Algorithmic Biology of Evolution and Ecology

Artem Kaznatcheev

Oriel College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2020

Abstract

Any process can be seen as an algorithm; its power and its limits can then be analysed with the techniques of theoretical computer science. To analyse algorithms, we divide the world in two: the problem space that shapes what might happen and the dynamics of what does happen. If we fix an idealised framework for one of the two, then we can obtain powerful general results by abstracting over the other. This "algorithmic lens" can be used to view both artificial and natural processes, including the natural processes of biological evolution.

In Part I, I idealize the space of evolution as a fitness landscape so that I can abstract over the possible evolutionary dynamics. I show that fitness landscapes can be represented by gene-interaction networks that encode the structure of epistasis.

For some landscapes, the epistatic structure produces a computational constraint that prevents evolution from finding even a local fitness optimum—thus contradicting the traditional assumption that local fitness peaks can always be reached quickly by natural selection. I introduce a distinction between easy landscapes, where local fitness peaks can be found in a moderate number of steps, and hard landscapes where finding any such local optimum requires an infeasible amount of time. Hard examples exist where strong-selection weak-mutation dynamics cannot find a local peak in polynomial time, even when it is known to be unique. More generally, I show that hard fitness landscapes exist where no evolutionary dynamics—even ones that do not follow adaptive paths—can find a local fitness optimum in polynomial time. Moreover, on hard landscapes, the fitness advantage of nearby mutants cannot drop off exponentially fast but must follow a power-law, similar to the one found by long-term evolution experiments, associated with unbounded growth in fitness. Thus, the constraint of computational complexity enables open-ended evolution on finite landscapes. I present candidates for hard landscapes at scales from single genes, to microbes, to complex organisms with costly learning (Baldwin effect) or maintained cooperation (Hankshaw effect). Finally, by looking closer at the fine structure of epistasis, I also extend the class of provably easy landscapes to include all those with tree-structured gene-interaction networks.

In Part II, I idealize the dynamics of evolution as replicator dynamics so that I can abstract over the space of ecologies (interactions between organisms). This requires replacing the fitness-as-scalar concept used in fitness landscapes by a fitness-as-function concept derived from evolutionary game theory. Since they have not been adequately defined or interpreted in the context of microscopic biology, I provide two interpretations of the central objects of game theory: one that leads

to what I call “reductive games” and the other to “effective games”. These interpretations are based on the difference between views of fitness as a property of tokens versus fitness as a summary statistic of types. Reductive games are typical of theoretical work like agent-based models. Effective games correspond more closely to experimental work and allow for empirical abstraction over poorly characterized interaction mechanisms like spatial structure.

This empirical abstraction allows me to analyse the in vitro evolution of resistance to cancer therapy. I develop a game assay to directly measure effective evolutionary games in co-cultures of non-small cell lung cancer cells that are sensitive vs resistant to the targeted drug Alectinib. I show that the games are not only quantitatively different between different environments, but that the presence of the drug or the absence of cancer-associated fibroblasts qualitatively switches the type of game being played by the in vitro population. This observation provides empirical confirmation of a central theoretical postulate of evolutionary game theory in oncology: we can treat not only the player, but also the game.

Thus through the whole thesis, I demonstrate how the algorithmic lens and abstraction can help us derive new ways of seeing and understanding both evolution and ecology.

Acknowledgements

Research is a maze, full of twists, turns, and dead ends. But hidden in this maze are amazing treasures. While wandering this maze, I have been fortunate enough to meet others who have shared their explorations, advice, and lent an ear to listen or a hand to help.

Thomas Shultz and Prakash Panangaden gave me the courage to first enter the maze. Julian Xue convinced me that biology is full of exciting questions to which I could find new answers. David Basanta took a gamble and brought me to the Moffitt Cancer Center, giving me a chance to interact with the brightest minds of mathematical oncology. In the second half of the DPhil, Dave Cohen reminded me of how much fun we can have arguing about math all day. Jacob Scott is the one that made the DPhil possible, through his support and passion for navigating the maze of research. Jake's excitement is contagious and lit the way for me. Peter Jeavons taught me that it is better to walk slowly along a well-chosen path than to sprint along an obvious one. Pete's mentorship, open-mindedness, careful reading, patience and attentiveness convinced me that this maze is worth exploring.

I am thankful to all these wanderers and everything they taught me. I am looking forward to continuing to wander with you and share in the treasures that we find.

Most importantly, I am thankful to Maylin for holding my hand and making me feel that not all who wander are lost. None of this would have been possible without you.

Contents

Chapter 1

Introduction: evolution as algorithm

Evolutionary biology and theoretical computer science are fundamentally interconnected. Already in the 19th-century work of Charles Darwin and Alfred Russel Wallace, we can see the emergence of concepts that theoretical computer scientists now hold as central to our discipline: concepts like asymptotic analysis, the role of algorithms in nature, distributed computation, and analogy from man-made to natural control processes. By recognizing evolution as an algorithm, we can continue to apply the mathematical tools of computer science to solve biological puzzles — to build an algorithmic biology (a term previously used in a similar setting by Richard Watson [225] and Seth Bullock [24]). That is my goal in this thesis.

In this introductory chapter, I take a historical view to motivate my approach to algorithmic biology. Although theoretical computer science is not yet being widely used to structure biological theory, I think there is a lot of historical precedent for building a tighter coupling between the mathematics of computer science and evolutionary biology. I outline the historic and current rational for this in the first four sections of this introduction. If you are not interested in a historic overview or if you are already convinced that *theoretical* computer science can offer a lot to biology then it is safe to skip these first four sections. In Section 1.1, I rehearse the Malthusian argument that prompted Darwin and Wallace to develop the concept of the struggle for existence as the engine of evolution. I stress that this argument has the same sort of asymptotic separation structure that theoretical computer scientists are used to seeing. In Section 1.2, I go into more detail about how the idea of natural selection was developed. I stress that natural selection is the abstraction of a practical algorithm that had been developed for animal breeding. In Section 1.3, I explore the difference between traditional computational biology and the algorithmic biology that is the focus of this thesis. In Section 1.4, I draw a parallel between the development of physics and my proposal for how this area of theoretical computer science should aim to develop as a field.

The remaining three sections of the introduction are about the general contribution of this thesis. I discuss the central methodological contribution in Section 1.5, where I make the distinction between idealization and abstraction in mathematical models. Abstraction is the primary unifying theme of this work, and here I introduce both the theoretical abstraction of Part I of the thesis (in Section 1.5.1) and the empirical abstraction of Part II (in Section 1.5.2). In Section 1.6, I shift from methodology to a discussion of the biological theme of this thesis: the structure of fitness. Finally, in Section 1.7 I discuss the work that serves as my contemporary motivation, set out the main contributions of this thesis, and give a detailed mapping between the chapters and my published work.

The rest of the thesis is structured in two main parts. Part I (Chapters 2 to 5) is about the computational complexity of fitness landscapes, and Part II (Chapters 6 to 8) is about the use of evolutionary game theory in oncology. Both are building on the legacy of Darwin, Wallace and the many evolutionary biologists since them.

1.1 Malthus and asymptotic arguments

Before I turn to Darwin and Wallace, let us read from one of the works that inspired them, Malthus' *Essay on the Principle of Population* [140]:

> Population, when unchecked, increases in a geometric ratio. Subsistence increases only in an arithmetical ratio. A slight acquaintance with numbers will show the immensity of the first power in comparison of the second

Computer scientists will recognise the above as an asymptotic argument. Although Malthus goes on to estimate the exact geometric ratio involved, he also recognises that specifics are not essential to his argument, and that given any geometric factor greater than 1, and any arithmetic factor, eventually the geometric growth would surpass the arithmetic. Malthus was using the fact that – asymptotically – exponentials grow faster than linear and then building a theory on top of this asymptotic separation.

Darwin and Wallace saw the importance of this observation. They also recognised that the essential aspect of it was the asymptotic separation. It did not matter which particular resources implemented the limiting factor. More importantly, it did not even matter what specific sub-exponential function those resources scale with – just that is was sub-exponential. They abstracted

Malthus’ principle as the basis for the struggle for existence. The also defined the multiply-realizable concept of *fitness* as the measure of which organisms would fare better or worse in this struggle. With this they provided an abstract cause and continued driver for natural selection.

These are also the central unifying themes for my thesis: abstraction, fitness, and open-ended evolution.

1.2 Bakewell to Darwin and ‘artificial’ to natural selection

Today, when we make appeals to evolution, we usually place natural selection as primary. For example, when we describe evolutionary medicine, we use terms like ‘using natural selection to achieve therapeutic goals’. In other words, we use a natural process to achieve our artificial goals. This is similar to how one might use the flow of the river to turn a mill.

But this was not the direction from which Darwin approached evolution. Instead, he started with domestication before moving on to variation in nature, laying out the struggle for existence, and only then – in Chapter 4 of *On the Origin of Species* [40] – did he finally define natural selection. It is only after this, late in Chapter 4, that he refers to domestication as ‘artificial’ selection.

Structurally, Darwin’s argument proceeds from looking at the selection algorithms used by humans and then abstracting it to focus only on the algorithm and not the agent carrying out the algorithm. He realised that the breeder’s role as selector can be replaced by another actor: the struggle for existence. He sees the importance of the algorithm of selection and that it can be implemented in many ways. In other words, he sees that evolution is realisable in multiple ways. It is only after we have already accepted Darwin’s explanation that we proceeded to reify natural selection and redefine or explain artificial selection in reference to it.

The flow of ideas clearly went from technology (animal husbandry) to theory (evolution). But there is a problem with my story: domestication of plants and animals is ancient. Old enough that we have transmissible cancers that arose in our domesticated canine helpers 11,000 years ago and persist to this day [155]. Domestication in general – the fruit of the first agricultural revolution – can hardly qualify as a new technology in Darwin’s day. It would have been just as known to Aristotle, and yet he thought species were eternal.

Why wasn’t Aristotle or any other ancient philosopher inspired by the agriculture and animal husbandry of their day to arrive at the same theory as Darwin?

The ancients did not arrive at the same view because it was not the domestication of the first agricultural revolution that inspired Darwin – it was something much more contemporary to him.

Darwin was inspired by the British agricultural revolution of the 18th and early 19th century.

Here, I want to sketch this connection between the technological development of the Georgian era and the theoretical breakthroughs in natural science in the subsequent Victorian era. The history will be brief as I will focus only on the part relevant to evolution as algorithm.

What was the British agricultural revolution? Was it even seen as a revolution in its own time or recognized only in hindsight? For this, we can turn to Victorian texts. We can see a description of the revolution directly in Darwin's writings. For example, in his October 1857 letter to Asa Gray, Darwin writes [25]:

> [s]election has been *methodically* followed in *Europe* for only the last half century

The emphasis is in the original and 'methodical' is key. The innovation that Darwin is alluding to originated in Leicestershire with Robert Bakewell, who built a mechanistic approach to agriculture. Not the replacement of farm workers by machines, but the methodical scientific approach to agriculture. The introduction of methodical inductive empiricism.

In particular, Bakewell is most remembered for the Dishley system – known today as line breeding or breeding in-and-in. Prior to Bakewell, livestock of both sexes were kept together in fields. This resulted in natural assortment between the livestock and did not easily produce systematic traits in the offspring – to the casual onlooker, the traits in the offspring of these populations would be diverse and seemingly random. Bakewell separated the sexes and only allowed deliberate, specific mating. This allowed him to more easily and rapidly select for desired traits in his livestock [80].

During the 35 years from Robert Bakewell inheriting his father's farm in 1760 to his own death in 1795, he developed several new breeds of livestock including new kinds of sheep, cattle, and horses [80, 204]. It was apparent to any observer that these were different variations on species. For example, they produced more wool, gained more weight more quickly, and were easier to work with than prior livestock. During Bakewell's lifetime, the average weight of bulls at auction is reported to have doubled [182, 204].

The Dishley system — i.e., Bakewell's algorithm — clearly produced new varieties. These new varieties raised two puzzles for the naturalists. The first puzzle was an algorithmic one: was a human breeder required to implement Bakewell's algorithm, or was this always taking place even without human intervention? The second puzzle was biological: can the variants established by this algorithm persevere and depart from each other indefinitely, or will they always revert to a common type when selection is relaxed? Darwin answered both these questions. I will discuss Darwin's answer to the algorithmic section here and save the biological question for Section 1.6.1.

In recognizing (artificial) selection, Darwin was not extracting an implicit algorithm from a long-held human practice. Rather, he was taking an explicit algorithm advocated and practiced by his contemporaries. In his *On the Origin of Species*, Darwin [40] explicitly acknowledges Bakewell's demonstration of variation under domestication, and even discusses the branching of Bakewell's variations under the breeding of different farmers (Buckley vs. Burgess; see Chapter 1 of Darwin [40]).

Darwin's contribution to Bakewell's algorithm was abstracting it: recognizing that the agent that implements the algorithm is irrelevant. We do not need to have Robert Bakewell or another agriculturalist do the selecting. Instead, we can have a distributed agent like the "struggle for existence". It is this algorithmic abstraction that allowed Darwin to revolutionize how we think about nature. But it was the latest technology of his day that led him there. Darwin took a human algorithm and asked if it can also explain nature.

In other words, Darwin's approach was to project onto nature the human algorithm corresponding to the actions of breeders. Like a computer scientist or mathematician today, he was using his understanding of human procedures to look at nature.

Bakewell's contribution to the technology of agriculture and influence on the future of evolutionary theory extends beyond breeding. And it extends beyond his direct influence on Darwin's philosophical contribution: Bakewell also established experimental plots on his farm to test different manure and irrigation methods. This practice was part of the inspiration for John Bennet Lawes' establishment of the Rothamsted Experimental Station in 1843 for carrying out long-term agricultural experiments [201].

Bakewell introduced the experimental approach to evolution that I will be continuing to build on theoretically in Chapter 2 and empirically in Chapter 7. In fact, it is not only inspiration that the Rothamsted Experimental Station offers: their 1856 Park Grass Experiment is still ongoing [201]. Although in this thesis I will rely on more contemporary experiments: in Chapter 2, I will turn to Lenski's 30+ year long-term evolution experiment with *E. coli* [231, 127]. In Chapter 7, I will discuss short evolutionary experiments in cancer carried out by my colleagues and me [119]. And I will link both to mathematics.

1.3 Imagination and (computational vs algorithmic) biology

Bakewell and Darwin did not mathematically formalise their insights in the way that theoretical computer scientists might today. For Darwin, this was probably because he did not see himself as a mathematician and even wrote in his autobiography [15] that:

> During the three years which I spent at Cambridge... I attempted mathematics... but got on very slowly. The work was repugnant to me, chiefly from my not being able to see any meaning in the early steps in algebra. This impatience was very foolish, and in after years I have deeply regretted that I did not proceed far enough at least to understand something of the great leading principles of mathematics, for [people] thus endowed seem to have an extra sense. But I do not believe that I should ever have succeeded beyond a very low grade. ... in my last year I worked with some earnestness for my final degree of B.A., and brushed up ... a little Algebra and Euclid, which later gave me much pleasure, as it did at school.

The Rothamsted Experimental Station that Bakewell inspired is perhaps best known for its theoretical contribution to evolutionary biology during the 14-year tenure (1919–1933) of Ronald Fisher. While at Rothamsted, Fisher developed the statistics and population genetics of the modern evolutionary synthesis to make sense of the data from these ongoing evolutionary experiments. Today, Fisher is remembered as an authority on both mathematics and biology.

In the preface to one of the first works of mathematical biology, Fisher [49] devotes some space to the differences between mathematicians and biologists. For Fisher, those early steps of algebra that Darwin found so repugnant are not the foundation of mathematics, but rather a practical technique. The manipulation of mathematical symbols is “comparable to the manipulation of the microscope and its appurtenances of strains and fixatives” [49]. He observed that this difference in tools of the trade is real, but superficial.

The substantive difference between mathematicians and biologists is in how their imagination was trained. The biologist’s imagination is trained on the complexity of the actual and the particular [49]:

> [biologists] are introduced early to the immense variety of living things; their first dissections, even if only of the frog or dog fish, open up vistas of amazing complexity and interest.

The mathematician’s imagination, instead, is trained on the elegance of the abstract [49]:

> The ordinary mathematical procedure in dealing with any actual problem is, after [idealizing] what are believed to be the essential elements of the problem, to consider it as one of a system of possibilities infinitely wider than the actual, the essential relations of which may be apprehended by generalized reasoning, and subsumed in general formulae, which may be applied at will to any particular case considered.

In other words, biologists see the actual particular complexity of the world, while mathematicians idealize ('the essential elements') and then abstract ('consider it as one of a system... infinitely wider') the world.

Darwin could see both the particular and the abstract structure in the world. I think that this ability to not lose the particular while reasoning abstractly is one of the important factors that allowed a break from Aristotelian biology. I do not need to rehearse Darwin's mastery of the particular – there is no doubt that he was a good naturalist. But I hope this chapter also highlights Darwin's mastery of the abstract – the ideas on which the algorithmic biology of this thesis aims to build.

Science had to wait from Darwin until Fisher and his contemporaries around the 1930s for the mathematical theory of evolutionary biology to take shape in the form of the modern synthesis. The modern synthesis developed at the same time as computer science. From the beginning there was interaction between theoretical computer science and biology – just look at Alan Turing's highly influential work on morphogenesis [218], or his unpublished work on neural networks and artificial life. Since then, however, the connection has largely focused on tools and practical concerns; on computational biology. In most cases, computational biology has largely taken computer science as offering a set of practical computational techniques to the working biologist. It has automated that repugnant algebra but it does not push for the different kind of imagination that Fisher identified. Computational biology has made the practical tools of computer science into a new kind of microscope for biologists – a set of techniques so ubiquitous that Markowetz [141] can even argue that all biology is computational biology.

But there is an alternative to traditional computational biology that uses theoretical computer science and the algorithmic lens. The algorithmic lens is not about computers or computer programs – in the same way that astronomy is not about telescopes and that thermodynamics is not about steam engines. Rather, the algorithmic lens highlights the fact that our theories, models and hypotheses are a kind of algorithm in their own right. Thus, we can use the conceptual tools built by theoretical computer scientists for analysing and designing algorithms to instead evaluate and refine our scientific theories, models, and hypotheses.

Evolution is such an algorithm. And we can analyse it using the tools of theoretical computer science. This is algorithmic biology.

Whereas computational biology is a practical branch of biology, algorithmic biology is a theoretical branch of biology. Algorithmic biology is a suite of mathematical techniques and a philosophical disposition taken from theoretical computer science and applied to the conceptual objects of evolutionary biology. This mirrors Fisher's view of the particular vs abstract imagination. A thesis in computational biology might feature simulations, data crunching, and computer programs as central characters. This thesis, especially in the second part, will feature some of this. Primarily, though, this text is a work of algorithmic biology and will thus be centred around theorems, lemmas, proofs and conceptual analysis.

As a branch of theoretical biology, it is important for algorithmic biology to engage with contemporary questions and models in biology. Although this introductory chapter explores history, the subsequent two parts of the thesis will engage the contemporary biological literature. I will introduce and discuss this literature in the chapters that it becomes relevant, rather than here.

1.4 From perpetual motion machines to the halting problem

There seems to be a longstanding tendency to use the newest technology of the day as a metaphor for making sense of our hardest scientific questions. These metaphors are often vague and imprecise. They tend to overly simplify the scientific question and also misrepresent the technology. This is not useful. And this is not what I aim to do with computation and biology in this thesis.

But the pull of metaphors to the latest technology does at least have a beneficial tendency to transform the technical disciplines that analyze this technology into fundamental disciplines that analyze our universe. This was the case for many aspects of physics, as I will discuss below, and I think it is currently happening with aspects of theoretical computer science. This is very useful – so this is what I aim to work towards with the algorithmic biology of this thesis.

If we go back to Darwin and Wallace, then the inspiration on evolution from technology was not limited to agriculture. Steam engines – the other new technology of their day – also make an appearance in the first publication of natural selection in 1858 [41]. In his section, Alfred Russel Wallace writes that the "action of [natural selection] is exactly like that of the centrifugal governor of the steam engine, which checks and corrects any irregularities almost before they become evident." He was proposing an analogy to another recent technology; the centrifugal governor – a self-feedback system – introduced into common usage by James Watt in 1788.

In the next section, we will go back in time to the birth of modern machines – to the water wheel and the steam engine. I will briefly sketch how the science of steam engines developed and how it dealt with perpetual motion machines. From here, we will jump to the analytic engine and the modern computer. I will suggest that the development of computer science has followed a similar path — with the *Entscheidungsproblem* and its variants playing the role of the perpetual motion machine. This will also let me introduce the sort of abstraction over arbitrary dynamics that I will rely on in the first part of the thesis, especially in Chapters 2 and 4.

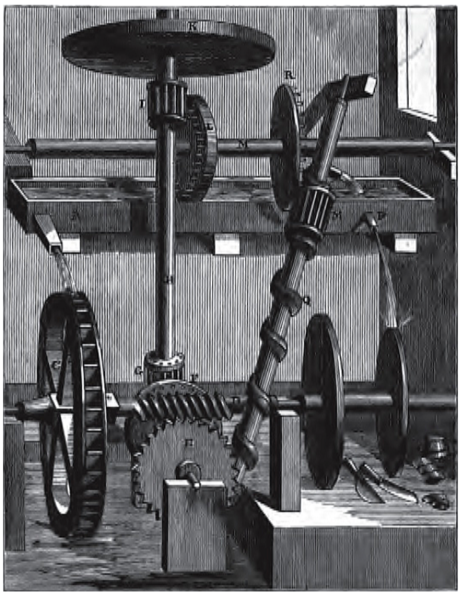


Figure 1.1: **Wood engraving of Robert Fludd's 1618 "water screw" perpetual motion machine [44, 5].** The machine was intended to perpetually drive a millstone. The idea was that water from the top tank turns a water wheel (bottom-left), which drives a complicated series of gears and shafts that ultimately rotate the Archimedes' screw (bottom-center to top-right) to pump water to refill the tank. The rotary motion of the water wheel also drives two grinding wheels (bottom-right) and is shown as providing sufficient excess water to lubricate them. If built, this machine would *not* work as intended and would *not* perpetually produce useful work.

The science of steam engines successfully universalized itself into thermodynamics and statistical mechanics. These are seen as universal disciplines that are used to inform our understanding across the sciences. Similarly, I think that we need to universalize theoretical computer science and make its techniques more common throughout the sciences. This thesis – especially the first part – will focus on the universal consequences and constraints of computation on biology.

1.4.1 Machines and the conservation of energy

As machines started to do more work for us, and as they became increasingly more efficient, it became natural to ask: will these machines ever stop? Can we make machines that do more work than we put in? Can we make perpetual motion machines?

We can see people already trying to make perpetual motion machines with water power as early as the start of the 1600s. For example, see Robert Fludd's 1618 sketch of a water screw perpetual motion machine in Figure 1.1. He imagined the top tank draining to turn a water wheel. The water wheel then cranked a shaft which both turned a millstone to do useful work and powered an Archimedes' screw that pumped water from the lower tank back up to the upper.

This certainly sounds, on first hearing, like it could work. We just need to get the gears running smoothly enough; right?

We find this laughable now, but that was the mindset for a lot of serious thinkers at the time.

As steam engines were developed and proliferated by the late 1700s, the excitement for perpetual motion machines only heightened. With so much mysterious power coming from coal, and newer and newer machines requiring less and less coal to do more and more work. It was easy to think that surely it would be possible to push past the point of 100% efficiency into free energy. It was easy to speculate about this at the time, since the steam engine itself was poorly understood. It lacked a solid theoretical and scientific grounding.

Of course, scientists were also very interested in these engines and they developed the groundwork for making sense of steam and other engines alongside the excitement for perpetual motion. But a modern science of steam engines was not really formed until around 1824 when Sadi Carnot published *Reflections on the Motive Power of Fire and on Machines Fitted to Develop that Power* [27]. This was the birth of the modern technical discipline: thermodynamics.

