE

by

Loizos Kounios

A thesis submitted in partial fulfilment for the degree of Master of Philosophy

December 2020



ABSTRACT

It is hypothesised that one of the main reasons evolution has produced such a tremendous diversity of amazing designs is because evolution has improved its own ability to innovate, a process called the 'evolution of evolvability'. Rupert Riedl, an early pioneer of evolutionary developmental biology, suggested that evolvability is facilitated by a specific developmental organisation that is itself a product of past selection. However, the construction of a theoretical framework to formalise such 'evolution of evolvability' has been continually frustrated by the indisputable fact that natural selection cannot favour structures for benefits they have not yet produced. Here we resolve this seeming paradox. Recent work shows that short-term selective pressures on gene interactions are functionally equivalent to a simple type of associative learning, well-understood in neural network research. This is important for the evolution of evolvability because this type of learning system can clearly change in a way that improves its performance on unseen, future test cases, without the need for the future to cause the past. Recognising a formal link with the conditions that enable such predictive generalisation in machine learning systems unlocks well-established theory that can be applied to understanding the evolution of evolvability. Here we use this to elucidate, and demonstrate for the first time, conditions where short-term selective pressures alter evolutionary trajectories in a manner that systematically improves long-term evolutionary outcomes.

DECLARATION OF AUTHORSHIP

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed by myself;
7. Parts of this work have been published as:

   L Kounios et al. (2016) Resolving the Paradox of Evolvability with Learning Theory: How Evolution Learns to Improve Evolvability on Rugged Fitness Landscapes. arXiv: 1612.05955 [q-bio.PE]

ACKNOWLEDGEMENTS

Where to begin?

THE OBVIOUS: There are no words that can express how thankful I am for being under the supervision of Dr Richard A. Watson. Richard has always been available to answer my questions and address all of my concerns. But that should be expected of any supervisor. More than that, Richard's enthusiasm is contagious and it is thanks to that that I found myself motivated and willing to do the work.

THE NOT-SO-OBVIOUS: Too many people to list by name. If you had any sort of interaction with me, you helped me maintain my sanity, so I owe you my thanks. There's the magnificent group of people in Southampton who made me feel welcome from day one, and whose company made the stay enjoyable. Many thanks to the TEXnicians at TEX.SX for sharing their expertise in TEX/LATEX and typography. If nothing else, I am proud of the fact that my thesis looks nice! Special thanks to Kostas for being a great friend and helping so much more than even he realises.

LASTLY: I would like to express my utmost gratitude to the people who helped proofread this report.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

dDE	discrete differential evolution.
DE	differential evolution.
evo–devo	evolutionary developmental biology.
GA	genetic algorithm.
G–P map	genotype–phenotype map.
GRN	gene-regulation network.
SBB	scalable building blocks.
VSM	variable structural modularity.

LIST OF SYMBOLS

$\boldsymbol{\phi}$	phenotype | vector.
$\boldsymbol{\phi}_a$	adult phenotype | vector.
$\bar{\boldsymbol{\phi}}_a$	population mean adult phenotype | vector.
$\mathbf{B}$	regulatory interactions | matrix.
$\bar{\mathbf{B}}$	population mean regulatory interactions | matrix.
$\mathbf{g}$	direct effects on phenotypic traits | vector.
$\bar{\mathbf{g}}$	population mean direct effects on phenotypic traits | vector.
H	Heaviside step function | function.
t_d	number of developmental timesteps | scalar.
w	fitness | scalar.

1 INTRODUCTION

How does evolution enable natural populations to adapt to the extraordinarily diverse and challenging selective environments that we observe? Rupert Riedl, an early pioneer of evolutionary developmental biology research, suggested that the evolution of complex organisms is facilitated by developmental architectures that are organised by natural selection to 'mimic' or 'imitate' the functional constraints on phenotypes [2–5]. Contemporary evolutionary developmental biology recognises that developmental organisation is both a product of natural selection and a factor that can significantly alter subsequent evolutionary outcomes – i.e. because developmental constraints and biases favoured by past selection can modify the likelihood of generating adaptive phenotypic variations that future selection can act on [2–4, 6]. It is therefore clear that natural selection can thereby produce heritable changes that modify the ability to evolve – a phenomenon called the 'evolution of evolvability' [4, 7, 8]. But our understanding in this area lacks a theoretical foundation that can answer questions about what types of developmental architectures are more evolvable, how they can be evolved via natural selection and under what conditions they facilitate the evolution of complex organisms.

Recent investigations into the evolution of evolvability have been successful in demonstrating that natural selection can change phenotypic distributions in a manner that modifies evolvability in several respects, including increasing developmental robustness and phenotypic heritability [9–11], increasing the rate of adaptation under directional selection [12], re-evolving previously evolved phenotypes more quickly [13], and enabling evolution to track changes in the environment more rapidly [13–15]. In some cases it can also accelerate the evolution of particular novel phenotypes, specifically, different combinations of modular features [15, 16]. These works have been careful to confirm that the changes to phenotypic distributions or the genotype–phenotype map (G–P map) that facilitate such evolvability evolve only because of 'short-term' adaptive benefits that affect differential growth or survival between genetic lineages. This is necessary: Although developmental organisations might influence long-term evolutionary outcomes (i.e. beyond these timescales that can be 'seen' by the selection that produced them), long-term benefits cannot be the reason that these organisations evolved.

Nonetheless, it remains legitimate to seek the general conditions where long-term evolvability (e.g. that results in complex organisms) can be facilitated by short-term selection. Although it is clear that modifications to developmental organisation can alter the path of evolutionary trajectories through phenotype space [17, 18], as well as their speed [13, 15, 16], it has not previously been shown that short-term selection can do this in a way that systematically changes future/long-term evolutionary trajectories for the better. For example, if we were able to re-run the tape of evolution with a different developmental organisation, even if we started from the same initial phenotype under the same selective conditions, it might be the case that evolution with one developmental organisation soon exhausts the production of beneficial variations and is thereby (by definition) trapped at a local optimum in the adaptive landscape, whereas evolution with another developmental organisation avoids or escapes many local optima and continues to produce adaptive

variations that ultimately lead to superior regions of phenotype space. In this report, we demonstrate that Riedl's hypothesis that evolvability is facilitated by a specific developmental organisation that is itself a product of past selection holds true.

The key to elucidating the necessary conditions for this result is the recognition that, given heritable differences in developmental organisation, evolution by natural selection can act not just as a conventional evolutionary system but as a learning system, with fundamental capabilities and limitations in common with other well-understood learning systems. Recent work [19] has shown that short-term selective pressures on phenotypic correlations (e.g. as produced by changes to the connections of a gene-regulation network (GRN)) are functionally equivalent to a simple type of learning mechanism, specifically associative or correlation learning, well-understood in neural network research. The developmental organisation that evolves thus internalises the structure of the selective environment (i.e. the functional constraints on phenotypes). This is consistent with other results examining how modularly-varying environments induce corresponding modularity in the G–P map [13, 14, 16, 20]. Furthermore, when evolved interactions control a non-linear and recurrent developmental process, such as a GRN, (rather than defining a linear G–P map) they can store and recall multiple distinct phenotypes that have been selected for in the past, spontaneously re-creating these patterns in adult phenotypes (i.e. without further selection) given random variation in embryonic phenotypes (e.g. produced by random genetic variation or environmental perturbation). This demonstrates the 'imitatory' nature of developmental organisation, as described by Riedl, very clearly.

How does this imitatory developmental organisation affect long-term evolvability? Recognising the equivalence with learning mechanisms illuminates the relationship between short-term and long-term evolvability because, in the context of learning systems, we already know a lot about how a system can generalise information gained from training data to predict the correct response to previously-unseen test data. This insight immediately defuses the mysterious connotations associated with long-term evolvability: Learning systems can exhibit predictive generalisation, not because they have a way for the future to cause the past, but simply because they are able to capture structural regularities in the environment that are (deep enough to be) time invariant, i.e. common to both the training data and the test data. A learning system that can only remember one thing at a time (like a one-to-one G–P map) cannot do this, but a learning system that can represent correlation structure, even simple pairwise correlations, can. Thus we hypothesise that, in the same sense, and with the same capabilities and limitations, an evolutionary process operating on a developmental organisation capable of imitating the structure of the selective environment can facilitate future adaptation. However, just as a learning system cannot exhibit predictive generalisation unless the structural regularities present in the environment can be represented in the type of structural regularities it can learn, short-term selection will not be successful in facilitating future evolution. Here we use this knowledge to demonstrate the consequences of Riedl's ideas for long-term evolvability of a GRN in highly-epistatic fitness landscapes.

REPORT STRUCTURE

The rest of this report is structured as follows.

In chapter 2, we review and discuss evolvability and relevant phenomena and concepts, as well as make the connection between learning and evolution while presenting the evidence that suggests that evolution is a learning process. In chapter 3, we introduce discrete differential evolution, a novel evolutionary algorithm that can be used to solve modular problems without learning. The algorithm uses differences between individuals in the population to increase the variation operators' evolvability as more individuals in the population reach areas of high fitness. Although this chapter is not closely related to the following chapters which focus on the evolution of evolvability in GRNs, this chapter is still relevant to the central theme of this thesis: evolvability. That is to say, though the work presented in this chapter does not help explain the relationship between the evolution of evolvability and learning, it showcases how a population of individuals can become more evolvable over evolutionary time. Additionally, the discussion of the literature on modularity and model-building algorithms is important to the chapters that follow. As such, we felt that the work presented in this chapter still fits the overall theme of the thesis. In chapter 4, we introduce and familiarise the reader with the GRN model used to investigate evolvability by reproducing parts of the work published in Watson et al. [19]. We then expand on their work by investigating how the evolution of a developmental memory affects future evolutionary trajectories. In chapter 5, we present a solution to the paradox of long-term evolvability from short-term selection by showing how evolution can learn to find increasingly fitter phenotypes on a multi-modal fitness landscapes over evolutionary time, and use learning theory to explain why the results are not surprising. In chapter 6, we discuss future research directions, including how we can use our model and learning theory model to improve our understanding of facilitated variation (with some preliminary results). Lastly, chapter 7 concludes the report.

2 THE EVOLUTION OF EVOLVABILITY

The modern synthesis is the current paradigm of evolution theory. Modern synthesis presents an account of evolutionary phenomena that can be best summarised with the phrase 'heritable variation in reproductive success' – meaning it relies on three fundamental mechanisms that copy (inheritance) minor variants of the genotypes (variation) of the better-performing phenotypes in a population (selection) to the next generation, a process which when repeated results in phenotypes adapted to their environment.

Although there is little doubt about the importance of these mechanisms, recent theoretical and empirical contributions on evolvability, evolutionary developmental biology (evo–devo), epigenetic inheritance and phenotypic plasticity raise questions about whether the modern synthesis provides a satisfactory account of evolution or whether it omits all-important details. In the past few years, attempts have been made to create a new framework that improves our understanding of evolution. The new framework is called the extended evolutionary synthesis [21–24].

In this report, I focus on one aspect of the extended evolutionary synthesis: evolvability, and specifically on the evolution of evolvability via the evolution of development and the G–P map.

2.1 WHAT IS EVOLVABILITY?

Darwin [25, p. 167] was the first to concede that 'our ignorance of the laws of variation is profound'. More than 150 years later, Darwin's statement remains true. Although variation is the raw ingredient upon which selection acts, we still have very little understanding of it. How is it that animals have evolved different anatomies and physiologies? What drives the generation of these variants? Half the answer is genotypic variation. The other half is the process that translates genotypic variation into phenotypic variation – i.e. development. We know a lot about the former, but very little about the origins of the latter and the effect it has on evolution.

Consider Darwin's finches, for example. Although they have evolved from a common ancestor, they differ in the shape of their beaks – each species' beak is adapted to its environment. It is obvious that natural selection has favoured each of the species' beaks because it enables them to consume the type of food available in their environment and convert it to energy to survive. But even though natural selection can explain how these variants propagated throughout the population, how did they come to be and why were they even possible to begin with? What were the driving forces that facilitated precise changes in the shape of the beaks? The simplest answer, and the one that Darwin himself thought to be true, is that a number of small, heritable variations accumulated in the species, thus resulting in the inter-species differences we observe. But is this answer plausible or are there other mechanisms involved?

It has been shown that it possible for the developmental process to produce structured phenotypic variation such that changes in the beak size and shape are a result of minor

Box 2.1: Case study: Limb co-variation in monkeys and apes.

The forelimbs and hindlimbs of monkeys tend to be equally long [30]. Since they are serially homologous and have similar functions, this is not surprising. Due to their common origin, 'the genetic and developmental architectures for the two sets of limbs [...] are often very similar' [32]. This means that mutations that affect the functionality and length of one set of limbs tend to also affect those of the other set. This is an example of a developmental process constraining and biasing phenotypic variation (i.e. a developmental constraint [33, 34]). Consequently, even if genotypic variation is random, the developmental process of monkeys only supplies a pre-determined amount (both in magnitude and direction) of phenotypic variation – specifically, the constraint enforces a one-to-one limb length ratio.

In a recent study, Young et al. [31] analysed correlations between the forelimbs and hindlimbs of monkeys and apes (including humans). Non-human apes differ from monkeys in that they have long arms used for tree locomotion, whereas humans have a significantly different limb structure with much longer hindlimbs (an adaptation to bipedal locomotion) [35, p. 370]. Young et al. [31] reported that within all examined species there exists a correlation between hindlimb and forelimb length. What is interesting is that although a correlation exists in apes (including humans), it is weaker than the correlation found in monkeys.

This suggests that the difference in limb co-variation and functionality between apes and monkeys could have arisen due to a relaxation of the developmental constraint governing limb co-variation in a common ancestor of the apes (but not the monkeys), resulting in the increased diversity observed. Young et al. [31] suggested that the relaxation of the constraint increased evolvability by enabling the independent evolution of forelimbs and hindlimbs both in terms of length co-variation and functionality. With the constraint relaxed, natural selection favoured and maintained the different limb co-variation constraints and functions evolved in apes (including humans).

changes in the genotype. A famous example is the *eyeless* gene in the *Drosophila*. When the gene is mutated during development, it causes the growth of an additional fully functional eye on the *Drosophila* body (with the exception being that the eye is not connected to the nervous system) [26]. Another example is of a single mutation on a gene causing the brain of the mouse to be three times its typical size [27]. Other examples include constraints on limb co-variation [28–31] (see box 2.1 for a brief case study of limb-covariation in monkeys and apes).

Phenomena such as these are of interest because they are not yet explained. Why is the G–P map structured in a way that enables this type of phenotypic variation? As GP Wagner and Altenberg [4] put it:

> [This] brings us to a level of phenomenon that is distinct from adaptation itself. It concerns the variational properties of the genome—the nature of phenotypic variation produced by genetic variation. (GP Wagner and Altenberg [4, p. 968])

The variational properties of populations cannot be treated as an afterthought – they are a requirement if adaptation via evolution by natural selection is to take place. Adaptation requires that changes to the genotype are somehow channelled into adaptive phenotypic changes. Obviously, adaptation does not require that *every* change is beneficial – that would imply directed evolution. For evolution to be able to produce adaptive changes, however, the distribution of phenotypes that is directly accessible via changes to the parental genotype must at least include some adaptive phenotypes. That is, there must exist a small probability that a variant is adaptive [4].

These are the type of phenomena evolvability attempts to explain – how genotypic variation is translated into phenotypic variation, and how the structure of variation itself evolved. More specifically, it is about the structure of phenotypic variation that can be induced following genotypic variation – and a key component of the structure of phenotypic variation is the G–P map. Evolvability is about understanding phenotypic variation, prior to all considerations of selection. Research on evolvability is particularly targeted at the capacity of organisms to produce adaptive phenotypic variation from random genotypic variation. Formally:

EVOLVABILITY An organism's capacity for adaptive evolution. An organism's propensity to generate adaptive phenotypic variation from random genotypic variation. Primarily a property of the G–P map since it is the G–P map that translates a genotype to a phenotype.

Under this definition, we can see the self-referential relationship between development and evolution. That is, the impact genotypic variation has on phenotypic variation depends on the phenotype itself, since the developmental process is both part of the phenotype *and* genetically encoded. In other words, the products of evolution change the process of evolution.

In the rest of this chapter, I survey the literature on evolvability and development. I begin with a brief history of evolvability (section 2.2), and proceed by analysing four important areas for understanding evolvability:

1. the difference between variability and variation (section 2.3),
2. the effect development and the G–P map have on variability (section 2.4),
3. whether robustness and evolvability are antagonistic forces (section 2.5),
4. whether evolvability can evolve (section 2.6).

Lastly, I briefly discuss the relationship between learning and evolution (section 2.7), an important aspect of the work that follows in the next chapters.

2.2 BRIEF HISTORY OF EVOLVABILITY

Since the first use of the term, evolvability has been used to describe a large number of related phenomena. For that reason, evolvability has been difficult to define.

The first use of the term 'evolvability' was in 1989 by Dawkins [37]. Dawkins [36] used his Biomorphs model (first introduced in 1986 [38]) to illustrate how a developmental process can enhance an organism's adaptability (evolvability), and how the developmental process itself may evolve (evolution of evolvability).

Although Dawkins was the first to use the term, a number of people had already investigated evolvability as defined by Dawkins, but simply never used the word 'evolvability'. Riedl [3], for example, discussed developmental constraints and body plans in the English version of his book *Order in Living Organisms* published in 1978 (for a detailed review of Riedl's work, written by two of his former students, see GP Wagner and Laubichler [5]), and so did Maynard Smith et al. [33] in 1985.

Since then, evolvability has been used to describe a large number of phenomena (figure 1). These phenomena are largely overlapping as they mainly differ in whether the

Figure 1: The many faces of 'evolvability' in evolutionary biology. Originally published in Brown [32].

The term 'evolvability' is used to refer to:	Reference(s)
(1) The capacity of populations to generate heritable phenotypic variation.	GP Wagner and Altenberg [4] and A Wagner [10]
(2) The capacity of the individuals within a population for adaptive phenotypic plasticity.	West-Eberhard [39]
(3) The intrinsic capacity of the individuals within a population to generate phenotypic variation in response to genotypic variation.	MW Kirschner and JC Gerhart [40]
(4) The potential of a population to produce novel mutations for use in the evolution of adaptations in the medium- to long-term.	Maynard Smith and Szathmáry [41] and Pigliucci [42]
(5) The current genetic variation in a population (rather than the prospective variation).	Houle [43]

focus is on genetic or environmental cues to facilitate evolvability, and on the time-scale they operate on (within populations, species, etc.) [32, 42].

It is mainly due to the lucid work of GP Wagner and Altenberg [4] and MW Kirschner and JC Gerhart [40] in the mid to late 1990s that research on evolvability started gaining traction. They provided straightforward definitions of evolvability, explained the role of development and regulatory interactions in facilitating it, and how other research areas in evolutionary biology are related to evolvability. Their definition is about the capacity of individuals and populations to produce adaptive phenotypic variation, and it encompasses a multitude of phenomena which are related to evolvability, such as development and the G–P map [12, 13, 16, 19, 44–46], mutation rates [47–49], modularity [12, 15, 19, 20, 50], and robustness [10, 11, 51–53].

2.3 VARIATION AND VARIABILITY

GP Wagner and Altenberg [4] brought together ideas from development, evolution of complexity and evolutionary computation in their discussion on evolvability (see also Toussaint and von Seelen [6] for a similar discussion).

They highlighted the distinction between variation and variability:

VARIATION The present phenotypic differences among individuals in a population.

VARIABILITY The propensity to vary. The phenotypic variation that may be generated via changes to the genotype, prior to all considerations of selection. A dispositional property.

GP Wagner and Altenberg [4] used 'solubility' as an example of property similar to variability. Solubility does not describe a substance's specific state but rather its expected behaviour when in contact with solvent. Similarly, variability describes a phenotypic trait's behaviour when influenced by environmental or genetic factors.

Since variability comes prior to all considerations of selection, it becomes possible to shift our focus from the process of selection towards the biological mechanisms responsible for the generation of phenotypes: mutation and development. The effect genotypic mutations have on phenotypes is dependent on the developmental process. Therefore, these two mechanisms determine an organism's variability.

Because the concept of variability is phenotype-centric and not gene-centric (the current paradigm in modern synthesis), it is possible to ask questions about the type of genetic and developmental architectures that can facilitate evolvability by channeling random genetic variation into partially non-random phenotypic variation. Since variability comes prior to all considerations of selection, it is fairly easy to quantify. Straightforwardly, variability is the probability distribution over all possible phenotypes given all possible mutations on the genotype [37]. The variability structure of an organism can thus be mathematically described by the shape of its variational distribution, i.e. the shape of the probability distribution over the set of all potential phenotypes [6, 37]. Note that this is different from the type of evolvability Houle [43] defines in that Houle's definition is about standing genetic variation in a population whereas this is about potential phenotypic variation that is yet to be realised.

It is important to note that beneficial variability structure is not merely about how many phenotypes are possible, but rather about *what* phenotypes are possible:

> The situation is analogous to obtaining a verse of Shakespeare from monkeys banging away on typewriters. Typewriters make this far more likely than if the monkeys had pencil and paper. The typewriters at least constrain them to produce strings of letters. Similarly, the genotype-phenotype map constrains the directions of phenotypic change resulting from genetic variation. (GP Wagner and Altenberg [4, p. 970])

If variability was merely the number of possible phenotypes, the pencil and paper organisms would enjoy better variability. But that is not the case. Although the number of possible phenotypes is significantly lower when using a typewriter, the underlying variability is clearly superior – every phenotypic trait is a letter. A good variability structure should constrain the phenotype space by preventing lethal mutations and focusing genotypic variation into potentially useful phenotypic variation.

By understanding the variability structure of an organism, it is possible to avoid statements such as 'phenotypic variation is random'. Since variability determines the potential range of variation, the existing variation is not random but is rather what the underlying variability structure allows it to be. Therefore, to explain evolvability, we need to understand the variability structure of an organism and what induces it: development and the G–P map.

2.4 DEVELOPMENT AND THE G–P MAP

Following a change in the genotype, the developmental process, which is responsible for translating a genotype to a phenotype, determines the phenotype produced. Thus, the variability structure of an organism is determined by the biases and constraints imposed by the developmental architecture. Therefore, some developmental architectures are more evolvable than others.

Box 2.2: Evo–devo and its relationship to evolvability.

Evo–devo emerged in the early 1980s to address the exclusion of developmental biology from the modern synthesis. Evo–devo attempts to bridge the gaps between evolutionary biology and developmental biology by offering explanations about the origin and evolution of developmental processes, and the consequences of developmental processes on evolutionary trajectories. Formally:

EVO–DEVO The study of how developmental processes affect evolutionary trajectories, and how developmental processes themselves have evolved and are being evolved. In particular, how constraints and biases in development affect evolution by altering the phenotypic distribution available to selection, and also how these constraints and biases are evolved. Of particular interest is understanding the evolutionary forces that facilitate changes to the developmental processes that enable morphological novelties.

As Müller [23, p. 947] put it, 'evo–devo moves the focus of evolutionary explanation from the external and contingent to the internal and inherent'.

Hendrikse et al. [45] explain that evo–devo should not simply answer developmental 'how' questions by, for example, investigating the evolution of the developmental processes of monkeys and apes and concluding that the evolved differences in development resulted in limbs with different functionalities and relative lengths between the species. Instead, evo–devo should ask questions about how the differences in the developmental processes of monkeys and apes affect their variability structure, which may in turn affect the evolutionary trajectories of monkeys and apes – a question that 'asks how properties of development interact with selective forces to produce evolutionary change' [45, p. 394].

From the above, it should be clear that evo–devo and evolvability are closely related. Hendrikse et al. [45] even suggest that the main objective of evo–devo is, in fact, to explain evolvability (see also von Dassow and Munro [54]). They point out that although evo–devo focuses on developmental biases and constraints (modularity, robustness, etc.), these are the exact phenomena which determine an organism's variability structure, and hence its evolvability.

It is important to understand what aspects of a developmental architecture make organisms more evolvable, and in particular what the underlying mechanisms that would give rise to potentially useful phenotypic variation are (see box 2.2 for a brief discussion on development, evolvability and evo–devo).

MW Kirschner and JC Gerhart [40] identified features of the developmental process which can be said to make organisms more evolvable and robust and integrated them under the theoretical framework of facilitated variation (also see references [7, 8, 55]). The facilitated variation theoretical framework attempts to explain the nature of phenotypic variation imposed by the developmental architecture and not just the amount. Facilitated variation is variation which is useful.

Some of the underlying mechanisms and organisational principles described in the facilited variation theoretical framework include modularity, robustness, weak regulatory linkage etc. Kirshner and Gerhart used the *homeobox* gene system genes in the *Drosophila melanogaster* to illustrate some of the above features. They showed that the *homeobox* gene alone is sufficient to determine the kind of body segment that is placed in each of the fly's compartment. More importantly, they explained that changes in the expression of *homeobox* genes produce non-random phenotypic changes by altering the ordering of body segments instead of producing 'intermediate' body segments and non-viable tissue.

These ideas and characteristics of the developmental process are related to the 'representation problem': how the phenotype space should be structured so that genotypic

variation has a chance to introduce adaptive variants. In their work, GP Wagner and Altenberg [4] emphasised the differences between successful and unsuccessful evolutionary computation experiments to illustrate that unstructured variation and selection are 'not universally effective in producing adaptation'. Toussaint and von Seelen [6] explained that evolution itself must have dealt with the representation problem. They argued that nature has developed a specific language to describe organisms and that this language is one out of infinite possibilities, making the idea that the language chosen is a mere coincidence incredibly unlikely. Further than that, the argument is that nature has evolved a language that solves its own representation problem by shaping variability in an appropriate manner.

Toussaint and von Seelen [6] also explain the relationship between model-building algorithms in evolutionary computation and the representation problem in evolutionary biology. Model-building algorithms have been shown to progressively improve their ability to explore the solution space to find better solutions. They do so by building a model of interactions between problem variables, that is representative of the structure and dependencies of the optimisation problem. Different algorithms utilise the built model in different ways (e.g. macro-variation, biased generation of phenotypes, etc.) to explore potentially useful areas of the solution space [56–65]. It may be possible to gain insight into how organisms deal with the representation problem, given what we know about the heuristics model-building algorithms use to estimate which areas of the solution space may be of interest.

There are of course issues to be addressed as the knowledge transfer from the literature on model-building algorithms to evo–devo is not seamless. Unlike work on evo–devo, model-building algorithms are not constrained by the 'laws of evolutionary biology'. For example, model-building algorithms may introduce a rule enforcing that variables a and b have the same value, given that they were often seen having the same value in the past. Whereas the possibility of introducing such a rule would be hard-coded in the model-building algorithm, evolution by natural selection is not capable of solving the problem easily. In evolution, not only would such a rule have to arise via a mutation to the genotype, but it would also have to be selected for. So even though there are clear similarities between model-building algorithms and the evolution of evolvability, how exactly do organisms deal with the representation problem to improve their evolvability?

The G–P map of an organism is what determines how it deals with the representation problem. The G–P map maps the genotype (that which is inherited) to the phenotype (that which is selected). Since the G–P map is encoded in the genotype, it is subject to variation and selection, and can therefore evolve. The key is understanding why G–P maps shape variability the way they do, and under what conditions and selective pressures G–P maps make individuals more evolvable by changing variability in an appropriate manner.

A large body of work has investigated how the phenotype space as induced by the G–P map may affect evolvability, robustness and related phenomena. It has been shown that modularly-varying goals may promote the evolution of modular networks [13, 20], and the evolution of more evolvable individuals by increasing their speed of adaptation [13] or even by showcasing facilitated variation properties such as weak regulatory linkage [16]. These results are explained via the existence of neutral networks (i.e. networks of genotypes which are connected via mutations, but produce the same phenotype). By alternating between such goals, evolution finds genotypes whose expressed phenotypes solve the problems and are neighbouring with respect to the genotype space. It has also been shown that the G–P map may promote robustness by causing neighbouring genotypes to

generate the same phenotype, and that this can in fact promote evolvability and innovation [9–11].

Nonetheless, it is not yet clear what the general conditions that facilitate long-term evolvability (e.g. that results in complex organisms) are and how or whether they can be facilitated by short-term selection (e.g. favouring a developmental organisation that mimics the selective constraints on phenotypes because it confers an immediate reduction in deleterious variation and/or increase in beneficial variation). Although it is clear that modifications to developmental organisation can alter the path of evolutionary trajectories through phenotype space [17, 18], previous work does not identify the conditions under which these developmental organisations evolve and why they alter the phenotype space in a beneficial manner. Recent work suggests that learning theory can help us identify these conditions.

In recent years, it has become commonplace in computer science to use evolutionary biology as inspiration for solving computational problems, but a recent paper by Watson et al. [19] suggests computer science can be used to answer evolutionary phenomena (see Watson et al. [66] for an extensive discussion on this topic). Watson et al. [19] showed using computational simulations that a gene-regulation network can form a memory of past phenotypes and even generalise from past phenotypes to learn a family of phenotypes. More than that, they showed an equivalence between the evolution of regulatory interactions and Hebbian learning [67, 68], and explained the implications that arise from the possibility that evolution is a learning process (discussed further in section 2.7). Recent work has expanded on this idea by investigating how the notions of under-fitting, over-fitting and induction in machine learning can be used to understand how evolution can learn to generalise via the evolution of the G–P map [69]. Our own work is on how development's ability to form a memory of past phenotypes and generalise may affect evolutionary trajectories [70] (see chapters 4 and 5).

2.5 ROBUSTNESS VERSUS EVOLVABILITY

Many closely-related definitions of robustness have been proposed, but here we adopt the definition put forth by A Wagner [11]:

ROBUSTNESS A phenotype is robust if it persists when perturbed. Perturbations include environmental perturbations such as changes in temperature or in available nutrients, and genetic perturbations via mutations to the genotype.

Like evolvability, robustness is a fundamental property of biological systems [7, 71, 72], with a large body of work in the literature researching the relationship between robustness and evolvability [9–11, 51–53].

Under the rubric of developmental biology and evo–devo, robustness is also known as 'developmental canalisation' [73]. A highly robust phenotype, for example, will translate to small genetic or environmental perturbations being ignored and development reproducing that same phenotype. 'Canalisation' thus refers to the fact that the developmental process will follow the same 'canal' to generate that same phenotype. But given a sufficiently large perturbation, development will move outside the canal. Waddington's epigenetic landscape [74] illustrates the concept of developmental canalisation very clearly (figure 2). The blue spheres in the two panels follow different canals to develop into different phenotypes and the shape of the landscapes makes it clear that it is difficult to move

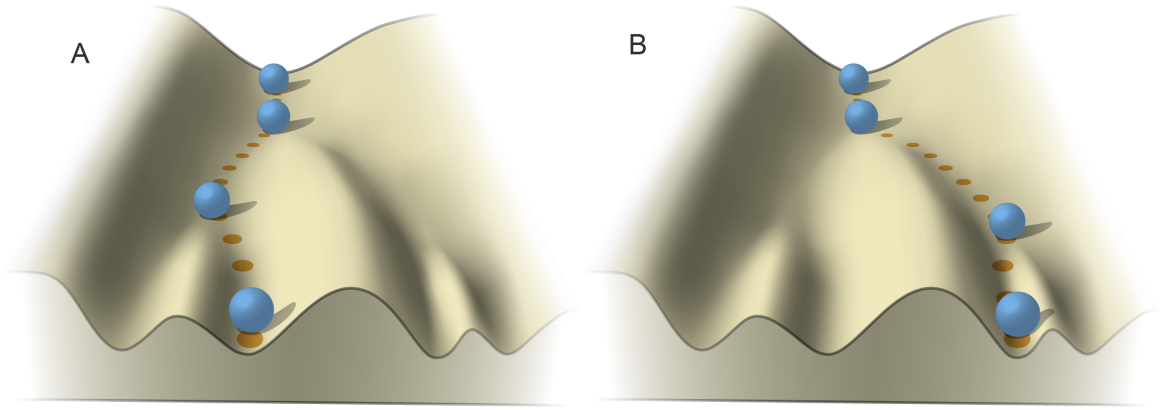


Figure 2: Waddington's epigenetic landscape [74]. Reproduced from KJ Mitchell [75].

from one canal to the other. Given the above definition, it seems obvious that conservative and myopic nature of natural selection would favour the canalisation of previously selected phenotypes in order to evolve a more reliable developmental process that can reproduce these phenotypes.