This did not stop inventors from working on perpetual motion machines, but more sober-minded scientists and engineers started to suspect that it might not be possible to ever build such machines. By 1856, Rudolf Clausius had formulated empirical principles which have since become the first laws of thermodynamics. From these empirical principles, one could finally argue that perpetual motion powering an external system was impossible.

But it was not clear how these empirical principles (or – in the terminology of Chapter 6 – effective theories) arose: maybe a new finding or a new type of engine could overturn them? Maybe we just needed to be more creative with the kinds of machines we considered. Just how widely could these empirical principles apply? Could they be explained or derived from simpler ideas? From the 1870s until the publication of his 1886 *Lectures on Gas Theory*, Ludwig Boltzmann developed a statistical mechanics to explain these empirical principles. He grounded these laws in statistics of the Newtonian laws that were seen at the time as foundational.

Finally, in 1918, Emmy Noether published her groundbreaking theorem that every differentiable symmetry of the action of a physical system has a corresponding conservation law [158]. Now we knew that the conservation of energy was not some odd empirical hypothesis open for challenge. Rather it was a consequence of the form of our physical laws. Conservation of energy was a consequence of invariance of our physical laws under time translations.

Putting all these ingredients together, we could be certain that perpetual motion machines were epistemically impossible. Their existence – in any form – is incompatible with our laws of physics.

But notice how these laws broadened. We started from reasoning about particular machines and particular experiments. We started from a science of steam engines and we ended at fundamental reality. Today, we use thermodynamics and statistical mechanics in all kinds of domains that have nothing to do with steam engines. A narrowly defined technical discipline has grown to be about the whole universe — and we now respect it as a useful tool and sanity check in all our other scientific disciplines.

1.4.2 Algorithms and the complexity of computation

A similar story has developed in computer science, except instead of steam engines, we have algorithms.

In the late 1800s, formal methods in mathematics were improving quickly. Just like improvements to steam engines emboldened the mechanics and inventors, these formal improvements emboldened mathematicians and logicians. After all, they were finding procedures for computing the solutions to more and more difficult mathematical problems. By 1928, Hilbert and Ackermann in *Grundzüge der Theoretischen Logik* asked the *Entscheidungsproblem* [77]:

> What is the procedure that determines for each logical expression for which domains it is valid or satisfiable?

This was the computer science equivalent of asking "What is the design for the perpetual machine?"

Thankfully for computer science, it took less time to find their version of Emmy Noether – i.e. Church, Turing, and Post. This was probably because mathematicians were already looking for formal rather than empirical answers. By 1936 these mathematicians showed the impossibility of Hilbert's dream: there exists no algorithm that can solve the *Entscheidungsproblem* [31, 217, 181].

In particular, it was shown that there were concrete problems – most notably the Halting problem – that no algorithm could solve, at least not in the general case. This was the computer science version of the conservation of energy: a barrier that prevented the wonders we desired and naively imagined as possible. Just like Noether, the computer scientists showed that this complexity limit was a consequence of our logical laws. An algorithm for solving the Halting problem – just like the perpetual motion machine – was epistemically impossible.

Since then, computer science has expanded our understanding of the limits of computation and we now have a richer web of belief on which problems are tractable – i.e., have algorithms that run in polynomial time – and which are intractable. Unfortunately, this web is still centred around a number of conjectures (like P vs NP, or the FP vs PLS conjecture I rely on in Chapter 4) that are strongly believed but not formally resolved [112].

Just like with thermodynamics and statistical mechanics breaking free of steam engines, computer science is rebelling against a view of itself as a specialized technical discipline dealing just with human-made 'algorithm engines'. As with thermodynamics' use of statistical mechanics to ground itself in Newtonian mechanics, the easiest way to universalize computer science was to ground itself in physicalism. This was achieved with Gandy's physicalist variant of the Church-Turing thesis [103, 51]. Its intuitive statement is that any function computable by a physical machine is computable by a Turing machine. A more operationalized statement might be that the statistics of measurement for any repeatable physical process can be approximated arbitrarily well by a Turing machine.

Of course, this is not the only way to universalize theoretical computer science. Personally, I prefer Post's cognitivist variant of the CT-thesis: Turing Machines or other equivalent forms of computation capture what is thinkable by us, and express the restriction of our finite understanding—[181, 110]. In other words, theoretical computer science is the ultimate tool for analyzing our theories, models, and hypotheses.

As theoretical computer science universalizes itself it seeks – just like thermodynamics and statistical mechanics before it – uses for its mathematical tools in the domains of other disciplines. If we recognize theoretical computer science as foundational, then we open a whole new toolbox for understanding the universe. This is a good resource for other sciences and also a great motivation for theoretical computer science. It is this ability of computer science to abstract over arbitrary algorithms – much how Noether could abstract over arbitrary physical machines – that allows me to consider computational constraints on arbitrary evolutionary dynamics in Chapters 2 and 4.

1.5 Idealization vs abstraction

As we saw with Darwin and Wallace, in defining the struggle for existence and fitness, they abstracted over the specific details that implement that struggle and fitness for any particular population. But this was in a verbal argument. In formal mathematical models, biologists tend to use idealization rather than abstraction.

To explain the distinction between idealization and abstraction, let us imagine the various populations studied in evolutionary biology as the miscellaneous collection of triangles in Figure 1.2. We can think of each of these triangles as representing a different biological process that implements evolution. In particular, let us think of each triangle as a different population with its own structure, demography, standing genetic variation, etc. and thus its own corresponding evolutionary dynamic. In the top left corner, the green triangle might correspond to a bee colony with a very specific and strange sex ratio. Over on the bottom right, the orange triangle might be a biofilm of slime mold with their complicated spatial structure. Maybe the blue triangle is a population of antelope undergoing range expansion. We could go on and on. For every empirical population studied by a

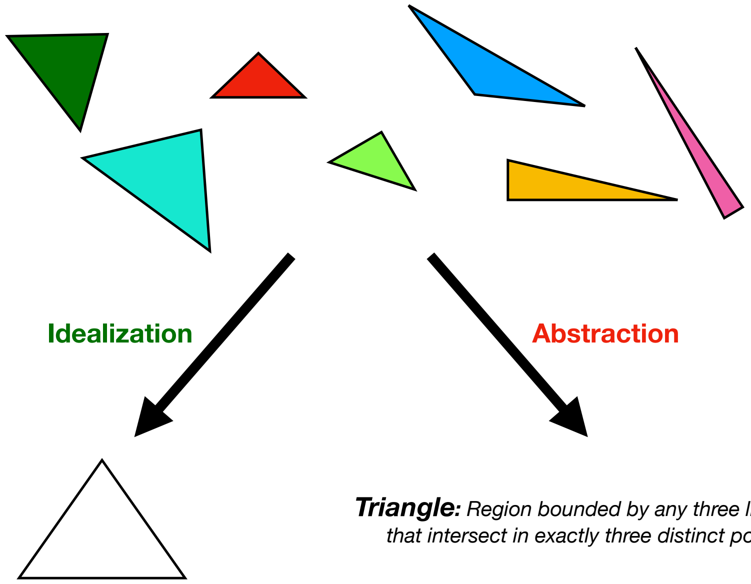

Idealization
Abstraction
Triangle: Region bounded by any three lines that intersect in exactly three distinct points

Figure 1.2: How different biological populations (triangles) can be treated via idealization vs abstraction. Figure from Kaznatcheev [105].

biologist, we might imagine a corresponding triangle.

The point is that they each have some very particular details. These can be very different for each evolutionary dynamic. Biologists deal with this complexity by idealizing: we pick a particularly simple or convenient evolutionary dynamic, one that we think is 'general'. Something like using an equilateral triangle as a stand-in for the mess of real triangles. We will often argue that this particularly simple model gets at the 'essence' of all evolutionary dynamics. But, in reality, our choice is often guided by our methods. We pick the equilateral triangle – for an actual mathematical biologist this might be something like the strong-selection weak-mutation dynamics – because we have the mathematical skills necessary to analyze it. And, from then on, we suppose that all evolutionary dynamics are an equilateral triangle and analyze them as such.

If we end up talking with more experimentally oriented colleagues, we might say: "oh yeah, this is kind of like the green bee colony". But our colleague might study slime molds and we would have to admit that it is not so much like the orange slime mold triangle. At that point, the resourceful modeler might offer to deform their idealized triangle to get one that looks more like the slime molds. So, we end up endlessly modifying our idealized models with various features that we want to add or take into consideration. In practice this is made extra difficult by our lack of knowledge about what kinds of triangles actually occur in nature.

From my experience this idealization approach is the more common approach in theoretical and mathematical biology. But it is not the only approach that we can take. The main contribution

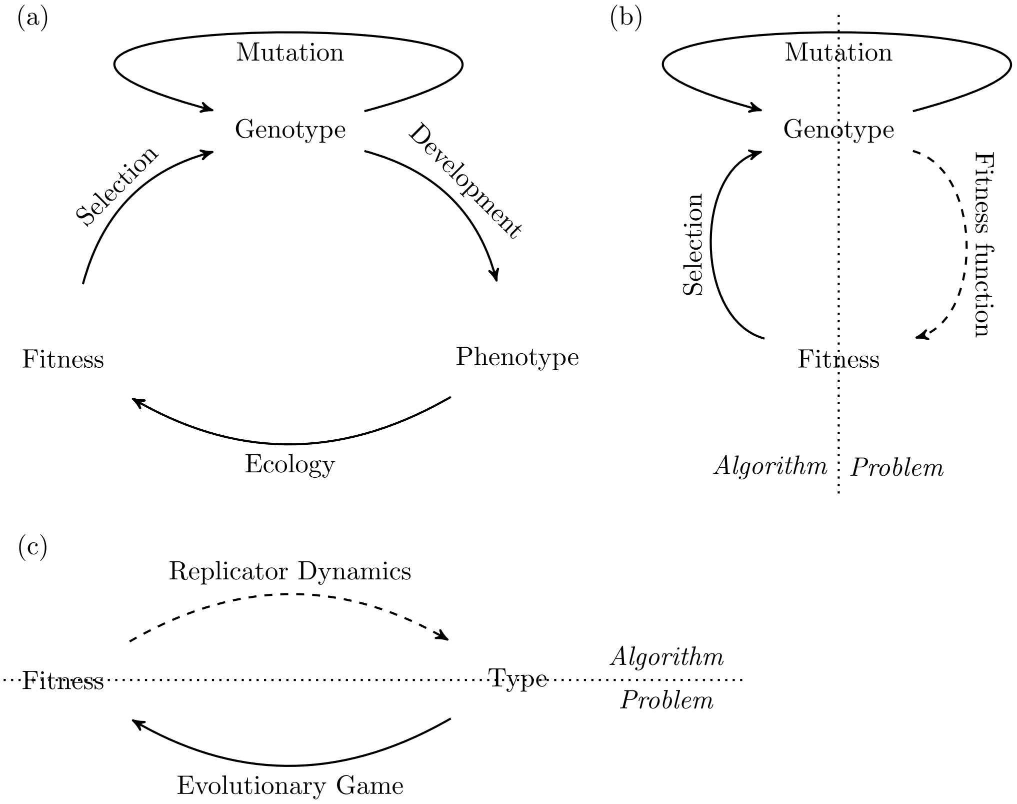


Figure 1.3: **Darwinian engine powering (a) eco-evolutionary dynamics alongside its theoretical (b) and (c) empirical abstractions.** The bottom cycle of (a) captures the struggle for existence (it is inspired by similar figures from presentations by Joachim Krug and Amitabh Joshi), and the top cycle of (a) captures the genesis of new variants. Panel (b) is the theoretical abstraction of (a) that I use in Part I of the thesis for studying evolution on fitness landscapes. Panel (c) is the empirical abstract of (a) that I use for Part II of the thesis to study cancer ecology by measuring evolutionary games. The idealizations that allow the projections are represented by dashed arrows. The dotted line shows how each projection can be divided into algorithm vs problem. This same figure repeats later as Figure 9.1.

and primary unifying theme of my thesis is to provide abstract as an alternative to idealization inspired by the techniques of theoretical computer science.

Of course, both idealization and abstraction are processes and they have to start somewhere. It is easy to point out, for example, that maybe real populations are not actually triangles because they have rounded corners or non-straight sides. As such, we need to take some space of possibilities as the starting point. In the case of this section, that starting point is triangles. In the case of the thesis, that starting point is the Darwinian engine in Figure 1.3 (revisited in more detail in Chapter 9). The engine in Figure 1.3a is made of two cycles that together change the distribution of genotypes. On the top is the genesis of new variants via the 1-arc mutation cycle. On the bottom is the struggle for existence via the 3-arc development-ecology-selection cycle. Figure 1.3a is already a simplification of the kinds of feedbacks that evolutionary biologists care about. Even in this simplification, a lot of details are hidden in each edge. This opens the door for both idealization (the replacement of a given edge by a 'typical' or easy to handle instantiation of it) and abstraction (reasoning about arbitrary instantiations of an edge). But first, let me return to triangles.

1.5.1 Theoretical abstraction and fitness landscapes

Instead of making an idealization, I want to follow the route of abstraction. To do this, we can just note that all the shapes we drew are triangles in Figure 1.2 and then see what we can conclude from properties that all triangles have in common. Any conclusion drawn in this way has the upside of being *multiply realizable*, the abstract conclusion can be implemented by many particular triangles and we do not need to know the details of each particular implementation to know that the conclusion will hold.

Unfortunately, abstraction comes with some downsides. First, it means that we cannot get certain specific results. We can say much more about a specific equilateral triangle than we can about an arbitrary triangle. Second, we lose some things. An equilateral triangle is a concrete triangle, it 'looks' like a triangle. An equilateral triangle 'resembles' the triangles it is modeling. The concept of triangle, however, is not a concrete triangle. It does not 'look' like anything. It does not 'resemble' the system it models. Rather, it specifies a language in which that system can be expressed – something like the logical specification: 'region bounded by any three lines that intersect in exactly three distinct points'. Thus, the abstraction can be of a different kind than the things it abstracts over and we need different tools for dealing with this.

How do we reason about arbitrary triangles? Or in the case of my thesis: how do we reason about arbitrary evolutionary dynamics with arbitrary population structures, etc? This is where the tools of theoretical computer science come in. In particular, I use theoretical abstraction. This is the first part of the thesis.

In this first part, I combine standard idealizations with novel abstractions. I visualize this as the projection of Figure 1.3a to the right as Figure 1.3b. In a standard move for evolutionary biologists, I idealize environments and abstract genotype-to-phenotype-to-fitness maps as fitness landscapes (the dashed line in Figure 1.3b) – i.e., as a map from genotypes to scalar fitness alongside some notion of genetic 'proximity'. This fitness landscape then serves as the 'problem' to be 'solved' by evolution. In a slight twist in Chapter 3, I train the mathematician's imagination on this standard view of fitness landscapes by focusing on all possible fitness landscapes that are expressible given a certain reasonable compact representation (based on valued constraint satisfaction problems [224, 84]).

The truly novel move for evolutionary biologists – a move familiar to theoretical computer scientists – is my abstraction over all possible evolutionary dynamics on these fitness landscapes.

Instead of reasoning about only specific idealized selection-mutation operations like fittest-mutant strong selection weak mutation dynamics, I prove results about any possible evolutionary dynamic. In Chapter 2, I introduce the ultimate constraint of computation that creates 'hard' fitness landscapes on which no evolutionary dynamic can find a local fitness peak in polynomial time. This chapter is largely conceptual with theorems referenced and interpreted but detailed proofs saved for later chapters. In Chapter 4, I present these proofs for arbitrary abstract evolutionary dynamics and I also present some concrete constructions of fitness landscapes that are hard for specific idealized evolutionary dynamics. Finally, in Chapter 5, I turn to the question of parametrized complexity and ask what restrictions have to be placed on our language for describing fitness landscapes in order that we can only describe 'easy' landscapes where local fitness peaks can be found quickly by any simple evolutionary dynamics.

Overall, this first part of the thesis is focused on theory. Although I do present some potential candidates for hard fitness landscapes occurring in nature, and even outline an algorithm in Section 3.7.1 for learning the representations of fitness landscapes from data, there are no actual experiments in Part I. The methodological theme of this part is largely theoretical: I use abstraction to expand the theorist's imagination.

1.5.2 Empirical abstraction and evolutionary games

But abstraction can also help with experiment, not just theory. That is the second part of the thesis.

In the language of triangles, we might care about some specific property of triangles like their area. Normally, we would find this area by measuring all three sides or measuring two sides and the angle between them. Then from these reductive measurements we would compute the effective area. In the context of evolutionary dynamics – especially in the context of evolutionary game theory (EGT; which I overview in Chapter 6) – a particular example might correspond to knowing the pairwise interaction between strategies and then running that interaction over some spatial structure to get some surprising prediction about which strategy comes to dominate the population following this particular spatially structured evolutionary dynamic. As I discuss in Chapters 6 and 8, this direction from reductive to effective has been the standard approach in much of evolutionary game theory.

But do we need to always measure these reductive details that identify a particular triangle? After all, many triangles have the same area. And if we only care about the area then we do not need to know which particular combination of side-lengths resulted in our area. Especially if we can come up with a clever way to measure area directly without measuring side lengths.

I do not know how to do this for the area of triangles, but I do develop the game assay in Chapter 7 to do this for effective evolutionary games. I do this by idealizing the selection-mutation process and the genotype-phenotype map as replicator dynamics, treating this as an algorithm and asking 'what problem is it solving?'. The answer to this becomes an abstraction over ecology that I express in the language of evolutionary games. I visualize this as the projection of Figure 1.3a to the bottom as Figure 1.3c.

This approach can be useful in cancer research where we might care about the outcome for a patient – a global effective property —– but not know the details of the interactions going on within the tumour – the local reductive game. In this case, we want a process like the game assay to measure the global effective game without first having to learn the reductive game and the details of the population structure that transforms it. That is why Chapter 7 concentrates on measuring the games played by non-small cell lung cancer.

Evolutionary game theory (EGT) also provides some of the clearest examples of multiple realizability of these abstract effective games. In Chapter 6, I focus on the multiple-realizability of the replicator equation – the central dynamic in EGT. In Chapter 8, I look at how the same effective games can be multiply realizable. The aspect of population structure that I focus on there is the heavily studied influence of spatial structure on dynamics. Effective games can 'absorb' both the reductive game and the population's spatial structure into a single measurement. Or in other words: the same effective game can be realized by many combinations of different reductive games and spatial structures. In this way, an effective game is an empirical abstraction that can be measured directly via the game assay.

This means that abstraction can be both theoretical and empirical and my thesis develops both methodologies.

1.6 The structure of fitness: from scalars to functions

I expect the methodological theme to be the most impactful contribution of this thesis. But my contribution is united by more than just method: the other unifying theme of this work is the structure of fitness.

1.6.1 Fitness scalars and open-ended evolution

In the first part of the thesis, I take the fitness $w(x)$ of genotype x as a given abstract scalar. The fitness landscape can then be looked at as specifying some notion of $\partial w_x / \partial x$ over an appropriately defined space of mutations. In this way, time is not dealt with explicitly, but the focus is instead on the number of fixations or introductions of new mutations. This is adequate, since we are looking to put a big lower bound on the time required to find local peaks and the time is lower bounded by the number of mutations. The goal is to show that for some landscapes this lower bound is so large – exponential in the size of the genome – that it cannot be realized even over evolutionary time-scales.

With this line of reasoning, I am looking to extend the understanding of open-ended evolution started by Darwin and Wallace. If we revisit Section 1.2, we can see that the primary innovation of Darwin and Wallace was not to show that organisms could change through selection – this was already evident from Bakewell's sheep. Rather it was to deal with the other empirical observation from Bakewell's sheep: when selection was relaxed the sheep appeared to 'return' to a common type. The naturalists at the time – like Karl Friedrich von Gaertner working on hybridization in plants [53] – called this the 'law of reversion'. In today's terminology, the 19th century wisdom was that species were at local peaks, and although human intervention could perturb them as a new variety away from that peak, once that intervention stopped the organisms would quickly return to the species peak. In other words, it was already clear to Darwin and Wallace's audience that new varieties could emerge. What was not clear was that these varieties could form new species (and the mechanism behind them). What was not clear was that it was possible for temporary perturbations to accumulate into permanent divergences and drastically different kinds – that evolution could continue indefinitely. In other words, what was not clear to the Victorians was open-ended evolution. And this is not always clear today, especially in the literature on static fitness landscapes where evolution is assumed to quickly find (and thus equilibrate at) local peaks.

Darwin and Wallace approached the challenge of open-ended evolution as geologists. They knew that the Earth was changing slowly but drastically. These changes could create new environments with each acting as a new 'Bakewell' and selecting for different traits. In modern terminology, this could be seen as a dynamic fitness landscape where the local fitness peak moves away due to environmental change and thus continues to give the populations opportunities to innovate and change. A second standard approach – used, for example, by Darwin in the special case of sexual selection – is to realize that a type's fitness depends on other types. Ecological feedback loops can form between types – such as evolutionary arms races – that force both to innovate.

In the first part of the thesis – especially Chapter 2 – I propose the constraint of computational complexity as a third driver of open-ended evolution. There I show that even in finite static landscapes, it can be effectively impossible to find local fitness peaks. This means that the population can continue to climb in fitness, and thus produce open-ended evolution, even without environmental change or ecological feedback. I link this to the inference of unbounded fitness gain in the *E.*

coli long-term evolution experiment [231, 127]; to the existence of beneficial point-mutations even in well-established ancient genes in yeast [129]; and to equilibrium related puzzles like Baldwin effect, Hankshaw effect, and larger debates in adaptationalism.

1.6.2 Fitness functions and the ecology of cancer

In the second part of the thesis, I start to question whether taking the fitness w as a scalar is sufficiently general. Especially in the context of cancer, the fitness of a given type of cells can depend very heavily on the distribution of other cell types in the environment. Thus, I consider a generalization of fitness from scalar to function. The collection of these "fitnesses as functions" for different types is summarized as an evolutionary game. In other words – as I discuss at the start of Chapter 6 – evolutionary games are a generalization of fitness landscapes. But with this more expressive model comes a price: it is hard to study a very large number of types.

As such, in the EGT setting, we usually zoom in on a few pre-existing types competing with each other. This can be thought of as a kind of ecological dynamics (although in the EGT literature itself, the word 'ecological' is usually reserved for fitness functions that are explicitly density- rather than just frequency-dependent). This zoomed-in view moves me away from looking at $\partial w/\partial x$ to instead focusing explicitly on the time dynamics of each type and its fitness – something closer to dw/dt.

This zoomed in view also forces me to examine more closely how fitness is defined. In Chapter 6, I build on the distinction between token vs type fitness to get two different kinds of evolutionary games: reductive vs effective. Most existing studies can be seen as focused on reductive games, so I instead focus Chapter 7 on effective games and how to measure them in cancer – my focus includes actually carrying out the new experiments required to measure the games played by non-small cell lung cancer. Finally, in Chapter 8, I describe how effective games can be multiple-realized by different kinds of reductive games and spatial structure. In terms of fitness, this means focusing on how type fitness is not a straightforward average over token fitness.

Thus the thesis as a whole abstracts, expands and elaborates our view of fitness without losing sight of particular experiments.

1.7 Contribution

I have already published most of the work in this text in blog posts, preprints and traditional journals. In this section, I want to state some of my main contributions, mention some of the key prior works that motivates those contributions, and provide a mapping between the chapters and my publications.

Chapter 1 borrows text from a series of blogposts that I wrote on my scientific blog *Theory, Evolutionary, and Games Group*, mostly inspired by various talks that I have given on my research.

Sections 1.1 & 1.3 are based on Kaznatcheev [98].

Section 1.2 is based on Kaznatcheev [94].

Section 1.4 is based on Kaznatcheev [104].

Sections 1.5-1.7 are not explicitly based on prior writing, but I developed the distinction between idealization and abstraction over a number of posts [92, 99, 105].

Chapter 2 is based on the main text of Kaznatcheev [96]. Here I develop the theory of hard fitness landscapes: the idea of computational complexity as an ultimate constraint on evolution and the distinction between easy vs hard landscapes. The goal of this chapter is to provide a biological motivation for why we should care about local peaks not being reachable in polynomial time.

I provide connections to existing biological data and sketch the surprising conclusions of the theory of hard fitness landscapes. Specifically, I consider hard fitness landscapes as a new approach to explaining open-ended evolution. For concrete experimental motivations, I consider (1) recent local fitness landscape measurements that found well established genes in certain wildtypes to be either at [183] or away from [129] fitness peaks in yeast and (2) the unbounded growth in the fitness of *E.coli* in the the long-term evolution experiments [231, 127]. On the theory side, I focus on how hard landscapes can resolve puzzles of adaptationalism [168] like the (3) maintenance of costly learning (Baldwin effect [13, 202]) and (4) maintenance of cooperation by hitchhiking (Hankshaw effects [70]). This chapter makes extensive forward references to results proved in the subsequent three chapters, but I save the formal definitions and proofs for those subsequent chapters.

Chapter 3 expands on parts of the appendix from Kaznatcheev [96] and the first half of Kaznatcheev, Cohen, and Jeavons [113]. The primary goal of this chapter is to provide the formal mathematical definitions and some of the techniques necessary for the proofs in the subsequent two chapters.

First, I transform the notions of magnitude, sign, and reciprocal sign epistasis developed by Weinreich, Watson, and Chan [228] and Poelwijk et al. [178], into a definition of three families of landscapes (i.e., classes of problem instances): smooth [39], semismooth, and rugged. The first and last of these are familiar to biologists, but semismooth landscapes are a new development with important consequences that I explore in Chapter 4. The popular biological representation for rugged landscapes is the (classic) NK-model [90, 91, 89]. In Section 3.3, I argue that this model is built on the antiquated one-gene-one-function view of biology, so I advocate for the generalized NK-model – which I call gene-interaction networks – as an alternative. These gene-interaction networks are equivalent to valued constraint satisfaction problem (VCSP) instances, which allows me to connect to a rich existing literature in AI.

In preparation for Chapter 5, I focus on binary Boolean VCSP instances (or, in biological terminology, biallelic gene-interaction networks without higher-order epistasis) since these are sufficiently expressive to capture the difference between easy vs hard fitness landscapes. I develop two kinds of equivalence classes among these instances: magnitude and sign equivalence. The first considers two gene-interaction networks as equivalent if they implement the same fitness function and the second considers two gene-interaction networks as equivalent if they implement the same fitness *graph* [39]. I show that both have a minimal normal form which allows us to unambiguously (i.e., independent of a particular representation) answer which genes interact epistatically in a given fitness landscape. Unfortunately, only the former normal form can be found efficiently: I show that it is NP-hard to find the minimal sign-equivalent gene-interaction network. This means that even if we have a gene-interaction network representation of a fitness landscape, we cannot always efficiently determine certain structural features like the existence of reciprocal sign epistasis. Finally, I also present an algorithm for learning fitness landscapes in Section 3.7.1 – a publication on this algorithm is currently in preparation, to be published in the future.

Chapter 4 is mostly based on the appendix from Kaznatcheev [96] (which itself builds on my old preprint [95]) and a part of Cohen, Cooper, Kaznatcheev, and Wallace [32]. My main contributions here are the proofs of the theorems that the theory of hard fitness landscapes relies on. This requires me to introduce new techniques from theoretical computer science into evolutionary biology. In particular, I unify and extend ideas developed separately by previous work in evolutionary computation, the analysis of simplex algorithms, and the computational complexity of polynomial local search.

Polynomial local search: That finding *global* optima in the classic NK-model of fitness landscape is hard was previously shown for $K \geq 3$ [227] and then $K \geq 2$ [233]. In Section 4.6, I use the hardness of polynomial local search [86] to prove the stronger result that finding any *local* optimum in the classic NK-model for $K \geq 1$ (or generalized NK-model for $K \geq 1$) is hard. I also connect Orlin, Punnen, and Schulz [164]'s approximately locally optimal solutions to the biologist's selection coefficient to argue that, on hard fitness landscapes, fitness can converge to a local optimum as a power-law but not as the exponential convergence typical of equilibration in physical systems. These general results apply to any evolutionary dynamic – even ones that do not follow adaptive paths, jump around, or even build and act on internal models of the fitness landscape – but require the fitness landscape to contain reciprocal sign epistasis (and thus be rugged).

Simplex algorithms: To show how simple – at least with respect to epistasis – hard fitness landscapes can be, I also show that semismooth fitness landscapes that lack reciprocal sign epistasis can be hard for particular evolutionary dynamics. In Section 4.3, I prove that semismooth fitness landscapes are equivalent to acyclic unique sink orientations of the Boolean hypercube that are studied in the analysis of simplex algorithms [208, 144]. By connecting semismooth fitness landscapes to AUSOs, I can adapt existing hardness results from the analysis of simplex algorithms to show hardness of particular semismooth fitness landscapes for particular evolutionary dynamics. Specifically, this allows me to restate Matousek and Szabo [144]'s results about the *RANDOM EDGE* simplex pivot rule in biological terminology: there exist semismooth fitness landscapes that are hard for random fitter-mutant strong-selection weak mutation (SSWM) dynamics. Thus, I show that although a short adaptive path exists from every genotype to the unique peak of a semismooth fitness landscape, evolution cannot in general find this path, but will end up on some long winding path instead.

Evolutionary computation: For the particular algorithm of fittest-mutant SSWM, the recursive *root2path* landscape construction was introduced by Horn, Goldberg, and Deb [79] as an example of a fitness landscape where fittest-mutant SSWM dynamics takes exponentially long to reach a fitness peak. Although this fitness landscape has a single peak, it is still rugged because it requires reciprocal sign epistasis to block potential short adaptive paths to the peak. I show that reciprocal sign-epistasis is not required for this kind of result by providing a recursive construction for a hard winding semismooth fitness landscape in Section 4.4. Although this semismooth fitness landscape does not require reciprocal sign epistasis, I prove, in Section 4.5, that both it and the *root2path* landscape would require dense gene-interaction networks if they were implemented directly instead of recursively. Cohen, Cooper, Kaznatcheev, and Wallace [32] prove that it is possible to build a fitness landscape that has a sparse gene-interaction network of bounded treewidth and is hard for fittest-mutant SSWM but I do not include this construction here.

Chapter 5 is based on and extends the second half of Kaznatcheev, Cohen, and Jeavons [113], Section 5.4 is not yet published. The main contribution here is to develop a better view of the boundary between gene-interaction networks that produce easy vs hard fitness landscapes. Whereas Chapter 4 aimed to classify how restrictive the model of fitness landscapes can be while still being able representing hard landscapes, this chapter aims for the other direction: what is the most expressive model that can only represent easy landscapes? In this way, the goal is to expand the class of provably easy fitness landscapes beyond smooth ones.

In this chapter, I use a structural measure of easiness: the nonexistence of any long adaptive paths to a local fitness peak. This was previously used in the optimization literature to classify easy instances of local MAX-CUT using span arguments, showing that MAX-CUT on graphs of degree less than 3 is easy [180, 153]. Since MAX-CUT instances can be seen as special kinds of VCSP instances, I generalize this span approach to all VCSPs by asking for the minimal number of distinct fitness levels in any VCSP that is sign-equivalent to the desired fitness graph. I show that all VCSP instances with a constraint network of degree less than 2 are easy. However, I also show that this span argument cannot extend to general degree ≥ 3 networks or even to trees of degree ≥ 4.

To overcome the limit on the span argument, I introduce a new proof technique based on encouragement paths in Section 5.2. I use the encouragement path technique to prove that any tree-structured binary Boolean VCSP has adaptive paths of at most quadratic length. I argue that this result is tight by providing examples of fitness landscapes with long adaptive paths that are represented by trees on domains of size 3, or Boolean with constraint graphs of tree-width 2. Together, all these results allow me to build a preliminary map of the boundary of hard vs easy fitness landscapes in Section 5.4 that acts as a summary of Part I.

Chapter 6 is based mostly the first half of Kaznatcheev [111], although I first presented the view of games as an abstraction of fitness landscapes with Peter Jeavons in the 2019 Mathematical Oncology Roadmap [188]. The primary goal of this chapter is to provide a brief overview of evolutionary game theory, especially as it has been used for the study of microscopic systems like cancer, and to develop my distinction between reductive vs effective games.

By starting with Abrams [1]'s distinction between token vs type fitness, I develop two different ways that economic game theory can be interpreted in evolutionary terms. If we take payoffs to be a token fitness then we arrive at reductive games and if we take payoffs to be a type fitness then we arrive at effective games. The latter of these is a new interpretation of evolutionary games that I argue is easier to link to experiments in microscopic systems. This chapter also showcases the multiple realizability of the central algorithm of EGT: replicator dynamics. I do this by collection in one place many known but conceptually different implementations of the replicator equation.

Chapter 7 is based on an interweaving of the main text and appendix of Kaznatcheev et al. [119]. The main contribution here is my development of the game assay and our first ever application of it to measure the ecology of non-small cell lung cancer.

Although there has been much interest and effort in making the evolutionary game theory of microscopic systems more data driven [220, 121, 219, 138, 66, 131, 8], most of this prior work follows a two-track approach. In the two-track approach, theory and experiment are done side-by-side and success is judged from (an often informal) hypothesis-testing or model-selection perspective by looking at agreement between the macroscopic output of a reductive theory and the experiment. In this chapter, I combine these two parallel tracks into a single track by experimentally operationalizing the effective games of Chapter 6 as an assayable hidden variable of a population and its environment.

Implementing the game assay in practice requires me to design and – together with my colleagues – carry out new kinds of experiments that I report in this chapter. Although not currently common in cancer biology, competitive fitness assays are a gold standard for studying bacteria. In a competitive fitness assay, two cell types are seeded in a petri dish at a known ratio (usually 1:1) and then the fitness of one or both types is measured. Typically, such a competitive fitness assay is conducted with a single initial ratio of two competing cell types. I define the experimental part of the game assay as the extension of the competitive fitness assay to a series of different initial seeding ratios. For the analysis part of the game assay, I show how to transform the outputs of these fitness measurements into a game point in a two-dimensional game space.

Heterogeneity in strategies for survival and proliferation among the cells that constitute a tumour is a driving force behind the evolution of resistance to cancer therapy [151, 74]. By carrying out the game assay in an experimental *in vitro* non-small cell lung cancer system, I am able to discover new things about the evolution of drug resistance in cancer. My colleagues and I measured this system to be playing either the LEADER or DEADLOCK game based on the presence or absence of environmental factors like fibroblasts and the drug *Alectinib*. The shift between these two qualitatively different game types confirms the previously theoretical postulate of EGT in oncology: it is possible to treat not just the player but also the game. The Deadlock game in the absence of fibroblasts and drug challenges the common theoretical assumption of the cost-of-resistance.

Chapter 8 combines ideas from the second half of Kaznatcheev [111] with Kaznatcheev [100] and includes figures from Kaznatcheev, Scott, and Basanta [115] and my contribution to the 2019 Mathematical Oncology Roadmap [188]. My main contribution here is to show the multiple realizability of effective games by changes in spatial structure.

Although it is well known that spatial structure can transform evolutionary game dynamics [159, 122, 73, 132, 163, 207, 196, 137], this transformation is not usually presented as a shift from a reductive to an effective game. The Ohtsuki-Nowak transform [162] is a notable exception to this – although it does not use the terminology of reductive vs effective games, it does present spatial structure as a game transformation. In this chapter, I use this transform to argue for inverting the direction of inference for EGT in microscopic system: instead of starting with an intuitive reductive game and adding spatial structure to get a surprising effective game, I suggest that we start with a measured effective game and 'subtract' spatial structure to get a surprising reductive game. I also show concrete examples of the multiple realizability of effective games by different combinations of space and reductive game (for example, the reductive Hawk-Dove game on 3-regular random graphs can produce the effective Leader game that we measured in Chapter 7) and warn against over-interpreting measured effective games because space can both create and hide frequency-dependent interactions.

Chapter 9 is the conclusion. Here I revisit the Darwinian engine from Figure 1.3 as a unifying theme of the thesis and catalog some of the questions for future work that the prior chapters opened. The chapter shares its structure with the thesis by following three famous quotes: "Nothing in biology makes sense, except in the light of evolution" [45]; "Nothing in evolution makes sense, except in the light of ecology" [69]; and "nothing in evolution or ecology makes sense, except in light of the other" [173]. The goal of this chapter and the thesis as a whole is to see new aspects of biology by the lights of evolution and ecology focused through the algorithmic lens.

Part I

Theoretical Abstraction: Evolution on Fitness Landscapes

> Nothing in biology makes sense except in the light of evolution
>
> Dobzhansky [45]

Chapter 2

Computational complexity as an ultimate constraint on evolution

Experiments show that evolutionary fitness landscapes can have a rich combinatorial structure due to inter-locus interaction known as epistasis [178, 30, 135, 166, 183, 129, 125, 209, 14, 46]. In this first part of the thesis, I will show that for some landscapes, this structure can produce a computational constraint that prevents evolution from finding local fitness optima – thus overturning the traditional assumption that local fitness peaks can always be reached quickly if no other evolutionary forces challenge natural selection. To do this, I introduce a distinction between the *easy* landscapes of traditional theory, where local fitness peaks can be found in a moderate number of steps (polynomial in the number of loci) and *hard* landscapes where finding local optima requires an infeasible amount of time.

In this chapter, I introduce this theory of hard landscapes, give an overview of its results, and discuss their importance for evolutionary biology. This is meant as an extended abstract and biological motivation for the proofs and detailed discussion of the results that I save for the subsequent three chapters of Part I:

Chapter 3 will formalize the notion of fitness landscapes and their representations;

Chapter 4 will formalize the theory of hard landscapes; and

Chapter 5 will refine the border between easy vs. hard landscapes.

This Chapter is intended to motivate the biological relevance of hard landscapes and uses common biological terminology that might be unfamiliar to some readers. If you prefer to see formal definitions and proofs before the biological consequences then I would recommend reading

Chapters 3, 4 and 5 before coming back to this chapter.

Throughout Part I, we will see that hard examples exist even among landscapes with no reciprocal sign epistasis (Definition 3.6); on these "semismooth" fitness landscapes, strong selection weak mutation dynamics cannot find the unique peak in polynomial time. More generally, on hard rugged fitness landscapes no evolutionary dynamics – even ones that do not follow adaptive paths – can find a local fitness optimum quickly. Moreover, on hard landscapes, the fitness advantage of nearby mutants cannot drop off exponentially fast but has to follow a power-law that long term evolution experiments have associated with unbounded growth in fitness. Thus, the constraint of computational complexity enables open-ended evolution on finite landscapes.

Knowing this constraint allows us to use the tools of theoretical computer science and combinatorial optimization to characterize the fitness landscapes that we see in nature. I present biological candidates for hard landscapes at scales from single genes, to microbes, to complex organisms with costly learning (Baldwin effect [13, 202]) or maintained cooperation (Hankshaw effect [70]).

After motivating evolutionary constraints as the mechanisms and phenomena that can keep a population from reaching a fitness peak, and after introducing the distinction between proximate vs ultimate constraints in Section 2.1, this chapter is structured by increasing abstraction. In Section 2.2, I will discuss concrete families of fitness landscapes in which specific evolutionary dynamics (i.e., algorithms) like random fitter-mutant and fittest-mutant strong-selection weak-mutation dynamics cannot find a unique local (and thus also global) fitness peak after a polynomial number of beneficial allele fixations. In Section 2.3, I will abstract to a broader class of evolutionary dynamics to see that there exist fitness landscapes in which all *adaptive dynamics* – that is, dynamics that follow *adaptive paths* (formally, Definition 3.1) and move populations strictly 'uphill' without any assumptions on which fitter mutant is fixed – require a super-polynomial number of beneficial allele fixations to find a local peak, or to even get close to an approximate peak. Finally, in Section 2.4, I will abstract to the level of any evolutionary dynamics – with no assumptions about following adaptive paths nor about respecting genetic distance – and still see the existence of hard fitness landscapes where local fitness peaks cannot be found in polynomial time.

There is a price that I will have to pay for more abstract and powerful results. I will make this payment in concreteness. Section 2.2 is based on specific constructions for hard landscapes. By Section 2.3, instead of providing concrete constructions for hard landscapes, we will see a sequence of transformations (reductions) that can be applied to existing constructions of circuits to, in principle, create concrete examples. Finally, in Section 2.4, I will have to rely completely on reductions and conjectured complexity class separations like FP $\neq$ PLS (see Section 4.6 for an explanation). Throughout this chapter, I will focus on providing an overview and suggesting some high-level consequences for biology of these hardness results. I will save the formal definitions for Chapter 3, proofs of hardness for Chapter 4, and proofs of tractability for Chapter 5.

For this price, abstraction will yield rewards. By expanding from strict fitness peaks to approximate peaks in Section 2.3, I will be able to better incorporate the numeric (rather than just combinatorial) structure of fitness landscapes and link to existing empirical observation of fitness traces in the *Escherichia coli* long-term evolution experiment [231]. This will allow the theory of hard landscapes to provide a new kind of explanation for the slow equilibration mode and open-ended evolution observed in these experiments. By abstracting to the level of arbitrary evolutionary dynamics in Section 2.4, I will be able to deal with the dynamics of specific complex population structures without the need for simplifying assumptions. This can be especially useful in populations that interweave Darwinian evolution and individual costly learning – such as in the Baldwin effect. Or in populations that interweave evolutionary and ecologically dynamics in spatially structured populations – such as the deme-structured populations (or, in spatial terms: a population distributed over islands with limited migration) of cooperators in the Hankshaw effect. In particular, I will be able to provide a new kind of explanation for how costly learning and cooperation can be maintained in these cases without needing to appeal to just-in-time environmental change. Finally, in Section 2.5, I will note some new general consequences that my theory of hard landscapes offers to adaptationism. And I will discuss how this offers different metaphors and raises new questions and lines of inquiry for biology, regardless of whether hard landscapes occur in nature or not. Just how ubiquitous hard landscapes (and the corresponding ultimate constraint on evolution) are in nature becomes an open empirical question for future work.

2.1 Proximate vs ultimate constraint

We usually imagine fitness landscapes (defined formally in Section 3.1) as hills or mountain ranges, and continue to assume – as Wright [234] originally did – that on an arbitrary landscape "selection will easily carry the species to the nearest peak". Biologists define a *constraint on evolution* as anything that keeps a population from reaching a local fitness peak. For those that view evolution as a sum of forces, with natural selection being only one of them, it is possible for other forces to act as a constraint when they overpower natural selection and keep the population away from a local fitness peak. Such cases are often associated with maladaptation [38] and are usually attributed to mechanisms like mutational meltdown [136], mutation bias [236, 235], recombination [133], genetic constraints due to lack of variation, or explicit physical or developmental constraints of a particular physiology [120]. I will refer to such situations, where non-selection forces (and/or aspects internal to the population) keep the population from reaching a local fitness peak, as ***proximate*** **constraints on evolution**.

In contrast, I will define a constraint as an ***ultimate*** **constraint on evolution** if it is due exclusively to features of the fitness landscape and is present in the absence of other forces, or even holds regardless of the strength of other forces.

All constraints are either proximate, ultimate, or a mix of the two. I introduce this terminology of proximate and ultimate *constraints* by analogy to Mayr's distinction between proximate and ultimate *causes* in biology [148]. Mayr considered as ultimate only those evolutionary causes that are due exclusively to the historic process of natural selection [11], so I consider as ultimate only those evolutionary constraints that are due exclusively to the fitness landscape structure of natural selection.

The distinction that I am making between proximate and ultimate constraints can be made clearer by reference to a distinction in computer science between algorithms and problems that I discuss in more detail in Section 3.5. I will consider the population structure, update rules, developmental processes, mutation operator or bias, etc as together specifying the algorithm that is evolution. In contrast, the families of fitness landscapes are like problems to be 'solved' by evolution and specific fitness landscapes on which populations evolve are problem-instances. A *proximate constraint* is any feature of the evolutionary algorithm that prevents the population from finding a local fitness optimum in polynomial time. For a classic example, consider a population with an extreme lack of genetic variation that cannot proceed to an adjacent fitter genotype because the allele that it differs in is simply not available in the population. In this case, the proximate constraint of lack of variation due to the details of this particular population's evolutionary algorithm prevents it from reaching a fitness peak. In contrast, an *ultimate constraint* is any feature of the problem (i.e. family of fitness landscapes) that prevents the population from finding a local fitness optimum in polynomial time. It is the goal of this chapter and Chapter 4 to show convincing examples of such constraints.

One candidate for an ultimate constraint on evolution – hysteresis or path-dependence – is already widely recognized. A local peak might not be the tallest in the mountain range, so reaching it can prevent us from walking uphill to the tallest peak. An example of this would be the needle-in-a-haystack landscape where all genotypes have fitness zero (the haystack) except one special genotype y (the needle) that has fitness one. In this case, the haystack is one giant plateau with most points (except those directly adjacent to y) as local optima. Being stuck in this fitness zero local-'peak' plateau prevents us from finding the global peak at y. A less drastic example might have two different peaks with separate basins of attraction, with the taller peak having a much smaller basin of attraction.

This constraint of hysteresis has directed much of the work on fitness landscapes toward (1) how to avoid sub-optimal peaks, or (2) how a population might move from one peak to another [234, 160]. Usually, these two questions are answered with appeals to the strength of other evolutionary forces. Although sometimes the second question is sidestepped by postulating that local fitness peaks are part of the same fitness plateau in a holey adaptive landscape and thus fitness valleys can be bypassed to move between different local optima in the plateau [56, 55]. But both of these types of questions implicitly assume that local peaks (or plateaus) are easy to reach and thus the norm for natural selection (or even the starting point in the case of the needle-in-a-haystack). When the constraint of hysteresis is active, being at one local optimum prevents the population from reaching other (higher) local optima. Thus, this candidate for an ultimate constraint is only partial: it prevents only certain – not all – local fitness optima from being found. In this case, it prevents evolution from finding the highest local peak: the global optimum. But, we seldom consider that even reaching *any* local optimum might be impossible in a reasonable amount of time.

In this chapter, I show that computational complexity is an ultimate constraint on evolution: it can prevent evolution from finding *any* local fitness peak (or local fitness plateau) – even low fitness ones. In other words, a careful analysis will show that the combinatorial structure of fitness landscapes can prevent populations from reaching any local fitness peaks. This suggests an alternative metaphor for fitness landscapes: fitness landscapes as mazes with the local fitness optima as exits. Natural selection cannot see far in the maze and must rely only on local information from the limited genetic variation of nearby mutants. I will show that, in hard mazes, we can end up following exponentially long winding paths to the exit because we cannot spot the shortcuts. In such cases, even if natural selection is the only force acting on the population, a fitness optimum cannot be found within even evolutionary timescales. Worse yet, the hardest mazes might not have any shortcuts and even the most clever and farsighted navigator will not know how to reach an exit in a feasible amount of time. In other words, even if the other evolutionary forces 'conspire to help' natural selection, in these cases a local fitness optimum cannot be found within even evolutionary timescales.

To establish these results, I will introduce into biology new techniques from theoretical computer science for analyzing the complexity of fitness landscapes. I embrace the randomness within the algorithm – i.e., the randomness of evolution. But instead of introducing a convenient-to-analyze distribution of possible fitness landscapes, I focus on worst-case analysis (for more justification, see Section 3.6). In this way, this part of the thesis can be seen as a contribution to the small but growing literature on population genetics and evolutionary biology through the algorithmic lens [133, 221, 229, 87, 95, 29, 134, 75, 124].

By focusing on worst-case analysis, I am constructing – sometimes implicitly – families of fitness landscapes that are consistent with the logical structure of our hypothesis class of conceivable fitness landscapes. I then show that in these hard fitness landscapes, computational complexity is an ultimate constraint. But this should not be interpreted as a claim that hard landscapes are ubiquitous or that computational complexity is a *major* or prevalent constraint. That would be an empirical question that depends on which fitness landscapes occur in nature. In this chapter, I suggest several candidates that I suspect correspond to hard landscapes, but the general empirical question of ubiquity is beyond the scope of this thesis.