If natural selection favours more robust developmental architectures, however, how can evolvability evolve? The two phenomena seem antagonistic on the surface. A more robust developmental architecture reduces the amount of phenotypic variation that can be produced, thus limiting the amount of phenotypic variation upon which selection can act [10]. But canalisation and evolvability are both quantitative properties. In terms of canalisation, this means that perturbations with respect to a phenotypic trait may or may not be neutral. Similarly, evolvability is not merely about increasing variability, but about channelling it in the right direction (e.g. by eliminating lethal mutations or facilitating adaptive phenotypic variation). Since both properties are quantitative and do not necessarily operate at the extremes, there is no reason the two cannot coexist. The variability structure of an organism is a result of both properties. Populations can be both robust and evolvable. For example, the existence of developmental canalisation can enable the accumulation of cryptic variation within a population; variation which can be beneficial when the environment changes, thus improving evolvability. Another example is the existence of neutral networks [71] as imposed by the developmental architecture which can greatly improve evolvability [16]. More generally, canalisation can constrain phenotypic variability in a manner which limits the number of deleterious or lethal mutations. Although this may be construed as antithetical to evolvability because potential phenotypic variation is decreased, the variability structure of such an organism would be highly evolvable since it is biased towards more promising regions of the phenotype space.

2.6 EVOLVABILITY AND FITNESS

One of the main arguments against evolvability is that it is inherently opposed to natural selection. Since selection is intrinsically myopic and acts solely on fitness differentials, how can evolvability, which may not be directly associated with fitness, be selected for? In other words, high-fitness individuals will be selected for regardless of their underlying evolvability.

Note that the argument is not against evolvability *per se*, but rather against the *evolution* of evolvability. It is clear that evolvable individuals will be selected in a situation where their improved evolvability can help them adapt rapidly and survive (such as when the environment changes), much like well-written source code will be preferable because it is more amenable to changes to customer requirements. But the question is how these evolvable individuals would arise to begin with given that there is no selective pressure for future benefits (i.e. evolvability)? If there is no selective pressure for evolvability, this means that evolvability has not been shaped over time, and hence there can be no evolution of evolvability. Individuals will differ in their evolvability regardless of selection, but if selection does not amplify these differences evolvability's impact will be insignificant.

Although the argument does have merit, it only does when not used as a straw-man argument. The argument for the evolution of evolvability has never been that it is an organismal adaptation, but rather that it is a side-effect of other evolutionary mechanisms. That is to say, evolvable individuals are not selected *because* they are evolvable; evolvability is an indirect result of selection.

As explained by GP Wagner and Altenberg [4], evolutionary biology asserts that all adaptations are the result of mutation and natural selection. But for that to be true, organisms *have* to be evolvable. Which leads to the question of *why* organisms are evolvable. Additionally, research suggests that G–P maps constrain and bias the direction of phenotypic variation [33, 34, 76, 77], that mutation and recombination rates are different between species [78, 79], and that organisms are extremely robust to environmental [71, 80] and genetic perturbations [71, 81] – all of which can be construed as properties related to evolvability. As Pigliucci [42] explains, these are not necessarily adaptations, but being able to explain them may have significant impact on our understanding of evolution. Therefore, the question is under what conditions these phenomena would arise.

An often-cited explanation is that improved evolvability comes from neutral mutations on the 'language of variation'. That is, evolvable individuals may arise due to mutations that do not (or minimally) affect their fitness, but instead have a significant impact on their potential phenotypic variation. Toussaint and von Seelen [6] use a grammar encoding scenario to illustrate this. In grammar encodings, many genotypes are developed into the same phenotype. For example, both `<CDCD>` and `<AA, A:CD>` yield phenotype `<CDCD>`. The variability of the two encodings is different, however. The former is a one-to-one encoding, whereas the latter has modular structure. Importantly, the transition from one encoding to the other is neutral – that is, the individual whose grammar changes via a mutation suffers no fitness effect at that point in time. But what changes is its underlying variability. They then proceed to illustrate how evolution changes grammar encodings in ways that facilitate the evolution of plant-like structures. Evolving the same phenotype with a one-to-one mapping and stepwise modifications would have been practically impossible. What is important is that evolution evolved and maintained a language suitable for the objective, without any direct selection for evolvability.

More interesting, perhaps due to its paradoxical nature, is the idea that evolvability may in fact evolve via selective pressures acting on robustness [7, 37, 40]. Natural selection would generally favour robustness over flexibility and thus we expect that previously selected phenotypes will be more resistant against genetic or environmental variation [49, 82]. Such selection for robustness decreases the phenotypic variability of the organism and is seemingly opposed to evolvability as it inhibits the discovery of potentially useful phenotypes in new, previously-unseen environments. Nonetheless, even if selective pressures were directly or indirectly acting on robustness, it is not immediately obvious how

this can result in the evolution of evolvability. After all, with selection acting on robustness, the organism would be expected to become extremely robust by fully canalising the currently selected phenotype. That is true, but new and useful phenotypic variation may arise when the environment changes throughout a population's evolution [13, 15, 16, 19, 20, 83]. The idea here is that cryptic genetic variation can be released under environmental stress leading to high phenotypic variation [10, 11]. More interestingly, selection acting on robustness over a distribution of a family of phenotypes can potentially learn structural correlations among phenotypic characters that alter the distribution of its potential phenotypes in ways that may facilitate evolvability [1, 69, 70] (see chapter 5).

As Pigliucci [42] explains, whether evolvability is a trait directly selected for or a byproduct of other evolutionary forces remains an open question whose resolution can have a significant impact on our understanding of the evolutionary process.

2.7 LEARNING AND EVOLUTION

2.7.1 What Is Learning?

Simply put, learning is the ability to have past experience affect future behaviour for the better with respect to a task [84]. A more appropriate definition for the work discussed here is as follows: learning refers to the performance improvement of a learning system against a given environment through the acquisition of information from having experienced similar environments in the past [85]. This definition has four key components: environment, performance experience and improvement. Environment refers to the problem or task that the learning system has to learn (e.g. distinguishing between blue and red balls). Performance is an objective assessment of the behaviour of the learning system in a given environment (e.g. the proportion of correctly identified balls). Experience refers to the way in which the learning system responded to the environment (e.g. the way in which a decision is made – whether a ball was blue or red). Improvement refers to a change in which the learning system responds to the environment that is beneficial (e.g. changes in the synaptic connections of the human brain that result in better performance).

2.7.2 Learning As a Search Problem

Learning can be interpreted as a search problem. A search problem is defined as searching for the best behaviour in a given environment (i.e. problem). Assume that we have a learning system that belongs to the family of neural networks. This learning system is capable of receiving input through retina-like sensors, process that input using an input–output function (its neural architecture), and provide an output in the form of an answer to the task at hand. In the blue and red balls problem described above, the search problem would be to find the set input–output function that translates the sensory input (i.e. a picture of a blue or red ball) into the correct output (i.e. correctly identifying whether the ball was blue or red). But in the space of all possible sets of input–output functions, how can a learning system improve its performance?

A simple scenario of how learning could take place in this case would be to present the learning system with a set of pictures of blue and red balls (i.e. training set) and let the learning system know what the right answer for each picture is (i.e. supervised learning).

The learning system is evaluated with respect to whether it can provide the right output for every given picture. For example, by presenting a learning system with a red ball and letting it know that it is in fact a red ball, the input–output function of a learning system will change in a way that makes it possible to identify red balls. Given a sufficiently large number of blue and red pictures, the learning system can potentially find an input–output function that correctly classifies the pictures it has seen into the blue and red classes. It is in this way that learning can be interpreted as a search problem – by searching over the set of possible input–output functions to find the one that improves its performance.

2.7.3 The Ability to Generalise

Further than that, the central aim of a learning system is to generalise. That is, not to only learn or memorise the training set, but to learn the regularities (i.e. general characteristics/features) that exist in the training set in order to be able to perform well on the test set (i.e. input that is not identical to the input the learning system was trained on). This is a much bigger problem than rote-learning (forming a memory). In learning systems, such generalisation ability is neither mysterious nor for granted. It is not really about the past or the future, but about generalising from the data you have seen to the test cases you have not.

Watson et al. [19] make the connection between the ability of a learning system to generalise and the ability an organism to adapt to novel environments and conditions. Later work by Kouvaris et al. [69] expands on that idea by highlighting the similarities between evolution and learning and showing that the transfer of knowledge between the two fields can help resolve the tension between robustness and evolvability in evolution and improve our understanding of conditions that facilitate the evolution of evolvability (also see Watson et al. [66] and chapter 5). In the following section, we summarise the findings of this body of work and provide evidence that evolution is a learning process.

2.7.4 Evolution Is a Learning Process

Here, we summarise the analogy and similarities between learning and evolution. We do so by showing the connection between the terminology used in the two fields (figure 3) and reporting on work or literature that supports that evolution is a learning process (figure 4) in table format:

SUMMARY

In this section, we presented a literature review on evolvability and related phenomena, along with highlighting the connection between evolvability and learning. We focused on the distinction between variation and variability, and explained how the developmental process may shape the variability structure and thus affect evolvability. We discussed whether robustness and evolvability are antagonistic forces and whether evolvability is evolvable. Lastly, we discussed the idea that evolution may be a learning process which will be the central focus of our work presented in subsequent chapters.

To summarise, evolvability is the capacity of organisms for adaptive phenotypic variation. Evolvability is primarily a property of the G–P map since it is the G–P map that is

Figure 3: Analogy between learning and evolution. Connecting the terminologies of the two fields. Adapted from personal communication with Richard A. Watson.

Learning Theory (Neural Networks):	Evolutionary Theory (e.g. gene-regulation networks):
Training set	Past selective environments/pressures (including current)
Test set	Future selective environments from the same class as past ones (e.g. modularly varying) that have not yet been seen
Hypothesis (set of weights)	Genotype (e.g. regulatory interactions, genes/mechanisms that regulate, etc.)
Hypothesis space	Genotype space
Input–Output function	Genotype–Phenotype map
Error minimisation	Fitness maximisation
Input	Embryonic phenotype (when development, and hence regulation, starts)
Actual output	Adult phenotype (when development, and hence regulation, ends)
Desired output	Target phenotype, or a local optimum after evolution follows fitness gradients in the fitness landscape
Neural architecture	Developmental architecture
External input	Environmental input/noise/perturbation
Activation dynamics	Developmental dynamics
Learning dynamics (search for good hypotheses)	Evolution of development (search for good phenotypes)
Memory (as attractor)	Adult phenotype (as attractor) [86]
Regularisation	Regulatory interactions cost; direct selective pressure on the number/weights of regulatory interactions
Error-correction and pattern completion in associative memories	Robustness; forming a memory of past phenotypes and recalling them given partial/noisy/incomplete embryonic phenotypes
What comes easy in machine learning: achieving good performance on the training set	What comes easy in evolution: finding good phenotypes for the present environment
The difficult part of machine learning: achieving good performance on the test set	The difficult part of evolution: finding good phenotypes in new selective environments or under different selective pressures (evolvability)

Figure 4: Predictions from learning theory. Predictions demonstrated in published work, in this report or discussed in this report but not yet demonstrated. Adapted from personal communication with Richard A. Watson.

Learning system (machine learning):	Developmental system (e.g. gene-regulation network):
'Neurons that fire together, wire together' [67]	'Genes that are selected together, regulate together' (Watson et al. [19] and chapter 4)
Associative memory: the ability to store and recall multiple patterns from the training set. The synaptic connections of the neural network are responsible for the encoding of the memory, whereas the activation dynamics of the neural network are responsible for the retrieval of stored memories [87].	Developmental memory: The ability to store and recall multiple phenotypic patterns from past selective environments/pressures. The regulatory interactions of the gene-regulation network are responsible for the encoding of the memory, whereas the developmental process (i.e. developing an embryonic phenotype into its adult form) is responsible for the retrieval of stored phenotypic patterns (Watson et al. [19] and chapter 4).
Pattern completion and error correction: the ability to retrieve memory from partial, corrupted or noisy sensory input.	The ability to retrieve full phenotypic patterns from partial, corrupted or noisy embryonic patterns (Watson et al. [19] and chapter 4).
Categorical perception: the ability to separate sensory inputs into discrete classes.	Switch-like phenotypic response to genetic changes (Watson et al. [19] and chapter 4).
Generalisation: the ability to produce an appopriate response to novel conditions	The ability to produce fit phenotypes in novel environments (not simply through recalling a stored memory) (Kouvaris et al. [69] and chapter 5)
Overfitting: achieving good performance on the training set at the expense of generalisation (i.e. good performance on the test set)	Robustness versus evolvability paradox: the myopic nature of selection favours what is currently fit (robustness) as opposed to what may be fit in the future, thus sacrificing the ability to anticipate future environments (evolvability) (Kouvaris et al. [69] and section 2.5)
Conditions that prevent overfitting	Conditions that facilitate/enhance evolvability (Kouvaris et al. [69] and section 6.4)
Becoming better at solving constraint satisfaction problems	Becoming better at evolving on highly-multimodal fitness landscapes composed of low-order epistatic interactions (Kounios et al. [70] and chapter 5)

responsible for translating a genotype to a phenotype. However, that does not consider how development itself evolves and therefore how evolvability itself evolves. When taking into account that the developmental process is encoded in the genotype and is therefore subject to natural selection, a self-referentiality is introduced in the Darwinian machine. Essentially, by changing the developmental process, the process of phenotypic variation is changed. In other words, the evolutionary process depends on the phenotype itself. Recent work shows that it is possible to improve our understanding of evolvability by showing that evolution itself is a learning process. We build on this work in subsequent chapters to show how evolution can learn to evolve better.

3 EVOLUTION WITHOUT LEARNING

ABSTRACT

The crossover operator in a genetic algorithm has been used to solve certain building-block problems that mutation-only methods cannot solve. However, it is well understood that this advantage of crossover is lost when optimising random-linkage problems, where the bits of each module are arbitrarily placed on the genome. Conventional crossover is therefore not applicable in engineering domains where the epistatic linkage map of a problem is not known *a priori*. This is because conventional crossover, such as uniform or one-point, does not respect the modular structure of a problem if epistatically dependent loci are not close together on the genome, thus preventing modules from being selected as a heritable units during crossover events. This motivates a large body of work on algorithms that aim to identify and exploit the linkage structure of a problem. Here, we introduce discrete differential evolution, a differential evolution variant converted for use on binary problems, that uses differences between evolved individuals to direct genetic variation. Differences between individuals provide information about subsets of loci that are epistatically dependent. These differences are super-imposed on individuals to introduce modular variation. Unlike prior work, this is achieved without explicit learning mechanisms that identify linkage information; rather it is possible because the macro-variation operator is able to exploit the modular structure of the solution space that is revealed implicitly by the distribution of individuals in the population. We show, with numerical simulation, that dDE is able to solve a number of random-linkage building-block problems in polynomial time.

3.1 INTRODUCTION

A large body of work in the genetic algorithm (GA) literature has been focused on investigating the ability or inability of crossover to exploit the principle of problem decomposition in problems with modular or building-block structure [88–92]. A polynomial versus exponential time complexity distinction has only been shown using tight-linkage crossover [89, 92]. However, the tight-linkage assumption (i.e. that epistatically dependent loci within a module are close together on the genome, enabling modules to be selected as a heritable units during crossover events, as per the building-block hypothesis [93–95]) compromises the applicability of this work.

In design-engineering and optimisation, the underlying structure of a problem is not generally available, meaning that there is no information about the correct ordering of the variables which would facilitate problem decomposition and effective crossover. This led to work that abandons building-block ideas altogether and utilises uniform crossover which does not assume tight-linkage and does not attempt to select on particular epistatically dependent subsets of loci. Another body of work was on linkage learning and estimation of distribution algorithms [58, 59]. Work on linkage learning attempts to overcome

this limitation by automatically identifying and exploiting problem structure, regardless of where across the genome epistatically dependent loci are situated (e.g. [58, 59, 61, 62, 96, 97]). This approach has been successful on practical problems in many domains. The idea that the GA is capable of implicitly exploiting problem structure is discarded, and, instead, this class of algorithms utilises machine learning techniques to explicitly learn and exploit the structure of the problem.

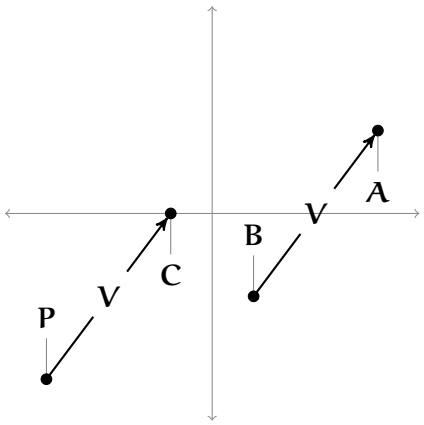
Here, we introduce discrete differential evolution (dDE), a differential evolution (DE) [98–103] variant converted for use on binary problems, that uses differences between evolved individuals to direct genetic variation. Differences between individuals provide information about subsets of loci that are epistatically dependent. These differences are super-imposed on individuals to introduce modular variation. Unlike prior work, this is achieved without explicit learning mechanisms that identify linkage information; rather it is possible because the macro-variation operator is able identify and exploit the modular structure of the solution space that is revealed implicitly by the distribution of individuals in the population.

Although DE is not naturally applicable to discrete spaces, its application has not been limited to that of continuous spaces, having been successfully applied to image processing, combinatorial optimisation, permutation and constraint satisfaction problems [104–109]. However, because the DE variation operator controls both the magnitude and the direction of genetic changes, it seems that the algorithm is unsuitable for use in binary optimisation problems, including those used to investigate the principles of building-block recombination in the GA [88–92].

Nonetheless, here we show that, in binary problem spaces, constructing a difference vector by taking the difference of two individuals, and super-imposing the differences onto another individual is sufficient for identifying epistatically dependent subsets of loci and crossing over coherent units. In continuous spaces, the emphasis of the DE variation mechanism is on how much and in which direction each variable should be changed. In binary spaces, where the magnitude of any change can only be 1, the principle of variation in DE can nonetheless do useful work by shifting the emphasis onto identifying which subset of variables should be changed simultaneously. That is, each locus in the difference vector takes either -1, 0, or 1. This determines which loci change and which do not, and whether they change from 0 to 1 or from 1 to 0. Thus, the magnitude of any change is fixed, but only the loci with a non-zero difference can affect the parent. Simply put, the difference of two individuals identifies a schema that is super-imposed onto another individual (figure 5).

In the remainder of this paper, we explore the usefulness of this simple idea and test how useful the principle of directed variation provided by DE can be in exploiting problem structure in binary problems. We utilise three distinct problems: the first is separable, the second is nearly-decomposable and the third is hierarchical; all of which are pathologically difficult for techniques relying on uninformed variation. Although the tight-linkage versions of these problems are solvable by a GA with one- or two-point crossover in polynomial time, the random-linkage or 'shuffled' versions are just as difficult for the GA (with any type of crossover) as they are for the uninformed methods. Here we show dDE can solve the problems reliably in polynomial time.

The rest of the paper is structured as follows. In section 3.2 we introduce dDE, and in section 3.3 we discuss an idealised scenario for problem decomposition and define the test problems. In section 3.4 we present the results of numerical simulations, including a time complexity comparison between dDE and uninformed search techniques. In section 3.5,



(a)

Vector	Bit-string
A	0010010010010
B	1010110000101
V	-000-000+0-+-
S	0 0 1 010
P	1001010100110
C	0001010110010

(b)

Figure 5: The principle of variation in differential evolution in continuous and binary spaces. (a) In continuous spaces of two variables (x and y), the differences between two individuals, **A** and **B**, selected at random from the population are used to generate a difference vector, **V**, that is then applied to a parent individual, **P**, to create a child individual, **C**. (b) The principle of differential evolution in a binary bit-string, introduced here, defines a tertiary (-1 shown as $-$, 0, $+1$ shown as $+$) difference vector, **V**, or, for clarity, an intermediate schema, **S**, that is super-imposed on a parent individual, **P**. This forces some bits of the parent to take the 0 allele, others to take the 1 allele, and the loci in the parent that already have the allele forced by the difference schema are left unchanged. To aid visualisation, the bits of the child that differ from its parent are shown in colour.

we discuss multiple aspects of our work, including why the problems we use for testing are difficult to solve, and what it means for dDE to 'learn'.

3.2 DISCRETE DIFFERENTIAL EVOLUTION: MODULAR VARIATION WITHOUT LEARNING

Here, we define the dDE algorithm and describe its key functions. Similar to DE, the algorithm is simple to understand and straightforward to implement. Much of its simplicity stems from the fact that no explicit learning takes place. Unlike previous work, we discard the idea of explicit linkage learning and revisit the abandoned idea of *implicitly* scaling search to search in the space of module solutions, or building blocks, as per the building block hypothesis [93–95]. In this paper, we investigate the ability of the DE algorithmic principles to globally optimise discrete, random-linkage, modular problems.

The exploration mechanism in dDE is a simple population of hill-climbers. To utilise the information the hill-climbers discover and achieve macro-variation, other algorithms require an intermediate step: the explicit learning of the epistatic linkage map of the problem, using techniques such as Bayesian networks [56, 57, 96] or Hebbian learning [67] and correlation matrices [62]. Our approach differs in that there is no intermediate learning step; macro-variation is achieved implicitly. This is possible because the macro-variation operator is able to identify and exploit the underlying modular structure of the solution space that is represented implicitly by the distribution of individuals in the population. The following is a high-level outline of the algorithm: Note the lack of intermediate learning and the two mutually exclusive modes of operation: bit-flip and macro-variation. The

bit-flip operator is used to locally optimise genotypes using the original search neighbourhood (i.e. Hamming space), whereas the macro-variation operator utilises information within the population when genotypes are locally optimized to produce targeted, simultaneous variable changes. By probabilistically switching between the two modes of operation, the 'greedy' selection scheme discards individuals when the individuals in the population (a) are not locally optimal, thus the macro-variation operator produces random rather than targeted variable changes; or (b) are locally optimal, thus the bit-flip operator only produces deleterious mutations. Consequently, the probabilistic switch between the two modes of operation in combination with the unique nature of the dDE macro-variation operator results in the implicit scaling of the search.

Overall Procedure

dDE relies on two control parameters: the population size n_{pop} and the probability of performing macro-variation, p_{mv}. The population is initialised to discrete values, and each generation consists of the exploration mechanism (bit-flip mutation) or the exploitation mechanism (differential crossover), and selection. The overall procedure is shown in pseudo-code format in figure 6. What follows is a detailed description of the stages of the algorithm.

Initialisation

Individuals in dDE must be initialised to discrete values. The operators used are insensitive to the choice of variable allocation conventions. That is, the population can be initialised to follow either evolutionary computation ($x \in \{0, 1\}$) or Ising spin conventions ($\sigma \in \{-1, +1\}$).

Bit-flip mutation

For the exploration aspect of higher-scale search, dDE uses a simple bit-flip mutation operator. The operator randomly selects a variable, $j = [1, n]$, in all n_{pop} individuals and flips it. Note that the operator could be modified to flip more variables per individual, but that is unnecessary because its objective is to identify module solutions; not to produce macro-variation.

Differential crossover

Central to the scaling of the search in dDE is the idea of taking the difference between two individuals and adding it to another individual – a process called differential mutation in the DE literature. Here, we show that, in the discrete rather than continuous space, adding the difference of two locally optimal configurations to another locally optimal configuration is sufficient for producing macro-variation, with the resulting configuration taking the variables required to produce a jump between local optima. By using the operator directly on a population of locally optimal configurations, the epistatic linkage map of the problem is emphasised by the differences between the individuals in the population, thus removing the need for explicit learning mechanisms that identify correlations between variables.

1: $\mathbf{X} \leftarrow$ Randomly initialise n_{pop} individuals to discrete values.
2: $\boldsymbol{w}^{\mathbf{X}} \leftarrow$ Evaluate the n_{pop} individuals in $\mathbf{X}$.
3: **while** ending criterion is not satisfied **do**
4: **if** $\text{rand}(0,1) \leqslant p_{mv}$ **then** ▷ Perform differential crossover.
5: **for** $i \leftarrow 1$ to n_{pop} **do**
6: $r_1^i \leftarrow \text{randi}(1, n_{pop})$ ▷ $r_1^i \neq i$
7: $r_2^i \leftarrow \text{randi}(1, n_{pop})$ ▷ $r_2^i \neq r_1^i \neq i$
8: $\mathbf{u}_i \leftarrow \gamma(\mathbf{x}_i + \mathbf{x}_{r_1^i} - \mathbf{x}_{r_2^i})$ ▷ Equation 1.
9: **end for**
10: **else** ▷ Perform bit-flip.
11: **for** $i \leftarrow 1$ to n_{pop} **do**
12: $j \leftarrow \text{randi}(1, n)$ ▷ Variable to flip.
13: $\mathbf{u}_i \leftarrow \mathbf{x}_i$ ▷ Copy the individual.
14: Flip u_{ij} ▷ Flip the variable.
15: **end for**
16: **end if**
17: $\boldsymbol{w}^{\mathbf{U}} \leftarrow$ Evaluate the n_{pop} individuals in $\mathbf{U}$.
18: **for** $i \leftarrow 1$ to n_{pop} **do** ▷ Selection.
19: **if** $w_i^{\mathbf{U}} \geqslant w_i^{\mathbf{X}}$ **then**
20: $\mathbf{x}_i \leftarrow \mathbf{u}_i$
21: $w_i^{\mathbf{X}} \leftarrow w_i^{\mathbf{U}}$
22: **end if**
23: **end for**
24: Increase generation count.
25: **end while**

Figure 6: Pseudo-code for dDE. n_{pop} is the population size; n is the problem size; $\text{rand}(x, y)$ is a uniform floating-point random number generator in the range $[x, y]$; and $\text{randi}(x, y)$ is a uniform integer random number generator in the range $[x, y]$.

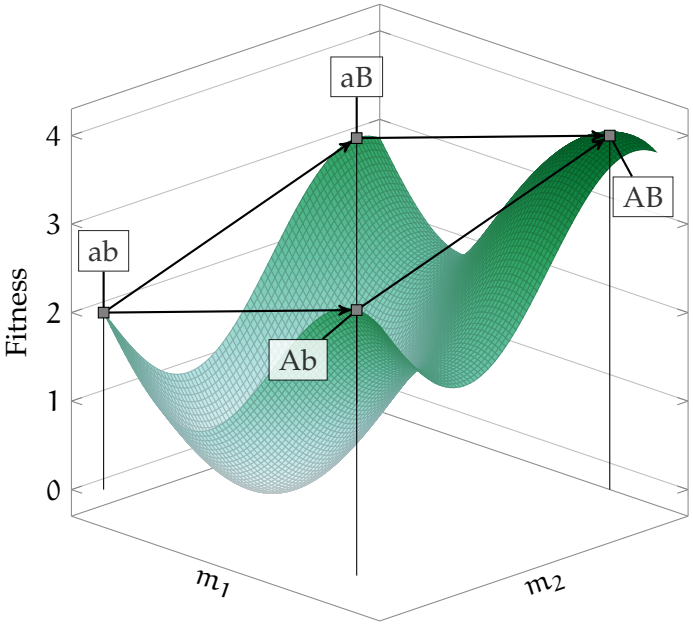


Figure 7: How differential crossover produces the variation required to jump between optima. In a population that includes individuals ab, Ab, and aB, the differences between ab and Ab identify the crossover points and allelic values that aB needs to take to jump directly to AB, thereby avoiding the intermediate fitness valley that includes sites of lower fitness.

Note that this operator does not introduce variation by flipping; instead, it identifies the crossover points and the allelic values required to substitute one module solution with another. Consider a problem with two modules with two module solutions each: m_1 with a and A and m_2 with b and B. Also assume that there are three individuals within the population: (1) ab; (2) Ab; and (3) aB. Say we take the difference between (1) and (2). The non-zero values in the difference vector are the crossover points. That is, (1) and (2) have the same module solution in module m_2, but not in m_1; (1) has a and (2) has A. The differences between module solutions a and A are also identified in the difference vector. Thus, the non-zero values in the difference vector are both the crossover points and the alleles (3) should take to move from a to A or stay at a (see figure 7). Because the value at each locus is inherited rather than flipped, we call this operator *differential crossover*.

Differential crossover is essentially the differential mutation operator, but bound by the constraints of search in binary spaces, as shown below:

$$\boldsymbol{\nu}_i = \gamma(\mathbf{x}_i + \mathbf{x}_{r_1^i} - \mathbf{x}_{r_2^i}), \tag{1}$$

where γ is a threshold function resetting the variables to legal binary values, $r = [1, n_{pop}]$ and $r_1^i \neq r_2^i \neq i$. The threshold function, γ, simulates the effect of super-imposing the non-zero values of the difference vector onto $\mathbf{x}_i$.

Selection

There are no differences between the DE and dDE selection stages: the ith trial vector, $\mathbf{u}_i$, takes the place of the ith target vector, $\mathbf{x}_i$, in the next generation if-and-only-if it is as fit or fitter. However, there are a few key aspects of it that enable the probabilistic switch between the bit-flip and differential crossover operators. First, because the fitness of each ith pair from the target and trial vectors populations is compared, by preceding

the selection stage with a bit-flip operator, the overall procedure is identical to that of a population of hill-climbers – meaning that each individual is likely to be exploring a different area of the fitness landscape which is a vital prerequisite for the success of the differential crossover operator. Second, the 'greedy' and steady-state replacement scheme discards individuals when the operators are not effective. That is true when bit-flip is performed on locally optimal configurations or when differential crossover is performed on non-locally optimal configurations.

3.3 MULTI-SCALE MODULAR PROBLEMS

Building-block recombination familiar in the GA literature introduces the idea of scaling up the search process from searching combinations of bits to searching combinations of building blocks [93–95]. This can be generalised to modular problems and searching combinations of module solutions. We define *module* to be a subset of variables with strong inter-dependencies and *module solution* as a specific configuration of a module whose fitness cannot be improved by any single-variable change. But contrary to the definition of building blocks, modules are not necessarily (a) of low order, nor is their order bounded by fixed values; or (b) of short defining length – i.e. co-dependent variables can be arbitrarily placed across the genome. The idea that a module solution is of particularly high fitness is retained. We can think about how searching combinations of module solutions facilitates effective search by thinking about how it transforms the neighborhood of the search space.

For any search process to be able to exploit multi-scale problems, an adaptive mechanism is required that enables a transformation in the fitness landscape neighbourhood from searching in the space of order-1 schemata to searching in the space of functional components of order-k, $k > 1$. In order for the process to be fruitful, the order-k schemata must form cohesive units which reflect the structure of the problem. Hence, the efficiency of such a mechanism will be at its maximum when the optimisation problem exhibits modular structure [92, 110] that will allow it to substitute one module solution, or sub-solution, with another.