2.2 Smooth vs semismooth and easy vs hard families of landscapes

What makes some fitness landscapes difficult to navigate is that the effects of mutations at different loci interact with each other. As I will discuss formally in Section 3.2, *epistasis* is a measure of the kind and amount of inter-locus interactions. A landscape without *sign epistasis* (Definition 3.5) – like the *Escherichia coli* β-lactamase fitness landscape measured by Chou et al. [30] – is called *smooth* (Sections 3.2.1 and 4.2; [228, 39]), and I will call a fitness landscape *semismooth* if it has no reciprocal sign epistasis (Definition 3.6; Sections 3.2.1 and 4.3). The *fitness graphs* ([39] and Section 3.1) of semismooth fitness landscapes are equivalent to acyclic unique sink orientations previously defined in a different context by Szabó and Welzl [208] for the analysis of simplex algorithms (Definition 4.7 and Proposition 4.8). Since reciprocal sign epistasis is a necessary condition for multiple peaks (Corollary 4.6 and Poelwijk et al. [179]), both smooth and semismooth fitness landscapes have a single peak x^*. Further, there are short adaptive paths in both: from any genotype x there always exists some adaptive path to x^* of length equal to the number of loci on which x and x^* differ (Theorem 4.9). This means that an omniscient navigator that always picks the 'right' adaptive point-mutation can be guaranteed to find a short adaptive path to the peak. But unlike smooth landscapes, in a semi-smooth landscape not every shortest path is adaptive and not every adaptive path is short. And since evolution does not have the foresight of an omniscient navigator, it is important to check which adaptive paths myopic evolutionary dynamics are able to find and follow.

Mutation is said to be *weak* when mutations arise so infrequently that we can assume that a population is always monomorphic except for a brief moment of transition as a new mutant fixes. Thus, we can represent the population as a single point on the fitness landscape with an evolutionary step corresponding to a selective sweep that moves the population to a neighbouring genotype. Selection is said to be *strong* when the only mutations that fix are those that increase fitness, in which case we can further assume that the evolutionary step takes us to a neighbouring genotype of higher fitness. The rule for selecting which neighbour ends up fixing depends on the details of our mutation operator and model of evolution (i.e. this rule specifies the algorithm). The set of algorithms corresponding to all such rules is known as **strong-selection weak mutation (SSWM) dynamics** (in biology) or as local search algorithms (in computer science).

A number of rules (or algorithms) for the SSWM dynamics have been suggested for which fitter neighbour will take over the population [166] – such rules correspond to different models of evolution. The two most common rules are to select a fitter mutant uniformly at random, or to select the fittest mutant. These rules capture the intuition of evolution proceeding solely by natural selection with other forces absent or negligible.

All SSWM rules will quickly find the (unique global) fitness optimum in a smooth fitness landscape. But there exist semismooth fitness landscapes such that, when starting from a random initial genotype, an exponential number of evolutionary steps will be required for either the random fitter-mutant ([144]; Theorem 4.10) or fittest-mutant (Theorems 4.16 and 4.20) dynamics to find the unique fitness optimum. For a small example on six loci, see Figure 2.1: the black arrows trace the evolutionary path that a population would follow under fittest mutant SSWM dynamics. Although two-step adaptive paths exist to the fitness peak (e.g., 000000 → 000001 → 000011), the myopic navigator cannot notice these shortcuts and ends up on a long winding path. In other words, even when there is a single peak and adaptive paths of minimal length to it, SSWM dynamics can take exponential time to find that peak.

These results show that the computational complexity due to the combinatorial structure of the fitness landscape can be enough to stop evolution from reaching a fitness optimum within a reasonable timescale, even in the absence of suboptimal local peaks. Computer scientists have found it helpful to distinguish between processes that require a time that grows polynomially with the size of the input – generally called tractable – and those that require a time that increases faster than any polynomial (super-polynomial) – which are called intractable. If the winding fitness landscapes of Figure 2.1 is generalized to $2n$ loci instead of just 6 (Section 4.4) then following fittest mutant SSWM dynamics to the peak is an intractable process since it scales exponentially, requiring $2^{n+1} - 2$ mutational steps. Although evolutionary time is long, it is not reasonable to think of it as exponentially long. For example, the above winding process with a genotype on just 120 loci and with new set of point-mutants and selective sweep at a rate of one every second would require more seconds than the time since the Big Bang.

To capture this infeasibility of super-polynomial scaling in time, I introduce a distinction between *easy* and *hard* families of fitness landscapes. If we can guarantee for any landscape in the family that a local fitness peak can be found by natural selection in a time that scales as a polynomial in the number of loci – as is the case for smooth fitness landscapes – then I will call that an ***easy* family of landscapes**. I will call a family of landscapes a ***hard* family of landscapes** if we can show that the family contains landscapes where finding a local fitness optimum requires a super-polynomial amount of time – as I described above for one particular family of semismooth fitness landscapes. Given that even for moderately sized genomes such large times are not realizable even on cosmological timescales, I will use "impossible" as a shorthand for "requiring an infeasible amount of time".

Given their exponential size, it is impossible to completely measure whole fitness landscapes on more than a few nucleotides. But with improvements in high-throughput second-generation DNA sequencing there is hope to measure local fitness landscapes of a few mutations away from a wildtype [183, 129]. And in Section 3.7.1, I will discuss how we might use these local fitness measurements to learn the gene-interaction networks that specify the whole fitness landscape.

As examples of existing local landscape measurements, Puchta et al. [183] estimated the fitness of 981 single-step mutations of a 333-nucleotide small nucleolar RNA (snoRNA) gene in yeast. They found no neighbours fitter than the wild type gene. This suggests that this gene is already at a fitness peak, and hence that the snoRNA gene's fitness landscape may be easy. In contrast, Li et al. [129] estimated the fitness of 207 single-step mutants of a 72-nucleotide transfer RNA (tRNA) gene, also in yeast, finding two neighbours that are significantly fitter than the wildtype and a number that are fitter but only within experimental noise. Thus, the wildtype tRNA gene is apparently not at a local fitness peak, which suggests this system as a candidate hard fitness landscape. Both studies also looked at many 2- and 3-step mutants, and the landscape of the tRNA gene was measured to have more than 160 cases of significant sign epistasis [129], with none in the snoRNA landscape [183], mirroring the difference between hard semismooth fitness landscapes and easy smooth landscapes that I am proposing here.

2.3 Rugged landscapes and approximate peaks

But there exist natural fitness landscapes that are even more complicated than semi-smooth ones. Rugged fitness landscapes – like the Lozovsky et al. [135] *Plasmodium falciparum* dihydrofolate reductase fitness landscape that I will discuss more in Section 3.2 – that contain reciprocal sign

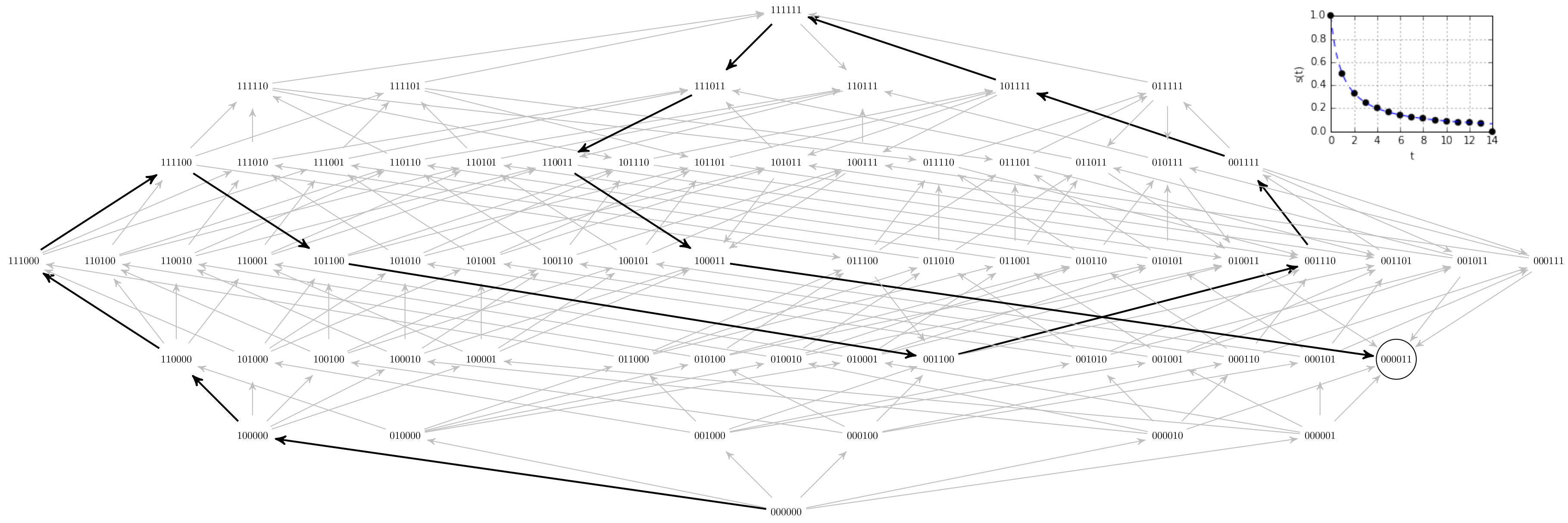


Figure 2.1: **Fittest mutant adaptive path in a winding semi-smooth fitness landscape.** An example on six loci of the winding semismooth fitness landscapes constructed in Section 4.4 on which the length of the path followed by fittest-mutant SSWM dynamics scales exponentially with the number of loci. Here the black arrows are the fittest available mutation, and the adaptive path takes 14 steps to reach the fitness peak at 000011. For the generalization of this landscape to $2n$-loci, it would take $2^{n+1} - 2$ steps for fittest mutant dynamics to reach the fitness peak at $(00)^{n-1}11$ (Theorem 4.16). Inset is the selection coefficient ($s(t) = \max_{y \in N(x_t) \cup \{x_t\}} \frac{w(y) - w(x)}{w(x)}$; Section 4.7) versus mutation step number (t) for the fittest mutant adaptive path.

epistasis and can have multiple peaks. As I discuss in more detail in Chapter 3, such rugged landscapes can be represented by either the classic NK-model (Section 3.3) or the generalized NK-model (that I introduce in Section 3.4). Both the classic NK-model for $K \geq 2$ and generalized NK-model for $K \geq 1$ can generate hard fitness landscapes where from some initial genotypes, any adaptive walk to any local peak is exponentially long (Corollary 4.26). On such landscapes, any adaptive evolutionary dynamic – including, but not limited to, all the SSWM dynamics we've considered so far – generally requires an exponential number of steps to reach a local fitness optimum. Even if an omniscient navigator could always choose the most clever adaptive single mutation to arise, the adaptive path would not reach a local fitness optimum within polynomial timescales.

To better integrate the numeric structure of fitness, let us consider a genotype x to be at an s-approximate peak [164] if each of x's mutational neighbours y have fitness $w(y) \leq (1 + s)w(x)$ (Definition 4.28). On the hard rugged fitness landscapes described above, fittest mutant dynamics will encounter an s-approximate peak with moderately small s in a moderate number of mutational steps (polynomial in n and $1/s$; Theorem 4.30).

However, on these same hard fitness landscapes, it is not possible to find an s-approximate peak for very small s in a feasible amount of time (i.e. not possible in time polynomial in n and $\ln 1/s$; Theorem 4.32). This (un)reachability of s-approximate fitness peaks is especially important to consider in discussions of nearly-neutral networks and approximate fitness plateaus [161, 55]. In an idealized, unstructured population, we can expect random drift to overcome selection when s drops below about $1/P$ where P is the number of individuals in the population. But certain structured populations can act as amplifiers of selection [171] and prevent drift from dominating until s is significantly closer to zero.

Given that the quantity s in the definition of an s-approximate peak is defined in the same way as the selection coefficient of population genetics [62], the above approximation results allow us to link the distinction between easy and hard fitness landscapes to the rich empirical literature on fitness traces and declining fitness gains in microbial evolution experiments [36]. On the hard rugged fitness landscapes described above – and even on the winding semi-smooth landscape of Figure 2.1 and Section 4.4 – this selective coefficient drops off at the slow rate of $s(t) \approx 1/t$ for fittest mutant dynamics. In general, on any family of landscapes – even the hardest ones – $s(t)$ can decay as fast as a power law. On easy landscapes, it can decay faster, but the power law decay in selection coefficient is the fastest decay possible on hard fitness landscapes. In particular, the selective coefficient, on hard landscapes, cannot decrease at the exponential rate (i.e. $s(t) \approx e^{-t}$; Corollary 4.33) that is typical of equilibration in non-biological systems. This slow decay in selection coefficient is consistent with the rule of declining adaptability observed in various microbial long-term evolution experiments [231, 127, 36], suggesting that at least some naturally occurring microbial fitness landscapes might be hard. Thus, a natural candidate for hard landscapes might be the landscapes with unbounded growth in fitness observed in the *E. Coli* long-term evolutionary experiment [231]. Whereas when one sees a power-law in allometry, one expects potential physical constraints [68, 120]; I propose that when one sees a power-law in selection strength or fitness, one should look for a computational constraint.

The existence of hard landscapes allows us to explain open-ended evolution as a consequence of the ultimate constraints of computational complexity. I am certainly not the first to note that populations might undergo unbounded increases in fitness and open-ended evolution. In fact, there is an extensive literature on the rate of adaptation [59, 60, 61, 167, 165, 231, 36] that seems to assume (at least implicitly) that a fitter mutation is always available. These models often directly build-in unbounded growth by treating mutations as independent random samples from a distribution of fitness effects that can always generate a higher fitness variant, albeit with low probability. So, although these models are also called fitness landscapes, they are not like the combinatorially structured fitness landscapes that I discuss in this thesis. To disambiguate, I will refer to these rate of adaptation models as Orr-Gillespie landscapes or unbounded tree landscapes.

If we want to imagine Orr-Gillespie landscapes in a way similar to the fitness landscapes that I study in this thesis then we have to create them over an infinite number of types (instead of the 2^n types in a biallelic fitness landscape on n loci) and usually give them an infinite branching factor (instead of the n for a fitness landscape on n loci) with a new branch for every possible sample from the mutation distribution. This approach corresponds to implicitly constructing a fitness landscape as an infinite unbounded tree that lacks the second-order and higher combinatorial structure that mutation-graphs provide. These unbounded tree models are currently better suited to empirical operationalization than (the exponentially large but) finite fitness landscapes and they make good effective theories on shorter timescale (where time – measured in the number of fixations – is significantly less than the size of the genome). But these models simply assume (often by reference to recent environmental change) that a beneficial mutation is always possible, rather than explain why such a mutation is always possible. Thus, unbounded tree landscapes presupposes that local fitness peaks cannot be reached since no local fitness peaks exist in these models. In this way, Orr-Gillespie landscapes build-in unbounded growth in fitness.

This is in stark contrast to the work I present in this chapter. To avoid building-in the unbounded growth in fitness that I aim to explain, I consider families of finite fitness landscapes. I show that these can be either easy or hard. In the hard families of landscapes, there is a computational constraint on evolution that ensures that beneficial mutations are available for effectively ever. Thus, this work can be read as an explanatory complement to the unbounded tree landscapes. Of course, given that I consider large but finite fitness landscapes, it is conceivable that a population will be found at a local fitness peak of a hard fitness landscape. This is conceivable in the same way as – according to the Poincaré recurrence theorem – all the oxygen molecules in a large room will eventually return arbitrarily close to the corner they were released from. But just as the Poincaré recurrence theorem does not invalidate the second law of thermodynamics [21], the existence of local peaks in finite static landscapes does not invalidate the general principles of open-ended evolution on hard fitness landscapes.

2.4 Arbitrary evolutionary dynamics: learning and cooperation

As we move from single genes [183, 129], to microbes [231, 36], and on to large organisms, a richer space of possible evolutionary dynamics opens up. To capture this rich space of possibilities, we need to abstract beyond adaptive dynamics by considering arbitrary mutation operators, demographies, population structures and selection functions – even ones that can cross fitness valleys and distribute the population over many genotypes. From the perspective of constraints on evolution, I want to now consider the effect of relaxing the selective constraint that confines populations to an adaptive path ([16]; Defintion 3.1). By allowing non-adaptive changes, I want to highlight the power of the constraint of computational complexity, even in the absence of the selective constraint. From the perspective of evolutionary forces, we have to allow for other strong forces that can potentially overpower or boost the force of natural selection. To make sure that we have considered all possibilities, I will model arbitrary evolutionary dynamics as the class of all polynomial-time algorithms. This takes us into the realm of the computational complexity class of polynomial local search (PLS; Johnson, Papadimitriou, and Yannakakis [86], Roughgarden [189] and Section 4.6). But even for these most permissive population-updating procedures, I will show in Chapter 4 that evolution will in general require an infeasible amount of time to find a local fitness peak in the classic NK-model with $K \geq 2$ or generalized NK-model with $K \geq 1$ (Theorem 4.25 and Corollary 4.27), or to find an s-approximate peak for very small s (Theorem 4.32). Evolution will still be trapped in the mazes of hard fitness landscapes and not reach anywhere near the 'exit' of a local fitness optimum. In other words, no proximate cause can overpower the ultimate constraint of computational complexity.

If one is accustomed to seeing results only for particular evolutionary algorithms, then the generality of the above results might seem fantastical. But these are exactly the kind of general results that are typical in computational complexity theory. By linking evolutionary biology to theoretical computer science, we can abstract over the details that implement particular evolutionary dynamics.

The strength of this ultimate constraint allows us to reason rigorously from disequilibrium to establish positive results. For instance, that costly learning (Baldwin effect [13, 202]) can remain adaptive, or that hitchhiking can maintain cooperation (Hankshaw effect [70]) effectively forever. In the case of costly learning, Simpson [202] noted: "[c]haracters individually acquired by members of a group of organisms may eventually, under the influence of selection, be reinforced or replaced by similar hereditary character". For Simpson [202] this possibility constituted a paradox: if learning does not enhance individual fitness at a local peak and would thus be replaced by simpler non-learning strategies, then why do we observe the costly mechanism and associated errors of individual learning?

A similar phenomenon is important for the maintenance of cooperation. Hammarlund et al. [70] consider a metapopulation that is not sufficiently spatially structured to maintain cooperation (see Chapter 8 for more on the effects of space on evolutionary games). They augment the metapopulation with a number of genes with non-frequency dependent fitness effects that constitute a static fitness landscape. If adaptive mutations are available, then cooperators are more likely to discover them due to the higher carrying capacity of cooperative clusters. This allows cooperation to be maintained by hitchhiking on the genes of the static fitness landscape. Hammarlund et al. [70] call this hitchhiking the Hankshaw effect and for them it constitutes a transient: since cooperation does not enhance opportunities for adaptive mutations at the fitness peak, then cooperators will be out-competed by defectors.

Currently, both the Baldwin and Hankshaw puzzles are resolved in the same way: by invoking just-in-time environmental change. Most resolutions of the Baldwin paradox focus on non-static fitness in rapidly fluctuating environments that are compatible with the speed of learning but not with evolutionary adaptation. Similarly, Hammarlund et al. [70] suggest making their transient permanent by focusing on dynamically changing environments. But, these just-in-time dynamic changes in the fitness landscape are not necessary if we acknowledge the existence of hard static fitness landscapes. Individual costly learning and higher densities of cooperative clusters leading to more mutational opportunities are two very different evolutionary mechanisms for increased adaptability. But they are both just polynomial-time algorithms. Regardless of how much these mechanisms speed-up, slow-down, guide, or hinder natural selection, the population will still not be able to find a local fitness optimum in hard fitness landscapes. Without arriving at a fitness optimum, the paradox of costly learning dissolves and the Hankshaw effect can allow for perpetual cooperation. This suggests that if we want a family of natural examples of evolution on hard fitness landscapes among more complex organisms, then good candidates might be populations with costly learning or persistent cooperation. More generally, the non-vanishing supply of beneficial mutations on hard landscapes can allow selection to act on various mechanisms for evolvability [16] by letting the evolvability-modifier alleles hitchhike on the favourable alleles that they produce.

2.5 General consequences for adaptationism

These examples can be seen as instances of a more general observation on adaptationism. It is standard to frame adaptationism as "the claim that natural selection is the only important cause of the evolution of most nonmolecular traits and that these traits are locally optimal" [168]. In this first path of the thesis, I show that these are two independent claims. Even if we assume that (1) natural selection is the dominant cause of evolution then – on hard fitness landscapes – it does not follow that (2) traits will be locally optimal. Given the popularity of equilibrium assumptions in evolutionary biology, I expect that future work could ease a number of other paradoxes and effects, in addition to the Baldwin effect and Hankshaw effect, by recognizing the independence of these two claims of adaptationism.

For those biologists who have moved on from debates about adaptationism and instead aim to explain the relative contribution of various evolutionary forces to natural patterns, I provide a new consideration: hard landscapes allow the force of natural selection on its own to explain patterns such as, for example, maladaptation. Prior accounts of maladaption rely on forces like deleterious mutation pressure, lack of genotypic variation, drift and inbreeding, and gene flow acting opposite to natural selection, resulting in a net zero force and thus a maladaptive equilibrium away from a fitness peak [38]. The ultimate constraint of computational complexity allows for perpetual maladaptive disequilibrium even in the absence of (or working against) these other forces.

Currently, finding a species away from a local fitness peak is taken as motivation for further questions on what mechanisms or non-selective evolutionary forces cause this discrepancy. In this context, my results provide a general answer: hard landscapes allow adaptationist accounts for the absence of evolutionary equilibrium and maladaptation even in experimental models with static environments – and/or the absence of strong evolutionary forces working against natural selection – like in the cases of the tRNA gene in yeast [129, 46] or the long-term evolutionary experiment in *E. coli* [231]. By treating evolution as an algorithm, we see that time can be a limiting resource even on long evolutionary timescales. These hard landscapes can be finite and deceptively simple

– having only limited local epistasis or not having reciprocal sign-epistasis – and yet allow for effectively unbounded fitness growth.

In contrast, a system found at a local fitness peak – like the snoRNA gene in yeast [183] – currently merits no further questions. The results in this chapter show that establishing evolutionary equilibrium should not be the end of the story. We need to also explain what features of the relevant fitness landscapes make them easy: i.e., explain why these fitness landscapes do not produce a computational constraint on evolution. For this, the tools of theoretical computer science can be used to refine our logical characterization of such fitness landscapes to guarantee that local peaks can be found in polynomial time. For example, we could consider limits on the topology of the gene-interaction network, or the type of interaction possible between genes [28] to separate easy from hard landscapes. In Chapter 5, I follow the first approach. I show that the longest adaptive path has at most $O(n^2)$ fixations for any family of landscapes where each biallelic gene interacts with at most two other genes (Theorem 5.6) or where the biallelic gene-interaction network is tree-structured (Theorem 5.15). Thus, these families of landscapes are provably easy. Classifying families as easy vs hard opens new avenues for both empirical and theoretical work. I develop a preliminary mapping of the boundary between easy vs hard landscapes in Section 5.4.

2.6 Summary and conclusion

In this chapter, I discussed the mathematical constructions for hard fitness landscapes (that are proved formally in Chapter 4) and suggested some empirical candidates. By doing this, I showed that computational complexity is an ultimate constraint on (our models of) evolution. But I did not establish that it is a *major* constraint in nature. Given the empirical candidates that I suggested, I expect it to play a major role. However, after future empirical investigations, it could be that we find no naturally occurring hard fitness landscapes. This would not be a disappointment. If our models of fitness landscapes allow for ultimate constraints but we do not see those ultimate constraints in nature, then we will know the direction in which to refine our models.

Given the limited – albeit growing [178, 30, 135, 209, 166, 183, 129, 125, 14, 46] – empirical data available on the distribution of natural fitness landscapes, it is tempting to turn to theoretical distributions of fitness landscapes. But we should be cautious here. As I will discuss in more detail in Section 3.6, the popular uniform distributions over gene-interaction networks and interaction components was introduced for ease of analysis rather than some foundational reason or empirical justification. With this distribution, I would expect hard instances to be scarce based on arguments similar to Tovey [215] and Hwang et al. [81]. However, instead of choosing a distribution for ease of analysis, we could instead choose one by Occam's razor: i.e. the Kolmogorov universal distribution (sampling landscapes with negative log probability proportional to their minimum description length). In the Occam case, I would expect the fraction of fitness landscapes that are hard to be significant based on results similar to Li and Vitányi [130] (see Section 4.8 for more discussion). I leave it as an open question for future work to determine what choices of distribution of fitness landscapes are most appropriate, and how average case analysis over those particular distributions compares to the distribution-free analysis that I have presented here and formalize in Chapter 4.

On easy landscapes, it is reasonable to assume that evolution finds locally-well-adapted genotypes or phenotypes. If we prove that a family of landscapes is easy (as I do for some families in Chapter 5) then we can continue to reason from fitness peaks (i.e. draw conclusions from the assumption that populations will easily reach or already are at a fitness peak), debate questions of crossing fitness valleys, and seek solutions to Wright [234]'s problem of "a mechanism by which the species may continually find its way from lower to higher [local] peaks".

But with hard landscapes, it is better to think of evolution as open-ended and unbounded. We will have to switch to a language of "adapting" rather than "adapted". We will have to stop reasoning from equilibrium – as I did in the discussion of maintaining costly learning and cooperation. Finally, we will have to stop asking about the basins of attraction for local peaks and instead seek mechanisms that select which unbounded adaptive path evolution will follow. It is tempting to read this language of disequilibrium and negation of "locally adapted" as saying that organismal traits are not well honed to their environment. But we must resist this mistake and we must not let better be the enemy of good. Finding local optima in the hardest landscapes is a hard problem for any algorithm, not just biological evolution. In particular, it is also hard for scientists: on hard landscapes we cannot find optimal solutions either, and so the adapting answers of evolution can still seem marvelously well-honed to us. Although I have focused on biological evolution, we can also look for hard landscapes in other fields. For example, these results translate directly to areas like business operation and innovation theory, where the NK-model is used explicitly [128, 186]. In physics, the correspondence between spin-glasses and the NK-model can let us look at energy minimization landscapes. In economics, classes of hard fixed-point problems similar to PLS are used as a lens on markets [189, 64]. In all these cases, theoretical computer science and combinatorial optimization offer us the tools to make rigorous the distinction between easy and hard landscapes. They allow us to imagine hard landscapes not as low-dimensional mountain ranges but as high-dimensional mazes that we will search for-effectively-ever.

Chapter 3

Representing fitness landscapes as gene-interaction networks

Genotype and fitness are two central concepts in evolutionary biology. Through its production of a phenotype and that phenotype's interactions with the biotic and abiotic environment, a given genotype has a certain fitness. A fitness landscape idealizes this relationship between genotypes (or phenotypes) and fitness.

In Section 3.1, I provide a formal definition of fitness landscapes and the idea of fitness graphs (Definition 3.2) to capture the combinatorial structure of the landscape. In Section 3.2, I discuss the idea of magnitude (Definition 3.4), sign (Definition 3.5), and reciprocal sign epistasis (Definition 3.6) and their corresponding families of smooth, semismooth, and rugged landscapes, alongside some empirical examples (Figure 3.3). The distinction between these different classes of landscapes will be central to the hardness results in Chapter 4. For rugged landscapes, I define the popular classical NK-model (Definition 3.7) in Section 3.3 and explain how it hard-codes the one-gene-one-function perspective. Given that the one-gene-one-function perspective is inelegant and outmoded, I introduce a generalized NK-model in Section 3.4 to remove the one-gene-one-function assumption. This generalized model is equivalent to instances of the valued constraint satisfaction problem (Definition 3.8) from artificial intelligence, which allows me to make powerful links to computer science in Chapter 5. But to make best use of these links, I have to introduce the computational distinction between problem vs algorithm (Section 3.5) and justify the use of worst-case analysis instead of assuming convenient-to-analyze distributions over fitness landscapes (Section 3.6).

An important feature of the generalized NK-model is the gene-interaction network as a compact representation of the fitness landscape. Unfortunately, without restrictions, this representation is not unique, so I have to investigate the minimal representations and the unique normal forms that they give rise to. First, in Section 3.7, I show that any fitness landscape has a minimal necessary magnitude-equivalent (see Definition 3.13) gene-interaction network that encodes the structure of magnitude (and higher) epistasis and that can be easily computed. As a bonus, we can use the correspondence between multilinear polynomials and these minimal magnitude-equivalent gene-interaction networks to learn fitness landscapes from local measurements (Section 3.7.1). Second, in Section 3.8, I introduce the idea of considering landscapes to be sign-equivalent (see Definition 3.18) if they have the same fitness graph. I show that a minimal normal form still exists, but it is NP-hard to compute (Theorem 3.26).

3.1 Fitness landscapes

In 1932, Wright introduced the metaphor of a fitness landscape [234]. The landscape is a genetic space where each vertex is a possible genotype and an edge exists between two vertices if a single mutation transforms the genotype of one vertex into the other. Fitness landscapes combine numeric fitnesses and a mutation-graph into a combinatorially structured space where each vertex is a possible genotype (or phenotype). The *numeric structure* is given by a function that maps each genotype to a fitness; typically represented as a non-negative real number and having different physical operationalizations in different experimental systems (as I will discuss in more detail in Chapter 6). The domain of this function has structure: A given genotype is more similar to some rather than other genotypes – giving us a notion of genetic distance or mutation-graph. The *mutation-graph* specifies which genotypes are similar, typically as edges between any two genotypes that differ in a single mutation. This provides the combinatorial structure.

The rest of this section explains the above terse definition more explicitly.

A genotype is a local fitness peak (or local fitness optimum) if no adjacent genotype in the mutation-graph has higher fitness. In Chapters 4 and 5, I will focus on the reachability of these peaks.

Formally, I will model the genotypes (or points), A, in a fitness landscapes as assignments to a collection of n genes (variables), indexed by the set of loci $[n] = 1, 2, \dots, n$, with domains of alleles $D_1, \dots, D_n$. Hence each point corresponds to a vector $x \in D_1 \times \cdots \times D_n$. We will generally focus on uniform domains (i.e., cases where $D = D_1 = \cdots = D_n$), where this simplifies to $x \in D^n$. In particular, we will often be interested in biallelic systems (i.e. Boolean domains), where $x \in \{0, 1\}^n$, so each genotype can be seen as a bit-vector.

To make this set, A, of genotypes into a space, we need a notion of genetic proximity or similarity. For this, I will define a **neighbourhood function** on the set of points A to be a function $N : A \to 2^A$. For simplicity, I will assume this function is symmetric in the sense that if $y \in N(x)$, then $x \in N(y)$, and I will call such a pair x and y **adjacent** points. Throughout the thesis, I will primarily focus on the case where the set of points A is the set of assignments $D_1 \times \cdots \times D_n$ and N is the **1-flip neighbourhood** defined as the unit Hamming ball by $y \in N(x)$ if and only if there is a variable position i such that $x_i \neq y_i$ and this is the only difference (i.e., $\forall j \neq i \quad x_j = y_j$). In the case of the Boolean domain, the graph of the function N, where the edges are the pairs of adjacent points, is the n-dimensional hypercube.

We can also think of this genetic space in terms of a mutation graph, where – for a biallelic system – a mutation can flip any loci from one allele to the other, thus two strings $x, y \in \{0,1\}^n$ are adjacent if they differ in exactly one bit. These two views of the combinatorial structure can usually be treated as equivalent, but I discuss a subtle and important distinction in Section 3.5.

The last ingredient, fitness, is given by a function that maps each string to a numeric value, usually a non-negative real or rational number. In this chapter, I will concentrate on functions that map to natural numbers. This is not a significant limitation, as fitness values can generally be re-scaled arbitrarily, but it simplifies some aspects of the presentation.

Given a set of genotypes, A, I define a **fitness function** on A to be an integer-valued function defined on A, that is, a function $f : A \to \mathbb{Z}$. Because I am modelling fitness, rather than cost, we want to *maximise* this objective functions in this thesis. But all results can also be carried over directly to the minimisation context (that is more popular in physics and machine learning) by flipping signs.

Populations of individual organisms can be thought of as inhabiting the vertices of the landscape corresponding to their genotype, with a polymorphic population distributed over many vertices. And we imagine evolution as generally trying to 'climb uphill' on the landscape by moving to vertices of higher fitness.

Definition 3.1. In a fitness landscape with fitness f, a path $x_1 ... x_t$ is called an **adaptive path** if for each $1 \leq i < t$, $x_{i+1} \in A$ is a neighbour of x_i and $f(x_{i+1}) > f(x_i)$.

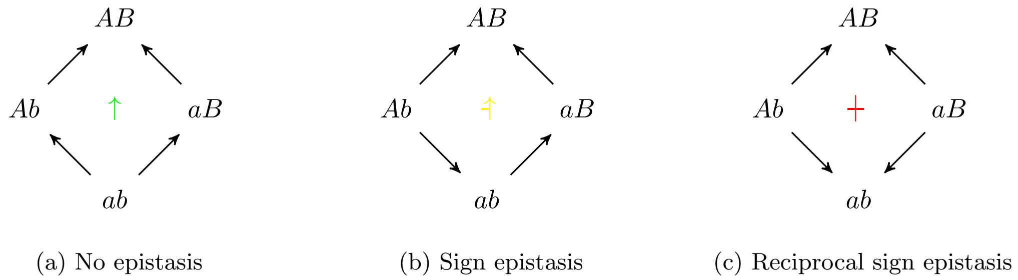
These are sometimes also called *accessible* paths, but I will avoid this terminology because for the most general evolutionary dynamics, the paths taken don't have to be strictly increasing in fitness; i.e. they don't have to necessarily be adaptive. In other words, for arbitrary evolutionary dynamics, evolution can follow (or access) non-adaptive paths. So non-adaptive paths are accessible to arbitrary evolutionary dynamics; and it would be awkward to say that non-accessible paths are accessible to arbitrary evolutionary dynamics. If some particular evolutionary dynamic produces only adaptive paths, though, then it is called an **adaptive dynamic**.

Sometimes it is useful to represent a fitness landscape as a *fitness graph* [42, 39] by replacing the fitness function by a flow on genotypes: for adjacent genotypes in the mutation-graph, direct the edges from the lower to the higher fitness genotype. This results in a characterization of fitness landscapes of a biallelic system as directed acyclic graphs on $\{0,1\}^n$. But generally, and more formally:

Definition 3.2 ([42, 39])**.** Given any fitness landscape (A, f, N), the corresponding **fitness graph** G has vertex set $V(G) = A$ and directed edge set $E(G) = \{xy \mid y \in N(x) \text{ and } f(y) > f(x)\}$.

Note that the edges of the fitness graph consist of all pairs of adjacent points which have distinct values of the fitness function, and are directed from the lower value of the fitness function to the higher value. If there are adjacent edges of equal fitness then it is customary to have no edge between them (although some people might draw an undirected edge in a different style). When we interpret the genetic space as the space of possible mutations, then the directed edges of the fitness graph represent the possible moves that can be made by adaptive dynamics or a local search algorithm. In such an interpretation (and especially in the context of the AI literature), the fitness graph could also be a called a *transition graph.* Finally, in the fitness graph representation of landscapes, fitness peaks correspond to sinks, and adaptive paths (or the traces of local search algorithms) correspond to paths that follow the edge directions of the directed graph. I will consider a population to be at evolutionary equilibrium if it is at a local peak in the fitness landscape (i.e., a sink in the fitness graph).

Crona, Greene, and Barlow [39] introduced the fitness graph representation explicitly into theoretical biology, but fitness graphs have been used implicitly in earlier empirical studies of fitness landscapes [42, 50, 67, 209]. Using fitness graphs is particularly useful empirically when it is difficult to quantitatively compare fitnesses across experiments. In this case, pairwise competition assays can be used to determine the edge direction. However, if pairwise competitions are used to build an empirical fitness graph, it is important to verify that the graph is transitive (acyclic) [223]. If the fitness graph is non-transitive then we know that fitness cannot be represented as a scalar – I discuss more general representations of fitness theoretically in Chapter 6 and engage with this empirically in Chapter 7. In theoretical work where fitness can be treated as a scalar, the fitness graph approach has made the proofs of some classical theorems relating local structure to global properties easier and shifts our attention to global algorithmic properties of evolution instead of specific numeric properties. The amenability of fitness graphs to proof will be most evident in



(a) No epistasis (b) Sign epistasis (c) Reciprocal sign epistasis

Figure 3.1: **Three different kinds of epistasis possible in fitness graphs.** Arrows are directed from lower fitness genotypes towards mutationally-adjacent higher fitness genotypes. Genes a, A and b, B are labeled such that fitness $f(AB) > f(ab)$. In the centre of each graph is a marker for the type of epistasis, the marker's various rotations and reflections cover the cases where AB does not have the highest fitness. For this more exhaustive classification and discussion see Figure 3.2

Chapters 4 and 5.

Throughout this thesis, I consider individual genotypes (or the corresponding phenotypes) as the domain of the fitness landscapes. Thus, I am focusing on micro-evolutionary processes. Given the extremely long time scales that I am considering in this report, it is also natural to consider generalizations where the vertices in the fitness landscape are interpreted as whole species and mutations as speciation events. For simplicity, I will not explicitly discuss such macro-evolutionary processes.

3.2 Epistasis and empirical fitness landscapes

An important structural feature of fitness landscapes is that the effects of mutations at different loci interact with each other. *Epistasis* is a measure of the kind and amount of inter-locus interactions. Here, I will follow Weinreich, Watson, and Chan [228] and Poelwijk et al. [178]'s classification of epistasis into three kinds: magnitude, sign, and reciprical sign. Consider two loci in some particular genetic background (i.e., the value of the loci outside the two were are focused on) with the first having alleles a or A, and the second b or B. For brevity, I will omit writing the particular genetic background (but I will start writing out the genetic background explicitly again in Section 3.8). Assume that the upper-case combination is more fit: i.e. $f(ab) < f(AB)$.

Definition 3.3. Two loci are **non-interacting** if the fitness effects are additive and independent of background: $f(AB) - f(aB) = f(Ab) - f(ab)$, $f(AB) - f(Ab) = f(aB) - f(ab)$.

In *magnitude epistasis* this additivity is broken, but the signs remain: $f(AB) > f(aB) > f(ab)$ and $f(AB) > f(Ab) > f(ab)$. More formally:

Definition 3.4. Two loci have **magnitude epistasis** if $f(AB) - f(aB) \neq f(Ab) - f(ab)$ and/or

$f(AB) - f(Ab) \neq f(aB) - f(ab)$ but $f(AB) > f(aB) > f(ab)$ and $f(AB) > f(Ab) > f(ab)$.

Note that the difference between non-interacting loci and magnitude epistasis is not invariant under strictly monotone transformations of the fitness function, thus fitness graphs will not distinguish between the two types. Hence, non-interacting loci and pairs with magnitude epistasis induce the same kind of subgraph in their fitness graph and this subgraph is shown in Figure 3.1a. Throughout this thesis, I will often use 'no epistasis' to cover both non-interacting loci and magnitude epistasis and I will put the '$\uparrow$' symbol between two loci to represent that they have no epistasis.
A system has *sign epistasis* if it violates one of the two conditions for magnitude epistasis. For example, if the second locus is b then mutation from a to A is not adaptive, but if the second locus is B then mutation from a to A is adaptive. More formally:

Definition 3.5 ([228])**.** Given two loci, if $f(AB) > f(aB) > f(ab) > f(Ab)$ then there is **sign epistasis** at the first locus.

In other words, if the fitness effect of a mutation $a \to A$ can have a different sign depending on the genetic background b or B of another locus then these two loci are said to have sign epistasis. This is called "sign" epistasis because the condition in Definition 3.5 could have equivalently been written as $\mathrm{sgn}(f(AB) - f(aB)) \neq \mathrm{sgn}(f(Ab) - f(ab))$. In terms of induced fitness (sub) graphs, that means that there is an adaptive path from ab to AB by mutating the second locus (i.e $ab \to aB \to AB$) but the path through the first locus is inaccessible to adaptive dynamics (i.e. $ab \not\to Ab \to AB$). Alternatively, if G is the fitness graph under consideration then this condition could also be written in terms of the fitness graph as $\{(ab)(aB), (aB)(AB), (Ab)(AB)\} \subseteq E(G)$ but $(ab)(Ab) \notin E(G)$. This is shown in Figure 3.1b: for $x_1 \in \{a, A\}$ and $x_2 \in \{b, B\}$, I introduce the symbol $x_1 {\dashv}\!{\uparrow} x_2$ to represent sign epistasis at the first locus and $x_1 {\uparrow}\!{\vdash} x_2$ for sign epistasis at the second locus.

Finally, a system has *reciprocal sign epistasis* if both conditions of magnitude epistasis are broken [178, 179, 39]. This is like having sign epistasis on both loci.

Definition 3.6 ([178])**.** Given two loci, if $f(AB) \geq f(ab)$ but $f(ab) > f(Ab)$ and $f(ab) > f(aB)$ then there is **reciprocal sign epistasis** between those two loci.

In other words, if both mutations have one sign on their own, but the opposite sign together – either bad + bad = good or good + good = bad – then the landscape has reciprocal sign epistasis [178, 179, 39]. A classic example of reciprocal sign epistasis is a lock-and-key, changing just one of the lock or the key breaks the mechanism, but changing both can be beneficial. In terms of induced fitness (sub)graphs, that means that there is no direct adaptive path from ab to AB through either the first or second locus: $ab \not\to Ab \to AB$ and $ab \not\to aB \to AB$. This is shown in Figure 3.1c: for

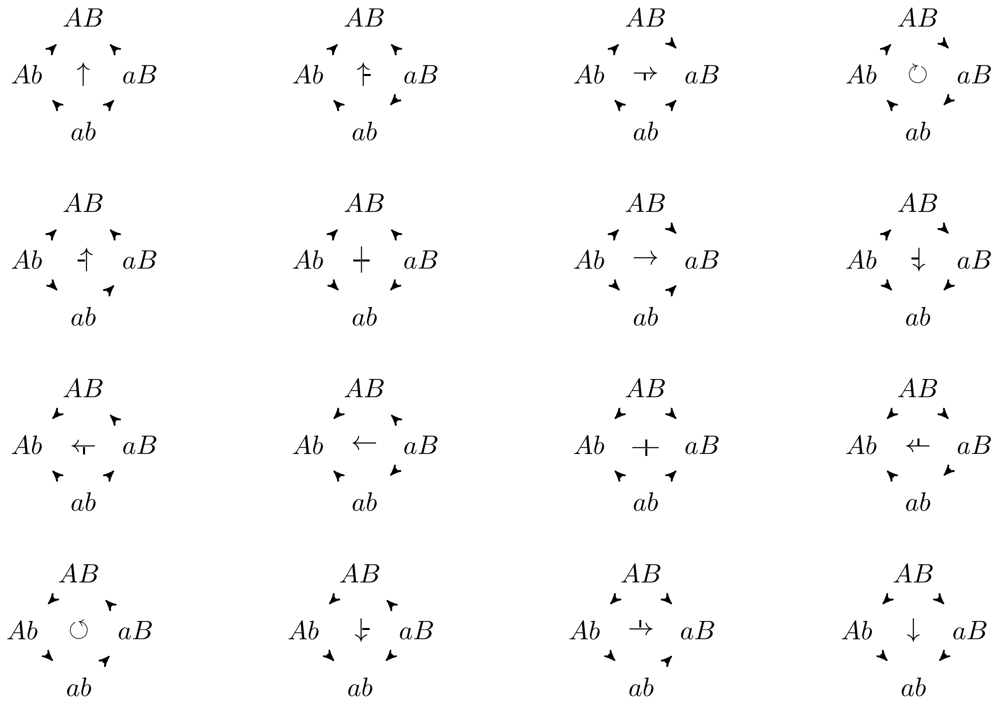


Figure 3.2: Three different kinds of epistasis possible in fitness graphs: no epistasis (↑), sign epistasis (↿,↾), and reciprocal sign epistasis (+). Arrows in the fitness graph are directed from lower fitness genotypes towards mutationally adjacent higher fitness genotypes. In the middle of each fitness graph is a symbol showing the kind (and orientation) of epistasis. Note that the bottom left (↺) and top right (↻) fitness graphs violate transitivity and cannot occur with scalar fitness values (I will introduce alternatives to scalar fitness in Chapter 6)

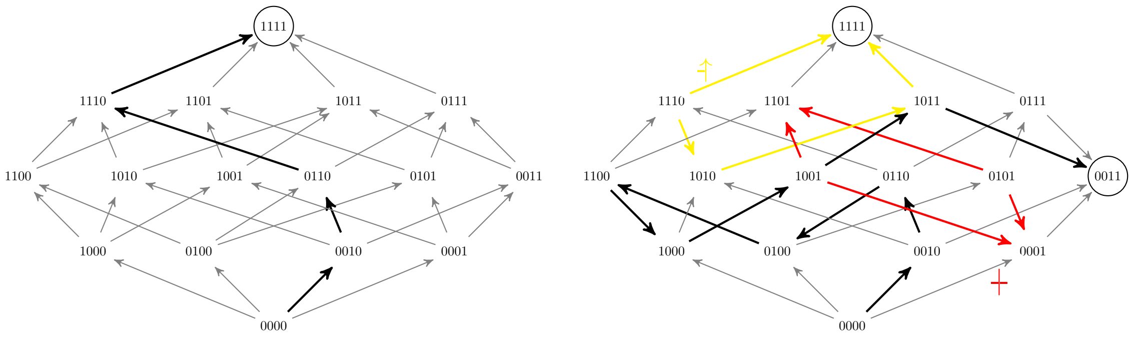
.

$x_1 \in \{a, A\}$ and $x_2 \in \{b, B\}$, I introduce the symbol $x_1 + x_2$ to represent reciprocal sign epistasis on this pair. As with sign epistasis the conditions in Definition 3.6 could be rewritten (in the obvious way) in terms of the sgn($\circ$) function or the fitness graph.

Finally, whereas Figure 3.1 visualized a prototypical example of each kind of epistasis, Figure 3.2 visualizes all the fitness graphs on two loci and categorizes the type of epistasis present by looking at the rotations of the symbols ↑, ↿, ↾, and +. Note the two cases of non-transitive fitness graphs in the corners of Figure 3.2 labeled by ↻ – these sort of graphs cannot be represented by scalar fitnesses, and I deal with them as games (or game landscapes) in Chapters 6 and 7.

3.2.1 Smooth, semismooth and rugged landscapes

The presence or absence of certain types of epistasis (and corresponding induced fitness (sub)graphs) produces different kinds of fitness landscapes. A landscape with non-interacting loci or only magnitude epistasis is called **smooth** [228, 39]. An empirical example would be the *Escherichia coli* β-lactamase fitness landscape measured by Chou et al. [30] and shown in Figure 3.3a.



(a) *E. coli* β-lactamase

(b) *P. falciparum* dihydrofolate reductase

Figure 3.3: **Two examples of empirical biallelic fitness landscapes on four loci.** Arrows are directed from lower fitness genotypes to higher and fitness optima are circled. Examples of adaptive dynamics are highlighted with thick black arrows. Figure 3.3a, based on the *E. coli* β-lactamase data of Chou et al. [30], is a smooth landscape with no sign epistasis. Thus, it contains a single optimum (1111). Figure 3.3b is based on Lozovsky et al. [135]'a *P. falciparum* dihydrofolate reductase growth rate data in the absence of pyrimethamine. It has two peaks (0011 & 1111) and both single sign (an example in yellow; ↥) and reciprocal sign epistasis (example in red; +). Based on Szendro et al. [209]'s Figure 1.