We define *macro-variation* to be variation that enables the substitution of module solutions and *macro-variation operator* as the mechanism that enables such variation. With the use of a macro-variation mechanism, search can escape solutions that would be locally optimal under conventional variation. Consider a problem with two modules with two module solutions, or local optima, each: A and a for m_1, and B and b for m_2. Additionally, we assume that the Hamming distance between the two module solutions, d, is large, meaning that a large number of simultaneous variable changes are required to jump between A and a or B and b. Since all combinations of module solutions are locally optimal and sit several variables apart, single-variable variation would result in a decrease in fitness. When the module solutions have been successfully identified, a macro-variation operator can enable the direct substitution of, for example, a for A, thereby avoiding the intermediate fitness valley which includes sites of lower fitness (assuming A is of higher fitness). Essentially, the macro-variation operator transforms the genotypic neighbourhood from one where neighbouring genotypes differ by a single bit to one where they differ by a module solution.

A number of conclusions can be reached from the above. In the modified genotypic neighbourhood resulting from an appropriate macro-variation operator, the substitution of one module solution for another corresponds to the transition between locally optimal configurations. The macro-variation operator produces multiple, neighbouring locally optimal configurations that can be compared by means of selection. Thus, the dynamics of the search process change from hill climbing in the space of single-variable changes to hill climbing in the space of module solutions. A hill climber in the original neighbourhood is only capable of finding the global optimum when search is fortunate enough to start at a genotype in its basin of attraction. Obviously, in a small problem, such as a two-module problem, a random-restart hill-climber will only require a few restarts. However, the number of local optima increases exponentially with the number of multi-modal modules, and consequently starting in the basin of attraction that optimally solves all sub-problems would require an exponentially growing amount of restarts [92].

A number of synthetic problems have successfully illustrated the idea of modularity and building blocks (e.g. [96, 111, 112]). These problems can, in general, be 'shuffled' to create a problem class of random-linkage problems whose structure cannot be exploited by linkage-preserving crossover techniques, such as one- or two-point crossover. Despite the fact that these problems have successfully shown what modularity-exploiting algorithms are good for, they all hold the assumption that the number of bits contained per module, or the number of modules contained in modules of later hierarchical levels, has to be small and fixed. This assumption can perhaps be attributed to the following: given that a hill-climber can solve a module and that the modules are separable, a hill-climber can solve all modules and hence the entire problem. On the other hand, given that a hill-climber cannot easily solve the modules, the solutions will have to be 'guessed'. Therefore, the required search to find the solution to a module would need to be exponential in the size of a module; so the modules have to be small. However, this reasoning omits the possibility that local search may be able to assist in finding solutions to modules but it cannot reliably find the solutions to modules [92]. Additionally, it does not allow previous work to show a polynomial-vs-exponential distinction between multi-scale and conventional search algorithms [92].

Although, in principle, searching with an appropriate macro-variation operator transforms the neighbourhood of the fitness landscape from one where the locally optimal configurations were separated by a number of lower-fitness intermediate sites to one where they are one variational step away, for this transformation to be successful, the automatic and reliable discovery of module solutions is essential. Therefore, the combination of an exploration mechanism that searches *for* module solutions and an exploitation mechanism that searches *in the space* of module solutions can significantly reduce the search space and the difficulty of problems that facilitate decomposition.

3.3.1 Scalable Building Blocks: A Decomposable Problem

The scalable building blocks (SBB) problem consists of m multi-peaked, order-k modules, with each peak having an appreciable basin of attraction. The full n-bit problem is constructed by concatenating, without between-module epistasis, all modules:

$$w(\mathbf{g}) = \sum_{i=1}^{m} f(b_i),$$

where $\mathbf{g} = \langle g_{r_1}, g_{r_2}, \ldots, g_{r_n} \rangle$ is the genotype of length n; $\mathbf{r} = \langle r_1, r_2, \ldots, r_n \rangle$ is a mapping between the variables and the randomised functional structure of the problem; $\mathbf{b}_i = \langle g_{r_{(i-1)k+1}}, g_{r_{(i-1)k+2}}, \ldots, g_{r_{ik}} \rangle$ is the ith module of the genotype containing k variables; $m = n/k$ is the number of modules; and f is the sub-function that determines the fitness contribution of each module:

$$f(\mathbf{b}_i) = \sum_{j=1}^{T_i} c(\mathbf{b}_i, \mathbf{t}_{ij}), \text{ where} \tag{2}$$

$$c(\mathbf{b}_i, \mathbf{t}_{ij}) = \begin{cases} w_{ij} & \text{if } |\mathbf{b}_i - \mathbf{t}_{ij}| = 0 \\ 1 + |\mathbf{b}_i - \mathbf{t}_{ij}|)^{-1} & \text{otherwise,} \end{cases} \tag{3}$$

where T_i is the number of optima within the ith module; $\mathbf{t}_{ij}$ is the jth target string (i.e. optimum) within the ith module; $|\mathbf{b}_i - \mathbf{t}_{ij}|$ is the Hamming distance between the module configuration and the target string; and $w_{ij} > 1$ is the fitness contribution, or weighting, of the target string $\mathbf{t}_{ij}$.

The fitness of each module is a sum of the fitness contributions, as defined by c, determined by the comparison of the module configuration with the set of target strings, $\mathbf{T}$. Maximal fitness contribution with regard to a particular target string can only be achieved when the module configuration matches the target string (i.e. $|\mathbf{b}_i - \mathbf{t}_{ij}| = 0$); otherwise, the fitness contribution is inversely proportional to the Hamming distance.

3.3.2 Variable Structural Modularity: A Nearly-Decomposable Problem

The second problem we use for testing is the variable structural modularity (VSM) problem [60, 62]. The test problem is defined by a randomly initialised, symmetric weight matrix, which results in a multi-modal, nearly-decomposable problem [113] of pair-wise constraints. There are two classes of the VSM problem: *random constraints* (RC), without explicit modularity; and *modular constraints* (MC) which is a macro-scale version of the RC class. Since we want to show an idealised scenario where an algorithm that can discover and exploit the epistatic linkage map of the problem can reliably solve it, we use the MC class.

The RC matrix is defined by a $m \times m$ symmetric matrix, R, whose diagonal elements all take the value $r_{ij} = d_i = 1$ if $i = j$, and the rest are uniformly initialised to fixed-magnitude, variable-sign values, $r_{ij} \in \{-d_e, d_e\}$:

$$r_{ij} = \begin{cases} d_i & \text{if } i = j, \\ \{-d_e, d_e\} \text{ with equal probability} & \text{otherwise,} \end{cases}$$

where d_i is the self-weight, and $d_e \leqslant d_i$ is the magnitude of the interaction between variables. The MC matrix is defined in terms of the RC matrix, which yields a symmetric, explicitly modular matrix, Ω, of size $n \times n$, where $n = mk$; m is the number of modules; and k is the length of each module. The MC weight matrix is initialised as follows:

$$\omega_{ij} = R\left(\left\lfloor \frac{i}{k} \right\rfloor, \left\lfloor \frac{j}{k} \right\rfloor\right).$$

The problems utilise state configurations represented in the Ising spin convention, $\sigma \in \{-1, +1\}$, $\boldsymbol{\sigma} = \langle \sigma_0, \sigma_1, \ldots, \sigma_{n-1} \rangle$, to calculate the fitness, which is given by:

$$w(\mathbf{g}) = -\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} g_i \omega_{ij} g_j, \text{ or} \tag{4}$$

$$w(\mathbf{g}) = -(\mathbf{g}\boldsymbol{\Omega}\mathbf{g}^T), \tag{5}$$

where $\mathbf{g}$ is the genotype, represented as a $1 \times n$ row vector, and $\mathbf{g}^T$ is its transpose. This creates a minimisation problem of weighted, pair-wise constraints, where the fitness contribution is beneficial only when the sign of the product of two variables (g_i and g_j) is the same as the sign of their weighting, ω_{ij} (i.e. when the constraint is satisfied).

3.3.3 hTrap: A Deceptive Hierarchical Problem

Hierarchical traps (hTrap) [114] is a combination of trap functions and a mapping that forms subsequent hierarchical levels. The length of the trap defines the overall problem size: $n = k^m$, where n is the overall problem size, $k \geqslant 3$ is the length of the trap, and m is the number of hierarchical levels.

At the lowest level, blocks of k bits contribute to the overall fitness:

$$\text{trap}_k(u) = \begin{cases} f_{\text{high}} & \text{if } u = k \\ f_{\text{low}} - u\frac{f_{\text{low}}}{k-1} & \text{otherwise,} \end{cases} \tag{6}$$

where u is the number of ones in the block, $f_{\text{high}} = 1$ and $f_{\text{low}} = 1 + 0.1/m$. For this choice of f_{high} and f_{low}, the global optimum of each module solution is a string of all zeroes.

Each k-bit configuration that is a solution to one of the traps is then mapped to a symbol on the subsequent level: 0^k is mapped to a 0; 1^k is mapped to a 1; and any other configuration is mapped to the null symbol, '-'. This mapping defines the bit-string at the following hierarchical level. The k-bit groupings formed at the subsequent level contribute to the fitness using the traps equation defined above only if they do not contain a null symbol.

For all levels, the fitness contribution is multiplied by k^i, where i is the current level (with $i = 1$ being the bottom level). This guarantees that the magnitude of the fitness contributions is the same across all levels. The process is repeated until the top level consisting of k bits is reached. The fitness contribution values are different at the top level: $f_{\text{high}} = 1$ and $f_{\text{low}} = 0.9$. This particular choice of fitness contributions means that although the module solution of all zeroes confers higher fitness than all ones at every other level, the global optimum is the string of all ones.

3.4 RESULTS

In this section, we provide empirical results for the behaviour and performance of dDE. In section 3.4.1, we investigate the behaviour of dDE and the differential crossover macro-variation operator on fixed-size SBB, VSM and hTrap problems. In section 3.4.2, we show how the performance of dDE compares to a number of conventional algorithms as the problem size increases.

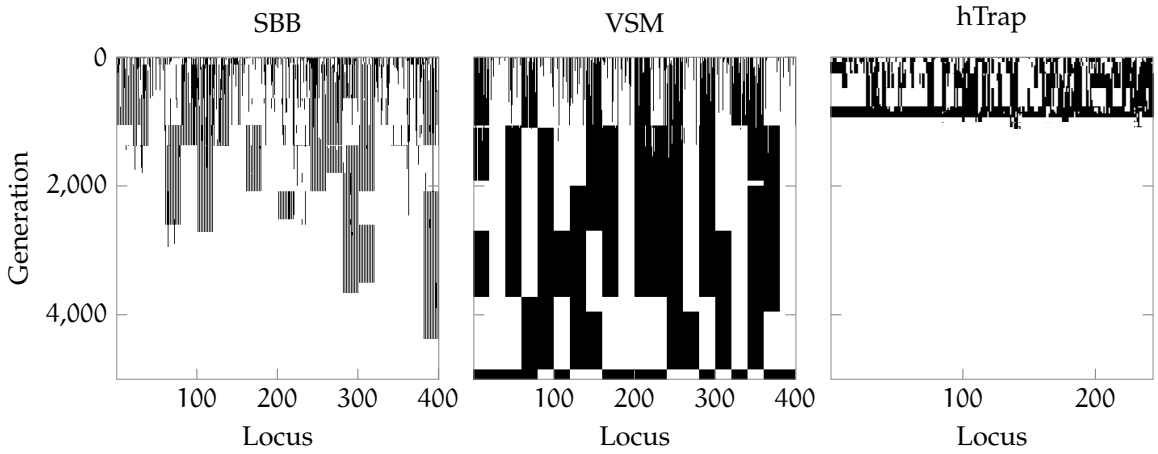


Figure 8: Genotype lineage diagrams for one individual of one example run of dDE on 400-bit SBB and VSM problems, with $m = k = 20$, and on a 243-bit hTrap problem, with $k = 3$, $m = 5$. dDE is capable of exploiting the problem structure and producing targeted, simultaneous variable changes. Initially, single-variable changes direct the search to locally optimal configurations until the exploration requirement of multi-scale search is satisfied. When there is information to exploit within the population (i.e. when individuals are locally optimal), differential crossover produces macro-variation as observed by the substitution of modules. The global optimum is found in all three instances.

3.4.1 Operation on Test Problems

Here, we investigate the behaviour of dDE on 400-bit SBB and VSM problems, with $m = 20$ modules of length $k = 20$ each ($n = mk$), and on a 243-bit Trap problems with traps of length $k = 3$ and $m = 5$ hierarchical levels ($n = k^m$). For the SBB problem, we use 2 optima per module: the globally optimal target string consists of all 1s and the locally optimal target string of 1s and -1s at every odd and even index respectively. The Hamming distance between the optima is 10 and the fitness weighting for the global and local optima is $w_{go} = 10$ and $w_{lo} = 1$ respectively. For the VSM problem, we set the within-module constraints to $d_i = 1$ and the between-module constraints to $d_e = 0.01$. For the hTrap problem, we set the trap length to $k = 3$, the fitness contribution of the all-ones solution to $f_{high} = 1$ and the all-zeroes fitness contribution to $f_{low} = 1 + 0.1/m$. At the top level, the all-ones and all-zeroes fitness contributions are changed to $f_{high} = 1$ and $f_{low} = 0.9$ respectively. This means that the globally optimal solution is an all-ones string. For all problems, the population size was set to $n_{pop} = 200$. We set p_{mv} to 0.2 for the SBB problem, and to 0.1 for the VSM and hTrap problems.

Figure 8 shows a genotype lineage diagram for one individual across 5000 generations of one example run of dDE on the three problems. The black and white colours translate to variables with the values -1 and $+1$ respectively. As can be seen from the single-variable changes, the optimisation process spends the initial stages searching for locally optimal configurations to identify module solutions and satisfy the exploration aspect of multi-scale search. When there are locally optimal configurations in the population, the search is transformed to a higher scale, with the differential crossover operator producing targeted, simultaneous variable changes. Effectively, the neighbourhood of the genotypes in the fitness landscape changes from one where the neighbouring genotypes differ by a single variable to one where they differ by a module solution. But unlike previous work utilising explicit learning mechanisms, searching in the space of module solutions is

achieved implicitly, by using differential crossover directly on the population to introduce variation.

There are a couple of things to note here. First: dDE is capable of solving decomposable (SBB), nearly-decomposable (VSM) and hierarchical (hTrap) problems. Second: the variables within a module are shown as contiguous blocks (i.e. tightly linked), but that is only done to show a better picture of the behaviour of the algorithm.

3.4.2 Time Complexity

Here, we examine empirically how the dDE scales with the problem size on the three test problems.

The problem parameters are as described in the previous section. The range of problem sizes tested was $m = k = [5, 20]$ for the SBB and VSM problems, and $m = [2, 5]$ for the hTrap problem. For the VSM problem, the weight matrix was randomised every execution (the algorithms are not tested on the same set of weight matrices). All locally optimal state configurations were enumerated and evaluated to identify the globally optimal configuration for each randomly generated weight matrix.

We ran dDE on 100 independent executions of each problem size and measured the mean number of function evaluations used before the global optimum was found. Population sizes were hand-chosen such that the solution probability criterion was satisfied: the global optimum was to be found in all 100 executions. We set the probability of performing macro-variation to $p_{mv} = 0.2$ for the SBB problem and $p_{mv} = 0.005$ for the VSM and hTrap problems.

For comparison, we also ran uninformed-variation algorithms on 30 independent executions of each problem size. We used a relaxed solution probability criterion for these algorithms: the global optimum was to be found in at least 27 out of 30 executions (90% probability) We only show data points for the search instances that met the criterion.

The uninformed search techniques we use are: uniform crossover evolution (UCE), memetic algorithm (MA) [115], and two restart hill-climbers (HC and HC′). UCE is used as a control experiment as it is identical to dDE in everything but the crossover method: dDE uses differential crossover and UCE uses uniform crossover. We hand-tune the population size. The MA is a hybrid uniform crossover GA and hill-climber. The premise is that although uniform crossover will not necessarily change all bits that would facilitate modular variation, the bits that do change may move an individual to a different basin of attraction, and hill-climbing from there will lead to a potentially better optimum. The control parameters were hand-tuned. Lastly, we use two restart hill-climbers. For the first one, HC, we set the mutation rate to $\mu = 1/n$, and restart the search every n consecutive generations without fitness improvement. We provide the second one, HC′, with problem-specific information. Specifically, we set the mutation rate, $\mu = d/n$, where d is the Hamming distance between the module solutions of each problem, and restart the search every $10n$ consecutive generations without fitness improvement.

Figure 9 shows the measured time complexities for the SBB problem. UCE, HC, and HC′ scale super-polynomially in n, as can be seen from the upwards curvature of the lines when plotted on log-log axes. Conversely, dDE has a distinctive advantage as its ability to search in the space of module solutions reduces the search space significantly. The time complexity for dDE is polynomial in n, as evident from the straight line on log-log axes. Note that the difficulty of the problem is relatively relaxed, with the Hamming distance

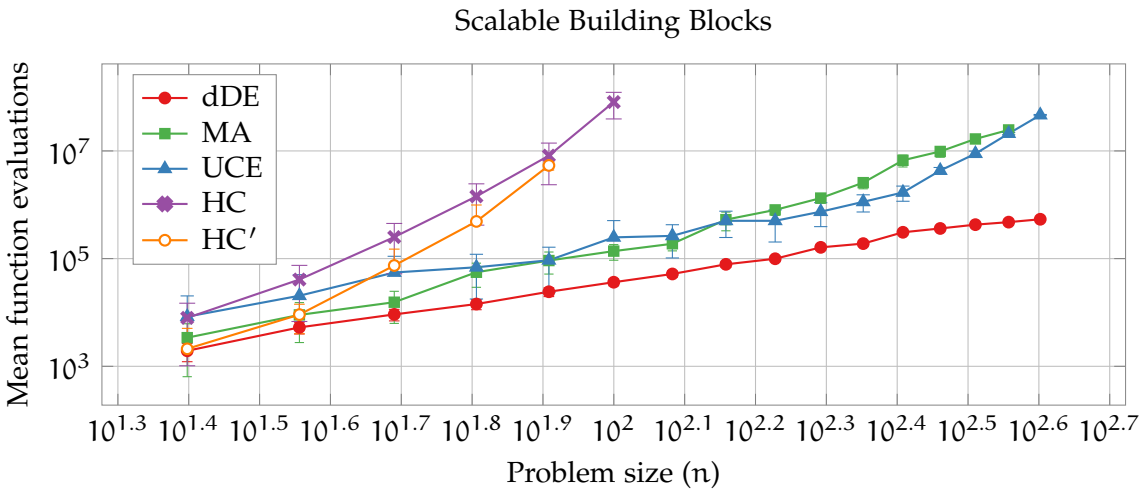


Figure 9: Mean number of function evaluations used by all algorithms to globally optimise the SBB problem. Data shown on log-log axes. The expected running time for uniform crossover evolution and the restart hill-climbers is super-polynomial in n, as evident from the upwards curvature of the line on log-log axes. Conversely, the expected running time of the MA and dDE is polynomial in n, with a straight line on log-log axes.

between optima not maximal at $\lfloor (k/2) \rfloor$. This increases the likelihood that uniform crossover retains modules from $\approx 2^{-2k}$ to $\approx 2^{-2\lfloor (k/2) \rfloor}$, and also increases the likelihood that HC′ produces the variation required to jump between optima. dDE is unaffected by the Hamming distance between optima because it uses informed methods to direct genetic variation.

Figure 10 shows the time complexities on the VSM problem. The results are qualitatively similar to those on the SBB problem despite the introduction of epistasis between the modules. UCE, HC and HC′ scale super-polynomially in n, as can be seen from the upwards curvature of the line when plotted on log-log axes. The performance of dDE is clearly superior because of its ability to identify and exploit the modular structure of the problem. Specifically, dDE scales polynomially in n, as evident from the straight curve when plotted on log-log axes. Note that the Hamming distance between optima is maximal at k, meaning that the probability that uniform crossover retains the linkage information and successfully substitutes a module solution is approximately $\approx 2^{-2k}$. As a result, the super-polynomial time complexity of UCE shows earlier than in the SBB problem, at $m = k = 11$.

Figure 11 shows the time complexities on the hTrap problem. dDE is capable of solving the hierarchical problem in polynomial time, with a time complexity of $\mathcal{O}(n^{1.9})$. Since we were unable to gather sufficient data for the uninformed-variation mechanisms, we show results from hBOA on the same problem instance as a comparison, adapted from the literature in Pelikan and Goldberg [116].

As the results show, dDE is capable of solving all three problems in polynomial time. Importantly, it does so without utilising any expensive learning mechanisms (see section 3.5). Through the simple idea of differential mutation, modular variation is achieved.

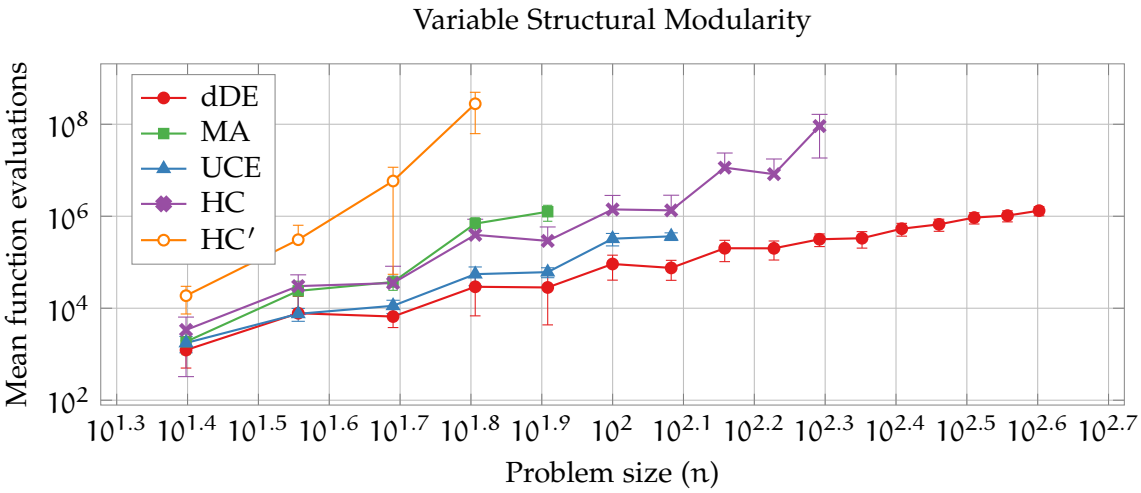


Figure 10: Mean number of function evaluations used by all algorithms to globally optimise the VSM problem. Data shown on log-log axes. The expected running time for uniform crossover evolution, the memetic algorithm, and the restart hill-climbers is super-polynomial in n, as evident from the upwards curvature of the line on log-log axes. Conversely, because dDE can exploit the modular structure of the problem and search in the space of module solutions, its expected running time is polynomial in n, with a straight line on log-log axes.

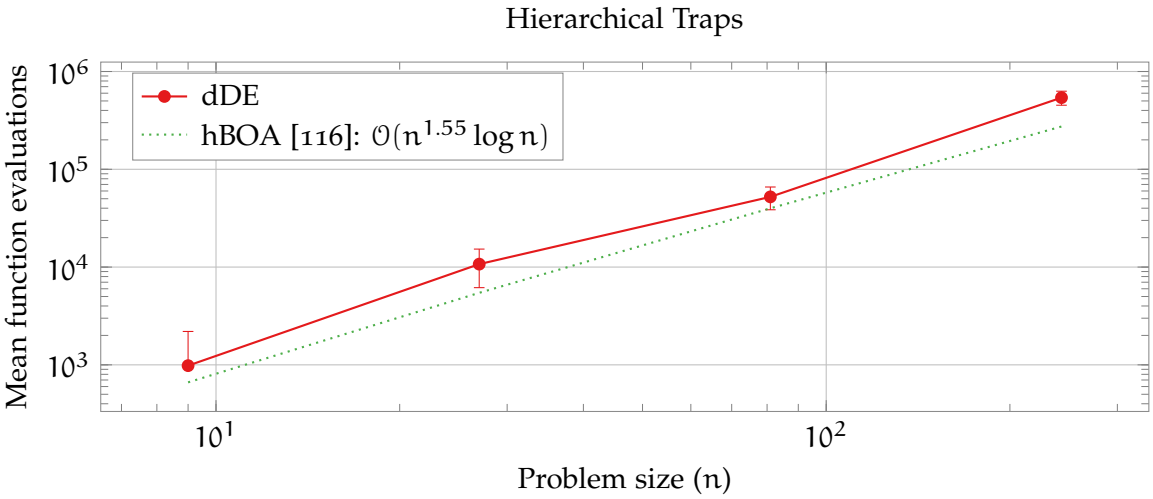


Figure 11: Mean number of function evaluations used to globally optimise the hTrap problem. Data shown on log-log axes. None of the uninformed-variation algorithms were capable of providing enough data on this problem. For comparison, we show data from hBOA optimizing the same instance of the problem, adapted from the literature in Pelikan and Goldberg [116]. dDE can exploit the modular structure of the problem and search in the space of module solutions at every hierarchical level, so its expected running time is polynomial in n, with a straight line on log-log axes. Surprisingly, despite its simplicity, the performance of dDE is similar to that of hBOA.

3.5 DISCUSSION

In this section, we discuss multiple aspects of our work, such as why the problems we used for testing are difficult, what it means for dDE to 'learn', and identify the limitations of our work.

3.5.1 Learning Correlations

Central to the successful scaling of the search is the process of exploring multiple areas of the fitness landscape to identify module solutions. To properly identify all module solutions without mistake (i.e. no false positive or negative correlations), a specific number of locally optimal configurations has to be sampled. In many of the current approaches, this number is controlled by the number of hill-climbing processes that are ran or with a learning rate control parameter [61, 62]. In the case of dDE, we do not explicitly set such a parameter because learning is not explicit. However, it is not difficult to see how the combination of the population size, n_{pop}, and probability of performing macro-variation, p_{mv}, control parameters is very similar to a learning rate.

Consider the following. During the hill-climbing process, alleles or module solutions at specific loci that confer high fitness will be selected. After these alleles or module solutions are discovered, the rate at which macro-variation is performed determines how quickly they are copied onto other individuals. To prevent early convergence, the population size must be sufficiently large with regard to the rate at which macro-variation is performed. That is, if the population size is too small or macro-variation is performed too often, diversity is lost quickly, meaning that the differences or similarities between the individuals in the population do not reflect the structure of the problem. This issue is very similar to that of tuning the learning rate value: setting it too high results in learned correlations that are not representative of the problem structure.

3.5.2 Future Work

In the problems we tested, each module had only $z = 2$ module solutions. Because the dDE macro-variation operator is a form of crossover, when more than 2 optima exist per module, a crossover event between three distinct module solutions is not guaranteed to produce another module solution. Despite this fact, dDE is capable of solving such problems for reasons we explain below.

Assuming that a macro-variation event produced a non-optimal module configuration, there are two possible outcomes: (1) the selection stage discards the child because it is not as fit; or (2) the child replaces the parent because other changes on the genome provide an overall fitness improvement (i.e., genetic hitch-hiking). In the first case, there is no issue because the child is simply discarded. As the generations pass, selective pressure will direct the search and reduce the number of module solutions within the population, which will result in meaningful macro-variation. In the second case, the non-optimal module configuration can potentially be replaced by another macro-variation operation. This, however, is not guaranteed for the same reason the non-optimal module configuration was produced: crossing over three distinct configurations does not necessarily yield a module solution. But because bit-flip or macro-variation are performed probabilistically, the bit-flip operator will repair the non-optimal module configuration. The secondary

role of the bit-flip operator is to fix the non-optimal module configurations that can be potentially produced by differential crossover. Consequently, although differential crossover is not guaranteed to produce module solutions when $z > 2$, dDE is capable of solving problems with more than two optima per module.

4 EVOLVABILITY ON SMOOTH LANDSCAPES

Recent investigations into the evolution of evolvability have been successful in demonstrating that natural selection can change phenotypic distributions in a manner that modifies evolvability in several respects, including increasing developmental robustness and phenotypic heritability [9–11], increasing the rate of adaptation under directional selection [12], re-evolving previously evolved phenotypes more quickly [13], and enabling evolution to track changes in the environment more rapidly [13–15]. In some cases it can also accelerate the evolution of particular novel phenotypes, specifically, different combinations of modular features [15, 16]. These works have been careful to confirm that the changes to phenotypic distributions or the G–P map that facilitate such evolvability do not require multi-level selection and evolve only because of 'short-term' adaptive benefits that affect differential growth or survival between genetic lineages. This is necessary: Although developmental organisations might influence long-term evolutionary outcomes (i.e. beyond these timescales that can be 'seen' by the selection that produced them), long-term benefits cannot be the reason that these organisations evolved.

In this chapter, we focus on one piece of work in particular. Watson et al. [19] used a GRN model to show that evolved phenotypes become more evolvable over evolutionary time by internalising structural information about the environments the phenotypes had been exposed to. This evolution of evolvability is possible because the regulatory interactions of the GRN evolve to reflect the phenotypic correlations that are favoured in each selective environment, and the developmental process used to produce phenotypes biases the phenotypic distribution towards said evolved correlations. For example, if gene a and gene b share the same allele in one environment, the regulatory interaction between genes a and b evolves to reflect this positive interaction between the two genes. In subsequent evolutionary timesteps, when a genotype is developed into a phenotype, the developmental process biases the corresponding phenes of genes a and b towards sharing the same allele. Watson et al. [19] have shown that this phenomenon of is functionally equivalent to a simple type of learning mechanism, specifically associative or correlation learning, well-understood in neural network research.

What has not been shown, however, is how these evolved regulatory interactions affect future evolutionary trajectories. In this chapter, we expand on this work by investigating how the evolution of a GRN in varying selective environments changes future evolutionary trajectories. We begin by introducing the GRN model and performing similar experiments as in Watson et al. [19] to familiarise the reader with the work and model. We then proceed with investigating how the evolution of regulatory interactions affects future evolutionary trajectories.

4.1 METHODOLOGY

4.1.1 Gene-Regulation Network Model

We simulate the evolution of a GRN using a non-linear, recurrent developmental process, as described in Watson et al. [19].

The phenotype of an individual is a gene expression profile, $\boldsymbol{\phi}$, described by a set of n gene expression potentials which are assumed to control phenotypic traits, naturally represented by a vector: $\boldsymbol{\phi} = \langle \phi_1, \phi_2, \ldots, \phi_n \rangle, \phi \in \mathbb{R}^n$. The genotype of an individual consists of two parts: a vector, $\mathbf{g} = \langle g_1, g_2, \ldots, g_n \rangle$, defining the gene expression potentials of the embryonic phenotype, and the elements, b_{ij}, of an interaction matrix, $\mathbf{B}$ (a similar setup was used in references [14, 19, 117–119]). Every element of the interaction matrix, $\mathbf{B}$, represents the magnitude and sign (excitatory or inhibitory) of the interaction between two traits – that is, whether a gene up-regulates or down-regulates another gene.

We use a non-linear, recurrent developmental process that maps a genotype to a phenotype, modeled as follows. At developmental time-step $t_d = 0$, the embryonic phenotype is set to $\boldsymbol{\phi}(t_d = 0) = \mathbf{g}$. For every subsequent developmental time-step, the phenotype vector is updated by a non-linear transformation determined by the weighted effect of each trait (or expression potential) on each other trait, (the interaction matrix) $\mathbf{B}$, and a degradation rate:

$$\boldsymbol{\phi}(t_d + 1) = \boldsymbol{\phi}(t_d) + \sigma\left(\mathbf{B} \times \boldsymbol{\phi}(t_d)\right) - \tau\boldsymbol{\phi}(t_d), \tag{7}$$

where $\tau = 0.2$ is the rate of degradation, and σ is a sigmoidal function (applied to all elements of the phenotype vector) that non-linearly limits the effect of interactions. The non-linearity is modeled with the hyperbolic tangent: $\sigma(x) = \tanh(x)$.