Since a smooth landscape is characterized by the absence of sign epistasis, I will analogously call a fitness landscape **semismooth** if it has no *reciprocal* sign epistasis. My naming of this new kind of fitness landscape will become more clear in Section 4.3, where I show its many structural commonalities with smooth landscapes (like a unique peak) and formal correspondence to the acyclic unique sink orientation graphs used in the analysis of the simplex algorithm for linear programming.

But in nature there exist natural fitness landscapes that are even more complicated than semismooth ones. For example, we know that some landscapes can contain reciprocal sign epistasis like the Lozovsky et al. [135] *Plasmodium falciparum* dihydrofolate reductase fitness landscape in Figure 3.3b. This is a rugged fitness landscape with two distinct fitness peaks at 0011 and 1111. In general, I will call a fitness landscape **rugged** if it has reciprocal sign epistasis (even if there are not multiple local peaks in the landscape). Although, as I discuss in Section 3.6, there is not enough data to justify postulating probability distributions over large landscapes, the standard biological intuition is that natural landscapes are at least a bit rugged and tend to have multiple peaks.

3.3 Classical NK-model and one-gene-one-function

The NK-model is a family of fitness landscapes [90, 91] that was introduced to study the landscape ruggedness that we expect to see in nature. This model allows tuning the amount of epistasis: the fitness contribution of each of the n loci depends not only on its gene, but also on the genes at up

Computational Terms	**Evolutionary Terms**
valued constraint satisfaction problem	generalized NK-model
variable	gene
variable index	locus
single variable assignment	allele
assignment to all variables	genotype
constraint	gene interaction
constraint graph	gene interaction network
arity	degree of epistasis
objective function	fitness

Table 3.1: Dictionary for translating between the languages of artificial intelligence and evolutionary biology. A preliminary version of this table appeared in Strimbu [206].

to K other loci.

Definition 3.7 ([90, 91, 89])**.** The (classic) *NK model* is a fitness landscape on $\{0,1\}^n$. The n loci are arranged in a *gene-interaction network* where each locus i is linked to K other loci $n^i_1, ..., n^i_K$ and has an associated fitness contribution function $f_i : \{0,1\}^{K+1} \to \mathbb{R}_+$ Given a vertex $x \in \{0,1\}^n$, we define the fitness $f(x) = \sum_{i=1}^n f_i(x_i; x_{n^i_1} ... x_{n^i_K})$.

By varying K we can control the amount of epistasis in the landscape. The model also provides an upper bound of $n\binom{K+1}{2}$ on the number of gene pairs that have epistatic interactions. Typically, in the biology and statistical physics literature, the fitness contributions f_i and sometimes the gene-interaction network are chosen uniformly at random from some convenient probability distribution. In contrast, the approach of theoretical computer scientists is to consider *arbitrary* rather than random choices for each f_i and the gene-interaction network. This is an important cultural difference in methodology between statistical physics and computer science that I discuss in more detail in section 3.6.

The more pressing issue with the classic NK-model, however, is that it seems to enshrine – or at least heavily favour – the one-gene one-function view of molecular biology. The easiest way to interpret the fitness components in the classic NK-model is as 'basic functions' that together add up to the total fitness of the organism. In this way, the fitness components serve as a rudimentary decomposition of the genotype to phenotype map. But in this interpretation, if each locus i is linked to a single fitness component f_i then we are linking one gene to one function. Sure, that function is mediated by K other genes, but there are still no more functions than there are genes.

3.4 Generalized NK-model as valued constraint satisfaction

We can avoid hard coding this strange one-gene-one-function assumption into the model by defining a generalized NK-model that does not force the number of fitness components to be equal to the number of genes. This generalized NK-model is not only more biologically plausible but is also equivalent to the valued constraint satisfaction model studied in computer science. For simplicity, I will present this model in both the language of computer science and biology, with a dictionary between the basic terms given in Table 3.1. I will use both languages interchangeably throughout, and present the general definition for a fitness function $f : A \to \mathbb{Z}$ with $A = D_1 \times \cdots \times D_n$ as given at the start of this chapter.

It is helpful to define a special notation for the restriction of a variable assignment (genotype) to some subset of variables, with indices (loci) in a set $S \subseteq [n]$, I will denote this by $x[S]$; so $x[S] \in \prod_{j \in S} D_j$. To reference the assignment to the variable at position i, I will usually write x_i unless it is ambiguous, in which case I will use the more general notation $x[i]$. If I want to modify x by changing a single variable, say the variable at position i, to some element $b \in D_i$, then I will write $x[i \mapsto b]$.

A **(valued) constraint** with scope $S \subseteq [n]$ is a function $C_S : \prod_{j \in S} D_j \to \mathbb{Z}$. For uniform domains, I will abbreviate $\prod_{j \in S} D_j$ as D^S (or for Boolean domains: $\{0,1\}^S$). The **arity of a constraint** C_S is the size $|S|$ of its scope. For unary (arity of 1) and binary (arity of 2) constraints I will omit the set notation and just write C_i for $C_{\{i\}}$ or C_{ij} for $C_{\{i,j\}}$. I will represent the values taken by a unary constraint C_i for each domain element by an integer vector of length $|D_i|$, and represent the values taken by a binary constraint C_{ij} for each pair of domain elements by an integer matrix, where x_i selects the row and x_j selects the column. A zero-valued constraint (of any arity) will be denoted by 0. In biological terminology, a valued constraint is a **fitness component** (often associated with a particular function) and its arity is that component's **degree of epistasis**. These valued constraints are a generalization of the fitness components f_i from the classic NK-model (Definition 3.7).

We can now define the generalized NK-model as a representation or implementation of a fitness landscape by a VCSP-instance:

Definition 3.8. An instance of the *valued constraint satisfaction problem (VCSP)* is a set of constraints $\mathcal{C} = \{C_{S_1}, \ldots, C_{S_m}\}$. I will say that a VCSP-instance $\mathcal{C}$ **implements** a fitness function f if $f(x) = \sum_{k=1}^{m} C_{S_k}(x[S_k])$.

In computer science, we are usually interested in the VCSP problem of maximizing f.

Note that instances of the classic NK-model form a subset of the VCSP-instances of the generalized NK-model. In particular, instead of an arbitrary number m of constraints, the classic NK-model would require $n = m$ and demand than for each $0 \leq i \leq m$, $i \in S_i$ and $|S_i| = K + 1$.

The arity of a VCSP-instance – or, in biological terminology, the degree of epistasis K of the generalized NK-model instance – is the maximum arity over its constraints; if this maximum arity is 2, then we will call it a *binary* VCSP-instance. Note that not all fitness functions can be expressed by binary VCSP-instances:

Example 3.9. *Needle-in-a-haystack fitness function* $\delta_y : \{0,1\}^n \Rightarrow \mathbb{R}$ *with* $y \in \{0,1\}^n$ *is given by:*

$$\delta_y(x) = \begin{cases} 1 & \text{if } x = y \\ \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

and cannot be expressed by any binary Boolean VCSP-instance if $n > 2$*.*

Given any VCSP-instance $\mathcal{C}$, we can take $A = D_1 \times \cdots \times D_n$ as the set of all possible assignments, $f_{\mathcal{C}}$ as the fitness function implemented by $\mathcal{C}$, and N as the 1-flip neighbourhood, to obtain an associated fitness landscape, $(A, f_{\mathcal{C}}, N)$, and hence an associated fitness graph, $G_{\mathcal{C}}$, by Definition 3.2. Note that the vertex set of $G_{\mathcal{C}}$ is the set of possible assignments (genotypes), A, and hence is exponential in the size of the instance, $\mathcal{C}$, in general.

The generalized NK-model can then be seen as the set of all fitness landscapes implementable by some VCSP-instance. Unlike the custom with traditional accounts of the (classic) NK-model, I will not simplify the analysis by just assuming some specific distribution over this space of these possible fitness landscapes. In Section 3.6, I will discuss in more detail why this can be a more useful (or at least different) perspective.

Each *binary* VCSP-instance also has an associated *constraint graph*, defined as follows, whose vertex set has the size of the number of loci and is thus polynomial in the size of the instance:

Definition 3.10. Given any binary VCSP-instance $\mathcal{C}$, the corresponding **constraint graph** has vertices $V(\mathcal{C}) = [n]$, edges $E(\mathcal{C}) = \{ij \mid C_{ij} \in \mathcal{C}, C_{ij} \neq 0\}$, constraint-neighbourhood function $N_{\mathcal{C}}(i) = \{j \mid ij \in E(\mathcal{C})\}$, and degree $d_{\mathcal{C}}(i) = |N_{\mathcal{C}}(i)|$.

In biological terminology, we can think of the constraint graph as the **gene-interaction network**. The strength of this name will be obvious after we see how the edges of the minimal gene-interaction networks correspond to the magnitude (Section 3.7) and sign epistasis (Section 3.8) in the fitness landscape.

Since each instance of the classic NK-model is an instance of the generalized model, we can look at the family of gene-interaction networks that the classic model generates. For $K = 0$, the loci are non-interacting and so the gene-interaction network has no edges. But the $K = 1$ case is a little bit more interesting:

Proposition 3.11. *The classic NK-model with* $K \leq 1$ *is equivalent to the binary generalized NK-*

model where the gene-interaction network has each connected component as a tree or a tree plus an extra edge.

Proof. ($\Rightarrow$): Draw a directed edge from each locus of the classic model to the locus that it interacts with (and no edge if its fitness component is unary). Now look at any connected component in this graph, say it has k vertices. Since it is connected, it must have at least $k-1$ edges (and thus be a tree). And since every edge has a parent, and each vertex is the parent of at most one edge, it must have at most k edges (and thus be a tree plus an extra edge).

($\Leftarrow$): Let us look at each connected component in the generalized model and show how to convert each into the classic model.

If a connected component with k vertices has $k-1$ edges then it is a tree. Pick any vertex as the root and direct edges towards the root. Now, each vertex will have at most one out-edge.

If a connected component with k vertices has k edges then it must contain a cycle on $l \in [3, k]$ vertices. Pick either direction around the cycle and direct all edges to follow that direction. If we then look at the remaining undirected edges in this connected component, they will form a forest of l trees, where each tree will contain exactly one vertex from the cycle. Go inside each tree and set the cycle vertex as the root and direct edges towards the root. Now, ever vertex in the tree will have one out-edge and the root will also have one out-edge from being on the cycle.

Repeat this for all the components in the graph. Now associate each constraint in the generalized model (i.e. now a directed edge in the graph) as a fitness component in the classic model corresponding to the locus that is the parent (or initial vertex) of each edge. □

In Chapter 5, I will look at other sub-classes of the generalized NK-model and prove that some sub-classes specify fitness landscapes with only short adaptive paths.

3.5 Landscapes vs dynamics as problems vs algorithms

By even discussing fitness landscapes, I am carving up nature into two parts:

1. a space on which things happen, and
2. the dynamics that happen.

Not everyone thinks this is a useful separation [57], but I think it can bring clarity and understanding. This separation is akin to the distinction that computer scientists carve between problems and algorithms. Families of fitness landscapes are like problems to be ‘solved’ by evolution and specific fitness landscapes on which populations evolve are problem-instances. In contrast, I will consider the population structure, update rules, developmental processes, mutation operator or bias, etc as together specifying the *algorithm* that is evolution. This allows me to keep the class of problems fixed while considering the performance of any algorithm (maybe within some reasonable class) on that class of problems. Or alternatively, as I do in Chapter 7, to fix an algorithm and thus define the problem as the thing that algorithm 'solves'. Philosophically, this distinction between problem and algorithm is like defining a separation between the environment and the population or system evolving within it. This is not always the best representation, but sometimes it can be a very fruitful one.

This distinction between algorithm and problem was already helpful in Chapter 2 when I defined the notion of proximal vs ultimate constraints on evolution. But it comes up in this chapter even though this chapter is largely concerned with understanding problems and how they are represented. For example, I will often refer to the adjacency graph of the genotype space as the mutation graph. This is because in practice, these two concepts are intertwined and defined circularly. We often think of mutations as occurrences that moved a genotype to a nearby point in genetic space, and we think of two points in genotype space as adjacent or nearby if a single mutation can move us between them. But technically, these spaces can be distinct. We could use a genetic notion of adjacency for defining structural features like local peaks or epistasis, without full knowledge of the possible mutations (or even more complicated moves in the case of recombination) in our system. From this, more general perspective, the genetic distance is a property of our problem, while the mutation distance is a property of our algorithm. In particular, in Chapter 4, we can think of the genotype space as purely a feature of the problem, since I will want to reason about any evolutionary dynamics (i.e. algorithm) on that space (and I will establish results like Corollary 4.27 that are robust over this feature). In particular, the evolutionary algorithm might use an arbitrary (potentially dynamic) mutation- or recombination-structure that is not the same as the genotype space. But in Chapter 5, I will focus on more specific algorithms where those algorithm's potential mutation neighbourhoods match the problem's genetic adjacency neighbourhoods.

3.6 Random vs concrete fitness landscapes

The classic NK model, and even variants like the generalized NK-model [81], are frequently studied through simulation and statistical analysis on random landscapes. The best current techniques in theoretical evolutionary biology come from the statistical mechanics of disordered systems and rely on “[t]he idea that unmanageable complexity can be replaced by randomness” [81]. This statistical approach uses randomness in two places:

1. the random mutations, birth-death events, and other physical and biological processes within the algorithm of evolution, and
2. the theoretical distributions of fitness landscapes themselves.

For a computer scientist, the first use of randomness corresponds to the analysis of randomized algorithms – certainly a good decision when thinking about evolution. The second use of randomness corresponds to average-case analysis over problem instances. In a typical biological treatment, the gene-interaction network is assumed to be something simple like a generalized cycle (where x_i is linked to $x_{i+1}, ...x_{i+K}$) or a random K-regular graph. The fitness contributions f_i are usually sampled from some convenient choice of distribution. This generates a random instance. But, unlike biologists, when the real-world distribution of problem-instances is unknown or hard to characterize, computer scientists are hesitant to pick a specific simple distribution just to analyze the algorithm. Instead, computer scientists usually specify a formal, logically-defined hypothesis class of conceivable problem-instances and then analyze their algorithm (for arbitrary distributions) over these instances. Usually, computer scientists move on to considering average case analysis over more restricted distributions or other complexity measures only after the worst-case analysis has been well characterized. As far as I know, in the case of fitness landscapes, such worst-case (or arbitrary distribution) analysis had not been fully explored before the jump to average case over easy and convenient distributions of problem instances was made. So in Chapters 4 and 5, I will give examples of such worst-case analysis.

Given a historical disconnect between theory and data [166, 125], the choice for distributions was usually made out of analytic convenience or (occasionally) out of the belief that a uniform distribution is akin to no assumption. Since there is no strong empirical or theoretically sound justification for the choice of particular distributions of large fitness landscapes, I avoid relying on a simple generating distribution and instead reason from only the logical description of the model. This can be thought of as worst-case analysis, or as analysis for arbitrary distributions of landscapes. By following this approach, we know that our results are features of the logic that characterizes a particular family of fitness landscapes and not artifacts of a particular simple sampling distribution. This is a standard method in theoretical computer science, but it is not as common in statistical physics or theoretical biology.

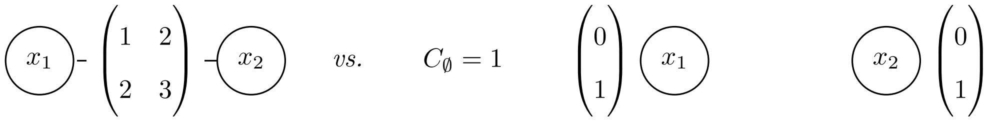
Although there is evidence for simple distributions on small fitness landscapes (on upto 8 genes; see Franke et al. [50] and Szendro et al. [209]) and a growing literature of measured fitness landscapes [178, 30, 135, 166, 183, 129, 125, 14, 46], there is little to no data on the distribution of large (i.e. on many loci) fitness landscapes in nature. And given the exponential size of fitness landscapes, it is unlikely that such data could be collected.

Thus, it is important to build theory that does not assume simple distributions over landscapes. In the future, if we discover a theoretical or empirical way of finding these distributions (that is more well-founded than just choosing a distribution that simplifies the analysis of our model) then we can consider how to extend this work to those distributions over landscapes. But that is outside of the scope of this thesis.

3.7 Magnitude-equivalence and simple VCSP instances

It is clear from Definition 3.8 that different VCSP-instances can implement the same fitness function.

Example 3.12. *Consider, the following two small VCSP-instances:*


x_1
$\begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$
x_2
vs.
$C_\emptyset = 1$
$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$
x_1
x_2
$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

where I draw the $C_\emptyset$ constraint separately, unary constraints next to the variable that selects their row, and binary constraints along the edge linking the variable that selects the matrix's row to the variable that selects the matrix's column.

Although these two instances have different constraint graphs, the fitness function they implement is $[f(00), f(01), f(10), f(11)] = [1, 2, 2, 3]$ *in both cases.*

We can capture the equivalence in Example 3.12 with the following definition:

Definition 3.13. If two VCSP-instances $\mathcal{C}_1$ and $\mathcal{C}_2$ implement the same fitness function f, then I will say they are **magnitude-equivalent**.

I will show in this section that for binary Boolean VCSP-instances each equivalence class of magnitude-equivalent VCSP-instances (or – in biological terminology – magnitude-equivalent gene-interaction network) has a *normal form*: a unique, minimal, and easy to compute representative member with special properties.

Definition 3.14. A binary Boolean VCSP-instance $\mathcal{C}$ is **simple** if:

- every unary constraint has the form $C_i = \begin{pmatrix} 0 \\ c_i \end{pmatrix}$ and

- every binary constraint has the form $C_{ij} = \begin{pmatrix} 0 & 0 \\ 0 & c_{ij} \end{pmatrix}$.

In drawings of constraint graphs of simple VCSP-instances I will often denote the unary constraint $\begin{pmatrix} 0 \\ c_i \end{pmatrix}$ by just c_i next to the variable, and the binary constraint $\begin{pmatrix} 0 & 0 \\ 0 & c_{ij} \end{pmatrix}$ by just c_{ij} on the edge between variables. Note that these c_i and c_{ij} can be either negative or positive, and their sign will be very important when I introduced sign-equivalence in Section 3.8.

I now give a direct proof of the following simplification result which is analogous to similar results using constraint propagation in the standard VCSP [34] and well-known for pseudo-Boolean functions [37]:

Theorem 3.15. *Any binary Boolean VCSP-instance $\mathcal{C}'$ can be transformed into a unique simple VCSP-instance $\mathcal{C}$ that is magnitude-equivalent to $\mathcal{C}'$. Moreover, $\mathcal{C}$ can be constructed from $\mathcal{C}'$ in linear time.*

Proof. First two key observations: (1) Any unary Boolean constraint $C'_i : \{0,1\} \to \mathbb{Z}$ can be rewritten as a linear function:

$$c'_i(x) = (1 - x_i)C'_i(0) + x_i C'_i(1) \tag{3.2}$$

and (2) any binary Boolean constraint $C'_{ij} : \{0,1\} \times \{0,1\} \to \mathbb{Z}$ can be rewritten as a multilinear polynomial of degree 2:

$$c'_{ij}(x) = (1 - x_i)(1 - x_j)C'_{ij}(0,0) + (1 - x_i)x_j C'_{ij}(0,1) + x_i(1 - x_j)C'_{ij}(1,0) + x_i x_j C'_{ij}(1,1). \tag{3.3}$$

From this, we can simplify $\mathcal{C}'$ just by simplifying polynomials:

$$\begin{align} f(x) &= C'_\emptyset + \sum_{i=1}^{n} C'_i(x_i) + \sum_{ij \in E(\mathcal{C}')} C'_{ij}(x_i, x_j) = C'_\emptyset + \sum_{i=1}^{n} c'_i(x) + \sum_{ij \in E(\mathcal{C}')} c'_{ij}(x) \tag{3.4} \\ &= C_\emptyset + \sum_{i=1}^{n} x_i c_i + \sum_{1 \leq i < j \leq n} x_i x_j c_{ij} \tag{3.5} \end{align}$$

where we note that the last part of Equation 3.4 is a sum of a constant, some linear functions, and some multilinear polynomials of degree 2, and is thus itself a multilinear polynomial of degree 2 (or less). Equation 3.5 follows from Equation 3.4 by multiplying out into monomials and then grouping the coefficients of each similar monomial. In particular, this gives us the following coefficients:

$$c_i = C'_i[1] - C'_i[0] \quad + \sum_{j \mid ij \in E(\mathcal{C}')} C'_{ij}[1,0] - C'_{ij}[0,0] \tag{3.6}$$

$$c_{ij} = C'_{ij}[0,0] - C'_{ij}[0,1] - C'_{ij}[1,0] + C'_{ij}[1,1] \tag{3.7}$$

The above calculation can be done in linear time. As the last step in the simplification, note that Equation 3.5 corresponds to a VCSP-instance $\mathcal{C}$ comprising a nullary constraint $C_\emptyset$, unary constraints $C_i = \begin{pmatrix} 0 \\ c_i \end{pmatrix}$, and binary constraints $C_{ij} = \begin{pmatrix} 0 & 0 \\ 0 & c_{ij} \end{pmatrix}$. □

The next theorem shows that a simple VCSP-instance has the minimal constraint graph of any binary instance that implements the same fitness function and the approach I take in its proof helps justify the term "magnitude":

Theorem 3.16. *Let $\mathcal{C}$ be a simple binary Boolean VCSP-instance. If the binary Boolean VCSP-instance $\mathcal{C}'$ is magnitude-equivalent to $\mathcal{C}$, then $E(\mathcal{C}) \subseteq E(\mathcal{C}')$.*

Proof. Let $e_i \in \{0,1\}^n$ be a variable assignment that sets the ith variable to one, and all other variables to zero. Similarly, let $e_{ij} \in \{0,1\}^n$ be a variable assignment that sets the ith and jth variables to one, and all other variables to zero. Let f be the fitness function implemented by $\mathcal{C}$. Since $\mathcal{C}$ is simple, we have:

$$f(e_{ij}) - f(e_i) - f(e_j) + f(0^n) = c_{ij} \tag{3.8}$$

where we take $c_{ij} = 0$ if $ij \notin E(\mathcal{C})$. Similarly, if $\mathcal{C}'$ also implements f, we have:

$$f(e_{ij}) - f(e_i) - f(e_j) + f(0^n) = C'_{ij}(1,1) - C'_{ij}(1,0) - C'_{ij}(0,1) + C'_{ij}(0,0) \tag{3.9}$$

If $ij \in E(\mathcal{C})$ then $c_{ij} \neq 0$, so $C'_{ij}(1,1) - C'_{ij}(1,0) - C'_{ij}(0,1) + C'_{ij}(0,0) \neq 0$ and hence $ij \in E(\mathcal{C}')$. □

Note that Equation 3.8 is a measure of magnitude epistasis. In particular, $c_{ij} = 0$ if and only if $f(e_{ij}) - f(e_i) = f(e_i) - f(0^n)$ and $f(e_{ij}) - f(e_j) = f(e_j) - f(0^n)$. In other words, a constraint between loci i and j is absent in the simplified (i.e., minimal magnitude-equivalent) gene-interaction network if and only if i and j are non-interacting loci in all genetic background. If i and j have magnitude epistasis or more (sign or reciprocal sign) then the constraint is present in all gene-interaction networks. Hence, simple networks track magnitude epistasis and higher. That is why I use the word 'magnitude' when I call these networks $\mathcal{C}$ and $\mathcal{C}'$ as "magnitude"-equivalent.