The gene expression profile of the adult phenotype determines the fitness of an organism. We define 'adult phenotype', $\boldsymbol{\phi}_a$, to be the phenotype vector after a fixed number of developmental time-steps, $t_{final} = 10$, has passed: $\boldsymbol{\phi}_a = \boldsymbol{\phi}(t_{final})$.

4.1.2 Evolutionary Model

All elements of both parts of the genotype, $\mathbf{g}$ and $\mathbf{B}$, are initialised to 0 (i.e, no regulatory interactions). Every generation, mutations are applied to $\mathbf{g}$. Specifically, a single randomly chosen trait of $\mathbf{g}$ is mutated every evolutionary time-step by adding to it μ_1, drawn from a uniform distribution in the range ± 0.1. All elements of $\mathbf{g}$ are hard-bounded in the range $[-1, 1]$. Mutations on the regulatory interactions, b_{ij}, are less frequent than on the direct effects. Specifically, every generation, with probability $\frac{1}{15}$, one randomly chosen regulatory interaction is mutated by adding to it μ_2, drawn from a uniform distribution in the range ± 0.02.

Our model operates under 'strong selection, weak mutation' assumptions [120], which means that a mutation is fixed or lost in the population before another mutation occurs. To accommodate these assumptions, the population mean genotype is represented by a single genotype ($\bar{\mathbf{g}}$ and $\bar{\mathbf{B}}$), which is mapped into the population mean adult phenotype, $\bar{\boldsymbol{\phi}}_a$ via the developmental process. The evolution of the population is modeled by introducing small mutations to the genotype ($\bar{\mathbf{g}}$ and/or $\bar{\mathbf{B}}$), which yields a mutant genotype ($\mathbf{g}'$ and $\mathbf{B}'$). The mutant genotype is developed into a mutant adult phenotype $\bar{\boldsymbol{\phi}}_a'$. The probability of the mutant fixing in a population is proportional to the selective advantage

it confers, but here we are interested in the direction of selection rather than the magnitude. As such, we assume a 'hill-climbing' model of selection (e.g. references [19, 119]), meaning that the selection coefficient is assumed to be large enough to cause beneficial and deleterious mutations to fix or not fix in the population respectively. Under these assumptions, the mutant genotype becomes the mean population genotype if the fitness of its corresponding phenotype is greater than or equal to the fitness of the existing mean population phenotype ($\bar{\mathbf{g}}(t+1) = \bar{\mathbf{g}}'$, $\bar{\mathbf{B}}(t+1) = \mathbf{B}'$); otherwise the genotype remains the same for the next generation.

The model is evolved in varying selective environments. In each selective environment, fitness is determined by a constant directional selection on each phenotypic character, defined by a selective environment vector $\mathbf{s} = \langle s_1, s_2, \ldots, s_n \rangle, s \in \{-1, 1\}$. Fitness increases when the signs of the characters of the adult phenotype and 'target' phenotype vector: (i) are aligned and the magnitudes of the characters of the adult phenotype are increased; and (ii) are misaligned and the magnitudes of the characters of the adult phenotype are decreased. Fitness decreases when the signs of the characters of the adult phenotype and 'target' phenotype vector: (i) are misaligned and the magnitudes of the characters of the adult phenotype are increased; and (ii) are aligned and the magnitudes of the characters of the adult phenotype are decreased. Specifically, fitness, w, is determined by the dot product of the adult phenotype and 'target' phenotype vectors:

$$w(\boldsymbol{\phi}_a) = \boldsymbol{\phi}_a \cdot \mathbf{s}. \tag{8}$$

Each element of $\mathbf{s}$ determines the direction of selection on each phenotypic character [12, 19].

4.2 EXPERIMENTAL SETUP

All values in $\mathbf{g}$ and $\mathbf{B}$ are initialised to 0. Every generation, mutations are applied to $\mathbf{g}$. Specifically, a single randomly chosen trait of $\mathbf{g}$ is mutated every evolutionary time-step by adding to it μ_1, drawn from a uniform distribution in the range ± 0.1. All elements of $\mathbf{g}$ are hard-bounded in the range $[-1, 1]$. Mutations on the regulatory interactions, b_{ij}, are less frequent than on the direct effects. Specifically, every generation, with probability 1/15, one randomly chosen regulatory interaction is mutated by adding to it μ_2, drawn from a uniform distribution in the range ± 0.02.

We alternate between two selective environments characterised by smooth fitness landscapes (similar to references [13, 15, 16, 19, 20, 83]). The two environments of size $n = 8$ are: $\mathbf{s}_a = \langle +, +, -, -, -, +, -, + \rangle$ and $\mathbf{s}_b = \langle -, +, -, -, -, -, +, + \rangle$, where $+ = 1$ and $- = 1$. Each simulation lasts 800000 generations, and the environment changes from one to the other every 2000 generations.

4.3 RESULTS

4.3.1 Evolving Regulatory Interactions and Developmental Memory

We begin by replicating parts of the work of Watson et al. [19]. In particular, we are showing how regulatory interactions evolve in varying selective environments and how

these regulatory interactions form a developmental memory. Figure 12a shows that, with evolutionary time, evolved regulatory interactions are separated into three classes. The first class consists of regulatory interactions that upregulate two genes (positive), the second class consists of regulatory interactions that downregulate two genes (negative), and the third class defines a lack of regulation between two genes (≈ 0). The pairs of genes that are upregulated are those that are positively correlated in *both* selective environments (i.e. $++$ or $--$ in *both* environments $\mathbf{s}_a$ and $\mathbf{s}_b$), whereas the ones that are negatively correlated in *both* selective environments (i.e. $+-$ or $-+$ in *both* environments $\mathbf{s}_a$ and $\mathbf{s}_b$). Pairs of genes that are not regulated have a different correlation across the two selective environments (e.g. $++$ in environment $\mathbf{s}_a$ and $+-$ in environment $\mathbf{s}_b$) and have a regulatory coefficient that is near zero on average (Watson et al. [19] and Kashtan et al. [119] report similar results). Figure 12b shows the evolved regulatory interaction matrix at the end of the simulation. Figure 12c shows the regulatory interaction matrix as predicted by Hebb's rule. It is calculated by taking the outer product of the target phenotype of the first selective environment and adding it to the outer product of the target phenotype of the second selective environment: $(\mathbf{s}_a \cdot \mathbf{s}_a) + (\mathbf{s}_b \cdot \mathbf{s}_b)$. Comparing figures 12b and 12c confirms that the pattern of regulatory interaction coefficients (positive, negative, and near zero) in the evolved matrix matches the pattern of regulatory interaction coefficients in the matrix predicted by Hebb's rule.

Although the target phenotypes used here are arbitrary bit patterns and include only eight features ($n = 8$), the results clearly demonstrate that the evolved regulatory interactions match the interactions predicted by Hebbian learning [19] (Figures 12b and 12c). Further than that, we can see that the evolved regulatory interactions bias the developmental process towards reproducing the previously seen phenotypes and their complements (figure 13).

These results were previously demonstrated in Watson et al. [19]. Here we reproduced parts of their work to elucidate how phenotypic distributions are affected by the evolution of regulatory interactions. In the following section, we expand on this work by investigating how these changes in phenotypic distributions caused by the evolution of a developmental memory affect evolutionary trajectories.

4.3.2 Evolution Becomes Faster at Re-Evolving Previously Seen Phenotypes

As the number of environmental changes increases, we find that the developmental process progressively canalises the two 'targets' and rapidly switches between them using pre-prepared responses. Figure 14 shows fitness trajectories before and after a change in environment at different stages of the simulation. We find that later in evolution, adaptation becomes faster, with the growth in fitness being sigmoidal. The reason for the sigmoidal growth is that there is a trade-off between robustness and evolvability. Since the developmental dynamics have formed attractors for both targets, the impact of deleterious mutations is reduced, but the impact of beneficial mutations is initially also reduced. After beneficial mutations have been accumulated and the genotype reaches the saddle point of the attractor landscape, beneficial mutations have significantly more impact. In fact, the accumulation of beneficial mutations results in a pre-prepared response that enables the rapid switch from target $\mathbf{s}_a$ to target $\mathbf{s}_b$ and vice versa. Although the developmental constraints canalise the two targets, adaptation to previously seen environments becomes faster.

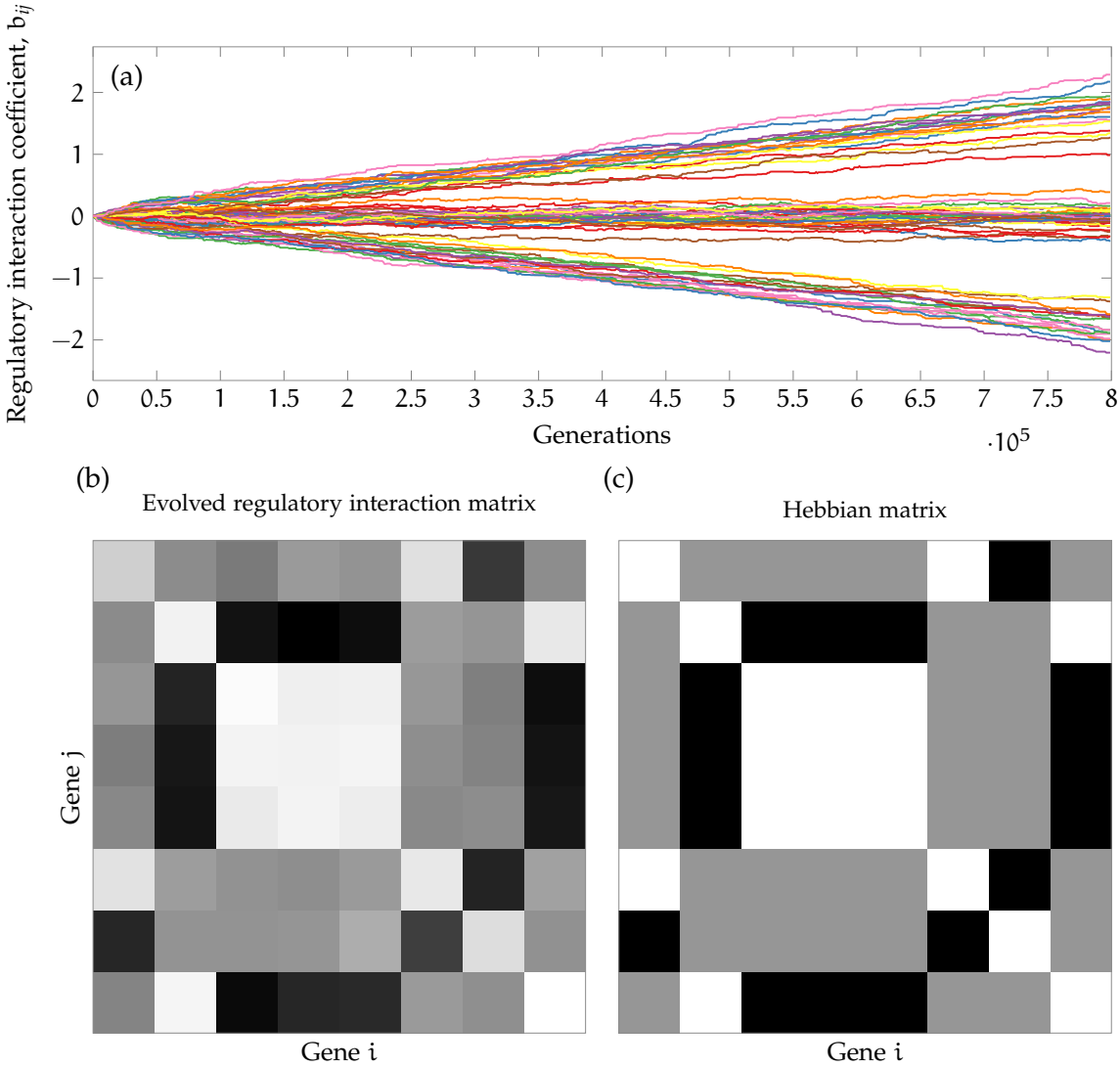


Figure 12: Evolved regulatory interactions for two changing environments. (a) With evolutionary time, evolved regulatory interactions are separated into three classes. The first class consists of regulatory interactions that upregulate two genes (positive), the second class consists of regulatory interactions that downregulate two genes (negative), and the third class removes regulatory interactions between specific genes (≈ 0). The pairs of genes that are upregulated are those that are positively correlated in *both* selective environments (i.e. $++$ or $--$ in *both* environments $\mathbf{s}_a$ and $\mathbf{s}_b$), whereas the ones that are negatively correlated in *both* selective environments (i.e. $+-$ or $-+$ in *both* environments $\mathbf{s}_a$ and $\mathbf{s}_b$). Pairs of genes that are not regulated have a different correlation across the two selective environments (e.g. $++$ in environment $\mathbf{s}_a$ and $+-$ in environment $\mathbf{s}_b$). (b) The evolved regulatory interaction matrix at the end of the simulation. White, black and grey shades are for genes that positively, negatively or not regulated respectively. (c) An interaction matrix derived from Hebb's rule rather than evolved. When comparing (b) and (c), it is clear that the two sets of regulatory interactions share the same pattern of interactions that upregulate, downregulate or not regulate pairs of genes.

After the environment changes, the allelic values of the loci of $\mathbf{g}$ which are not aligned with the current target phenotype will slowly change towards its direction. We want to

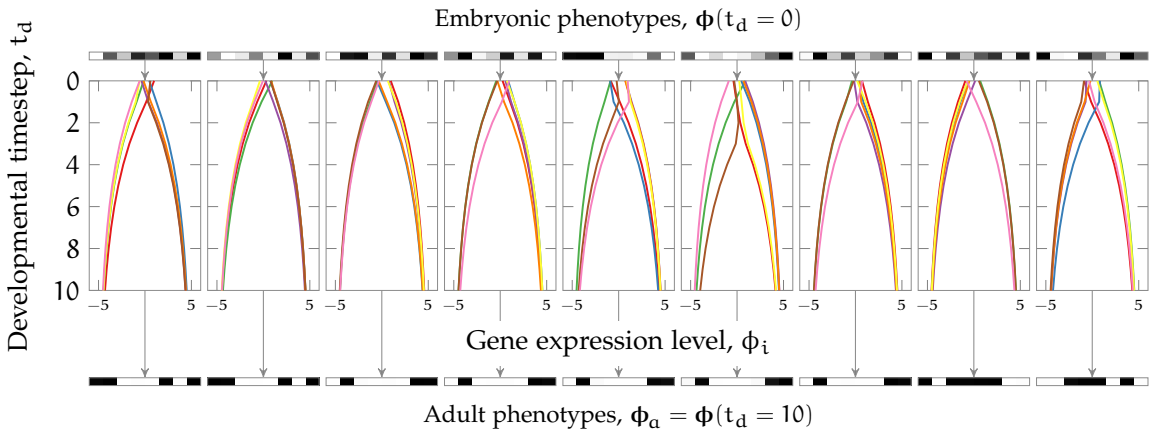


Figure 13: Gene expression levels change over developmental time. Nine independent developmental trajectories each starting from a random embryonic phenotype ($\boldsymbol{\phi}(t_d = 0) = \mathbf{g}$). The embryonic phenotypes are depicted in the top row. The middle row shows how the gene expression levels change over developmental time given the evolved regulatory interaction matrix from figure 12b. The bottom row shows the adult phenotypes produced at the end of development ($\boldsymbol{\phi}_a = \boldsymbol{\phi}(t_{final})$).

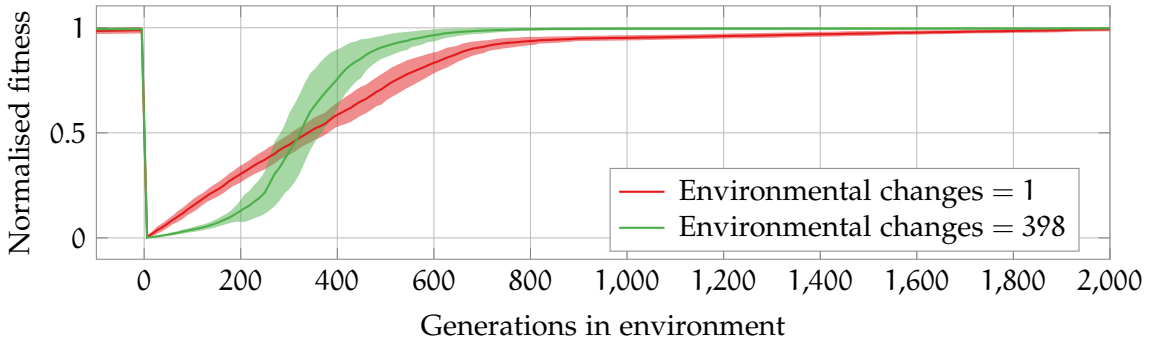


Figure 14: Organisms become more evolvable by recovering fitness faster when exposed to systematically varying selective environments characterised by smooth fitness landscapes. Results shown are from 30 replicates of the experiment, with each experiment lasting 800000 generations and the environment changing every 2000. Vertical axis is mean normalised fitness; horizontal axis shows generations in environment, with the change in environment happening at generation 0. As the number of environmental changes increases, adaptation to the new (but previously seen) environment becomes faster. The sigmoidal trend in fitness growth suggests that there is a trade-off between robustness and evolvability. The fitness benefit from initial mutations is reduced due to the developmental constraints further canalising the target phenotypes, but that does not preclude the accumulation of beneficial mutations which results in a pre-prepared response that enables the rapid switch from target $\mathbf{s}_a$ to target $\mathbf{s}_b$.

compare what happens to the fitness landscape as $\mathbf{g}$ changes when the G–P map is one-to-one and when it is evolved. To do so, we ran another experiment in which we (1) set the regulatory interaction matrix, $\mathbf{B}$, to be either an identity matrix (one-to-one), or equal to the regulatory interaction matrix evolved at the end the experiment shown in figure 14; and (2) controlled the direct effects on gene expression potentials, $\mathbf{g}$, by setting the allelic values of some of its loci to be equal to the allelic values of the corresponding loci in target phenotype $\mathbf{s}_a$, and having the remaining loci vary in the range $[-1, 1]$. We chose loci 1

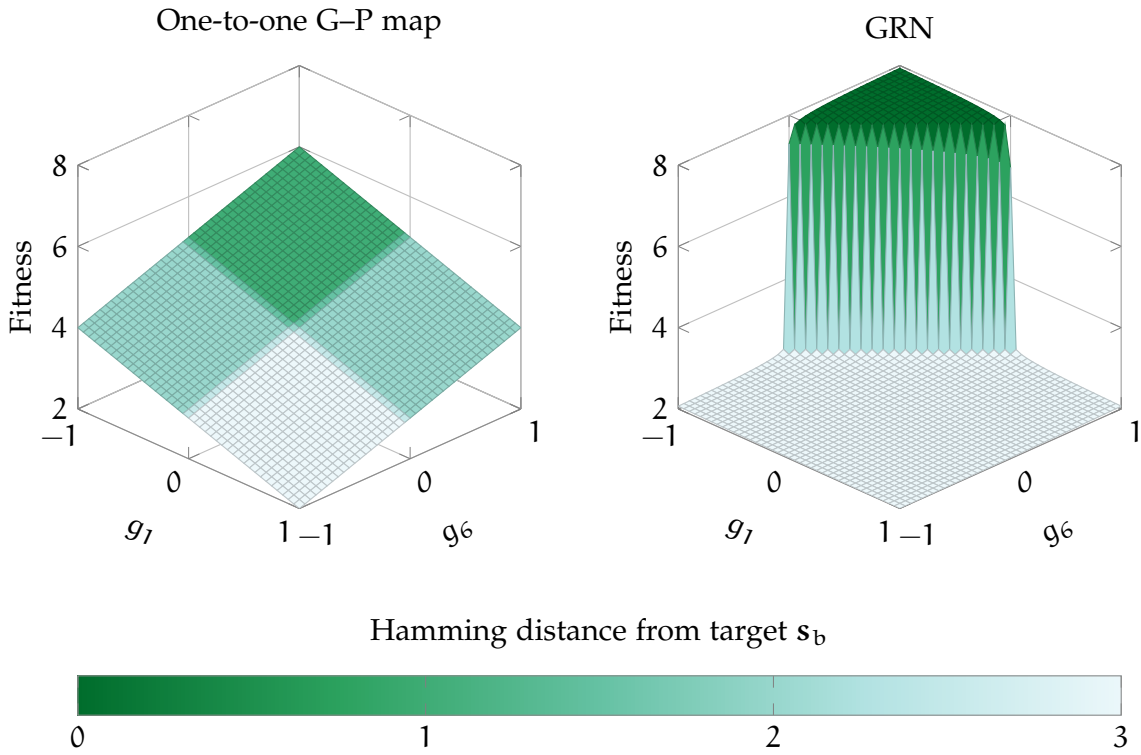


Figure 15: How the evolved G–P map affects the fitness landscape. Left: one-to-one G–P map; right: evolved G–P map. Evaluating how the fitness of the adult phenotypes and their Hamming distance with respect to target phenotype $\mathbf{s}_b$ are affected as the allelic values of two of the loci (g_1 and g_6) that differ between $\mathbf{s}_a$ and $\mathbf{s}_b$ move towards the direction of $\mathbf{s}_b$. The gene expression levels of adult phenotypes are normalised before evaluation, and the colour denotes the Hamming distance between the adult phenotype and target $\mathbf{s}_b$. With a one-to-one G–P map, fitness is increased linearly and the Hamming distance is decreased by one when an allele's sign is changed to match the corresponding allele of target $\mathbf{s}_b$. With an evolved G–P map, there is a non-linear increase in fitness and the Hamming distance is either the Hamming distance between the two target phenotypes or 0. Sign-wise, the adult phenotype produced is always one of targets $\mathbf{s}_a$ and $\mathbf{s}_b$. As the two alleles change towards the direction of target $\mathbf{s}_b$, the evolved G–P map produces fitter phenotypes than those produced with a one-to-one G–P map. This is because although only two of three alleles are changed to match target $\mathbf{s}_b$, the embryonic phenotype resembles target $\mathbf{s}_b$ more than it does $\mathbf{s}_a$, and therefore development produces target $\mathbf{s}_b$.

and 6 to be the loci that vary, as the two target phenotypes, $\mathbf{s}_a$ and $\mathbf{s}_b$, have different allelic values in those loci. Multiple $\mathbf{g}$ vectors are developed into adult phenotypes, and those adult phenotypes have (1) their fitness evaluated with respect to target phenotype $\mathbf{s}_b$; and (2) their Hamming distance to target phenotype $\mathbf{s}_b$ measured. In doing so, we are able to understand why the robustness versus evolvability trade-off exists and how the fitness landscape is affected by an evolved G–P map as the direct effects on gene expression potentials, $\mathbf{g}$, move towards the direction of the current target phenotype.

With a one-to-one G–P map, we observe a linear increase in fitness and a linear decrease in Hamming distance as a single allele's sign is changed to match the corresponding allele of target $\mathbf{s}_b$ (figure 15; left column). With an evolved G–P map, as shown in the right column of figure 15, we observe a sigmoidal increase in fitness, and a Hamming distance

that is either (a) the Hamming distance between targets $\mathbf{s}_a$ and $\mathbf{s}_b$ ($d = 3$); or (b) 0 (i.e. $H(\boldsymbol{\phi}_a) = H(\mathbf{s}_b)$) (where $H(.)$ is the Heaviside step function). The sigmoidal increase in fitness is due to the trade-off between robustness and evolvability, as described earlier. The effect on the Hamming distance, however, is due to the developmental interactions constraining the discrete-space phenotypic neighbourhood to only two phenotypes: targets $\mathbf{s}_a$ and $\mathbf{s}_b$. As the two alleles move towards the direction of target $\mathbf{s}_b$, the evolved G–P map produces fitter phenotypes than those produced with a one-to-one G–P map. In other words, the same embryonic phenotype is developed into two different adult phenotypes of different fitnesses. That is because with a one-to-one G–P map, only two of the three loci that differ between the targets have their allelic values change to match the current target phenotype. With the evolved G–P map, however, changes to the allelic values of two loci are channelled through the developmental process, causing the third locus to change its allelic value also. More specifically, the evolved G–P map produces the phenotype that is more close in euclidean space to the embryonic phenotype, and when the allelic values of two of the three loci change towards the direction of target $\mathbf{s}_b$, the allelic value of the third locus is also changed in the direction of $\mathbf{s}_b$.

4.4 DISCUSSION

In this chapter, we have demonstrated that the evolution of developmental memory changes evolutionary trajectories for the better. Not only do phenotypes become more robust, they become more evolvable at the same time by enabling evolution to rapidly switch between memorised phenotypes using pre-prepared responses. This leads to a speed-up in evolution in terms of the number of generations required to reach a high-fitness phenotype when changing from one environment to the other.

These results are not entirely novel, however. Recent investigations have shown that systematically varying environments can enhance evolvability by making it possible to switch between phenotypes faster [13–16]. This type of short-term evolvability makes it so that evolution is capable of rediscovering previously seen phenotypes faster or so that evolution can discover generalisations of previously seen phenotypes and evolve to them rapidly without having encountered them before.

In the following chapter, we investigate whether what we term as 'long-term evolvability' can evolve. We define long-term evolvability as the type of evolvability that enables evolution to find phenotypes that are objectivelly fitter than the ones it encountered before. Note that this is not the same as the sort of evolvability that results from generalising across previously seen phenotypes encountered on smooth fitness landscapes (for example, references [13, 15, 16, 19]). In that body of work it was necessary to hand-design a family of single-peaked selective landscapes sharing common structural regularities. Instead we utilise the observation that the local peaks present in a single multi-peaked landscape naturally share common structural regularities because they derive from the same set of underlying epistatic constraints [61, 121]. This does not depend on any explicit modularity or other contrivance in the problem structure. This also has the advantage that the fitnesses of different local peaks are comparable because they are on the *same* landscape, which is important because it enables us to assess not merely whether evolution evolves a given phenotype *more quickly*, but also whether evolution can find a different

phenotype that is *fitter*. In chapter 5, we investigate whether it is possible for evolution to find progressively better phenotypes on such fitness landscapes.

5 EVOLVABILITY ON RUGGED LANDSCAPES

The work that follows is from the following papers:

1. Kounios L et al. (2016) Resolving the Paradox of Evolvability with Learning Theory: How Evolution Learns to Improve Evolvability on Rugged Fitness Landscapes. arXiv: 1612.05955 [q-bio.PE].
70. Kounios L et al. (in preparation[b]) Resolving the Paradox of Evolvability with Learning Theory: How Evolution Learns to Improve Evolvability on Rugged Fitness Landscapes. *PLoS Computational Biology*.

The following are contributions from authors in the aforementioned papers. Loizos Kounios, Richard A. Watson, Jeff Clune, Kostas Kouvaris, Günter P. Wagner and Daniel M. Weinreich conceptualised the experiments. Loizos Kounios, Richard A. Watson, Jeff Clune and Kostas Kouvaris designed the experiments and analysed data. Loizos Kounios performed the research. Richard A. Watson wrote most of the introduction and section 5.4.3. Loizos Kounios wrote everything else. Jeff Clune and Günter P. Wagner provided relevant literature and helped improve the writing. Daniel M. Weinreich and Mihaela Pavlicev helped with revising the writing. Finally, Freddy Nash contributed by performing research, producing figures, and further additions to the text, but those changes are only included in the as-of-yet unpublished Kounios et al. [70] and not in this thesis.

ABSTRACT

It has been hypothesised that one of the main reasons evolution has been able to produce such impressive adaptations is because it has improved its own ability to evolve – 'the evolution of evolvability'. Rupert Riedl, for example, an early pioneer of evolutionary developmental biology, suggested that the evolution of complex adaptations is facilitated by a developmental organisation that is itself shaped by past selection to facilitate evolutionary innovation. However, selection for characteristics that enable future innovation seems paradoxical: natural selection cannot favour structures for benefits they have not yet produced; and favouring characteristics for benefits that have already been produced does not constitute future innovation. Here we resolve this paradox by exploiting a formal equivalence between the evolution of evolvability and learning systems. We use the conditions that enable simple learning systems to generalise, i.e. to use past experience to improve performance on previously unseen, future test cases, to demonstrate conditions where natural selection can systematically favour developmental organisations that benefit future evolvability. Using numerical simulations of evolution on highly epistatic fitness landscapes, we illustrate how the structure of evolved gene regulation networks can result in increased evolvability capable of avoiding local fitness peaks and discovering higher fitness phenotypes. Our findings support Riedl's intuition: Developmental organisations that 'mimic' the organisation of constraints on phenotypes can be favoured by short-term selection and also facilitate future innovation. Importantly, the conditions that enable the

evolution of such surprising evolvability follow from the same non-mysterious conditions that permit generalisation in learning systems.

5.1 INTRODUCTION

The ability of natural populations to exhibit adaptation depends on the production of suitable phenotypic variation that natural selection can act on. Such variability in turn depends, amongst other things, on the properties of the G–P map or the organisation of developmental processes that constrain and bias the distribution of possible phenotypic variants. Contemporary evolutionary developmental biology recognises that developmental organisation is both a product of natural selection and a factor that can significantly alter subsequent evolutionary outcomes [2–4, 6]. It is therefore clear that natural selection can produce heritable changes that modify the ability to evolve – a phenomenon called 'evolution of evolvability' [4, 7, 8]. But can natural selection systematically improve its own ability to evolve? It has been argued that natural selection increases evolvability over time and that this is important in explaining the amazing diversity and complexity of the natural world [2, 3, 122]. Yet, how natural selection, which operates only on short-term fitness differences, can improve long-term evolvability remains one of the most important, open questions in evolutionary biology [2–4, 7, 8, 122–127].

Rupert Riedl, an early pioneer of evo–devo research, provided key concepts and ideas on the topic of evolvability. He suggested that the evolution of complex adaptations is facilitated by developmental architectures that are organised by natural selection to 'mimic' or 'imitate' the functional constraints on phenotypes [2–5]. However, his ideas have not been previously demonstrated in an explicit mechanistic model. Such a model needs to explain exactly what form such imitation takes and, more challengingly, explain why imitating the constraints experienced in the past facilitates evolutionary innovation in the future. We provide a resolution to this problem by drawing a formal analogy with well-established knowledge from another discipline; namely, learning theory. A simple analogy between learning and evolution has been noted many times [128, 129], but the link has recently been formalised and deepened extensively [19, 66, 69, 130–133]. Here we argue that Riedl's notion of an imitatory developmental organisation is directly analogous to a learning system that internalises a model of its environment, and the link with learning also provides the missing explanatory component and mechanistic principles to demonstrate how this facilitates future innovation. Specifically, in learning systems the idea that past experience can be used to generalise to future, previously unseen, test cases is not at all paradoxical. This enables us to update Riedl's notion of development that *imitates* to a process of development that *innovates* in a simple but predictable manner.

5.1.1 Generalisation and Evolvability

Here, we develop the intuition that evolvability is to evolution as generalisation is to learning [66, 133]. That is, learning without generalisation is essentially just remembering what you have already seen; it is obvious that evolution by natural selection can do this in the sense that genotype frequencies are determined by past selection. But evolvability suggests an ability to go beyond this; not merely to favour the phenotypes that have already

been rewarded but to facilitate innovation – an ability to produce new phenotypes that are fit even though they have not been previously subject to selection. The evolution of a high mutation rate in a rugged fitness landscape is potentially a simple way to provide this – implicitly exploiting the prediction that high degrees of epistasis in the past might be indicative of high degrees of epistasis in the future. But a parameter that simply controls the amount of variability (e.g. mutation rate) has little or no ability to accumulate specific information from past selection and is, in any case, not easy to evolve [49]. Learning systems show that predicting the future (in specific ways) from past experience is possible in simple mechanistic systems. Learning systems do not really 'see the future', of course; they are simply finding underlying structural regularities that are invariant over time [130]. For this to be non-trivial, it is essential that the future is not simply the same as the past, but rather shares invariant structural properties – properties that are, in some sense, beneath the surface. In this paper, by transferring existing knowledge from learning systems, we explore the conditions where natural selection can achieve this same type of generalisation and thus demonstrate evolvability that facilitates innovation. Unlike models based on the evolution of mutation rates that attempt to increase evolvability simply by enabling more (random) variability, our models increase evolvability by enabling 'smarter' variability; i.e. by evolving developmental organisations that create a distribution of variants with specific adaptive structure.

To learn structural regularities requires an ability to accumulate information from multiple past examples. A learning system that models each feature independently cannot do this because each new example overwrites the features of the previous example (to the extent that they are different). This is analogous to an evolutionary system where phenotypes are directly encoded by genotypes – i.e. each phenotypic trait is controlled by a single gene, also known as 'one-to-one'. Given that genetic variation is undirected, a one-to-one mapping produces undirected variability in phenotypic characteristics and is therefore incapable of 'remembering' more than one thing (except in the degenerate sense of representing a simple average), or generalising [69, 133]. The evolutionary significance of this is that it is unable to use information from past selective environments to modify the future response to selection except by changing its *current* phenotype – there is no possibility of representing tacit information from past selection. In contrast, a learning system that generalises requires an ability to represent the underlying structural regularities of a problem domain, e.g. to observe which features 'go together' in a set of examples, and this requires a more flexible input–output mapping. Representing structural relationships between multiple features of observed solutions enables a learning system to exploit similarities between multiple past solutions (and possible future solutions) even though their individual features may be different or contradictory.

This is analogous to a G–P map that constrains and biases the combinations of phenotypic traits produced by genotypes, e.g. via a GRN. This is capable of representing information about structural regularities observed in past selected phenotypes and generalising by facilitating phenotypic variation which reflects the regularities represented within the network [19, 69]. Recent work shows that it is possible for a GRN to represent information about structural regularities observed in past selected phenotypes and also describes in detail how information is introduced into such networks by past selection in the same way that an associative learning system acquires information from experience [19, 69]. Specifically, this work shows that the effect of natural selection on mutations that change the strength of connections in a GRN is formally equivalent to learning mechanisms that alter the strength of synaptic connections in a neural network [19]. The potential for

neural networks to generalise via such simple learning mechanisms is well characterised. Thus, a simple G–P map based on gene-regulatory interactions is capable of representing such correlations in the same way that a neural network is. Here we show for the first time that this ability of natural selection to shape the organisation of gene networks can improve evolvability on rugged fitness landscapes, enabling future evolution to move in a modified space of phenotypes and thereby avoid local fitness peaks that prevent evolution with one-to-one mappings from finding higher fitness phenotypes. We also show that the evolving gene network achieves this result by creating pleiotropic interactions that mimic the structure of the epistasis, inherent in the fitness landscape, acting on phenotypic traits, as Riedl predicted; but crucially, we also show that it is thus capable of generalisation necessary to improve future innovation. It is in this way that natural selection is able to recognise and exploit structural regularities in the selective environment that facilitate evolutionary innovation.

To explain this evolvability we need to explain the congruence between short-term fitness benefits (arising from organisations that mimic the structure of selective constraints on the phenotype) and long-term fitness benefits (i.e. an ability to innovate on rugged fitness landscapes). This is neither guaranteed nor mysterious; it depends on the same conditions where generalisation is possible in learning systems. Specifically, that learnable regularity is present in the training data. To demonstrate this, we utilise the observation that the phenotypes at local peaks in a multi-peaked epistatic fitness landscape are not arbitrarily different to one another (except in special cases). It is, of course, possible to construct truly random landscapes where high-fitness phenotypes have no shared commonalities. But this will not be the case whenever phenotypes have fitnesses that result from a sum of low-order epistatic fitness contributions (e.g. pairwise epistasis, but strictly, anything less than order-n epistasis where n is the number of traits). In general, phenotypes at local fitness peaks share common structural regularities because they derive from the same set of underlying epistatic constraints [61, 121]. The presence of such regularities is of no use to natural selection with a one-to-one G–P map, since such a map is incapable of representing or exploiting them. But we show that evolution on such a landscape, using phenotypes produced by an evolving GRN, can 'learn' to exploit these regularities. This learning–evolution analogy is more than the simple idea that selection 'rewards' fit phenotypes by increasing their frequency in the population. Rather this is a formal functional equivalence meaning that natural selection has the ability to induce an internal structural organisation (in this case, via the evolution of gene-regulatory interactions) that mimics the structure of the epistasis acting on phenotypes. This 'internalised model' of the environment produces phenotypes that respect the epistatic interactions acting on those phenotypes and is thus enriched for fit phenotypes – including novel phenotypes. This causes future evolutionary trajectories to be biased toward parts of the fitness landscape with structural regularities that are 'familiar' from past evolutionary experience. This has the effect of causing future evolution to avoid low-fitness peaks (because they have less structural commonality with high-fitness phenotypes experienced in the past) and thus find high-fitness phenotypes faster and more reliably. We demonstrate this by comparing evolutionary trajectories with more evolved and less evolved gene networks, from many different starting points in phenotype space, and measuring the long-term fitnesses they attain. We are particularly interested in cases where evolutionary trajectories with a more evolved mapping systematically arrive at different local optima of higher fitness than evolutionary trajectories with a less evolved mapping. Passing this test is important because it indicates that the more evolvable mapping enables the discovery of new high-

fitness phenotypes that cannot be discovered with the less evolvable mapping (unless evolution moves against selective gradients or depends on multiple specific genetic mutations occurring simultaneously). Accordingly, we compare multi-generation evolutionary trajectories with different G–P maps: (a) a direct encoding where the phenotype is the same as the genotype (i.e. a 'one-to-one' or identity mapping); (b) evolved G–P maps at different stages of evolution. Specifically, the direct encoding has no gene-regulatory interactions such that each gene controls exactly one phenotypic trait, whereas the evolved mappings introduce gene-regulatory interactions. We assess the fitness levels attained and whether different fitness peaks (with phenotypes of higher fitness) are discovered in a multi-peaked fitness landscape.

Recent investigations into the evolution of evolvability have demonstrated that short-term natural selection can increase robustness [9–11], increase the rate of adaptation under directional selection [12], re-evolve previously evolved phenotypes more quickly [13], and enable evolution to track changes in the environment more rapidly [13–15]. Some of this work is evidence of the imitatory features that Riedl described, e.g. modularity, or other structural features, that mimic the structure of the environment [15, 16]. Clune et al. [15], in work on the evolution of neural networks, go further to demonstrate that conditions favouring modular networks not only mimic the modularity of the task environment but also facilitate more effective evolvability and superior evolutionary outcomes. Here, investigating evolvability in a gene network (rather than a neural network), we show that there is sufficient learnable regularity in much more general fitness landscapes (without designed-in modularity). More importantly, we explain the mechanism by formally linking the conditions for this result to generalisation in learning theory.

5.1.2 Evolvability on Rugged Fitness Landscapes

To understand how evolutionary trajectories, starting from the same phenotype on the same fitness landscape, can be different from one another, we need to understand (a) that different G–P maps can produce different distributions of phenotypic variation [2–4, 6–8, 122–126]; and (b) how this distribution interacts with selection to determine the direction of movement in phenotype space [17, 18]. Given that *genetic* variation is undirected, a one-to-one mapping produces undirected variability in phenotypic characteristics also, and evolutionary trajectories are thus shaped by selective gradients only. Mappings that produce non-uniform phenotypic distributions may cause differences in subsequent evolutionary trajectories [17, 18]. Under constant directional selection, if phenotypic variability in the direction of selection is decreased, this will decrease the rate of adaptation. Conversely, if the phenotypic variation in the direction of selection is increased this will accelerate evolution [12]. Pavlicev et al. [12] showed that the effect of short-term selection is to favour genetic lineages that increase phenotypic variation in the direction of selection precisely because they accelerate adaptation in this manner. Moreover, biases in the distribution of phenotypic variants can not only increase or decrease the rate of adaptation but can also alter the direction of evolutionary trajectories through phenotype space [17, 18]. In much the same way that a keel on a sailing boat can make the direction of travel differ from the direction of the wind, a biased phenotypic distribution, by making evolutionary change in some dimensions easier than others, can make the path of evolutionary change differ from the current direction of selection and thus cause evolutionary trajectories to visit different regions of phenotype space.

In any single-peaked landscape, this may not alter long term evolutionary outcomes; assuming that a G–P map is not so constrained as to remove any of the necessary variability altogether, evolution will ultimately attain the same phenotypic peak and hence the same fitness in the long term, whether it be via a direct or circuitous route [17, 18]. In a multi-peaked fitness landscape, if a G–P map changes the regions of phenotype space that are explored, this causes natural selection to find different fitness peaks – thus altering long-term evolutionary outcomes. Figure 16 demonstrates this using data from the experiments that follow. The figure shows how evolution over a fixed set of possible mutations given one G–P map quickly exhausts the production of beneficial variation and becomes trapped at a local optimum in the adaptive landscape (Figure 16a). Nonetheless evolution over the same set of possible mutations with a different G–P map continues to produce adaptive variations that ultimately lead to superior regions of phenotype space (Figures 16b and 16c).

To explain a systematic mechanism for the evolution of evolvability of this type requires that we understand the kind of knowledge that natural selection can 'learn' from past experience (i.e. the kind of information that the organisation of the G–P map can hold) and its correspondence with the knowledge that is needed to evolve high-fitness phenotypes. This is what Riedl could not provide. Here we show that when the G–P map is defined by a network of gene-regulatory connections, it is capable of learning the structure of pairwise epistatic interactions in the fitness landscape. This biases the distribution of phenotypes produced by genetic mutations to suppress the expression of trait combinations that conflict with high-fitness phenotypes and amplify the expression of trait combinations that are compatible with high-fitness phenotypes. These biases are a simple way of generalising the information (combinations of phenotypic traits) found at multiple easy-to-find local peaks of average (for peaks) fitness, it is thereby able to predict the location of much rarer local peaks of exceptionally high fitness.

5.1.3 Experimental Setup

We simulate the evolution of a GRN controlling a set of phenotypic traits, $\boldsymbol{\phi}$, via a set of gene-expression potentials [19] (see *section 5.4.1*). The genotype defines the embryonic state of each gene expression potential with a vector, $\mathbf{g}$, and an interaction matrix, $\mathbf{B}$, whose elements represent the magnitude and sign of the regulatory interactions between one gene and another in the GRN [19]. In the absence of evolved regulatory interactions, $\mathbf{B}$, the G–P map is one-to-one, and the sign and magnitude of elements in $\boldsymbol{\phi}$ correspond directly to the elements of $\mathbf{g}$. Thus, in the absence of $\mathbf{B}$, mutations to $\mathbf{g}$ produce a uniform ball of variation in phenotype space on average. In the presence of evolved regulatory interactions, $\mathbf{g}$ is mapped into an adult phenotype using a non-linear, recurrent developmental process defined by $\mathbf{B}$.

Developmental constraints, $\mathbf{B}$, are assumed to evolve slowly. Specifically, the correlations between traits, represented by the interaction matrix $\mathbf{B}$, are assumed to evolve slowly relative to changes in the magnitudes of the traits themselves, represented by the elements of $\mathbf{g}$ (e.g. the number of mutational sites affecting the expression level of a single gene is greater than the number of sites affecting its co-regulatory interaction with another specific gene [19]). We refer to the timescale over which developmental constraints evolve as 'deep' evolutionary time, and the more rapid timescale of evolution *given* those developmental constraints as microevolutionary time. We are interested in how the genetic

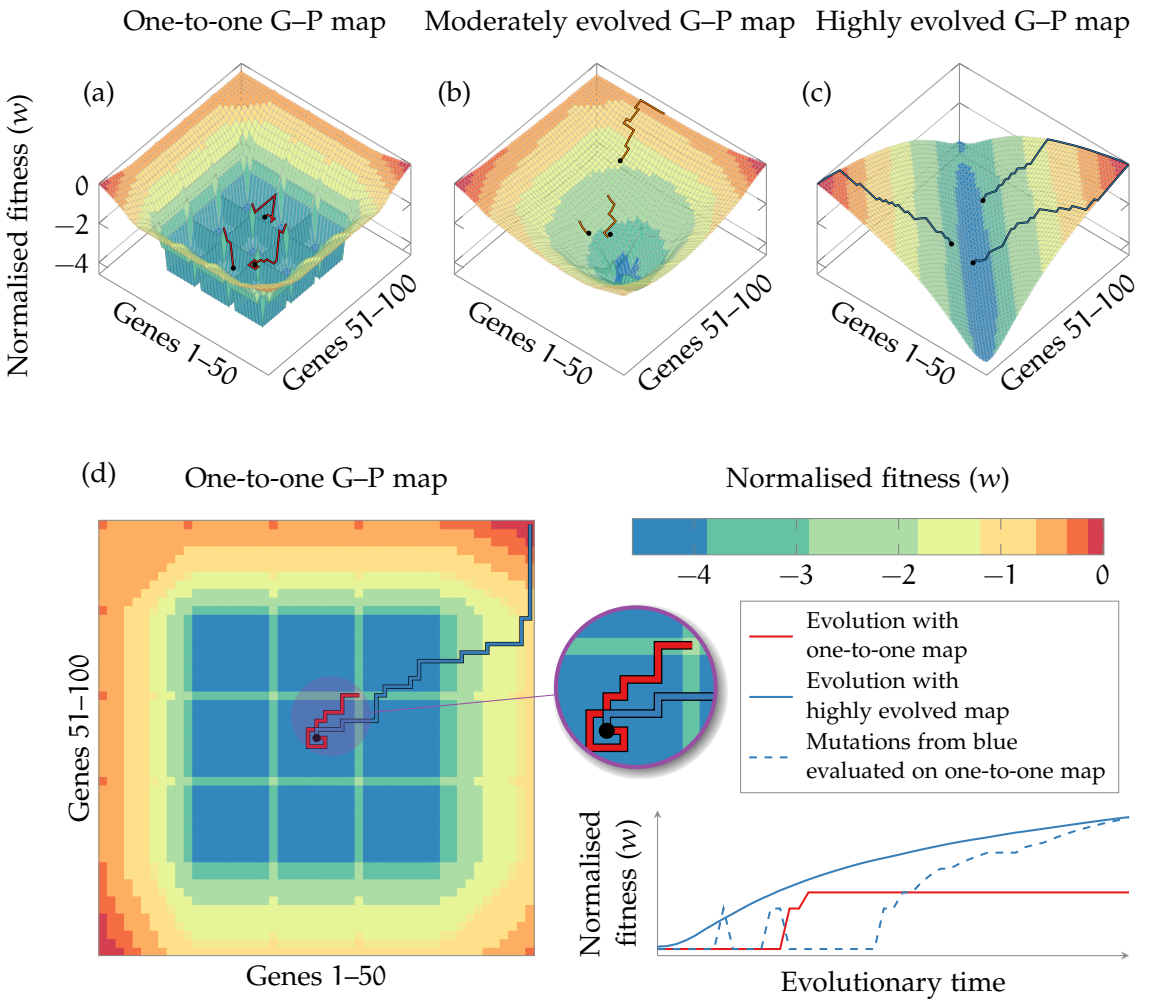


Figure 16: Highly evolved G–P maps enable microevolution to find fitter phenotypes by changing the fitness landscape. The fitness landscape for a set of genetic mutations is modified by evolved changes in the genetic background defining the G–P map. Each mapping determines the phenotype (i.e. expression profile of 100 gene products) produced by a set of 100 genes. The initial mapping is a direct encoding (with no gene-regulatory interactions) such that each of the phenotypic traits is determined by the state of exactly one of the genes. Genetic mutations under this 'one-to-one' or 'identity mapping' have no pleiotropy or phenotypic epistasis, but the adaptive landscape exhibits widespread epistasis for fitness. The other mappings are the result of evolutionary changes (at other genetic sites) that introduce gene-regulatory interactions. In each case the surface shown depicts the same two-dimensional cross-section through the highly-epistatic high-dimensional fitness landscape (each dimension is a particular cross-section through a subspace of genotypes; see *section 5.4.5*). This cross-section maintains the property that local optima on this surface (i.e. where all neighbouring points are lower in fitness) are also true local optima in the original high-dimensional fitness landscape. The two globally optimal phenotypes in this landscape are located at opposite corners of this cross-section. Fitness reported is in logarithmic scale. (Continued on next page.)

Figure 16: (Continued from previous page.) (a) With the one-to-one G–P map, the space of phenotypes corresponds directly to the underlying space of genotypes. Trajectories show simulation data of paths on the fitness surface taken by evolution using this direct encoding (start points drawn randomly in genotype space fall in low fitness regions of this surface with high probability). In this rugged fitness landscape these trajectories quickly become trapped when a local optimum is encountered. Evolution fails to find a global optimum in 1000 attempts. Three example evolutionary trajectories are shown (black circles denote starting locations); to enhance visualisation, trajectories which revisit previously seen phenotypes via neutral changes are not shown. (b) The fitness surface with a moderately evolved G–P map. As the genetic background defining the parameters of the G–P map (i.e. evolving gene-regulatory interactions) evolve under natural selection, the phenotypes produced over the same space of 100 genes change (in predictable ways, see text) and thus have different fitnesses. Some points in genotype space which were local optima in (a) are no longer local optima in (b), and thus evolutionary trajectories from the same starting points as (a) sometimes discover higher fitness phenotypes. This is because the evolved regulatory interactions change the phenotypic neighbourhood in a way which enables evolutionary trajectories to follow fitness gradients to different local optima. (c) The fitness surface with a highly evolved G–P map. This mapping produces phenotypes that can always be improved by small mutations, and evolutionary trajectories thus reliably find the global optima when starting from any point (1000 random starting positions tested, all succeed). (d) Comparing the fitness effects of movements in the same genetic space given different mappings (enlarged area and line plots). The plot shows one trajectory from (a) (red) and one trajectory from (c) (blue), both overlaid on the fitness surface resulting from the G–P map in (a). The depicted evolutionary trajectory (red) gets stuck at a local optimum with low fitness. Trajectories through genotype space with the evolved mapping in (c) (blue) include mutations that would have been deleterious with the direct encoding (see fitness decrease in dashed blue line plot). However, these genetic mutations are accepted by natural selection because they produce beneficial changes to phenotypes given the evolved mapping (see monotonically increasing fitness in blue line plot). Unlike the case with the one-to-one mapping, mutations that decrease the genetic distance to the fittest genotype always increase fitness with the evolved mapping. The evolved mappings achieve this via the evolution of regulatory interactions that bias the distribution of phenotypes produced by genetic mutations such that the expression of trait combinations that conflict with high-fitness phenotypes is suppressed and the expression of trait combinations that are compatible with high-fitness phenotypes is amplified.

constraints of the GRN (defined by $\mathbf{B}$) change over deep time and how they modify the mapping between $\mathbf{g}$ and $\boldsymbol{\phi}_a$, the adult phenotype; and in particular, how the distribution of phenotypes produced under mutations to $\mathbf{g}$ (for a given $\mathbf{B}$) thus have the potential to modify microevolutionary trajectories through phenotype space.

It is not possible for evolution, or any other adaptive process, to infer the underlying structure of fitness a landscape from a single point in that landscape. The prior work of Alon and colleagues thus elaborates the idea that varying goals – changing repeatedly from one selective target to a different (but structurally similar) selective target – enhances the evolution of evolvability to previously seen environments [13, 15, 16, 83] and in some cases (e.g. modularly-varying goals) produces generalisation to previously-unseen environments [15, 16, 19, 69, 83]. In that work it was necessary to hand-design a family of single-peaked selective landscapes sharing common structural regularities. Instead we utilise the observation that the local peaks present in a single multi-peaked landscape

naturally share common structural regularities because they derive from the same set of underlying epistatic constraints [61, 121]. This does not depend on any explicit modularity or other contrivance in the problem structure. This also has the advantage that the fitnesses of different local peaks are comparable because they are on the same landscape. This is important because it enables us to assess not merely whether evolution evolves a given phenotype *more quickly*, but also whether evolution can find a different phenotype that is *fitter*. This ability cannot be assessed when the landscape (at any one time) is single-peaked.

Thus, rather than changing between multiple single-peaked fitness landscapes, we retain the condition of variable selective pressures via a different method; namely, repeated exposures to the single, multi-modal fitness landscape (e.g. a population is repeatedly, but intermittently, exposed to a stress such as an alternate habitat, toxin, or predator population). As per Kashtan et al. [13], this creates a scenario in which evolution alternates between a 'target' selective environment (in this case, multi-peaked) and a 'null' selective environment where evolution is neutral. Kashtan et al. [13] found that alternating between the target and null environments showed no speedup in most cases because there were no structural similarities between the two environments for the developmental constraints to internalise. In our case, however, structural similarities do exist; not between the null and target environments, but between the different local peaks of the single target environment. The null environment is important, however, in enabling evolution to sample multiple local peaks of the target fitness landscape. Interim periods, where selection on the relevant traits is neutral, allows their values to drift and causes the next exposure to start from a different phenotype (i.e. a different location in the fitness landscape). We term each (re-)evolution in the target landscape a microevolutionary 'episode' and we approximate the effects of exposure to the null environment by randomizing $\mathbf{g}$ between episodes (but retaining the slower-evolving developmental constraints, $\mathbf{B}$). In this manner, the slow evolution of developmental constraints responds to the phenotypic correlations selected over a distribution of different local peaks in a multi-peaked fitness landscape. Selection occurring at multiple local peaks of multi-modal fitness landscape is conceptually similar to the notion of selection occurring in multiple, single-peaked fitness landscapes (i.e. the multiple landscapes can be viewed as different local peaks of the same underlying fitness landscape) – but methodologically the former has the advantage that we do not need to hand-design structural similarity (e.g. modularity) into multiple phenotypic targets.

We examine two different classes of fitness landscapes each defined by the superposition of many pairwise sign-epistatic interactions. Natural fitness landscapes contain widespread epistasis [134] (pairwise, or 'order two', epistasis being the minimal case), including sign epistasis – i.e. where a change in one trait may be either beneficial or deleterious depending on the value of another trait [135]. Multiple epistatic interactions can create epistatic constraints that are difficult or impossible to resolve simultaneously (we say that an 'epistatic constraint' between two traits is 'resolved' if the fitness contribution conferred by their epistatic interaction is maximised; see *section 5.4.4*). Local fitness peaks occur when any change that resolves one epistatic constraint causes one or more other constraints, of greater total fitness effect, to be violated, resulting in a net decrease in fitness. Different fitness peaks in the resultant fitness landscape tend to resolve different (but not unrelated) subsets of epistatic constraints and, in general, confer different fitness values.

A least-assumptions model of multi-modal landscapes is provided by random constraints, making no assumptions about the organisation of low-order epistatic interactions.

Note that although the organisation of the epistatic interactions is random, the fact that they are low-order (in this case, pairwise) means that the resultant fitness landscape, although highly multi-modal, is not arbitrary [121]. Whether it is, in principle, possible to simultaneously resolve all epistatic constraints depends on the *consistency* of the constraints in a problem [136] – i.e. in a consistent problem, there exists a configuration that resolves all constraints. With random constraints, consistency is low; many epistatic constraints cannot be simultaneously resolved by *any* phenotypic configuration. Prior work with optimisation using neural network learning [61] shows that the consistency of constraints will influence the success of learning (here, the ability of evolution to improve evolvability). Accordingly, in a second landscape class, *consistent constraints*, we control consistency by making all constraints consistent with a particular target phenotype. Again this landscape exhibits many locally optimal configurations where, although there exist higher-fitness configurations, fitness improvements can require changes to many traits simultaneously. Like the randomly organised problem, this describes a scenario where high-fitness configurations over subsets of traits are incompatible with high-fitness configurations over different but overlapping subsets of traits. Globally optimal configurations where all constraints are resolved exist in the consistent case (but not the random case) but are nonetheless very rare and difficult to find by incremental change (in this class there are only two configurations of 2^n possible configurations that resolve all epistatic constraints simultaneously).

Thus our experiments simulate the evolution of a gene-regulation network on rugged fitness landscapes composed of many low-order sign-epistatic interactions. We study the adaptation of developmental biases and constraints, accumulated over multiple evolutionary episodes that visit multiple peaks on such fitness landscapes over deep evolutionary time. Specifically, we examine the capability of natural selection to find generalised developmental organisations that reflect the structure of these landscapes and thus modify microevolutionary trajectories in a manner that enables evolution to find different adaptive peaks of higher fitness.

5.2 RESULTS

In several of the experiments that follow we compare microevolutionary episodes (from random starting phenotypes): (a) with unbiased development, i.e. a one-to-one G–P map (created by using the identity matrix as the regulatory interactions matrix, $\mathbf{B}$); (b) with an evolved developmental process, i.e. a G–P map controlled by evolved regulatory interactions (itself evolved over many past evolutionary episodes). By comparing the end-points of these episodes we can assess whether the phenotypic distribution evolved over deep time is causing microevolutionary trajectories to find different fitness peaks in the fitness landscape and whether they are higher or lower in fitness.

5.2.1 Evolved Genotype–Phenotype Maps Improve Evolvability

We begin by examining the behavior of a GRN evolving on a multi-modal fitness landscape built from either (i) consistent epistatic constraints, or (ii) random epistatic constraints. Simulations for (i) and (ii) last for 1600 and 4000 evolutionary episodes, and each episode lasts for 5000 and 2000 generations respectively.

We begin by taking advantage of the fact that fit and unfit phenotypes can be easily distinguished by eye in the consistent constraints problem. We define the problem constraints to be consistent with a recognisable image (without loss of generality, in this case, concentric squares). Constraints are defined between neighbouring pixels in the image, but finding the globally optimal phenotype requires evolution to satisfy all these local constraints simultaneously – which is only possible by finding the target image or its complement (see *section 5.4.4*). Fitness trajectories over microevolutionary time on the landscape with consistent epistatic constraints are shown in figure 17a. Here we show microevolutionary trajectories from episodes 1–5, 996–1000 and 1596–1600. The phenotypes found at the end of each microevolutionary episode using GRNs from different stages of deep evolutionary time are shown in the top row of figure 17a. The fitnesses of phenotypes at the start of every microevolutionary episode are similar both early and late in deep evolution, but we see from the end-points of earlier and later microevolution that fitter phenotypes are found later in deep time. Early in deep evolution, phenotypes found by microevolution have small patches that are locally consistent with the target image, but neighbouring patches do not agree, leaving unresolved constraints at the boundaries between these patches. Later in deep evolution, microevolution can find larger patches that are consistent with the target image, i.e. larger subsets of constraints are being resolved simultaneously because developmental biases have learned which combinations of traits work well together.

To further illustrate this progressive improvement in the ability to find high-fitness peaks, we plot the fitnesses of the phenotypes found at the end of every microevolutionary episode for one instance of the consistent and random constraints problems (figures 17c and 17e). We find that during the evolution of the GRN over deep time, microevolution becomes better at discovering high-fitness phenotypes. Although the evolving GRN does not find high-fitness phenotypes reliably early in deep evolution, it slowly becomes better at evolving. Later evolutionary episodes find better configurations by learning phenotypic correlations from locally optimal phenotypes it found in the past. That is, evolution slowly internalises the problem structure that can be inferred from previously visited peaks into its G–P map, and subsequent microevolution is then biased to recreate the structures that have been learned (as shown in the top row of figure 17a).

To highlight the effect these evolved biases of the G–P map have on subsequent microevolution, we compare the evolved G–P map's behavior to that of a one-to-one G–P map (figures 17b and 17d for the consistent and random constraints problems respectively). With a one-to-one G–P map, past experience cannot alter the developmental process (there is no trend over deep time) and microevolution simply follows the fitness gradients of the fitness landscape. Although the evolving GRN and one-to-one G–P map models find phenotypes of similar fitness early in deep evolution, the evolving GRN increases its ability to find fit phenotypes over time (figures 17b to 17e). This clearly illustrates that the evolved biases and constraints in the G–P map enable evolution to find higher-fitness phenotypes than those found by an unbiased evolutionary process (i.e. a one-to-one G–P map). This is true for both problem classes tested, and the results are consistent over 30 independent replicates of each experiment, with 30 different landscapes in the random constraints case (figure 18).

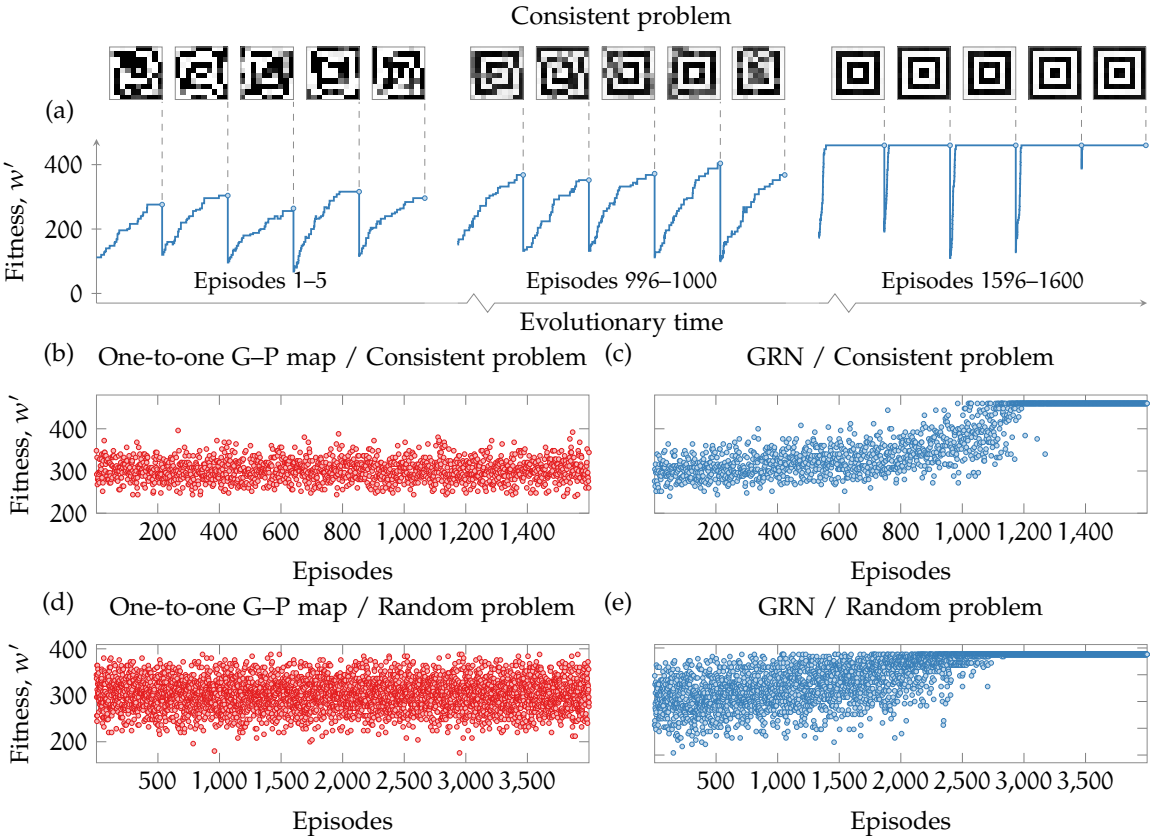


Figure 17: Evolution finds G–P maps that improve evolvability. (a) Fitness trajectories for the GRN over micro-evolutionary time given consistent epistatic constraints. The top row in (a) shows that evolution finds different phenotypes with every new evolutionary episode. With evolutionary time, evolution learns deep structural regularities common to the phenotypes it has previously encountered that allow it to increase evolvability by generalising and successfully resolving the epistatic constraints in the fitness landscape. Although the fitnesses of the initial phenotypes are similar both early and late in deep evolutionary time, micro-evolution at later evolutionary periods finds fitter phenotypes. The phenotypes evolve larger patches that agree with the target image with evolutionary time. By the end, only the globally optimal phenotypes are found. Evolution over deep time given consistent epistatic constraints is shown in (b)–(c), and for random epistatic constraints in (d)–(e). Each point in (b)–(e) shows the fitness of the phenotype at the end of a micro-evolutionary episode (each starting from a random $\mathbf{g}$ phenotype). (b) and (d) show results for a one-to-one G–P map (red), and (c) and (e) for a GRN (green). Since the developmental process does not evolve with the one-to-one G–P map, the distribution of locally optimal phenotypes found shows no trend during the whole simulation. In the same landscapes, with a G–P map evolving slowly over deep time, evolution becomes better at discovering high-fitness phenotypes, eventually finding a very high-fitness phenotype from any starting phenotype. The results shown here are consistent across 30 replicates of the experiments (figure 18).