3.7.1 Measuring fitness landscapes by learning simple gene-interaction networks

Perhaps more importantly, Equation 3.8 specifies a simple algorithm for inferring gene-interaction networks from existing data of local fitness landscapes. Whole fitness landscapes are exponentially large in the number of genes and thus, even for a moderate number of genes, directly measuring every genotype's fitness becomes unimaginable – even for a theorist. But instead we can measure local fitness landscapes (or for a computer scientist, query a specific set of fitness values as set out below). A **local fitness landscape measurement** of depth d around a *wildtype* $w \in A$ (i.e., around an 'original' or 'default' type, this is an experiment-dependent definition but the wildtype is usually the typical form that occurs in nature) is a measurement of the fitness of w and all other genotypes $x \in A$ that are d or fewer mutations away from w. For a Boolean fitness landscape, a local fitness landscape measurement of depth d will require on the order of $\sum_{i=1}^{d} \binom{n}{i}$ fitness assays. Thus, this measurement is possible in theory for constant depth d. And in practice, it has been measured to depth 2 for a 333-nucleotide small nuclear RNA gene [183], and for a 72-nucleotide transfer RNA gene [129] – both in yeast.

If we take the wildtype as 0^n in Equation 3.8 and assume that fitness landscape is expressible by a binary gene-interaction network then these local fitness landscape measurements provide enough data to get each c_{ij} by iterating over every pair $1 \leq i < j \leq n$. Note that because we assume the fitness landscape is expressible by some binary gene-interaction network, this requires only a quadratic number of queries to the fitness function and not an exponential number. We can further get $C_\emptyset = f(0^n)$ and $c_i = f(e_i) - f(0^n)$ (for a further n queries to the fitness function), thus getting all of the constraints in the simple gene-interaction network. This is actually equivalent to just over-fitting a polynomial; i.e. to specifying a multilinear polynomial with $1 + n + \binom{n}{2}$ parameters from $1 + n + \binom{n}{2}$ datapoints. Thus, when we query a gene-interaction network learned in this way on a point x outside the training set then it is better to think of the result as a polynomial interpolation rather than a prediction. And in this way, it is better to interpret the learned gene-interaction network itself as a way to summarize – or represent in a different way – the local fitness landscape measurement. This is akin to how I propose that we think of the game assay in Chapter 7 for summarizing the ecology of a different kind of evolutionary dynamic.

Since my above proposal for learning gene-interaction networks from local fitness landscape measurements is equivalent to linear regression, it will also benefit from our rich understanding of how measurement noise can be handled and confidence intervals propagated in linear regression. Further work can explore this learning algorithm more deeply and apply it to experimental fitness landscapes, but such work is beyond the scope of this thesis.

3.8 Sign-equivalence and trim VCSP instances

In the previous section we looked at the equivalence class of all VCSP-instances which implement precisely the same fitness function. However, when investigating the performance of arbitrary adaptive dynamics or local search algorithms, the exact values of the fitness function are not always relevant. This is because different particular evolutionary dynamics or search heuristics will handle particular fitness differences differently. But the structure of adaptive paths matters to all adaptive dynamics, since they are (by definition) restricted to choosing between just these adaptive paths. As such, it can be sufficient to consider only the fitness graph.

It is also important to focus on fitness graphs for experimental reasons. This is essential for experimental systems where fitness cannot be directly measured numerically and competition assays can only establish relative fitness ordering of the genotypes (i.e., direction of fitness graph edges). However, even in experimental models where fitness can be measured directly, the specific numbers can be a little arbitrary. Often, any strictly monotonic transformation of the measured values would also be a reasonable alternative fitness measure. But such strictly monotonic transformations can introduce magnitude epistasis, and so change the simple gene-interaction network of Section 3.7. This ambiguity has been previously considered for smooth fitness landscapes under the concept of global epistasis [169]. But in this section, I will provide a more general account by noticing that even though magnitude epistasis might change, the fitness graph itself does not change under strictly monotonic transformations of the fitness function. As such, it is important to focus on a very convenient representation of the fitness graph.

Unfortunately, different fitness functions implemented by simple VCSPs that are not magnitude-equivalent and have different gene-interaction networks can still implement the same fitness graph.

Example 3.17. *Let f be a fitness function implemented by a simple VCSP-instance $\mathcal{C}$, where the fitness values of all adjacent genotypes are distinct, but there is at least one pair i, j of positions with no constraint C_{ij}.*

Now consider the new fitness function $f'(x) = 2f(x) + C_{ij}(x_i, x_j)$ where $C_{ij} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$. Since all adjacent fitness values given by $2f(x)$ differ by at least 2, every edge is still present in the new fitness graph, and no directions are changed by the new constraint. Thus, the fitness graph corresponding to f' is unchanged, but we cannot eliminate this new C_{ij} constraint (and associated magnitude epistasis) without changing the precise values of the fitness function.

To capture the similarity between f and f' in Example 3.17, I introduce a more abstract equivalence relation:

Definition 3.18. If two VCSP-instances $\mathcal{C}_1$ and $\mathcal{C}_2$ give rise to the same fitness graph, then I will say they are **sign-equivalent**.

Note that sign-equivalence is an even larger (or more coarse-grained) equivalence class than what would be defined by strictly monotonic transformations of the fitness function (as shown by Example 3.17).

As with magnitude-equivalence, I will show that for binary Boolean VCSP-instances it is possible to define a normal form or minimal representative member of each equivalence class of sign-equivalent VCSP-instances with a unique minimal constraint graph (or, in biological terminology, gene-interaction network). Unfortunately, we will see that, unlike the situation for the easy-to-compute minimal magnitude-equivalent constraint graph, this minimum sign-equivalent constraint graph is NP-hard to compute (Theorem 3.26).

To get started, it is useful to more formally elaborate the notion of sign-epistasis from Definition 3.5 to high dimensional fitness graphs:

Definition 3.19. In a Boolean fitness graph G with vertex set $\{0,1\}^n$, I will say that i **sign-depends** on j if there exists an assignment $x \in \{0,1\}^n$ such that:

$$(x, x[i \mapsto \overline{x}_i]) \in E(G) \quad \text{but} \quad (x[j \mapsto \overline{x}_j], x[i \mapsto \overline{x}_i, j \mapsto \overline{x}_j]) \notin E(G) \tag{3.10}$$

Note that i **sign-depends** on j if and only if, for any fitness function f that corresponds to the fitness graph G, there exists $x \in \{0,1\}^n$ such that:

$$\mathrm{sgn}(f(x[i \mapsto \overline{x}_i]) - f(x)) \neq \mathrm{sgn}(f(x[i \mapsto \overline{x}_i, j \mapsto \overline{x}_j]) - f(x[j \mapsto \overline{x}_j])). \tag{3.11}$$

Equivalently, we can say that i sign-depends on j if and only if there is at least one genetic background x such that $i \mathrel{-\!\!\!\uparrow} j$ or $i \mathrel{-\!\!\!\!+\!\!\!\!-} j$ (i.e., i and j have sign epistasis or reciprocal sign epistasis).

I will say that i and j **sign-interact** if i sign-depends on j or j sign-depends on i (or both). If i and j do not sign-interact, then I will say that they are **sign-independent**.

Definition 3.20. A simple binary Boolean VCSP-instance $\mathcal{C}$ with associated fitness graph $G_{\mathcal{C}}$ is called **trim** if for all $ij \in E(\mathcal{C})$, i and j sign-interact in $G_{\mathcal{C}}$.

A sign-equivalent analog of Theorem 3.15 guarantees a normal form:

Theorem 3.21. *Any simple binary Boolean VCSP-instance $\mathcal{C}'$ can be transformed into a trim VCSP-instance $\mathcal{C}$ that is sign-equivalent to $\mathcal{C}'$.*

To prove Theorem 3.21, I now establish two propositions: Proposition 3.22 connects the magnitude of constraints with their effect on fitness graphs, and Proposition 3.23 connects the magnitude of constraints to sign-interaction.

Proposition 3.22. *Given a simple binary Boolean VCSP-instance $\mathcal{C}$ implementing a fitness function f, if removing the constraint C_{ij} changes the corresponding fitness graph, then for at least one $k \in \{i,j\}$ there exists some $x \in \{0,1\}^n$ with $x_i = x_j = 1$ such that:*

$$c_{ij} \geq f(x) - f(x[k \mapsto 0]) > 0 \quad \textit{or} \quad c_{ij} \leq f(x) - f(x[k \mapsto 0]) < 0 \tag{3.12}$$

Proof. Without loss of generality (by swapping i and j in the variable numbering if necessary), we can suppose that $k = i$. Consider two cases:

Case 1 ($c_{ij} > 0$): If removing C_{ij} changes the fitness graph, then there exists some $x \in \{0,1\}^n$ with $x_i = x_j = 1$ such that:

$$f(x) > f(x[i \mapsto 0]) \quad \text{but} \quad f(x) - c_{ij} \leq f(x[i \mapsto 0]). \tag{3.13}$$

We can re-arrange Equation 3.13 to get $c_{ij} \geq f(x) - f(x[i \mapsto 0]) > 0$ where the strict inequality follows from the left clause of Equation 3.13.

Case 2 ($c_{ij} < 0$): This is the same as case 1, except that the direction of the inequalities in Equation 3.13 are reversed. □

Proposition 3.23. *Given a simple binary Boolean VCSP-instance $\mathcal{C}$ implementing a fitness function f, if there exists a constraint C_{ij} in $\mathcal{C}$, some assignment $x \in \{0,1\}^n$ with $x_i = x_j = 1$, and some $k \in \{i,j\}$ such that:*

$$c_{ij} \geq f(x) - f(x[k \mapsto 0]) > 0 \quad \textit{or} \quad c_{ij} \leq f(x) - f(x[k \mapsto 0]) < 0 \tag{3.14}$$

then i sign-depends on j in the associated fitness graph $G_{\mathcal{C}}$.

Proof. As in the proof of Proposition 3.22, we can suppose that $k = i$ (by swapping i and j in the variable numbering if necessary). Also, as in the proof of Proposition 3.22, the case for $c_{ij} < 0$ is symmetric (by flipping the direction of inequalities) to $c_{ij} > 0$. Thus, we will just consider the case where $k = i$ and $c_{ij} > 0$:

Given that Equation 3.14 tells us that $f(x) > f(x[i \mapsto 0])$ (i.e., that $x[i \mapsto 0]x \in E(G_\mathcal{C})$), to establish that i sign-depends on j per Definition 3.19, we need to show that $f(x[j \mapsto 0]) \leq f(x[i \mapsto 0, j \mapsto 0])$ (i.e., that $x[i \mapsto 0, j \mapsto 0]x[j \mapsto 0] \notin E(G_\mathcal{C})$). So, let us look at the difference of the latter:

$$f(x[j \mapsto 0]) - f(x[i \mapsto 0, j \mapsto 0]) = f(x) - f(x[i \mapsto 0]) - c_{ij} \leq 0 \tag{3.15}$$

where the equality follows from Definition 3.8 ($\mathcal{C}$ implements f) and Definition 3.14 ($\mathcal{C}$ is simple), and the inequality follows from the first part of Equation 3.14. □

Now, I can assemble Propositions 3.22 and 3.25 into a proof of Theorem 3.21:

Proof of ***Theorem 3.21***. Note that Equations 3.12 and 3.14 specify the same conditions, hence the negation of this condition:

$$\begin{gathered}\text{for all } x \in \{0,1\}^n \text{ with } x_i, x_j = 1 \text{ and all } k \in \{i,j\}, \text{ we have that:} \\ f(x) - f(x[k \mapsto 0]) > c'_{ij} \geq 0 \quad \text{or} \quad f(x) - f(x[k \mapsto 0]) < c'_{ij} \leq 0\end{gathered} \tag{3.16}$$

can be used to glue together the contrapositives of Proposition 3.23 (if i and j are sign-independent then Equation 3.12 does not hold) and Proposition 3.22 (if Equation 3.14 does not hold then C'_{ij} can be removed from $\mathcal{C}'$ without changing the corresponding fitness graph). So we can convert $\mathcal{C}'$ to a trim VCSP-instance that is sign-equivalent to $\mathcal{C}'$ by simply removing all $C'_{ij} \in \mathcal{C}'$ where i and j are sign-independent in the associated fitness graph $G_{\mathcal{C}'}$. □

The next result is the sign-equivalence analog of Theorem 3.16. It shows that a trim VCSP-instance has the minimal constraint graph of any binary instance with the same associated fitness graph:

Theorem 3.24. *Let $\mathcal{C}$ be a trim binary Boolean VCSP-instance. If the binary Boolean VCSP-instance $\mathcal{C}'$ is sign-equivalent to $\mathcal{C}$, then $E(\mathcal{C}) \subseteq E(\mathcal{C}')$.*

To prove Theorem 3.24, we just need to show that if i and j sign-interact in a fitness graph G, then any VCSP-instance that has the same associated fitness graph G must have an edge between i and j in its constraint graph. Or, in other words, constraints between sign-interacting positions cannot be removed while preserving sign-equivalence. That is, we just need the following proposition:

Proposition 3.25. *Let $\mathcal{C}$ be a binary Boolean VCSP-instance with associated fitness graph $G_\mathcal{C}$. If i, j sign-interact in $G_\mathcal{C}$, then the constraint C_{ij} in $\mathcal{C}$ is non-zero.*

In other words, if $i \neq j$ sign-interact somewhere in the fitness graph, there must a gene-interaction between i and j in any gene-interaction network that describes that fitness graph.

Proof. Without loss of generality, assume that we have an edge in $G_\mathcal{C}$ from $x[i \mapsto \overline{x_i}]$ to x. Thus, the fitness function f implemented by $\mathcal{C}$ must satisfy the following two inequalities:

$$f(x) > f(x[i \mapsto \overline{x_i}]) \quad \text{and} \quad f(x[j \mapsto \overline{x_j}]) \leq f(x[i \mapsto \overline{x_i}, j \mapsto \overline{x_j}]) \tag{3.17}$$

Define $g_i(x_i) = C_i(x_i) + \sum_{k \neq j} C_{ik}(x_i, x_k)$ and similarly for g_j. Also let $K_{ij}(x)$ be the part of f independent of x_i, x_j: i.e., $f(x) = K_{ij}(x) + g_i(x_i) + g_j(x_j) + C_{ij}(x_i, x_j)$. Rewriting (and simplifying) the two parts of Equation 3.17, we get:

$$g_i(x_i) + C_{ij}(x_i, x_j) > g_i(\overline{x_i}) + C_{ij}(\overline{x_i}, x_j) \tag{3.18}$$

$$g_i(x_i) + C_{ij}(x_i, \overline{x_j}) \leq g_i(\overline{x_i}) + C_{ij}(\overline{x_i}, \overline{x_j}) \tag{3.19}$$

These equations can be rotated to sandwich the g_i terms:

$$C_{ij}(x_i, x_j) - C_{ij}(\overline{x_i}, x_j) > g_i(\overline{x_i}) - g_i(x_i) \geq C_{ij}(x_i, \overline{x_j}) - C_{ij}(\overline{x_i}, \overline{x_j}) \tag{3.20}$$

which simplifies to $C_{ij}(x_i, x_j) - C_{ij}(\overline{x_i}, x_j) > C_{ij}(x_i, \overline{x_j}) - C_{ij}(\overline{x_i}, \overline{x_j})$ and – due to the strict inequality – establishes that C_{ij} is non-zero. □

3.8.1 Hardness of sign-minimization

However, unlike with magnitude-equivalence, it is NP-hard to determine a minimal sign-equivalent VCSP-instance, as the next result shows:

Theorem 3.26. *The problem of deciding whether i and j sign-interact in a given simple binary Boolean VCSP-instance is NP-complete.*

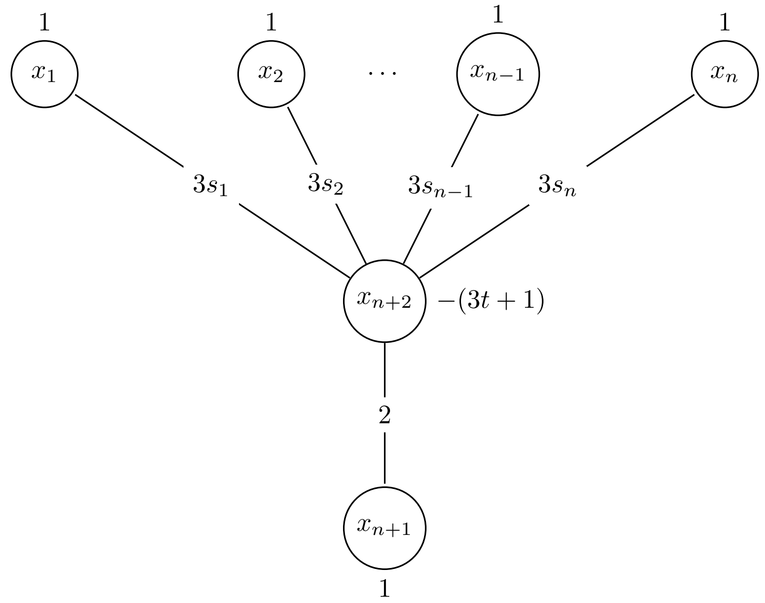
Notice that the above means that finding the minimal sign-equivalent gene-interaction network is hard because by Proposition 3.25 a constraint C_{ij} cannot be trimmed only if i and j sign-interact. This means that, for example, even if we could use the technique in Section 3.7.1 to infer the minimal magnitude-equivalent gene-interaction network specifying a fitness landscape, we would

would not be able to then produce the minimal sign-equivalent gene-interaction network. In other words, we cannot in general figure out efficiently (i.e., in time polynomial in the number of loci) if two loci have sign epistasis (or reciprical sign epistasis) somewhere in a fitness landscape or if the two loci just have magnitude epistasis.

Proof. To see that this problem is in NP, note that we can provide a variable assignment x as a certificate and check that under that variable assignment either i sign-depends on j or j sign-depends on i (or both).

I will establish NP-hardness by reduction from the SubsetSum problem, which is known to be NP-complete [52]: A set of integers $\{s_1, \dots, s_n\}$ and a target t is a yes-instance of the SubsetSum problem if there exists some subset $S \subseteq [n]$ such that $\sum_{i \in S} s_i = t$.

Now consider a simple binary Boolean VCSP-instance $\mathcal{C}$ on $n + 2$ variables, that implements fitness function f and has associated fitness graph $G_{\mathcal{C}}$, whose constraint graph has the shape of a star, with central variable position $n + 2$:



In words, the constraints of $\mathcal{C}$ are given by:

- unary constraints $C_i = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ for all $i \leq n + 1$ and
- unary constraint $C_{n+2} = \begin{pmatrix} 0 \\ -(3t+1) \end{pmatrix}$ on the central variable $n + 2$;

- binary constraints $C_{i,n+2} = \begin{pmatrix} 0 & 0 \\ 0 & 3s_i \end{pmatrix}$ between the central variable $n+2$ and variable i, for $1 \leq i \leq n$; and
- binary constraint $C_{n+1,n+2} = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}$ between $n+1$ and $n+2$.

Claim: $\langle \{s_1, \ldots, s_n\}, t \rangle$ is yes-instance of SubsetSum if and only if $n+1$ and $n+2$ sign-interact.

We clearly have that for all $x \in \{0,1\}^{n+2}$, $f(x[n+1 \mapsto 1]) > f(x[n+1 \mapsto 0])$, so $n+1$ does not sign-depend on $n+2$. Thus my claim becomes equivalent to verifying the conditions under which $n+2$ sign-depends on $n+1$. Let's look at the two directions of the if and only if in the claim:

Case 1 ($\Rightarrow$): If $\langle \{s_1, \ldots, s_n\}, t \rangle \in$ SubsetSum, then there is a subset $S \subseteq [n]$ such that $\sum_{i \in S} s_i = t$. Let $e_S \in \{0,1\}^n$ be the variable assignment such that for any $i \in S$, $e_S[i] = 1$ and for any $j \notin S$, $e_S[j] = 0$. We have that:

$$f(e_S01) = |S| - 1 \qquad f(e_S11) = |S| + 2$$

$$f(e_S00) = |S| \qquad f(e_S10) = |S| + 1$$

By Equation 3.11, these imply that $n+2$ sign-depends on $n+1$.

Case 2 ($\Leftarrow$): If $\langle \{s_1, \ldots, s_n\}, t \rangle \notin$ SubsetSum, then for any $S \subseteq [n]$ we either have $\sum_{i \in S} s_i \leq t - 1$ or $\sum_{i \in S} s_i \geq t + 1$. Thus, given an arbitrary assignment $e_S \in \{0,1\}$ we have two subcases:

If $\displaystyle\sum_{i \in S} s_i \leq t - 1$ then:

$$f(e_S01) - f(e_S00) \leq -4$$

$$f(e_S11) - f(e_S10) \leq -2$$

Or, if $\displaystyle\sum_{i \in S} s_i \geq t - 1$ then:

$$f(e_S01) - f(e_S00) \geq 2$$

$$f(e_S11) - f(e_S10) \geq 4$$

In either subcase, $\text{sgn}(f(e_S01) - f(e_S00)) = \text{sgn}(f(e_S11) - f(e_S10))$, so by Equation 3.11, $n+2$ does not sign-depend on $n+1$. □

The above has focused on sign-equivalence for fitness landscapes that are implementable by binary constraints. But there are fitness landscapes like Example 3.9 and even fitness graphs that are not implementable by binary VCSPs. In general, to handle such cases, I would need to generalize the concept of gene-interaction networks from graphs to hypergraphs where an edge can be incident on more than two vertices. Such a generalization is relatively straightforward but beyond the scope of this thesis. However, even in these higher arity cases, the structure of the trim gene-interaction network can be useful. In particular, if we abandon the idea of assigning a specific constraint to each edge then the minimal sign-equivalent gene-interaction graph can still be unambiguously defined (in a way that is useful for Chapter 5) by saying that an edge exists between loci i and j if there is some genetic background in which i and j have sign epistasis or reciprocal sign epistasis. In this way, the minimal sign-equivalent gene-interaction graph can still provide an unambiguous summary of which genes are epistatically linked in a given fitness graph, even if that fitness graph cannot be implemented by a binary VCSP-instance. Of course, when the fitness graph cannot be implemented by a binary VCSP then specific constraint matrices cannot be assigned to the edges of the gene-interaction graph to make it into a network.

3.9 Summary

Fitness landscapes are a central metaphor in evolutionary biology. In Section 3.1, I formalized this metaphor by defining fitness landscapes as mathematical objects that we can analyze. I divided these landscapes into three categories (smooth, semismooth, rugged) in Section 3.2 based the maximum kind of epistasis (none, sign, reciprocal sign) that appears within the landscapes. These three broad categories will serve as the structure for the coarse hardness results in Chapter 4. To establish the more refined easy vs hard classification of Chapter 5, I will need to look at the structure of the gene-interaction networks that I introduced in Section 3.4 as a representation of fitness landscapes. Thus, this chapters direct impact on this thesis is to set up the tools we will need to prove the various biologically significant results of Chapters 4 and 5.

But the definitions I set out here are not limited to just what they can do for Chapters 4 and 5. They can be used beyond this thesis. Thankfully, the gene-interaction networks that I introduced here have a number of nice features, including unique minimal representations. The minimal magnitude-equivalent representation can be found efficiently (unlike the minimal sign-equivalent representation) and even used as a way to interpolate whole fitness landscapes from local measurements (Section 3.7.1). For now, this approach to measuring gene-interaction networks is in its early stages but I hope to develop it in future work as a practical technique in a similar way to how I develop the game assay in Chapter 7. This way, we can one day transform fitness landscapes from being just a theoretical abstraction to also providing an empirical abstraction.

Chapter 4

Formalizing the theory of hard fitness landscapes

One of the biggest advantages of linking biology to theoretical computer science is the latter's insistence on formal definitions, theorems, and mathematical proofs. Although this formalization process can be slow going at times, I think it is well worth the effort. By focusing on formal mathematics, we can add to the study of evolution the new tool-set of algorithmic biology that extends the techniques of computational biology beyond the more practical data crunching, simulation and computational metaphors. Here, I will formalize the theory of hard fitness landscapes.

In Chapter 2, I focused on the biological importance, interpretation, and implication of the theory of hard landscapes. In Chapter 3, I gave definitions for the representation of these fitness landscapes. In this chapter, I focus on the evolutionary dynamics and provide the formal proof of the hardness results. In Section 4.1, I discuss the difference between local and global optima. In the rest of the chapter, I prove the theorems on which the conclusions of Chapter 2 are based. The rest of the chapter is then structured around the different epistasis-based families of landscapes defined in Section 3.2:

Smooth fitness landscapes: although highly restricted, these landscapes are the source of a lot of intuition and early models of fitness landscapes. So, in Section 4.2, I briefly remind us of some important properties of smooth landscapes.