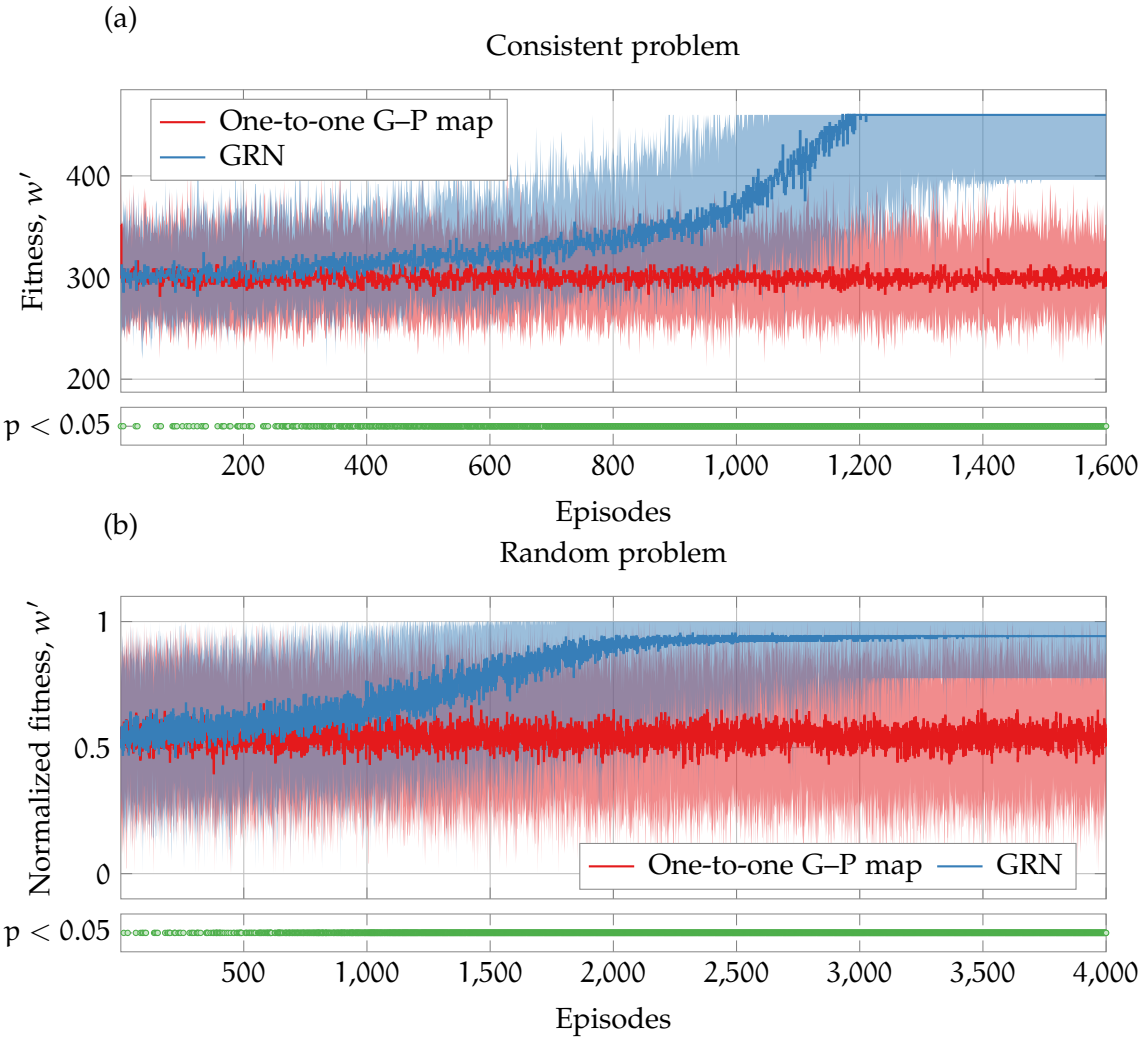


Figure 18: Evolution finds G–P maps that improve evolvability. These plots show the median fitness of the phenotypes found at the end of every microevolutionary episode (each starting from a random phenotype) over 30 replicates of the experiments. The shaded area shows the minimum and maximum fitness found by any of the 30 replicates. For the random constraints experiments, we normalise fitness for each problem instance to the range $[0, 1]$ with respect to the minimum and maximum fitness found by either of the two treatments. p-values shown are from the Mann–Whitney U test. For both problem classes tested, the distribution of phenotypes found by the one-to-one G–P map shows no trend during evolutionary time. This is because there is no developmental process, and thus no G–P map that can learn across deep evolutionary time. When the GRN is evolved on these same problem classes, however, the distribution of phenotypes that evolution finds shows a trend of increasing fitness. With a GRN evolving slowly over deep time, evolution becomes better at discovering high-fitness phenotypes, eventually finding very high-fitness phenotypes from any starting phenotype.

5.2.2 The Evolved Genotype–Phenotype Map Avoids Low-Fitness Local Peaks in the Fitness Landscape

We verify that the increased evolvability of the evolved GRN is due to an ability to avoid low-fitness local optima in the fitness landscape by comparing microevolutionary trajectories given (a) a one-to-one G–P map; and (b) G–P maps taken from different stages of the

previous experiments. We create 1000 individuals with different random $\mathbf{g}$ values (and hence random phenotypes) and then evolve $\mathbf{g}$, using these different G–P maps. To ensure that these differences are really caused by finding different local optima, not merely by finding the same phenotypes faster, we run each microevolutionary episode until phenotypes show no further improvement.

We confirm that phenotypes found with the evolved G–P maps are fitter than those found with the one-to-one G–P map, and that the longer the G–P map is evolved, the better its evolvability (figures 19 and 20). Note that these experiments properly distinguish between individuals that are highly evolved and individuals that have high evolvability. That is, when $\mathbf{g}$ is randomised, phenotypes have similar fitness regardless of the G–P map. Yet, evolution with the evolved G–P maps finds fitter phenotypes (higher end points) and it finds them faster (steeper initial slope). Such concrete evidence for the evolution of evolvability in multi-peaked fitness landscapes has not previously been demonstrated. We also verified that G–P maps that can achieve this level of performance are extremely rare in the space of possible G–P maps (figure 21). This is strong evidence that natural selection has optimised the G–P map of the evolving GRN. Despite the fact that mutations to the G–P map are selected only on the basis of their immediate, short-term fitness consequences, the regulatory interactions that are evolved represent the problem structure, and the evolved developmental biases facilitate the production of high-fitness phenotypes.

5.2.3 The Evolved Genotype–Phenotype Maps Internalise Structural Information About the Selective Environment

To confirm that the evolved G–P map is internalising structural information about the selective environment, as Riedl suggested [2, 3], we analyse the structure of the evolved regulatory interactions after evolving the GRN on the consistent problem. The structural information within the selective environment can be seen in terms of the epistatic interactions defined in the problem matrix (figure 22, first frame; white and black for $+$ and $-$ epistatic interactions between gene i and gene j respectively, grey for no interactions). In episode 1, the regulatory interactions evolved reflect the traits in the one local optimum discovered at the end of that episode. In episode 2, a different local optimum was visited, so conflicting regulatory interactions were evolved. Thus, interactions which existed in both phenotypes were reinforced (black and white), whereas interactions which existed only in one phenotype were weakened (grey). With evolutionary time, after a number of different local optima has been visited, regulatory interactions closely match the structural information which exists in the selective environment (episodes 100 and 800). Both positive and negative regulatory interactions have evolved along the diagonals, but the rest of the regulatory interactions remain close to zero, as is the case in the problem matrix. At this stage, the developmental architecture has successfully internalised structural information about the selective environment.

We find that this developmental architecture serves as an intermediate phase between learning the problem structure and finding a system of constraints that enables evolution to find the global optimum reliably. Since we know what the globally optimal phenotype is in the consistent problem, we also know what the optimal regulatory interaction matrix is: the matrix whose regulatory interactions match the pixel correlations in the concentric squares pattern. In a multi-modal fitness landscape, such as the one we use here, the phenotype found by evolution at the end of every microevolutionary episode is, in practically

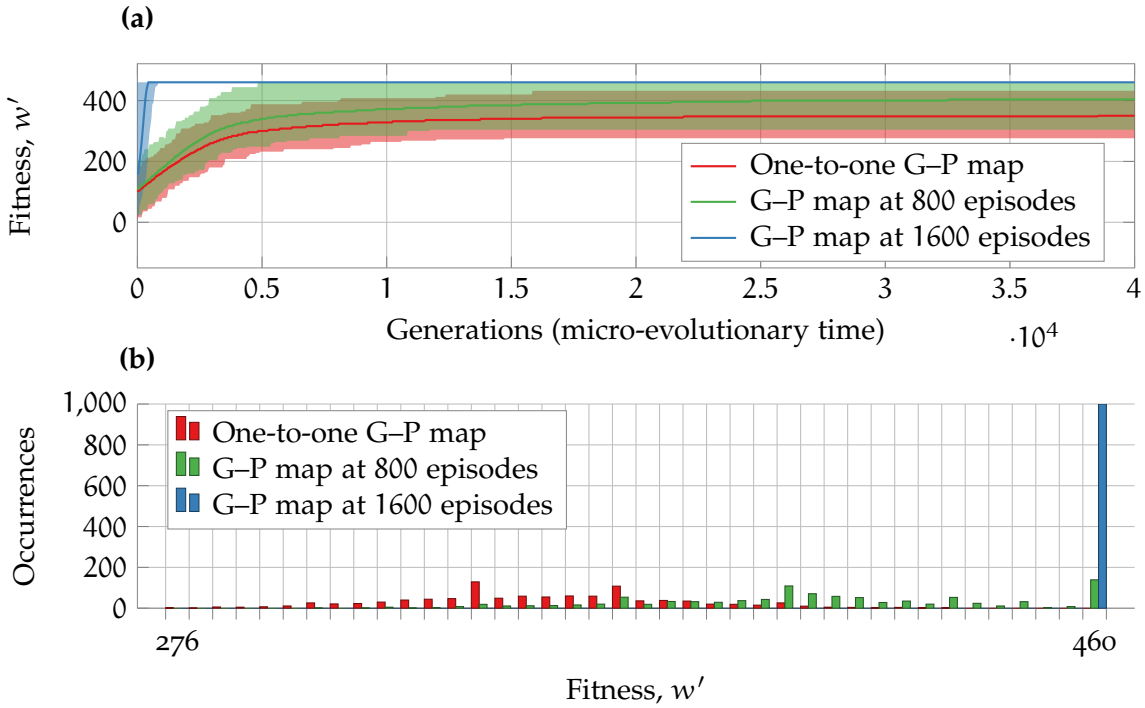


Figure 19: The evolved G–P map improves the ability of microevolution to avoid low-fitness local peaks in the consistent constraints fitness landscape and find higher-fitness peaks. This figure shows evolution over microevolutionary time, given different G–P maps, rather than only the fitnesses found at the end of microevolutionary episodes as in figure 17. We use G–P maps taken from the experiment shown in figure 17c, freeze their deep evolution (i.e. freeze the evolution of the G–P map $\mathbf{B}$, but allow evolution of the genotype $\mathbf{g}$), and probe the effect they have on subsequent microevolution. We randomly initialise 1000 $\mathbf{g}$ vectors drawn from a uniform distribution in the range $[-1, 1]$ and let them microevolve for 40000 generations (instead of 5000 as in the previous experiment) using three G–P maps: a one-to-one mapping, and two mappings taken from episodes 800 and 1600 from the experiment shown in figure 17c. (a) Lines show median fitness, with the shaded areas showing the minimum and maximum fitness found by any of the 1000 individuals for every evolutionary time-step. Microevolution with the less-evolved G–P maps is stuck on lower-fitness phenotypes, but microevolution with more-evolved G–P maps avoids these local peaks in the fitness landscape to find different, higher-fitness phenotypes. Additionally, microevolution with the more-evolved G–P maps requires less evolutionary time to find the phenotypes it settles on as can be seen from the steeper slope. (b) End points of (a) shown in histogram format (40 equally sized buckets). Both evolved mappings find a fitter distribution of phenotypes than the one-to-one mapping. The evolved maps find phenotypes that are not just of higher fitness on average, but of higher fitness than *all* phenotypes found with the one-to-one G–P map.

every case, not the global optimum. However, the different local optima found at the end of every microevolutionary episode provide evolution with different, partially contradictory and partially consistent, information about combinations of phenotypic traits that are fit. Given sufficiently many examples, the evolving GRN begins to average-out the contradictory information and amplify the consistent information. Therefore, we hypothesise that the evolved regulatory interaction matrix will, in the medium term, become similar to the problem matrix – because the constraints in the problem matrix reveal themselves

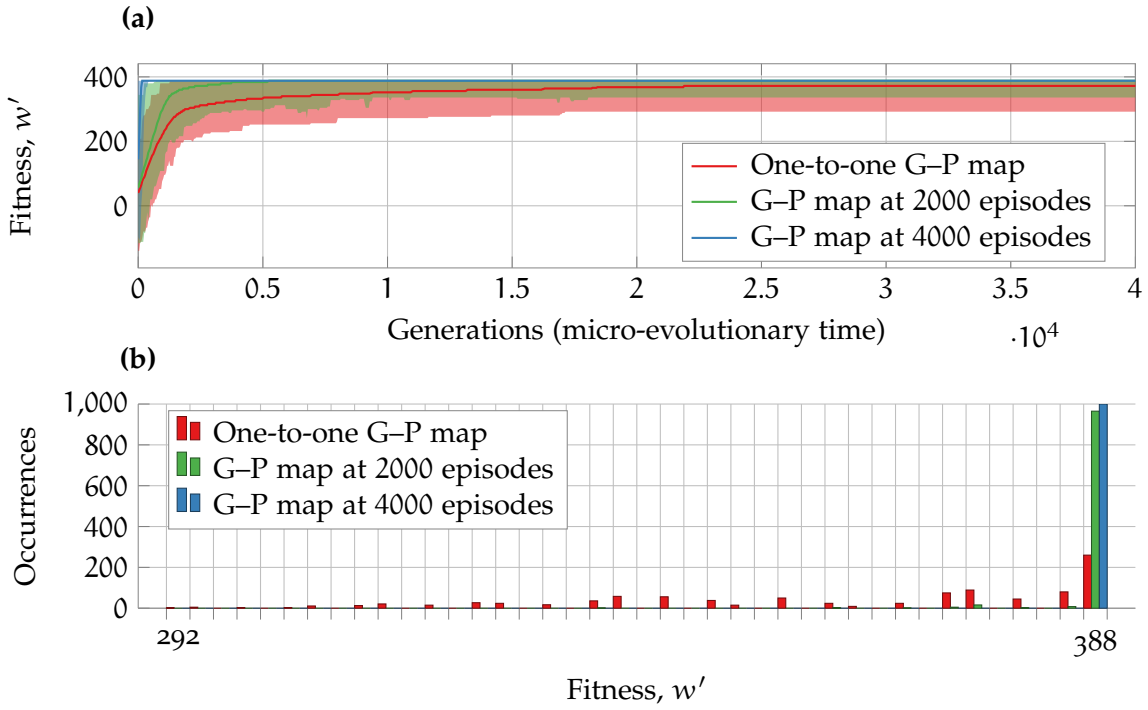


Figure 20: The evolved G–P map improves the ability of microevolution to avoid low-fitness local peaks in the random constraints fitness landscape and find higher-fitness peaks. This figure shows evolution over microevolutionary time, given different G–P maps, rather than only the fitnesses found at the end of microevolutionary episodes as in figure 17. We use G–P maps taken from the experiment shown in figure 17e, freeze their deep evolution (i.e. freeze the evolution of the G–P map $\mathbf{B}$, but allow evolution of the genotype $\mathbf{g}$), and probe the effect they have on subsequent microevolution. We randomly initialise 1000 $\mathbf{g}$ vectors drawn from a uniform distribution in the range $[-1, 1]$ and let them microevolve for 40000 generations (instead of 2500 as in the previous experiment) using three G–P maps: a one-to-one mapping, and two mappings taken from episodes 2000 and 4000 from the experiment shown in figure 17e. (a) Lines show median fitness, with the shaded areas showing the minimum and maximum fitness found by any of the 1000 individuals for every evolutionary time-step. Although microevolution with the less-evolved G–P maps is able to find higher-fitness phenotypes than in the earlier experiment due to it being allowed to microevolve for longer (figure 17), microevolution with the more-evolved G–P maps avoids local peaks in the fitness landscape more reliably and finds higher-fitness phenotypes on average. When comparing the slope of the trajectories, we can see that microevolution with the more-evolved G–P maps settles on high-fitness phenotypes faster than microevolution with the less-evolved G–P maps. (b) End points of (a) shown in histogram format (40 equally-sized buckets). Both evolved mappings find a fitter distribution of phenotypes than the one-to-one mapping. The evolved maps find the best phenotypes found by the one-to-one G–P map more reliably.

in the fitness landscape as the most consistent 'signal' that evolution can identify (this is the imitatory organisation that Riedl described). But later, we would expect the evolved GRN to 'fill-in' entries in this matrix that were unspecified in the problem matrix – and eventually, to identify values that correspond to the optimal solution matrix (figure 22, last frame). Although the global optimum might not yet have been visited, this filling-in of values that agree with the global optimum is nonetheless possible because these inter-

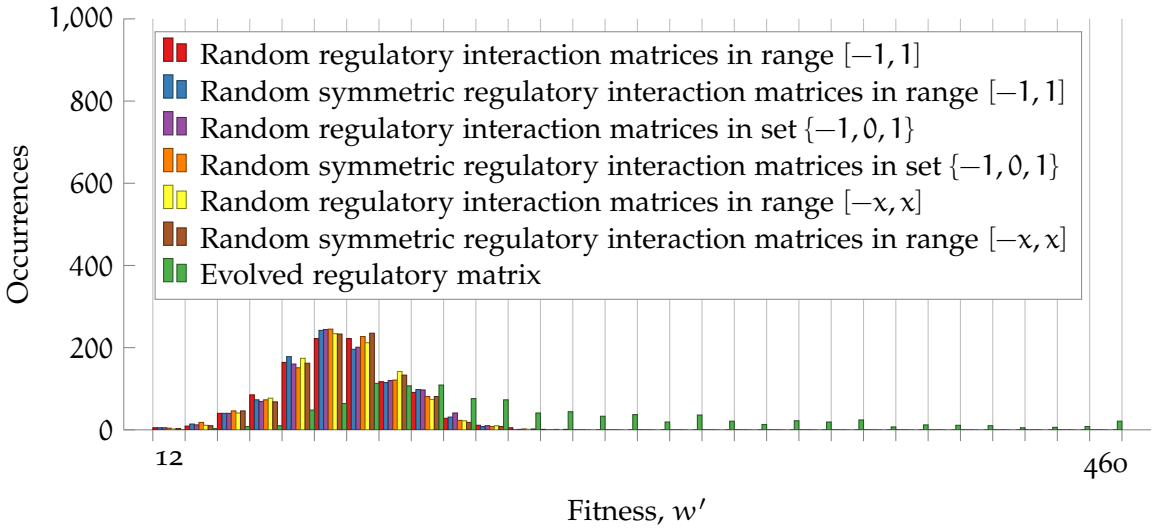


Figure 21: Natural selection has optimised the G–P map for long-term evolvability. We randomly initialise 1000 $\mathbf{g}$ vectors drawn from a uniform distribution in the range $[-1, 1]$ and evaluate the adult phenotypes they develop into using different regulatory interaction matrices. For each $\mathbf{g}$ vector, we randomly initialise a regulatory interaction matrix $\mathbf{B}$ in different ranges and sets (see legend; x is the mean value of all magnitudes in the final, evolved matrix coming from figure 17c), and evaluate the phenotype $\mathbf{g}$ is developed into given the interaction matrix $\mathbf{B}$. Since selective pressures in the fitness landscape are symmetric and our evolved matrices are also approximately symmetric, we force some of the regulatory interaction matrices we generate to be symmetric. For comparison, we evaluate the adult phenotypes that are produced from the same 1000 $\mathbf{g}$ vectors, but using the regulatory interaction matrix taken from the end of the experiment shown in figure 17c. We find no significant difference in performance between symmetric and non-symmetric randomly generated matrices. On the other hand, the evolved G–P map develops phenotypes of higher fitness on average. This is because natural selection optimised the G–P map of the evolving GRN. Even though it was selected for immediate, short-term fitness advantages, the regulatory interactions that are evolved represent the problem structure, and the evolved developmental biases facilitate the production of high-fitness phenotypes.

actions are the ones that are most reliably consistent with the correlations observed over a distribution of local optima [19].

Our data confirms that this behavior, previously studied in a learning neural network [61], also occurs in the evolving GRN. At the end of the simulation, the evolved regulatory interaction matrix closely resembles the addition of the problem and optimal matrices as described above (figure 22; second and last frames). A clear separation between positive and negative regulatory interactions persists along the diagonals, and the rest of the regulatory interactions also evolve to up-regulate or down-regulate in a way which is consistent with the globally optimal phenotype. The evolving G–P map thus goes beyond internalising structural information about the selective environment, finding a system of constraints that enables evolution to find the global optima reliably.

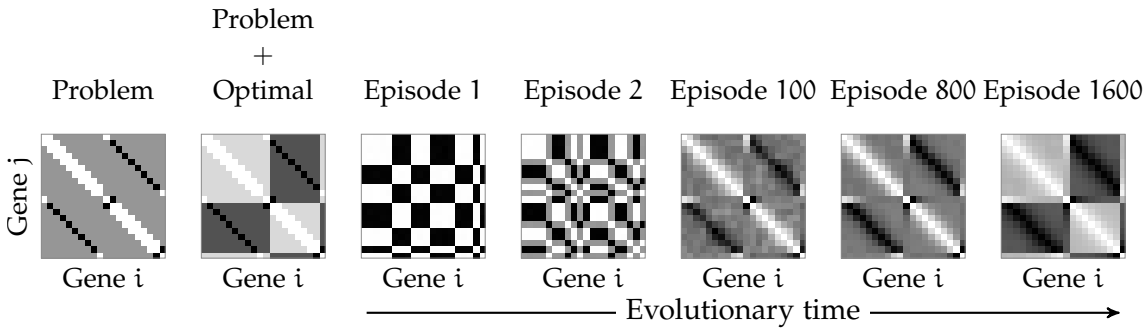


Figure 22: The evolving G–P map goes beyond internalising structural information about the selective environment, finding a system of constraints that enables evolution to find the global optima reliably. The first frame shows a subset (20×20 out of 100×100) of the consistent problem matrix ($\mathbf{C}$; see *section 5.4.4*), and frames 3–7 show visual representations of a subset of the regulatory interaction matrices at different stages of deep evolution (also 20×20 subsets; white for maximally positive numbers at that episode, black for maximally negative numbers and greyscale for intermediate values). The second frame shows a 20×20 subset of the regulatory interaction matrix we approximate evolution will evolve due to visiting a large number of local optima and not only the global optima (see text). This set of regulatory interaction enables evolution to find the global optima reliably. The evolved G–P maps at episodes 1 and 2 are consistent with the local optima discovered at the end of each respective episode, but differences between the two phenotypes cause evolution to reinforce interactions which exist in both phenotypes (black and white) while weakening interactions which exist in only one (grey). After a number of local optima have been visited, the evolved regulatory interactions (episodes 100 and 800) closely resemble the problem matrix (first frame). Both positive and negative regulatory interactions have evolved along the diagonals (darker and lighter colors), but the rest of the regulatory interactions remain close to 0, as is the case in the problem matrix. At this point, the evolved G–P maps have internalised structural information about the selective environment – i.e. what values patches of neighbouring pixels should take to confer fitness benefits, but not how these patches should be combined at the boundaries to produce the globally optimal phenotypes. This developmental architecture serves as an intermediate phase between learning the problem structure and finding a system of constraints that enables evolution to find the global optima reliably. At the end of the simulation (episode 1600), the evolved regulatory interaction matrix closely resembles this approximated optimal matrix (second frame). As is the case earlier in evolution, both positive and negative regulatory interactions have evolved along the diagonals (darker and lighter colors). However, the rest of the regulatory interactions don't remain close to zero, but instead evolve to up-regulate or down-regulate in a way which is consistent with the globally optimal phenotypes.

5.2.4 Evolvability Evolves as a Systematic Byproduct of Short-term Selection for Fitness

Here we begin to unpack how the evolution of evolvability occurs in these experiments. Since the long-term fitness consequences cannot be the reason that natural selection evolves these highly evolvable G–P maps, we need to identify the immediate selective pressures in operation and how they have this effect on long-term evolvability.

To understand how short-term selective pressures affect long-term evolvability, we set $\mathbf{g}$ to match the concentric squares image and evolve only the regulatory interactions $\mathbf{B}$.

We examine how the fitness of the adult phenotype produced changes as the regulatory interactions evolve. Note that $\mathbf{g}$ has to develop according to the regulatory interactions, $\mathbf{B}$, and these interactions are initially absent and slowly evolve. Because of that, the evolution of $\mathbf{B}$ has a fitness effect even though $\mathbf{g}$ already matches the concentric squares image. We also examine how the developmental basin of attraction of the concentric squares phenotype changes – i.e. the number of $\mathbf{g}$ vectors that map into the concentric squares phenotype.

When GRN interactions evolve they increase the fitness of the phenotype; otherwise they would not be selected by short-term selection. But they also have an effect on the distribution of phenotypes produced by that GRN under subsequent mutations to $\mathbf{g}$. Specifically, both the fitness of the adult phenotype and the developmental basin of attraction of the concentric squares phenotype increase with evolutionary time (figure 23). That is, regulatory interactions that increase immediate fitness necessarily have the systematic side-effect of increasing the number of genotypes that map into that same phenotype. This is because the evolved regulatory interactions reinforce correlations which exist in the current phenotype and these correlations bias the distribution of phenotypes that can be produced by the developmental process towards reproducing these same evolved correlations [19]. In this sense, we say that the genetic robustness of the selected phenotype is increased – the number of $\mathbf{g}$ vectors that map into a specific phenotype is increased. This increase in the developmental basin of attraction of the phenotype is functionally equivalent to the development of an associative memory for the current phenotype [19].

This robustness does not, in itself, explain the evolution of evolvability – indeed, an inability to produce different phenotypes opposes evolvability. But what happens to developmental basin sizes when regulatory interactions evolve slowly over multiple selected phenotypes as in our main experiments?

Given that the sum of all basins of attraction has a fixed size (the total configuration space of $\mathbf{g}$), an increase in one basin implies a decrease in the combined size of the others. Thus, when a GRN is evolved over a distribution of phenotypes, not all of their basins can increase indefinitely and, to first approximation, the phenotypes that are selected for most often will eventually out-compete all other phenotypes. However, evolving interactions store information about phenotypic correlations rather than the specific values of individual phenotypic traits [19]. Accordingly, when one basin increases others will decrease, but the basins of phenotypes that are structurally similar to the currently selected phenotype will reduce only a little compared to the basins of phenotypes that are structurally unrelated. Over a distribution of selected phenotypes, the GRN learns a generalised model of those phenotypes – i.e. it learns their correlation structure – and the phenotypes whose basins are enlarged the most are those that agree with the most phenotypic correlations observed over that sample of selected phenotypes (see Watson et al. [61] for an illustration of this in neural network learning).

We use the consistent problem to confirm that as the GRN evolves, the size of the basin of local optima decreases and the size of the basin of the global optimum increases even though the global optimum has not yet been visited (figure 24). This, combined with the fact that evolved regulatory interactions are consistent with the globally optimal phenotype (figure 22), explains how microevolutionary trajectories are able to follow fitness gradients and find the global optimum even though phenotypes of similar fitness (i.e. starting in the same place) with a one-to-one or less-evolved developmental G–P map may have instead followed fitness gradients to a local optimum (as depicted in figures 16 and 17a).

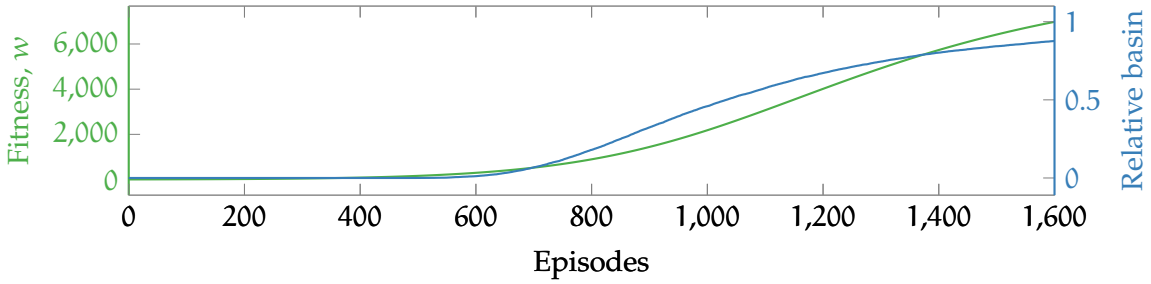


Figure 23: Selection for fitness increases the developmental basin of attraction for the target phenotype. Without loss of generality, we make $\mathbf{g}$ equal to the target phenotype of the consistent constraints problem, and evolve *only* the regulatory interactions under natural selection for 1600 evolutionary episodes. We measure how the fitness of phenotypes developed from $\mathbf{g}$ changes with evolutionary time and also how the developmental basin of attraction of the target phenotype changes with time. The fitness of the adult phenotype produced increases with evolutionary time (green; left y-axis) due to the developmental interactions evolving in the direction of the target phenotype, thus amplifying the adult phenotype's gene expression levels (this figure does not use fitness of discretised phenotypes; see *section 5.4.4*). The proportion of the developmental basin of attraction of the target phenotype also changes with evolutionary time. This is measured by randomly sampling the $\mathbf{g}$ space (here, sample size is set to 10000 in a uniform distribution in the range $[-1, 1]$), developing all $\mathbf{g}$ vectors using G–P maps taken from the end of every evolutionary episode, and reporting the proportion of $\mathbf{g}$ vectors that develop into the target phenotype. The basin of attraction of the target phenotype increases with evolutionary time (blue; right y-axis). The target phenotype's basin of attraction does not see an immediate increase because earlier in evolution the regulatory interactions are too 'weak' to solely determine the adult phenotype produced. As the regulatory interactions become stronger with evolutionary time, the basin of attraction increases. By the end of the simulation, the basin of attraction of the target phenotype enlarges to encompass almost the entire space, such that $\approx 90\%$ of the sampled $\mathbf{g}$ vectors develop into the target phenotype.

5.3 DISCUSSION

Our results demonstrate conditions where short-term selection adapts developmental constraints such that they facilitate long-term adaptation on rugged fitness landscapes. This is possible when the landscape is composed of low-order epistatic interactions because this presents structural regularities (i.e. phenotypic correlations) that a GRN can represent and can acquire through selection. We show that evolvability is an indirect but necessary consequence of selection for immediate, short-term fitness increases. Robustness of the currently selected phenotype is also a side effect of such changes. But counter to naive intuitions, robustness does not oppose evolvability – in fact, canalisation (of the right things) is necessary for evolvability [3, 7, 8, 69]. In this work we show that a GRN can evolve to favour fit phenotypic correlations, rather than merely favouring individual phenotypic traits, if it is exposed to selection that is representative of fit correlations over evolutionary time. And we have shown that this is the case when a GRN is repeatedly exposed to a multi-modal fitness landscape causing it to visit a distribution of locally optimal phenotypes. We speculate that negative frequency dependence, causing a population to continually explore phenotype space [137], may produce similar results to the model of repeated exposures to an otherwise static stress environment used here.

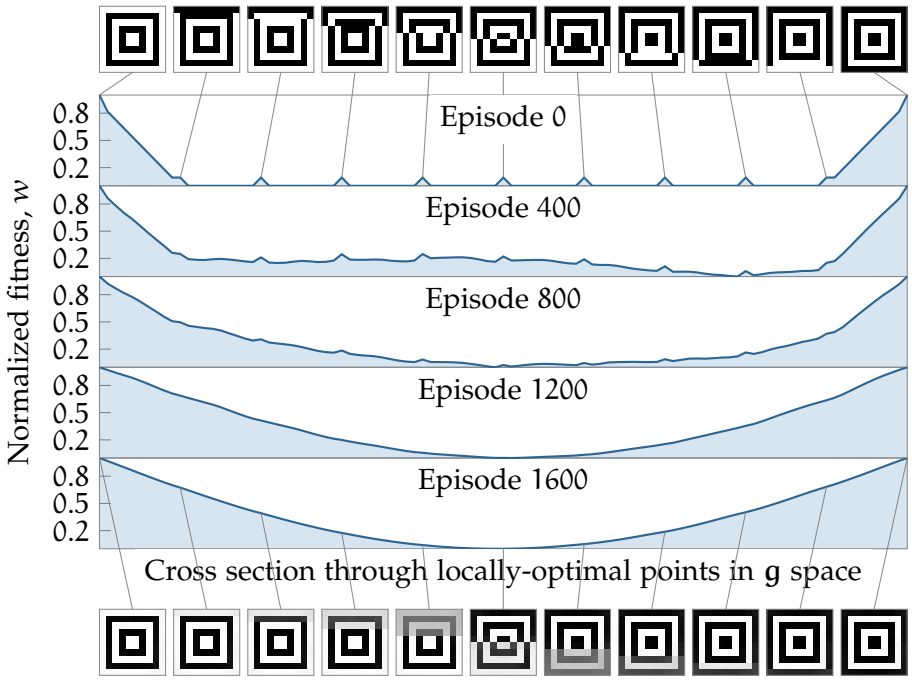


Figure 24: The evolving G–P map enhances evolvability by enlarging the basins of the global optima and removing basins for local optima. We evolve a GRN on the consistent constraints problem for 1600 episodes, and store all G–P maps from the end of every episode. We then use a cross section through the space of $\mathbf{g}$ values that starts from the global optimum (concentric squares) and passes through a number of local optima in the fitness landscape. The genotypes in this cross-section are developed into adult phenotypes using all 1600 G–P maps from the previous simulation, and the adult phenotypes are evaluated. The fitnesses of phenotypes developed with each G–P map, in each cross-section, are normalised for comparison. In the case of a one-to-one map (first frame; episode 0), the adult phenotypes produced are equal to the $\mathbf{g}$ vectors, and the fitness landscape is multi-modal, meaning that evolution can get trapped on local optima. The top row shows the locally optimal adult phenotypes produced. Over many evolutionary episodes the GRN changes. This does not change the mapping from phenotypes to fitness, but it does change the mapping from genotypes ($\mathbf{g}$) to phenotypes, as evident from the differences in the phenotypes depicted at the top and bottom rows. Accordingly, the fitness landscape for this cross-section through $\mathbf{g}$ changes over evolutionary time. Specifically, the fitness landscape produced by the phenotypes developed from this cross-section of $\mathbf{g}$ becomes progressively smoother, eliminating local optima and enlarging the basin of attraction of the global optimum. Thus the fitness landscape available to selection slowly becomes effectively uni-modal when augmented by the evolving G–P map.

As evolved developmental biases and constraints internalise the structure of the fitness landscape, the distribution of phenotypic variation produced by development changes to favour these correlations (enriches for the production of phenotypes that contain these correlations). This has the consequence that high-fitness phenotypes are found faster and with greater probability. Although natural selection is myopic, selective pressures under these conditions result in a G–P map that acts like a longer-term memory and allows subsequent evolution to effectively 'ignore' locally optimal configurations that could potentially trap evolution with an unbiased G–P map (figure 16). Thus, our results show the

evolution of long-term evolvability as a consequence of selection for immediate fitness differences.

These behaviors can be explained by simple learning concepts. If the past is a good indicator of the future, learning from the past can improve a model's ability to perform well in environments not previously experienced. This requires, however, that natural selection has an appropriately powerful 'model space', capable of representing the way in which phenotypes that were fit in the past are 'similar' to novel high-fitness phenotypes [133]. A one-to-one G–P map cannot do this unless the similarity is simply based on Euclidean distance in phenotype space. But an evolving GRN is capable of internalising information about phenotypic correlations. When this is a good model of the structure present in the selective environment, as it is when the environment contains low-order epistasis, this results in improved evolvability. More generally, the relationship between the evolution of evolvability and generalisation in learning systems helps us understand the conditions for, and limitations of, the evolution of evolvability [69]. This has enabled us to demonstrate the first formal model for the evolution of evolvability capable of showing and explaining the evolution of evolvability that facilitates future innovation from short term selection pressures. Contrary to popular assertions, this is not impossible – just as predicting the future by generalising from past experience is not impossible in learning systems. This is an example of how the transfer of machine learning theory to evolutionary theory [19, 66, 69, 132, 133] has potential to demystify the evolution of evolvability, and help us understand how it works and the conditions that enable it.

5.4 METHODOLOGY

5.4.1 Gene-Regulation Network

We simulate the evolution of a gene-regulation network (GRN) using a non-linear, recurrent developmental process, as described in section 4.1.1.

5.4.2 Evolutionary Model

The evolutionary model follows the same setup described in section 4.1.2, with some differences. A description of the differences follows.

Instead of mutations on the regulatory interactions, $\mathbf{B}$, occurring probabilistically every generation, mutation and selection on $\mathbf{B}$ is approximated with a selection-limited model of evolution applied at the end of every evolutionary episode (see *section 5.4.3*).

Additionally, rather than changing between multiple single-peaked fitness landscapes, we repeatedly alternate between a 'null' environment (where the genotype drifts) and a multi-modal fitness landscape (see *section 5.1.3*).

5.4.3 Selection-Limited Evolution

Under *directional* selection, the cumulative effect of a large number of small mutations is equivalent to the effect of a small number of large mutations when controlling for variance. Assuming that the number of generations required for $\mathbf{g}$ to align itself with a local optimum and stabilise is small compared the total number of generations within an

evolutionary episode, the effect natural selection has in any evolutionary episode can be modeled as follows. Hill-climbing selection is applied to all elements in $\mathbf{B}$ (random order; no replacement) by testing two mutations: one for positive directional selection drawn from a distribution with mean q and one for negative directional selection drawn from a distribution of $-q$, each with a standard deviation of σ. Only one mutation can be selected for under directional selection. Experiments use $q = 3 \times 10^{-6}$ and $q = 2 \times 10^{-6}$ for the consistent and random constraints problem classes respectively. A standard deviation of $\sigma = 0.01q$ is used for both problem classes. This combination of control parameters was chosen empirically. Multiple different combinations were used before we settled on this combination which enables the discovery of high-fitness phenotypes without 'over-fitting' to a locally optimal, low-fitness peak [69].

Essentially, we are simulating the cumulative effect of and 40^2 and 100^2 discrete mutations for the random and consistent problem classes respectively (coming from the number of pairwise interactions in each problem class), each with equal probability of being accepted given directional selection, all occurring before each new evolutionary episode. Note that there is a trade-off between (a) the number of mutations that occur before each new evolutionary episode; and (b) the number of evolutionary episodes that are simulated in total. If we increase the number of evolutionary episodes, the number of mutations can be decreased, but this means that more mutation–selection cycles must be simulated. For our purposes, we find that 1600 evolutionary episodes are sufficient for the consistent constraints problem, and 4000 evolutionary episodes for the random constraints problem.

5.4.4 Epistatic Fitness Landscapes

We utilise multi-modal selective environments defined by the sum of a large number of pairwise sign-epistatic interactions. The fitness, w, of a phenotype, $\boldsymbol{\phi}_\alpha$, is defined by a matrix of epistatic interactions, $\mathbf{C}$, as follows:

$$w(\boldsymbol{\phi}_\alpha) = \boldsymbol{\phi}_\alpha \mathbf{C} \boldsymbol{\phi}_\alpha^{\mathsf{T}}. \tag{9}$$

$\mathbf{C}$ is an $n \times n$ symmetric matrix whose elements, c_{ij}, determine the nature of the epistatic interaction between traits i and j. All features of the fitness landscape are thus defined by the epistatic constraints defined by $\mathbf{C}$. We say that the 'epistatic constraint' between two traits is 'resolved' if and only if the fitness contribution from their interaction is maximised. This is easily achieved for a single epistatic interaction (either by "$++$" or "$--$" for positive epistasis, i.e. $c_{ij} > 0$, or by "$+-$" or "$-+$" for negative epistasis, i.e. $c_{ij} < 0$, where $+$ and $-$ represent the maximal positive and negative traits values respectively), but multiple epistatic interactions between many traits can create epistatic constraints that are difficult or impossible to resolve simultaneously.

We utilise two different classes of such fitness landscapes. Random constraints uses a problem matrix whose diagonal elements take the value 1, $c_{ii} = 1 \forall i$, and the rest are randomly initialised to a fixed-magnitude, variable-sign value with equal probability: $c_{ij} \in \{-d, d\}$. Here, we use $d = 1$ and set the problem size to $n = 40$. This results in a problem with low consistency, meaning that, for example, there exists no phenotype that resolves the constraints between A and B, and B and C, and also resolves the constraint between C and A. In this fitness landscape, the globally optimal phenotype tends not to be able to resolve all epistatic constraints.

In the consistent constraints problem class, the global optimum resolves all constraints. Here, $\mathbf{C}$ is defined from a given target phenotype in a way that makes the target phen-

otype and its complement the globally optimal phenotypes. Without loss of generality, we choose the target phenotype to be a 10×10, black-and-white image of concentric squares. In doing so, we are able to provide an easily-interpretable visual representation of phenotypes (see figure 17a). The problem matrix is generated by calculating the outer product of the flattened image vector, $\mathbf{x}, x_i \in \{-1, 1\}$, and setting all elements of the resulting matrix that are non-neighbouring pixels with respect to the target image to zero. Because the problem matrix contains information about neighbouring pixels only (since all non-neighbouring pixels are set to 0), local optima are caused by conflicting constraints arising when local patches of constraints are resolved in mutually incompatible ways, leaving unresolved conflicts at the boundaries between these patches.

Whenever the sign of a trait resolves more constraints than it violates, equation (9) defines directional selection that rewards increases in the trait magnitude. This creates a sustained selective pressure for genetic mutations that alter the G–P map [12, 61]. The hypotheses tested in this paper, however, concern the ability to discover different phenotypes and, in particular, different local optima in the fitness landscape. Accordingly, we want to know whether the phenotypes that are found are fitter merely because they have traits with larger magnitudes (which can receive higher fitness under the directional selection of equation (9)) or whether they have a different pattern of traits present in the phenotype satisfying a different combination of epistatic constraints. In particular, the GRN has $n = |\mathbf{g}|$ evolvable parameters that can affect the magnitude of each trait (as well as the correlations between traits), whereas the one-to-one G–P map can only alter the elements of $\mathbf{g}$. Accordingly, to remove any unfair advantage, our results report fitness using

$$w'(\boldsymbol{\phi}_a) = w(\theta(\boldsymbol{\phi}_a)), \tag{10}$$

where $w(x)$ is the fitness function given by equation (9) and $\theta(y) = 1$ if $y > 0$ else -1.

This ignores fitness differences created by merely increasing the magnitude of a trait and instead counts the number of satisfied epistatic constraints. It is in this sense that we refer to 'different' phenotypes, and different local optima in the fitness landscape, i.e. disregarding differences in magnitudes that do not change the number of satisfied epistatic constraints. This ensures that fitness improvements are due to finding different phenotypic patterns and not merely due to higher expression levels – because it has found good phenotypes by learning what traits work well together, rather than learning to produce large traits.

5.4.5 Cross-Sections Through Genotype Space

To generate the cross-sections through genotype space that are used in figures 16 and 24, we utilise the global optimum of the fitness landscape: $\mathbf{s} = \langle s_1, s_2, \ldots, s_n \rangle, s \in \{-1, 1\}^n$, where n is the number of gene expression potentials.

Figure 24 utilises a one-dimensional cross-section through a subspace of $\mathbf{g}$ vectors. Here, we want to generate a cross-section that begins from $\mathbf{g}_s$ which corresponds to the global optimum $\mathbf{s}$ and ends at $\mathbf{g}'_s$ which corresponds to the global optimum's complement $\mathbf{s}'$, by flipping one bit at a time. So the cross-section consists of $(n + 1)$ $\mathbf{g}$ vectors:

$$\mathbf{g}_i = \langle g_1, g_2, \ldots, g_n \rangle,$$

where $i = [\![1, 2, \ldots, n+1]\!]$. The individual bits of $\mathbf{g}_i$ for a given i are determined by

$$g_{ik} = \begin{cases} -s_k & \text{for } i > k \\ s_k & \text{otherwise.} \end{cases}$$

where $k = [\![1, 2, \ldots, n]\!]$. Thus, $\mathbf{g}_1 = \mathbf{g}_s$ and $\mathbf{g}_{n+1} = \mathbf{g}'_s$. This cross-section maintains the property that local optima on the surfaces depicted in figure 24 are true local optima in the original high-dimensional fitness landscape also.

Figure 16 utilises a two-dimensional cross-section of similar construction, with each dimension being a particular cross-section through a subspace of $\mathbf{g}$ vectors. First, we split the global optimum, $\mathbf{s}$, in two:

$$\mathbf{s}_a = [\![s_1, s_2, \ldots, s_{\frac{n}{2}}]\!]$$

$$\mathbf{s}_b = [\![s_{\frac{n}{2}+1}, s_{\frac{n}{2}+2}, \ldots, s_n]\!]$$

Similar to the one-dimensional cross-section, we generate a cross-section for each target ($\mathbf{s}_a$ and $\mathbf{s}_b$) by flipping one bit at a time. The first cross-section begins from $\mathbf{s}_a$ and ends at its complement, $\mathbf{s}'_a$. The second cross-section begins from $\mathbf{s}_b$ and ends at its complement, $\mathbf{s}'_b$. So each cross-section consists of $\frac{n}{2} + 1$ vectors:

$$\mathbf{x}_i = \left\langle x_1, x_2, \ldots, x_{\frac{n}{2}} \right\rangle \text{ and}$$

$$\mathbf{y}_j = \left\langle y_1, y_2, \ldots, y_{\frac{n}{2}} \right\rangle,$$

where $i = j = [\![1, 2, \ldots, \frac{n}{2} + 1]\!]$. The individual values of $\mathbf{x}_i$ and $\mathbf{y}_j$ for a given i and j are determined by

$$x_{ik} = \begin{cases} -s_a^k & \text{for } i > k \\ s_a^k & \text{otherwise.} \end{cases}$$

$$y_{jk} = \begin{cases} -s_b^k & \text{for } j > k \\ s_b^k & \text{otherwise.} \end{cases}$$

where $k = [\![1, 2, \ldots, \frac{n}{2}]\!]$. By concatenating two vectors, one from each of the two cross sections, we can form a $\mathbf{g}$ vector that can be developed into an adult phenotype. Similar to the one-dimensional cross-section described earlier, the local optima on the surfaces depicted in figure 16 are also true local optima in the original fitness landscape.

6 FUTURE WORK

6.1 SOLVING HARDER PROBLEMS

Recent work has shown that the tension between over-fitting and generalisation in machine learning, and the tension between robustness and evolvability in evolution are equivalent [69]. More than that, it is shown that machine learning techniques that alleviate over-fitting to promote generalisation (e.g. by preferring sparsity via connection minimisation) can also be used to alleviate extreme robustness and promote evolvability.

In the typical GRN model, the regulatory interactions are represented by a fully connected matrix of pairwise interactions. Due to the high connectivity in the interaction matrix, over-fitting to the wrong phenotype early during evolution is a common occurrence. To counteract that, extremely low mutation rates are used. Not only does this requirement raise questions about the generality of the results, but it also prohibits testing on harder problems due to the prohibitive number of generations simulations would need to run for to yield meaningful results.

But we now know that it is possible to circumvent the mutation rate requirement by utilising regularisation techniques from machine learning to prevent over-fitting [69]. Higher mutation rates should enable the GRN model to solve harder, perhaps more impressive, problems, and also alleviate concerns about the generality of the results.

6.2 GENERALISING FROM MANY TO ONE

In chapter 5, we show that the GRN model evolved over a distribution of locally-optimal phenotypes on a multi-modal fitness landscape composed of low-order, sign-epistatic interactions can generalise to learn one phenotype: the global optimum. But generalisation is not usually associated with the exhibited behaviour.

When machine learning advocates, and recently computational biologists, think of the word 'generalisation', they associate it with the idea of learning the regularities of a sample of data (training set) to be able to generalise over another sample consisting of unseen data (test set). This is not what we are showing. The GRN *does* learn the regularities of a small sample (locally optimal phenotypes), but it actually generalises to one specific phenotype (globally optimal phenotype). So the question is whether this is in fact generalisation.

An alternative word to describe this behaviour would be 'specialisation'. Since the population is repeatedly exposed to the same fitness landscape (i.e. the same task), the fact that the developmental process always produces the global optimum suggests that the GRN has achieved specialisation – it has learned the optimal behaviour to achieve the task.

But does 'specialisation' only describe the end product – the fact that the GRN has found the globally optimal phenotype? Is the progression towards the global optimum

generalisation or is it still specialisation because evolution with the GRN progressively finds fitter phenotypes (figure 17)?

Evidently, there is a lack of conceptual clarity about what exactly it is the results are showing. Although in recent years it has become commonplace in computational biology to associate evolvability with generalisation (mainly by evolving towards modularly-varying goals), it is difficult to see an argument against the notion that finding the globally optimal phenotype in a multi-modal fitness landscape can be construed as evolvability. But the question is whether this is a new type of evolvability.

6.3 RELATIONSHIP WITH MODEL-BUILDING ALGORITHMS

A common theme in evolutionary computation is the notion of model-building. Model-building algorithms iteratively improve their ability to optimise a problem by building a model of interactions between problem variables. This model is then used in different ways (e.g. macro-variation, biased generation of phenotypes, etc.) to assist the evolutionary process in finding fitter phenotypes [56–65].

Two such model-building algorithms [61, 62] are of particular interest because of similarities they share with the GRN model: they utilise a non-linear, recurrent network and build a model of interactions in a Hebbian manner [67, 68]. They use a generate-and-test hill-climbing process which accepts variations in the state variables only when they increase fitness. Both algorithms build a model by repeatedly exposing a single individual from different initial conditions to a fitness landscape, and enforcing that changes in the interaction matrix are in a Hebbian direction with respect to the state variables. By repeating this process, the interaction matrix internalises inter-variable interactions, i.e. whether pairs of variables should co-occur or co-vary, that best describe the structure of the problem being optimised. All these characteristics are shared by the GRN model and the work presented in chapters 4 and 5.

But the way these two algorithms utilise the learned model is different. The first, a restart Hopfield network with selective associations (rHN-S) [61], modifies the 'test' aspect of the generate-and-test method. The hill-climbing process still produces the same phenotypes, but the selection process may accept changes in state variables that would have otherwise been rejected – the phenotype gets initially worse until it finally gets better. Selection is not changed arbitrarily, however. Learned correlations alter selection in such a way that, more often than not, the initial drop in fitness will result in an increase in fitness. Essentially, selection enables the hill-climber to ignore local optima. The second, a restart Hopfield network with generative associations (rHN-G) [62], modifies the 'generate' aspect of the generate-and-test method. Instead of producing small variations which are accepted by selection, this algorithm produces targeted, macro-scale variation by, for example, changing all state variables necessary to switch from one module solution to another. Essentially, the updated variation operator ensures that hill-climbing occurs at a higher level (that of modules instead of single variables), but the selection aspect remains unchanged.

The question is whether the GRN model is equivalent to one (or both) of the above, or something different altogether. Considering that the GRN does not perform macro-variation at the genotype level like rHN-G, it is not immediately obvious how the two may be connected. On the other hand, both rHN-S and the GRN improve their ability to

find high-fitness phenotypes exclusively via point mutations on the genotype. But there is a very important distinction between the two, which can also be used to explain why one is more biologically plausible than the other.

Although rHN-S and the GRN both 'ignore' local optima to reach the global optimum, rHN-S does so by accepting mutations that would have otherwise been deleterious. In other words, the phenotype that is exposed to the fitness landscape is less fit than it was in the previous generation – a proposition which would make any evolutionary biologist suggest biological implausibility. On the other hand, the GRN ignores local optima by altering the genotype–fitness relationship. Evolution with the GRN may accept mutations on the genotype that would have been deleterious earlier, but the phenotype which is accepted is different; it is only accepted when it is fitter, meaning that such progression towards fitter phenotypes is not contradictory to natural selection. Put differently, the GRN follows fitness gradients that form an adaptive walk, not the illusion of one like rHN-S does.

Upon closer inspection, there is also the possibility of the GRN being similar to rHN-G. Although the GRN does not perform macro-variation on the genotype, it is possible to perform macro-variation on the phenotype via pleiotropic effects of point mutations on the genotype. In particular, a genotype that sits at a saddle point of the attractor landscape (internalised in the developmental interactions of the GRN) will necessarily fall into one of the attractors when a mutation occurs. Although this mutation would only be a minor mutation on the genotype, a large number of phenotypic traits may change simultaneously, a phenomenon resembling macro-variation on the phenotype.

This of course makes the question even harder to answer. It is clear that the GRN model is similar to both algorithms. So is it something different altogether? Is it the combination of the two? One could argue that looking at the GRN model from a generate-and-test perspective does not put any emphasis on development (genotype to phenotype), an important part of evolution that sits between the generate (mutation) and test (phenotype evaluation) elements of an abstract evolutionary process. If we assume that 'generate' refers to the genotype and 'select' to the phenotype, perhaps it only makes sense to look at the GRN model from a different perspective – not a generate-and-test one, but a generate-develop-and-test one instead. Moreover, it is not as if the rHN-S and rHN-G algorithms are not already looked at from a generate-develop-and-test perspective, but rather that their 'develop' aspect is inconsequential since there is no genotype–phenotype distinction.

This is a question that needs to be addressed more formally, as it could potentially help transfer knowledge from model-building algorithms to the evolutionary process.

6.4 FACILITATED VARIATION AND HIERARCHICAL ORGANISATION

The key observation in the work of Uri Alon and colleagues is that environments in nature do not vary at random, but rather consist of different sub-functions that need be satisfied in different combinations in every environment [13, 16, 20]. In their work published in 2008, they utilise this important observation to showcase how facilitated variation and its components can evolve in organisms by evolving in environments that change in a systematic manner [16]. However, there is a lack of conceptual clarity in their work due to the models they are using.

Since there is no distinction between genes that determine the existence of phenotypic traits and genes that regulate in their work, it is difficult to know what mutations which improve evolvability actually do. The reason this distinction is important is because it can enable us to understand how changes to the regulatory parts of the genotype shape the phenotype space and hence alter future phenotypic trajectories. The existence of neutral networks in the phenotype space, for example, cannot be solely attributed to genes directly coding for traits. Without genes that regulate or a complex developmental process, the G–P map would be one-to-one and such evolvability via the existence of neutral networks would not have been possible. But were these neutral networks there as part of their developmental process that translates a genotype to a logic gate or RNA secondary structure and remained unaffected throughout evolution, or were they altered during evolution via mutations to the regulatory parts of the genotype?

With the GRN model we utilise in chapters 4 and 5, we have a clear distinction between genes coding for traits and regulatory genes. Using this model, we can show how facilitated variation evolves when varying systematically between environments, and in particular, how neutral networks can evolve which will facilitate the evolution of facilitated variation components such as weak-regulatory linkage and modularity in the developmental architecture. This allows us to show, for example, how two phenotypes which were not initially connected via neutral networks and sitting many mutational sites apart can evolve to be only a few mutational sites away from each other via the evolution of neutral networks in phenotype space. In the simplest case where a memory of two phenotypes is formed in the developmental process (see chapter 4), we know that there exist two neutral networks: one for each phenotype. We also know that these neutral networks are connected to one another and it is possible to traverse them such that any of the two phenotypes can evolve.

What we have not been able to show until recently, however, is how to evolve the equivalent of what Parter et al. [16] refer to as 'genetic triggers': small genetic changes that can trigger large, pre-prepared phenotypic responses. This phenomenon would evolve in our work in a very basic manner in that when the embryo would reach the saddle point of the attractor landscape internalised in the developmental process, a small mutation could serve as a genetic trigger to change multiple phenotypic traits at the same time in a way that reflects the change in environment (figure 15). In this scenario, changes to the phenotype space which would bring two phenotypes which were far apart in phenotype space earlier during evolution closer together were limited in how far apart these phenotypes would be able to be placed. In particular, the minimum number of mutational sites that would need to be changed to change from one phenotype to the other is approximately $x/2 + 1$, where x is the Hamming distance between the two phenotypes. This translates to getting enough mutations to reach the saddle point of the attractor landscape and then one more mutation to fall into the attractor of the other phenotype.

Recent work has shown that selective pressures for sparse networks can improve evolvability and that in cases of extreme sparsity, it is possible to for evolution to evolve a developmental architecture that allows the expression of one gene to affect multiple phenotypic traits simultaneously [69]. This relies on utilising a cost–benefit fitness function such that the fitness of an organism is not determined solely by its performance in the environment, but also by the number and / or magnitude of its regulatory interactions. The relative weight between the cost and the benefit is controlled by the parameter λ.

Here, we utilise such a cost–benefit fitness function to show how different developmental architectures can evolve and that they can have a significant impact on phenotype

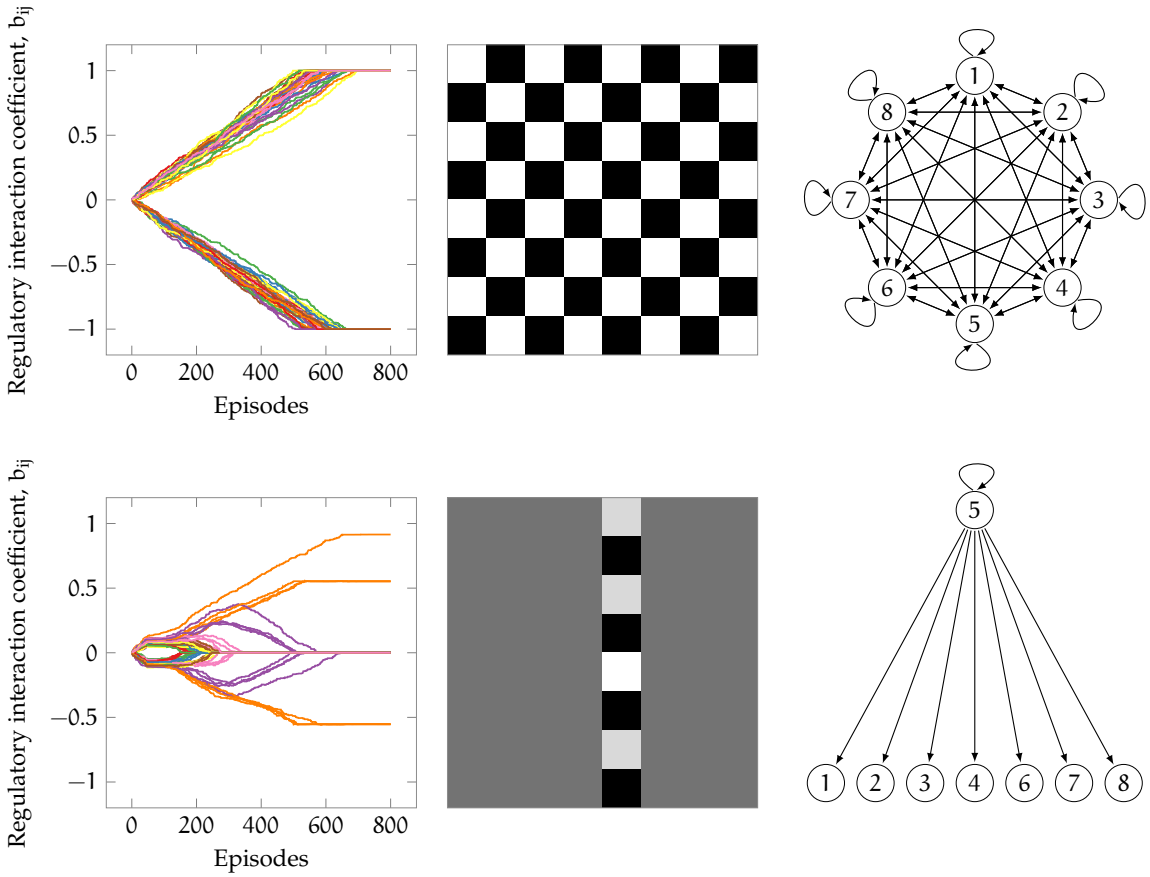


Figure 25: Hierarchical organisations evolve with the presence of a connection cost. The first column shows the evolutionary trajectory of the regulatory interactions; the second column shows a matrix representation of the final regulatory interactions; and the third column shows a network representation of the final regulatory interactions. First row: In the abscence of a connection cost ($\lambda = 0$), regulatory interactions evolve to reinforce correlations in the environment. The evolved correlations saturate, since this enables development to produce phenotypes of high magnitude which are highly fit. In this sort of network architecture, every gene has approximately the same control over the expression of all the other genes. The developed gene expression levels are determined by an averaging effect of all regulatory interactions acting on each gene. Second row: When an appropriate connection cost is in place (here $\lambda = 1$), regulatory interactions evolve to reinforce correlations in the environment but the evolved networks are sparse. The evolved correlations do not saturate, since increasing the magnitude of the connections is costly. By the end of the simulation, only a small number of connections remain. In this sort of network architecture, one gene controls the expression of all the other genes in the phenotype. Essentially, a trigger gene has evolved: changes to this specific gene have a large phenotypic effect by changing multiple phenotypic traits simultaneously.

space and potential phenotypic trajectories. We compare the evolution of regulatory interactions when alternating between two target phenotypes of length $n = 8$ (as in chapter 4) using two different values of λ: 0 and 1. This enables us to show the evolution of genetic triggers which are not affected by the Hamming distance between phenotypes (figure 25). When $\lambda = 1$, trigger genes are evolved. Large, non-random phenotypic effects in the form

of modular variation are one mutation away. This can also be seen as a simple example of the evolution of hierarchy, where one gene controls the expression of multiple other genes. Expanding on this work will allow us to showcase the evolution of facilitated variation features such as weak regulatory linkage and modularity in the developmental architecture. Further than that, this will allow us to show how evolution can achieve modular variation and globally optimise a different class of multi-modal fitness landscape than the one we utilise in chapter 5.

7 CONCLUSIONS

This report presents work carried out in an attempt to improve our understanding of evolvability and its evolution.

In chapter 2, we explained the significance and motivation behind understanding and explaining evolvability within the context of evolutionary biology, and touched upon some of the open questions, some of which were investigated in this report. In a nutshell, evolvability is an essential ingredient if evolution by natural selection is to produce the level of complexity, well-adaptedness and diversity observed in nature. The simple mechanisms of variation, inheritance and selection typically used in evolutionary computation are incapable of matching the power of the evolutionary process in the natural world. This is because the mechanisms, although necessary, are not sufficient to explain the biological world as we see it. Evolvability is the missing ingredient. Evolvability describes the capacity of organisms for adaptive evolution, and this capacity stems from their inherent properties – in particular, their developmental process and G–P map. Understanding how developmental processes evolve and how they affect future evolutionary trajectories is an essential part of understanding the evolutionary process in terms of forms rather than simply changes in gene frequencies.

In chapter 3, we introduced discrete differential evolution, a novel evolutionary algorithm based on differential evolution that can be used to solve modular problems. The algorithm achieves this without any explicit learning model. Instead, it relies on information that exists inherently within the population in order to direct evolution towards areas of high fitness.

In chapter 4, we demonstrated that the evolution of developmental memory changes evolutionary trajectories for the better. Not only do phenotypes become more robust, they become more evolvable at the same time by enabling evolution to rapidly switch between memorised phenotypes using pre-prepared responses. This leads to a speed-up in evolution in terms of the number of generations required to reach a high-fitness phenotype when alternating between environments.

Chapter 5 presented a solution to the paradox of long-term evolvability. More spefically, we utilised a GRN model to investigate how the evolution of developmental interactions affect future evolutionary trajectories. We investigated how evolution can learn from the past to assist the evolutionary process in the future, similar to how machine learning algorithms learn from the training set to generalise to the test set. We showed that evolution produces GRNs that possess long-term evolvability. This is essentially due to the regulatory interactions internalising the problem structure that can be inferred from locally optimal phenotypes in a multi-modal fitness landscape. The problem structure internalised within the regulatory interactions is then used to help evolution avoid low-fitness peaks altogether. Importantly, GRNs evolve long-term evolvability although selection is only in the short-term.

Lastly, chapter 6 provided a brief outline of future work directions, with particular emphasis on the relationship between the GRN and a family of model-building algorithms

and on how our current model can help us improve our understanding of facilitated variation.

BIBLIOGRAPHY

1. Kounios L et al. (2016) Resolving the Paradox of Evolvability with Learning Theory: How Evolution Learns to Improve Evolvability on Rugged Fitness Landscapes. arXiv: 1612.05955 [q-bio.PE].
2. Riedl R (1977) A Systems-Analytical Approach to Macro-Evolutionary Phenomena. *The Quarterly Review of Biology* 52(4):351–370.
3. Riedl R (1978) *Order in Living Organisms: A Systems Analysis of Evolution* (Wiley: New York).
4. Wagner GP, Altenberg L (1996) Complex Adaptations and the Evolution of Evolvability. *Evolution* 50(3):967–976. DOI: 10.2307/2410639.
5. Wagner GP, Laubichler MD (2004) Rupert Riedl and the Re-Synthesis of Evolutionary and Developmental Biology: Body Plans and Evolvability. *Journal of Experimental Zoology Part B: Molecular and Developmental Evolution* 302B(1):92–102. DOI: 10.1002/jez.b.20005.
6. Toussaint M, von Seelen W (2007) Complex Adaptation and System Structure. *Bio Systems* 90(3):769–782. DOI: 10.1016/j.biosystems.2007.03.004.
7. Kirschner M, Gerhart J (1998) Evolvability. *Proceedings of the National Academy of Sciences of the United States of America* 95(15):8420–8427. DOI: 10.1073/pnas.95.15.8420.
8. Gerhart J, Kirschner M (2007) The Theory of Facilitated Variation. *Proceedings of the National Academy of Sciences of the United States of America* 104(Suppl. 1):8582–8589. DOI: 10.1073/pnas.0701035104.
9. Wagner A (2005) Robustness, Evolvability, and Neutrality. *FEBS Letters* 579(8):1772–1778. DOI: 10.1016/j.febslet.2005.01.063.
10. Wagner A (2008) Robustness and Evolvability: A Paradox Resolved. *Proceedings of the Royal Society B: Biological Sciences* 275(1630):91–100. DOI: 10.1098/rspb.2007.1137.
11. Wagner A (2012) The Role of Robustness in Phenotypic Adaptation and Innovation. *Proceedings of the Royal Society B: Biological Sciences* 279(1732):1249–1258. DOI: 10.1098/rspb.2011.2293.
12. Pavlicev M, Cheverud JM, Wagner GP (2011) Evolution of Adaptive Phenotypic Variation Patterns by Direct Selection for Evolvability. *Proceedings of the Royal Society B: Biological Sciences* 278(1713):1903–1912. DOI: 10.1098/rspb.2010.2113.
13. Kashtan N, Noor E, Alon U (2007) Varying Environments Can Speed Up Evolution. *Proceedings of the National Academy of Sciences of the United States of America* 104(34):13711–13716. DOI: 10.1073/pnas.0611630104.
14. Lipson H, Pollack JB, Suh NP (2002) On the Origin of Modular Variation. *Evolution* 56(8):1549–1556. DOI: 10.1111/j.0014-3820.2002.tb01466.x.
15. Clune J, Mouret JB, Lipson H (2013) The Evolutionary Origins of Modularity. *Proceedings of the Royal Society B: Biological Sciences* 280(1755):20122863. DOI: 10.1098/rspb.2012.2863.

16. Parter M, Kashtan N, Alon U (2008) Facilitated Variation: How Evolution Learns from Past Environments to Generalize to New Environments. *PLoS Computational Biology* 4(11):e1000206. DOI: 10.1371/journal.pcbi.1000206.
17. Arnold SJ, Pfrender ME, Jones AG (2001) The Adaptive Landscape As a Conceptual Bridge Between Micro- and Macroevolution. *Genetica* 112–113(1):9–32. DOI: 10.1023/A:1013373907708.
18. Arnold SJ, Bürger R, Hohenlohe PA, Ajie BC, Jones AG (2008) Understanding the Evolution and Stability of the G-Matrix. *Evolution* 62(10):2451–2461. DOI: 10.1111/j.1558-5646.2008.00472.x.
19. Watson RA, Wagner GP, Pavlicev M, Weinreich DM, Mills R (2014) The Evolution of Phenotypic Correlations and "Developmental Memory". *Evolution* 68(4):1124–1138. DOI: 10.1111/evo.12337.
20. Kashtan N, Alon U (2005) Spontaneous Evolution of Modularity and Network Motifs. *Proceedings of the National Academy of Sciences of the United States of America* 102(39):13773–13778. DOI: 10.1073/pnas.0503610102.
21. Rose MR, Oakley TH (2007) The New Biology: Beyond the Modern Synthesis. *Biology Direct* 2(30). DOI: 10.1186/1745-6150-2-30.
22. Pigliucci M (2007) Do We Need an Extended Evolutionary Synthesis? *Evolution* 61(12):2743–2749. DOI: 10.1111/j.1558-5646.2007.00246.x.
23. Müller GB (2007) Evo-Devo: Extending the Evolutionary Synthesis. *Nature Reviews Genetics* 8(12):943–949. DOI: 10.1038/nrg2219.
24. Pigliucci M, Müller GB, eds. (2010) *Evolution - The Extended Synthesis* (MIT Press: Cambridge).
25. Darwin C (1859) *On the Origin of Species by Means of Natural Selection or the Preservation of Favoured Species in the Struggle for Life* (John Murray: London).
26. Halder G, Callaerts P, Gehring WJ (1995) Induction of Ectopic Eyes by Targeted Expression of the Eyeless Gene in Drosophila. *Science* 267(5205):1788–1792. DOI: 10.1126/science.7892602.
27. Chenn A, Walsh CA (2002) Regulation of Cerebral Cortical Size by Control of Cell Cycle Exit in Neural Precursors. *Science* 297(5580):365–369. DOI: 10.1126/science.1074192.
28. Capdevila J, Belmonte JCI (2000) Perspectives on the Evolutionary Origin of Tetrapod Limbs. *Journal of Experimental Zoology* 288(4):287–303.
29. Hallgrímsson B, Willmore K, Hall BK (2002) Canalization, Developmental Stability, and Morphological Integration in Primate Limbs. *American Journal of Physical Anthropology* 119(S35):131–158. DOI: 10.1002/ajpa.10182.
30. Young NM, Hallgrímsson B (2005) Serial Homology and the Evolution of Mammalian Limb Covariation Structure. *Evolution* 59(12):2691–2704. DOI: 10.1111/j.0014-3820.2005.tb00980.x.
31. Young NM, Wagner GP, Hallgrímsson B (2010) Development and the Evolvability of Human Limbs. *Proceedings of the National Academy of Sciences of the United States of America* 107(8):3400–3405. DOI: 10.1073/pnas.0911856107.
32. Brown RL (2013) What Evolvability Really Is. *The British Journal for the Philosophy of Science*. DOI: 10.1093/bjps/axt014.
33. Maynard Smith J et al. (1985) Developmental Constraints and Evolution: A Perspective from the Mountain Lake Conference on Development and Evolution. *The Quarterly Review of Biology* 60(3):265–287.

34. Brakefield PM (2006) Evo–Devo and Constraints on Selection. *Trends in Ecology & Evolution* 21(7):362–368. DOI: 10.1016/j.tree.2006.05.001.
35. Futuyma DJ (2013) *Evolution*. 3rd ed. (Sinauer Associates Inc., U.S.: Sunderland).
36. Dawkins R (1989) The Evolution of Evolvability. In: *Proceedings of the Conference on the Synthesis and Simulation of Living Systems (Artificial Life)*. Ed. by C Langton. Santa Fe Institute Studies in the Sciences of Complexity 6. (Addison–Wesley), 201–220.
37. Gallagher A (2009) Evolvability: A Formal Approach. PhD thesis. University of Oxford. URL: http://ora.ox.ac.uk/objects/uuid:d3b0511e-bee5-4778-8822-703c514c1c1d.
38. Dawkins R (1986) *The Blind Watchmaker* (Longman: Harlow).
39. West-Eberhard MJ (2003) *Developmental Plasticity and Evolution* (Oxford University Press: Oxford).
40. Kirschner MW, Gerhart JC (2006) *The Plausibility of Life: Resolving Darwin's Dilemma* (Yale University Press: New Haven).
41. Maynard Smith J, Szathmáry E (1995) *The Major Transitions in Evolution* (Oxford University Press: Oxford).
42. Pigliucci M (2008) Is Evolvability Evolvable? *Nature Reviews Genetics* 9(1):75–82. DOI: 10.1038/nrg2278.
43. Houle D (1992) Comparing Evolvability and Variability of Quantitative Traits. *Genetics* 130(1):195–204.
44. Beldade P, Brakefield PM (2002) The Genetics and Evo–Devo of Butterfly Wing Patterns. *Nature Reviews Genetics* 3(6):442–452. DOI: 10.1038/nrg818.
45. Hendrikse JL, Parsons TE, Hallgrímsson B (2007) Evolvability As the Proper Focus of Evolutionary Developmental Biology. *Evolution & Development* 9(4):393–401. DOI: 10.1111/j.1525-142X.2007.00176.x.
46. Draghi J, Wagner GP (2008) Evolution of Evolvability in a Developmental Model. *Evolution* 62(2):301–315. DOI: 10.1111/j.1558-5646.2007.00303.x.
47. Bedau MA, Packard NH (2003) Evolution of Evolvability via Adaptation of Mutation Rates. *Bio Systems* 69(2–3):143–162. DOI: 10.1016/S0303-2647(02)00137-5.
48. Earl DJ, Deem MW (2004) Evolvability Is a Selectable Trait. *Proceedings of the National Academy of Sciences of the United States of America* 101(32):11531–11536. DOI: 10.1073/pnas.0404656101.
49. Clune J et al. (2008) Natural Selection Fails to Optimize Mutation Rates for Long-Term Adaptation on Rugged Fitness Landscapes. *PLoS Computational Biology* 4(9):e1000187. DOI: 10.1371/journal.pcbi.1000187.
50. Hansen TF (2003) Is Modularity Necessary for Evolvability?: Remarks on the Relationship Between Pleiotropy and Evolvability. *Bio Systems* 69(2–3):83–94. DOI: 10.1016/S0303-2647(02)00132-6.
51. Aldana M, Balleza E, Kauffman S, Resendiz O (2007) Robustness and Evolvability in Genetic Regulatory Networks. *Journal of Theoretical Biology* 245(3):433–448. DOI: 10.1016/j.jtbi.2006.10.027.
52. Ciliberti S, Martin OC, Wagner A (2007) Innovation and Robustness in Complex Regulatory Gene Networks. *Proceedings of the National Academy of Sciences of the United States of America* 104(34):13591–13596. DOI: 10.1073/pnas.0705396104.
53. Draghi JA, Parsons TL, Wagner GP, Plotkin JB (2010) Mutational Robustness Can Facilitate Adaptation. *Nature* 463(7279):353–355. DOI: 10.1038/nature08694.

54. Von Dassow G, Munro E (1999) Modularity in Animal Development and Evolution: Elements of a Conceptual Framework for EvoDevo. *Journal of Experimental Zoology* 285(4):307–325.
55. Gerhart J, Kirschner M (1997) *Cells, Embryos, and Evolution: Toward a Cellular and Developmental Understanding of Phenotypic Variation and Evolutionary Adaptability* (John Wiley and Sons Ltd.: Oxford).
56. Pelikan M, Goldberg DE, Cantú-Paz E (1999) BOA: The Bayesian Optimization Algorithm. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Ed. by W Banzhaf et al. (Morgan Kaufmann), 525–532.
57. Pelikan M, Goldberg DE (2000) Hierarchical Problem Solving and the Bayesian Optimization Algorithm. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Ed. by LD Whitley et al. (Morgan Kaufmann), 267–274.
58. Larrañaga P, Lozano JA, eds. (2001) *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation* (Kluwer Academic Publishers: Boston).
59. Pelikan M, Kumara S, Cantú-Paz E, eds. (2006) *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*. Studies in Computational Intelligence 33 (Springer Berlin Heidelberg: Berlin). DOI: 10.1007/978-3-540-34954-9.
60. Mills R (2010) How Micro-Evolution Can Guide Macro-Evolution: Multi-Scale Search via Evolved Modular Variation. PhD thesis. University of Southampton. URL: http://eprints.soton.ac.uk/156549/.
61. Watson RA, Buckley CL, Mills R (2011a) Optimization in "Self-Modeling" Complex Adaptive Systems. *Complexity* 16(5):17–26. DOI: 10.1002/cplx.20346.
62. Watson RA, Mills R, Buckley CL (2011b) Transformations in the Scale of Behavior and the Global Optimization of Constraints in Adaptive Networks. *Adaptive Behavior* 19(4):227–249. DOI: 10.1177/1059712311412797.
63. Cox CR, Watson RA (2014) Solving Building Block Problems Using Generative Grammar. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Ed. by C Igel. (ACM), 341–348. DOI: 10.1145/2576768.2598259.
64. Mills R, Jansen T, Watson RA (2014) Transforming Evolutionary Search Into Higher-Level Evolutionary Search by Capturing Problem Structure. *IEEE Transactions on Evolutionary Computation* 18(5):628–642. DOI: 10.1109/TEVC.2014.2347702.
65. Kounios L, Mills R, Watson RA (in preparation[a]) The Ability of Differential Evolution Principles to Solve Random-Linkage Modular Problems. *IEEE Transactions on Evolutionary Computation*.
66. Watson RA et al. (2016) Evolutionary Connectionism: Algorithmic Principles Underlying the Evolution of Biological Organisation in Evo-Devo, Evo-Eco and Evolutionary Transitions. *Evolutionary Biology* 43(4):553–581. DOI: 10.1007/s11692-015-9358-z.
67. Hebb DO (1949) *The Organization of Behavior: A Neuropsychological Theory* (Wiley and Sons: New York).
68. Ackley DH, Hinton GE, Sejnowski TJ (1985) A Learning Algorithm for Boltzmann Machines. *Cognitive Science* 9(1):147–169. DOI: 10.1207/s15516709cog0901_7.
69. Kouvaris K, Clune J, Kounios L, Brede M, Watson RA (2017) How Evolution Learns to Generalise: Using the Principles of Learning Theory to Understand the Evolution of Developmental Organisation. *PLoS Computational Biology* 13(4):e1005358.

70. Kounios L et al. (in preparation[b]) Resolving the Paradox of Evolvability with Learning Theory: How Evolution Learns to Improve Evolvability on Rugged Fitness Landscapes. *PLoS Computational Biology*.
71. Wagner A (2007) *Robustness and Evolvability in Living Systems*. Princeton Studies in Complexity (Princeton University Press: New Jersey).
72. Wagner A (2011) *The Origins of Evolutionary Innovations: A Theory of Transformative Change in Living Systems* (Oxford University Press: Oxford).
73. Waddington CH (1942) Canalization of Development and the Inheritance of Acquired Characters. *Nature* 150(3811):563–565. DOI: 10.1038/150563a0.
74. Waddington CH (1957) *The Strategy of the Genes: A Discussion of Some Aspects of Theoretical Biology* (George Allen & Unwin: London).
75. Mitchell KJ (2007) The Genetics of Brain Wiring: From Molecule to Mind. *PLoS Biology* 5(4):e113. DOI: 10.1371/journal.pbio.0050113.
76. Brigandt I (2007) Typology Now: Homology and Developmental Constraints Explain Evolvability. *Biology & Philosophy* 22(5):709–725. DOI: 10.1007/s10539-007-9089-3.
77. Braendle C, Baer CF, Félix MA (2010) Bias and Evolution of the Mutationally Accessible Phenotypic Space in a Developmental System. *PLoS Genetics* 6(3):e1000877. DOI: 10.1371/journal.pgen.1000877.
78. Harrison F, Buckling A (2007) High Relatedness Selects Against Hypermutability in Bacterial Metapopulations. *Proceedings of the Royal Society B: Biological Sciences* 274(1615):1341–1347. DOI: 10.1098/rspb.2006.0408.
79. Schoustra SE, Debets AJM, Slakhorst M, Hoekstra RF (2007) Mitotic Recombination Accelerates Adaptation in the Fungus *Aspergillus nidulans*. *PLoS Genetics* 3(4):e68. DOI: 10.1371/journal.pgen.0030068.
80. Meiklejohn CD, Hartl DL (2002) A Single Mode of Canalization. *Trends in Ecology & Evolution* 17(10):468–473. ISSN: 0169-5347. DOI: 10.1016/S0169-5347(02)02596-X.
81. Kitano H (2004) Biological Robustness. *Nature Reviews Genetics* 5(11):826–837. DOI: 10.1038/nrg1471.
82. Wagner GP, Pavlicev M, Cheverud JM (2007) The Road to Modularity. *Nature Reviews Genetics* 8(12):921–931. DOI: 10.1038/nrg2267.
83. Crombach A, Hogeweg P (2008) Evolution of Evolvability in Gene Regulatory Networks. *PLoS Computational Biology* 4(7):e1000112. DOI: 10.1371/journal.pcbi.1000112.
84. Simon HA (1983) Why Should Machines Learn? In: *Machine Learning: An Artificial Intelligence Approach*. Ed. by RS Michalski, JG Carbonell, TM Mitchell. Symbolic Computation. (Springer Berlin Heidelberg: Berlin), 25–37. DOI: 10.1007/978-3-662-12405-5_2.
85. Langley P (1996) *Elements of Machine Learning* (Morgan Kaufmann: San Fransisco, CA).
86. Kauffman SA (1993) *The Origins of Order: Self Organization and Selection in Evolution* (Oxford University Press: New York).
87. Hopfield JJ (1982) Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences of the United States of America* 79(8):2554–2558.
88. Watson RA (2001) Analysis of Recombinative Algorithms on a Non-Separable Building-Block Problem. In: *Proceedings of the Workshop on Foundations of Genetic*

Algorithms (FOGA). Ed. by WN Martin, WM Spears. (Morgan Kaufmann), 69–89. DOI: 10.1016/B978-155860734-7/50087-1.

89. Watson RA (2004) A Simple Two-Module Problem to Exemplify Building-Block Assembly Under Crossover. In: *Proceedings of the Conference on Parallel Problem Solving from Nature (PPSN)*. Ed. by X Yao et al. (Springer), 161–171. DOI: 10.1007/978-3-540-30217-9_17.
90. Prügel-Bennett A (2004) When a Genetic Algorithm Outperforms Hill-Climbing. *Theoretical Computer Science* 320(1):135–153. DOI: 10.1016/j.tcs.2004.03.038.
91. Jansen T, Wegener I (2005) Real Royal Road Functions—Where Crossover Provably Is Essential. *Discrete Applied Mathematics* 149(1–3):111–125. DOI: 10.1016/j.dam.2004.02.019.
92. Watson RA, Jansen T (2007) A Building-Block Royal Road Where Crossover Is Provably Essential. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Ed. by H Lipson. (ACM), 1452–1459. DOI: 10.1145/1276958.1277224.
93. Holland JH (1975) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (University of Michigan Press: Ann Arbor, Michigan).
94. Goldberg DE, Holland JH (1988) Genetic Algorithms and Machine Learning. *Machine Learning* 3(2–3):95–99. DOI: 10.1023/A:1022602019183.
95. Forrest S, Mitchell M (1992) Relative Building-Block Fitness and the Building-Block Hypothesis. In: *Proceedings of the Workshop on Foundations of Genetic Algorithms (FOGA)*. Ed. by LD Whitley. (Morgan Kaufmann), 109–126.
96. Pelikan M, Goldberg DE, Cantú-Paz E (2000) Linkage Problem, Distribution Estimation, and Bayesian Networks. *Evolutionary Computation* 8(3):311–340. DOI: 10.1162/106365600750078808.
97. Yu TL, Goldberg DE (2006) Conquering Hierarchical Difficulty by Explicit Chunking: Substructural Chromosome Compression. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Ed. by M Cattolico. (ACM), 1385–1392. DOI: 10.1145/1143997.1144210.
98. Storn R, Price K (1995) *Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces*. Tech. rep. TR-95-012. International Computer Science Institute, University of Berkeley. URL: ftp://ftp.icsi.berkeley.edu/pub/techreports/1995/tr-95-012.pdf.
99. Storn R, Price K (1996) Minimizing the Real Functions of the ICEC'96 Contest by Differential Evolution. In: *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC)*. (), 842–844. DOI: 10.1109/ICEC.1996.542711.
100. Storn R (1996) On the Usage of Differential Evolution for Function Optimization. In: *Proceedings of the Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS)*. (), 519–523. DOI: 10.1109/NAFIPS.1996.534789.
101. Storn R, Price K (1997) Differential Evolution - A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces. *Journal of Global Optimization* 11(4):341–359. DOI: 10.1023/A:1008202821328.
102. Vesterstrøm J, Thomsen R (2004) A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems. In: *Proceedings of the IEEE Congress on Evolutionary Computation*. Vol. 2. (), 1980–1987. DOI: 10.1109/CEC.2004.1331139.

103. Price KV, Storn RM, Lampinen JA (2005) *Differential Evolution: A Practical Approach to Global Optimization* (Springer: Berlin).
104. De Falco I, Della Cioppa A, Tarantino E (2006) Automatic Classification of Handsegmented Image Parts with Differential Evolution. In: *Proceedings of the Conference on Applications of Evolutionary Computing*. (Springer-Verlag), 403–414. DOI: 10.1007/11732242_36.
105. Pan QK, Tasgetiren MF, Liang YC (2008) A Discrete Differential Evolution Algorithm for the Permutation Flowshop Scheduling Problem. *Computers & Industrial Engineering* 55(4):795–816. DOI: 10.1016/j.cie.2008.03.003.
106. Wang L, Pan QK, Suganthan PN, Wang WH, Wang YM (2010) A Novel Hybrid Discrete Differential Evolution Algorithm for Blocking Flow Shop Scheduling Problems. *Computers & Operations Research* 37(3):509–520. DOI: 10.1016/j.cor.2008.12.004.
107. Sauer JG, dos Santos Coelho L (2008) Discrete Differential Evolution with Local Search to Solve the Traveling Salesman Problem: Fundamentals and Case Studies. In: *Proceedings of the IEEE International Conference on Cybernetic Intelligent Systems (CIS)*. (), 1–6. DOI: 10.1109/UKRICIS.2008.4798955.
108. Randall M (2011) Differential Evolution for a Constrained Combinatorial Optimisation Problem. *International Journal of Metaheuristics* 1(4):279–297. DOI: 10.1504/IJMHEUR.2011.044302.
109. Das S, Suganthan PN (2011) Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation* 15(1):4–31. DOI: 10.1109/TEVC.2010.2059031.
110. Mills R, Watson RA (2007) Variable Discrimination of Crossover Versus Mutation Using Parameterized Modular Structure. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Ed. by H Lipson. (ACM), 1312–1319. DOI: 10.1145/1276958.1277206.
111. Goldberg DE, Deb K, Horn J (1992) Massive Multimodality, Deception, and Genetic Algorithms. In: *Proceedings of the Conference on Parallel Problem Solving from Nature (PPSN)*. Ed. by R Männer, B Manderick. (Elsevier), 37–48.
112. Watson RA, Hornby GS, Pollack JB (1998) Modeling Building-Block Interdependency. In: *Proceedings of the Conference on Parallel Problem Solving from Nature (PPSN)*. Ed. by A Eiben, T Bäck, M Schoenauer, HP Schwefel. (Springer), 97–106. DOI: 10.1007/BFb0056853.
113. Watson RA, Pollack JB (2005) Modular Interdependency in Complex Dynamical Systems. *Artificial Life* 11(4):445–457. DOI: 10.1162/106454605774270589.
114. Pelikan M (2002) Bayesian Optimization Algorithm: From Single Level to Hierarchy. PhD thesis. University of Illinois at Urbana-Champaign. URL: http://medal-lab.org/files/2002023.pdf.
115. Prügel-Bennett A (2010) Benefits of a Population: Five Mechanisms That Advantage Population-Based Algorithms. *IEEE Transactions on Evolutionary Computation* 14(4):500–517. DOI: 10.1109/TEVC.2009.2039139.
116. Pelikan M, Goldberg DE (2006) Hierarchical Bayesian Optimization Algorithm. In: *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*. Ed. by M Pelikan, S Kumara, E Cantú-Paz. Studies in Computational Intelligence 33. (Springer Berlin Heidelberg: Berlin), 63–90. DOI: 10.1007/978-3-540-34954-9_4.

117. Lande R, Arnold SJ (1983) The Measurement of Selection on Correlated Characters. *Evolution* 37(6):1210–1226. DOI: 10.2307/2408842.
118. Jones AG, Arnold SJ, Bürger R (2007) The Mutation Matrix and the Evolution of Evolvability. *Evolution* 61(4):727–745. DOI: 10.1111/j.1558-5646.2007.00071.x.
119. Kashtan N, Mayo AE, Kalisky T, Alon U (2009) An Analytically Solvable Model for Rapid Evolution of Modular Structure. *PLoS Computational Biology* 5(4):e1000355. DOI: 10.1371/journal.pcbi.1000355.
120. Gillespie JH (1984) Molecular Evolution Over the Mutational Landscape. *Evolution* 38(5):1116–1129.
121. Watson RA, Buckley CL (2011) Global Adaptation in Networks of Selfish Components: Emergent Associative Memory at the System Scale. *Artificial Life* 17(3):147–166. DOI: 10.1162/artl_a_00029.
122. Conrad M (1998) Towards High Evolvability Dynamics. In: *Evolutionary Systems: Biological and Epistemological Perspectives on Selection and Self-Organization*. Ed. by G Van de Vijver, SN Salthe, M Delpos. (Springer Netherlands), 33–43. DOI: 10.1007/978-94-017-1510-2_4.
123. Conrad M (1972) The Importance of Molecular Hierarchy in Information Processing. *Towards a Theoretical Biology* 4:222–228.
124. Conrad M (1979) Bootstrapping on the Adaptive Landscape. *Bio Systems* 11(2):167–182. DOI: 10.1016/0303-2647(79)90009-1.
125. Conrad M (1990) The Geometry of Evolution. *Bio Systems* 24(1):61–81. DOI: 10.1016/0303-2647(90)90030-5.
126. Altenberg L (1995) Genome Growth and the Evolution of the Genotype-Phenotype Map. In: *Evolution and Biocomputation: Computational Models of Evolution*. Ed. by W Banzhaf, FH Eckman. Lecture Notes in Computer Science 899. (Springer Berlin Heidelberg), 205–259. DOI: 10.1007/3-540-59046-3_11.
127. Díaz Arenas C, Cooper TF (2013) Mechanisms and Selection of Evolvability: Experimental Evidence. *FEMS Microbiology Reviews* 37(4):572–582. DOI: 10.1111/1574-6976.12008.
128. Skinner BF (1965) *Science and Human Behavior* (Simon and Schuster).
129. Maynard Smith J (1986) *The Problems of Biology* (Oxford University Press: Oxford).
130. Valiant LG (2007) Evolvability. In: *Mathematical Foundations of Computer Science 2007*. Ed. by L Kučera, A Kučera. Lecture Notes in Computer Science 4708. (Springer), 22–43. DOI: 10.1007/978-3-540-74456-6_5.
131. Chastain E, Livnat A, Papadimitriou C, Vazirani U (2014) Algorithms, Games and Evolution. *Proceedings of the National Academy of Sciences of the United States of America* 111(29):10620–10623. DOI: 10.1073/pnas.1406556111.
132. Power DA et al. (2015) What Can Ecosystems Learn? Expanding Evolutionary Ecology with Learning Theory. *Biology Direct* 10(1):1–24. DOI: 10.1186/s13062-015-0094-1.
133. Watson RA, Szathmáry E (2016) How Can Evolution Learn? *Trends in Ecology & Evolution* 31(2):147–157. DOI: 10.1016/j.tree.2015.11.009.
134. Weinreich DM, Lang Y, Wylie CS, Heckendorn RB (2013) Should Evolutionary Geneticists Worry About Higher-Order Epistasis? *Current Opinion in Genetics & Development* 23(6):700–707. DOI: 10.1016/j.gde.2013.10.007.
135. Weinreich DM, Watson RA, Chao L (2005) Perspective: Sign Epistasis and Genetic Constraint on Evolutionary Trajectories. *Evolution* 59(6):1165–1174. DOI: 10.1111/j.0014-3820.2005.tb01768.x.

136. Barahona F (1982) On the Computational Complexity of Ising Spin Glass Models. *Journal of Physics A: Mathematical and General* 15(10):3241–3253. DOI: 10.1088/0305-4470/15/10/028.
137. Watson RA, Ebner M (2014) Eco-Evolutionary Dynamics on Deformable Fitness Landscapes. In: *Recent Advances in the Theory and Application of Fitness Landscapes*. Ed. by H Richter, A Engelbrecht. Emergence Complexity and Computation 6. (Springer), 339–368. DOI: 10.1007/978-3-642-41888-4_12.
138. Ridley M (2003) *Evolution*. 3rd ed. (John Wiley and Sons Ltd: Oxford).
139. Carroll SB (2005) *Endless Forms Most Beautiful: The New Science of Evo Devo and the Making of the Animal Kingdom* (W. W. Norton & Co.: New York).
140. Alon U (2006) *An Introduction to Systems Biology: Design Principles of Biological Circuits* (Chapman & Hall/CRC: Boca Raton, FL).
141. Sansom R, Brandon RN, eds. (2007) *Integrating Evolution and Development: From Theory to Practice* (MIT Press: Cambridge).
142. Valiant L (2013) *Probably, Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World* (The Perseus Books Group: New York).
143. Lipson H, ed. (2007) *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. (ACM).