Semismooth fitness landscapes: these landscapes share many properties in common with smooth fitness landscapes and I go over these similarities and differences in Section 4.3. I prove a characterization (Theorem 4.9) that is structured in a similar way to smooth landscapes. However, computationally, semismooth landscapes, unlike smooth ones, can be hard. In Section 4.3.1, I use the equivalence of semismooth fitness landscapes and acyclic unique-sink orientations of hyper-cubes to adapt hardness results from the analysis of simplex algorithms. This provides hard landscapes for random fitter-mutant SSWM dynamics. In the subsequent section, I show how to recursively construct hard fitness landscapes for fittest-mutant SSWM dynamics from specific start position (Section 4.4) and random start position (Section 4.4.1). I call this the winding semismooth fitness landscape. Finally, in Section 4.5, I show that for any implementation of the winding semismooth fitness landscape as a generalized NK-model, the resulting gene-interaction network will be "complicated" (have unbounded treewidth; Corollary 4.23).

Rugged fitness landscapes: these fitness landscapes can – unlike the previous two – have many peaks. And I study them via the classic and generalized NK models (for a reminder of definitions, see Sections 3.3 and 3.4). To analyze these models of landscapes, in Section 4.6, I review the complexity class PLS, show that both the classic NK-model for $K \geq 2$ and generalized NK-model for $K \geq 1$ are PLS-complete (Theorem 4.25) and discuss the generality of the results. In Section 4.7, I discuss the hardness of s-approximate peaks (Definition 4.28) and nearly-neutral networks. Finally in Section 4.8, I provide an intuition for why the assumption of simple distributions of fitness landscapes in prior work might have made the existence of hard families more difficult to spot earlier. But I save a full discussion of easy instances of fitness landscapes for Chapter 5.

Throughout the chapter, I argue that local fitness optima may not be reachable in a reasonable amount of time – even when allowing progressively more general and abstract evolutionary dynamics. For this generality, we pay with increasing complication in the corresponding fitness landscapes. This progression of results is summarized in Table 4.1 (which also serves as a guide for navigating the chapter). If we restrict our evolutionary dynamics to random fitter-mutant SSWM or fittest-mutant SSWM, then just sign epistasis is sufficient to ensure the existence of hard landscapes. If we allow any *adaptive* evolutionary dynamics, then reciprocal sign epistasis in the classic NK model with $K \geq 2$ or generalized NK model with $K \geq 1$ is sufficient for hard landscapes. If we want to show that arbitrary evolutionary dynamics cannot find local fitness optima, then we need $K \geq 2$ and the standard conjecture from computational complexity that FP $\neq$ PLS.

Landscape	**Max epistasis**	**Hardness of reaching local optima**	**Proved in...**
smooth	magnitude (↑)	Easy for all strong-selection weak-mutation (SSWM) dynamics	Section 4.2
semismooth	sign (↑̵,↑̵)	Hard for SSWM with random fitter-mutant or fittest-mutant dynamics	Theorems 4.10, 4.16, & 4.20
rugged	reciprocal sign (┼)	Hard for all SSWM dynamics: initial genotypes with all adaptive paths of exponential length Hard for all evolutionary dynamics (if FP $\neq$ PLS) Easy for finding approximate local peaks with moderate optimality gap: selection coefficients can drop-off as power law Hard for approximate local peaks with small optimality gap: selection coefficient cannot drop-off exponentially	Corollary 4.26 Theorem 4.25 Theorem 4.30 Theorem 4.32 Corollary 4.33

Table 4.1: **Summary of main results of Chapter 4.** Each landscape type (column 1) is characterized by the most complicated permitted type of epistasis (column 2; see 3.2). Based on this, there are families of this landscape type that are easy or hard under progressively more general dynamics (column 3), which is proved in the corresponding part of the chapter (column 4).

4.1 Local vs global fitness peaks

For the majority of this chapter – with the exception of Section 4.7 – the exact fitness values or their physical interpretations do not matter; only the rank-ordering of fitness and the structure of adaptive paths matters (see Definition 3.1 and Section 3.8).

In general, adaptive paths can continue until they reach a local fitness optimum:

Definition 4.1. A genotype u is a *local fitness optimum* (sometimes also called a *(local) fitness peak*) if for all adjacent genotypes v, we have $f(v) \leq f(u)$.

Note the above definition of a genotype as a local fitness optimum allows for adjacent genotypes of equal (or lesser) fitness. In particular, this means that points within a *fitness plateau* can be local fitness optima. At times, I will assume for simplicity that no two adjacent genotypes have exactly the same fitness to avoid considering fitness plateaus; but this is not an important restriction and all the hardness results can be reproved without it. A local fitness optimum is a *global fitness optimum* if all other genotypes in the whole of the genetic space (not just neighbours) have the same or lower fitness (i.e., if no other local fitness optimum in the whole of the genetic space has a higher fitness).

4.2 Smooth fitness landscapes

As I discussed in Section 3.2.1, if a fitness landscape has no sign epistasis then it is a smooth landscape and has a single peak x^* [228, 39]. Every shortest path from an arbitrary x to x^* in the mutation-graph is an adaptive path – a flow in the fitness graph – and every adaptive path in the fitness graph is a shortest path in the mutation graph [39]. Thus, evolution can quickly find the global optimum in a smooth fitness landscape, with an adaptive path taking at most n steps: that is, all smooth fitness landscapes are easy landscapes. For an example, see the smooth *Escherichia coli* β-lactamase fitness landscape measured by Chou et al. [30] shown in Figure 3.3a.

Proposition 4.2 ([228, 39])**.** *If there is no sign epistasis in a fitness landscape, then it is called a* smooth landscape *and has a single peak x^*. Every shortest path (ignoring edge directions) from an arbitrary genotype x to x^* is an adaptive path, and every adaptive path from x to x^* is a shortest path (ignoring edge directions).*

Where by ‘shortest path (ignoring edge directions)’, I mean any shortest path in the mutation-graph, irrespective of if the fitness along the edges of that path increases (‘up arrow’) or decreases (‘down arrow’). In other words, an arbitrary shortest path between x and x^* corresponds to an arbitrary swapping of genes at the loci on which x and x^* from the value they have in x to the value in x^*. This means, for example, that if x and x^* differ on d loci then there are $d!$ many shortest paths between them and by Proposition 4.2 those are also the $d!$ adaptive paths between them.

4.3 Semismooth fitness landscapes

Since a smooth landscape is always easy, let’s introduce the minimal amount of epistasis: sign epistasis, without any reciprocal sign epistasis.

Definition 4.3. A *semismooth fitness landscape* on $\{0,1\}^n$ with fitness function f is a fitness landscape that has no reciprocal sign epistasis. Such a fitness function f is also called semismooth.

For some of the following proofs, it will be useful to define sublandscapes.

Definition 4.4. Given a landscape on n bits, a sublandscape spanned by $S \subseteq [n]$ is a landscape on $\{0,1\}^S$ where the alleles at the loci (indices) in S can vary but the indices in $[n] - S$ are fixed according to some string $u \in \{0,1\}^{[n]-S}$.

Note that the whole landscape is a sublandscape of itself (taking $S = [n]$). For any $S \subset [n]$, there are $2^{n-|S|}$ many sublandscapes on S corresponding to the possible $u \in \{0,1\}^{[n]-S}$. Reciprocal sign epistasis between bits i and j corresponds to a sublandscape on $\{i,j\}$ that has two distinct peaks.

Now, I can note a couple of important properties of semismooth landscapes. The first important property of semismooth landscapes is a slightly more general statement of Poelwijk et al. [179]'s theorem about the necessity of reciprocal sign epistasis.

Theorem 4.5. *A fitness landscape on $\{0,1\}^n$ has some sublandscape with more than one distinct peak if and only if it has reciprocal sign epistasis.*

The proof will show that a minimal multi-peak sublandscape must have size 2. I will do this by considering longest walks in a sublandscape. This proof technique is distinct from Poelwijk et al. [179].

Proof. ($\Leftarrow$): This direction is obvious because the two loci reciprocal sign epistasis fitness subgraph has two peaks (see 3.1c and Section 3.2). The real work is in the other direction.

($\Rightarrow$): Let's consider a minimal sublandscape L that has more than one distinct peak: that means that if this sublandscape is spanned by S (i.e., $\{0,1\}^S$) then no sublandscape spanned by $T \subset S$ has multiple peaks.

Since L is minimal, its peaks must differ from each other on each bit in S, for if there was a bit $i \in S$ on which two peaks agreed then that bit could be fixed to that value and eliminated from S to make a smaller sublandscape spanned by $S - \{i\}$ with two peaks. Thus, the minimal multipeak sublandscape has precisely two peaks. Call these peaks x^* and y^*.

Claim: In a minimal multipeak sublandscape, from each non-peak vertex, there must be a path to each peak.

Let's prove the claim by contradiction: Consider an arbitrary non-peak vertex x, and suppose it has no path to the x^* peak. Since any path from x in L must terminate at some peak, take the longest path from x to the peak y^* that it reaches, and let y be the last step in that path before the peak. Notice that y must only have one beneficial mutation (on bit i), the one to the peak. For if it had more than one beneficial mutation, it could take the non-peak step to y' and then proceed from y' to y^* (x^* is not an option by assumption, and there are only two peaks in L) and thus provide a longer path to the peak. Now consider the landscape on $S - \{i\}$, with the ith bit fixed to y_i. Since y_i is the same as x^*_i (both are opposite of y^*_i), x^* is still a peak over $S - \{i\}$, but so is y (since it's only beneficial mutation was eliminated by fixing i to y_i). But this contradicts minimality, so no such x exists.

Now that we know that we can reach each peak from any vertex x, let us again consider the longest path from x to y^* with y as the last step in that path before the peak, and i as the position of the last beneficial mutation. Since all non-peak vertices must reach both peaks, there must be some other beneficial mutation j from y to x' that eventually leads to x^*. But if x' is not a peak then it must also have a way to reach y^*, but then we could make a longer path, contradicting the construction of y. Thus x' must be the peak x^*.

This means that x^* and y^* differ in only the two bits i and j. But in a minimal multipeaked sublandscape they must differ in all bits, so $S = \{i, j\}$ (i.e., this sublandscape is an example of reciprocal sign epistasis). □

Corollary 4.6. *A fitness landscape without reciprocal sign epistasis has a unique single peak.*

Proof. This follows from the contrapositive of Theorem 4.5, since the whole landscape is a sublandscape of itself. □

The above results can also be restated in the terminology used to analyze simplex algorithms.[208, 144]

Definition 4.7. A directed acyclic orientation of a hypercube $\{0, 1\}^n$ is called an *acyclic unique sink orientation (AUSO)* if every subcube (face; including the whole cube) has a unique sink.

This makes the contrapositive of Theorem 4.5 into the following proposition:

Proposition 4.8. *A semismooth fitness graph is an AUSO*

Now, if we let $x \oplus y$ mean XOR between x and y and let $||z||_1$ mean the number of 1s in z then we can state the main theorem about semismooth fitness landscapes:

Theorem 4.9. *A semismooth fitness landscape has a unique fitness peak x^* and for any vertex x in the landscape, there exists a path of length $||x^* \oplus x||_1$ (Hamming distance to peak) from x to the peak.*

Proof. The unique peak x^* is just a restatement of Corollary 4.6. To show that there is always a path of Hamming distance to the peak (i.e., $||x^* \oplus x||_1$ – a length equal to the number of bits that x^* and x differ on), I will show that given an arbitrary x, we can always pick a mutation k that decreases the Hamming distance to x^* by 1.

Let S be the set of indices that x and x^* disagree on, $|S| = ||x^* \oplus x||_1$. Consider the sublandscape on S with the other bits fixed to what x and x^* agree on. In this sublandscape x^* is a peak, thus by Theorem 4.5 x isn't a peak and must have some beneficial mutation $k \in S$. This is the k we were looking for. □

Note that this proof specifies an algorithm for constructing a short adaptive walk to the fitness peak x^*. However, this algorithm requires knowing x^* ahead of time (i.e., seeing the peak in the distance). But evolution does not know ahead of time where peaks are, and so cannot carry out this algorithm. Even though a short path to the peak always exists, evolutionary dynamics might not follow it.

The equivalence between semismooth fitness landscapes and AUSOs allows us to prove – in certain cases – that finding these short paths to the peak is difficult. In the computer science literature, it is believed that AUSOs are computationally difficult to solve (see, Fearnley et al. [48] for a formal treatment in the case of the closely related USOs). In the literature on the analysis of simplex algorithms, it is believed that for almost any pivot rule there will exist AUSO where that pivot rule will take a long time to find the peak and explicit constructions of AUSOs are known for the intractability of specific pivot rules [123, 85, 65, 3, 208, 193, 144]. These constructions can be reinterpreted in biological terminology to give us hard semismooth fitness landscapes, as I do below.

4.3.1 Hard semismooth landscape for random fitter-mutant SSWM

The simplest evolutionary rule to consider is picking a mutation uniformly at random among ones that increase fitness. This can be restated as picking and following one of the out-edges in the fitness graph at random (i.e., this is equivalent to the random-edge simplex pivot rule [144]). Proposition 4.8 allows me to use the hard AUSOs constructed by Matousek and Szabo [144] as a family of hard semismooth landscapes.

Theorem 4.10 (Matousek and Szabo [144]'s Theorem 1 in biological terminology)**.** *There exist semismooth fitness landscapes on $\{0,1\}^n$ such that random fitter-mutant SSWM dynamics starting from a random vertex, with probability at least $1 - e^{-\Omega(n^{1/3})}$ follows an adaptive path of at least $e^{\Omega(n^{1/3})}$ steps to evolutionary equilibrium.*

Proof. Theorem 1 by Matousek and Szabo [144] states that:

> There are positive constants c, c_1 such that for all sufficiently large n there exists an acyclic unique-sink orientation (AUSO) of the n-dimensional cube $[0,1]^n$ such that the algorithm *RANDOM EDGE*, started at a randomly chosen vertex, with probability at least $1 - e^{-c_1 n^{1/3}}$ makes at least $e^{cn^{1/3}}$ steps before reaching the sink.

This translates to the biological statement of Theorem 4.10 by noting that AUSOs are semismooth fitness landscapes (Proposition 4.8) and that *RANDOM EDGE* is the same algorithm as random fitter-mutant SSWM dynamics. □

In other words, multiple peaks – or even reciprocal sign-epistasis – are not required to make a complex fitness landscape. In fact, AUSOs were developed to capture the idea of a linear function on a polytope (although AUSOs are a slightly bigger class). It is not surprising to find the simplex algorithm in the context of semismooth landscapes, since we can regard it as a local search algorithm for linear programming (where local optimality coincides with global optimality). Linear fitness functions are usually considered to be some of the simplest landscapes by theoretical biologists; showing that adaptation is hard on these landscapes (or ones like them) is a surprising result. One of the advantages of formally connecting evolutionary biology to theoretical computer science and combinatorial optimization is that we can get these sort of surprising results "straight out of the box", as I did above by translating the theorem of Matousek and Szabo [144] into biological terminology.

4.4 Winding landscape: Recursive construction of hard semismooth landscape for fittest-mutant SSWM

One might object to taking random fitter mutants because sometimes the selected mutations are only marginally fitter than the wildtype. It might seem natural to speed-up evolution by always selecting the fittest possible mutant. Here I show that, in general, this does not help.

Consider a fitness landscape on $\{0,1\}^m$ with semismooth fitness function f that, if started at 0^m, will take k steps to reach its evolutionary equilibrium at x^*. I will show how to grow this into a fitness landscape on $\{0,1\}^{m+2}$ with semismooth fitness function f' that if started at 0^{m+2} will take $2(k+1)$ steps to reach its evolutionary equilibrium at 0^m11.

For simplicity of analysis, let us define the following functions and variables for all points in $\{0,1\}^m$ that are not an evolutionary equilibrium under f (i.e., all except x^*). Let

$$s^+(x) = \max_{y \in N(x) \text{ s.t. } f(y) > f(x)} f(y) - f(x) \tag{4.1}$$

$$s^-(x) = \min_{y \in N(x) \text{ s.t. } f(x)+s^+(x) > f(y) > f(x)} f(y) - f(x). \tag{4.2}$$

where $N(x)$ are the neighbours of x in the mutation graphs (i.e., genotypes that differ from x in one bit).

Now overload these into constants, as follows: define $s^+ = \min_x s^+(x)$ and $s^- = \min_x s^-(x)$. Suppose that f is such that $s^- < s^+$; otherwise set $s^- = s^+/2$ (do this also, if $N(x)$ s.t. $f(x) + s^+(x) > f(y) > f(x)$ is empty for some non-equilibrium x).

Let $x \oplus y$ mean the XOR between x and y. Consider the 'reflected' function $f(x \oplus x^*)$. I call this function *reflected* because we could visualize $f(x \oplus x^*)$ as the same function as $f(x)$, except it is mirrored (swapping the fitness effects of the 0 and 1 allele) along each locus i where $x^*_i = 1$. Note that if $f(x)$ is semismooth then so is $f(x \oplus x^*)$, since it just relabels the directions of some dimensions. The reflected function preserves all the important structure. In particular, if under $f(x)$ it took k steps to go from 0^m to x^* then under $f(x \oplus x^*)$ it will take k steps to from from x^* to 0^m.

Definition 4.11. Given a fitness landscape $f : \{0,1\}^m \to \mathbb{R}$, its *wind-up* $f' : \{0,1\}^{m+2} \to \mathbb{R}$ is:

$$f'(xab) = \begin{cases} f(x) & \text{if } a = b = 0 \\ f(x) + s^- & \text{if } a \neq b \text{ and } x \neq x^* \\ f(x^*) + s^- & \text{if } a = 0, b = 1 \text{ and } x = x^* \\ f(x^*) + s^+ & \text{if } a = 1, b = 0 \text{ and } x = x^* \\ f(x \oplus x^*) + f(x^*) + 2s^+ & \text{if } a = b = 1 \end{cases} \tag{4.3}$$

Basically the $x00$ subcube is the original landscape, the $x10$ and $x01$ subcubes serve as 'buffers' to make sure that the walk doesn't leave the first subcube before reaching x^*00, and the $x11$ is the original landscape reflected around x^* that takes us from x^*11 to 0^m11.

Notice, that f' has the same s^+ and s^- as f.

Now we just need to establish some properties:

Proposition 4.12. *Fittest-mutant SSWM dynamics will not leave the* $\{0,1\}^m00$ *subcube until reaching* x^*00.

Proof. By definition, the fittest mutant (i.e., neighbour over $\{0,1\}^m$) from each genotype $x \in \{0,1\}^m$ that isn't x^* in f, has a fitness advantage of s^+ or higher. Hence adding two extra edges from $x00$ to $x10$ and $x01$, each with fitness advantage $s^- < s^+$ will not change the edge that fittest-mutant SSWM picks. □

Proposition 4.13. *SSWM dynamics will not leave the* $\{0,1\}^m11$ *subcube after entering it.*

Proof. This is because f' has strictly greater fitness on the $\{0,1\}^m11$ subcube than on the other three subcubes. Confirming this, note that for every $x \in \{0,1\}^m$:

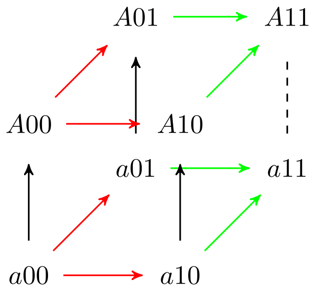
$$\begin{aligned}
f(x \oplus x^*) + f(x^*) + 2s^+ &\geq f(x^*) + 2s^+ && \text{since } f \text{ is non-negative} && (4.4)\\
&\geq f(x^*) + s^+ && \text{since } s^+ > 0 && (4.5)\\
&\geq f(x^*) + s^- && \text{since } s^+ > s^- && (4.6)\\
&\geq f(x) + s^- && \text{since } x^* \text{ is fitness peak of } f && (4.7)\\
&\geq f(x) && \text{since } s^- > 0 && (4.8)
\end{aligned}$$

□

Proposition 4.14. *If f on $\{0,1\}^m$ has no reciprocal sign-epistasis then (since $s^- > 0$) f' on $\{0,1\}^{m+2}$ has no reciprocal sign-epistasis.*

Proof. Consider any pair of genes $i, j \in [m]$. Among these first m genes, depending the last two bits, we are looking at landscapes on $\{0,1\}^m00$, $\{0,1\}^m01$, $\{0,1\}^m10$, or $\{0,1\}^m11$, with the fitness given by $f(x)$,$f(x) + s^-$,$f(x) + s^-$, or $f(x \oplus x^*) + f(x^*) + 2s^+$ (respectively). All these landscapes have isomorphic combinatorial structure to f and thus the same kinds of epistasis. Since f has no reciprocal sign-epistasis, all these subcubes lack it, too.

Now, let's look at the case of where the gene pair goes outside the first m genes. Consider an arbitrary gene $i \in [m]$, let $u \in \{0,1\}^{i-1}$, $v \in \{0,1\}^{m-i}$ be arbitrary. Label $a, A \in \{0,1\}$ such that $f(uav) < f(uAv)$ and look at the subcube $u\{0,1\}v\{0,1\}^2$:



The solid black edges have their directions from the definition of a and A. The red edges have their direction because $s^+ > s^- > 0$. The green edges have their direction because of Proposition 4.13. The direction of the dotted black edge will depend on if x^* contains 0 (point up) or 1 (point down) at position i, but regardless of the direction, no reciprocal sign epistasis is introduced. □

Corollary 4.15. *Given f' on $\{0,1\}^{m+2}$, the fittest-mutant SSWM dynamics starting at 0^{m+2} will take $2(k+1)$ steps to reach its unique fitness peak at 0^m11.*

Proof. By Proposition 4.12, the walk will first proceed to x^*00 taking k steps. From x^*00, there are only two adaptive mutations x^*10 or x^*01, and the first is fitter. From x^*10 there is only a single adaptive mutation (to x^*11), taking us to $k+2$ steps. From x^*11, by Proposition 4.13, it will take us k more steps to reach 0^m11; totaling $2(k+1)$ steps. □

Theorem 4.16. *There exist semismooth fitness landscapes on $2n$ loci that take $2^{n+1}-2$ fittest mutant steps to reach their unique fitness peak at $0^{2(n-1)}11$ when starting from 0^{2n}.*

Proof. We will build the family of landscapes inductively using our construction, starting from an initial landscape:

$$\begin{tikzcd}
 & f_1(11) = 6 & \\
f_1(10) = 4 \arrow[ur, green] & & f_1(01) = 3 \arrow[ul] \\
 & f_1(00) = 2 \arrow[ul, green] \arrow[ur] &
\end{tikzcd}$$

The resulting path length T_n will be given by the recurrence equation: $T_{n+1} = 2T_n + 2$ with $T_1 = 2$. This recurrence is solved by $T_n = 2^{n+1} - 2$.

□

Call the landscapes constructed as in the above proof, *winding landscapes.* A visual example of the winding landscape construction on 6 loci ($n = 3$ in Theorem 4.16) is given in Figure 2.1. The winding landscapes construction is similar to Horn, Goldberg, and Deb [79]'s *Root2path* construction, except their approach introduced reciprocal sign epistasis despite having a single peak. In fact, we can generate instances of the *Root2path* landscapes by setting $s^- < 0$ in the winding induction stem from Equation 4.3. This will, of course, make Proposition 4.14 false by introducing reciprocal sign-epistasis, but the other proofs in this section would still be valid and thus establish an exponential steepest ascent.

Of course, the landscape in the proof of Theorem 4.16 is an arbitrary initial fitness landscape and any semismooth landscape can be used as a starting point; the walk would still scale exponentially, but there would be a different initial condition. Further, the winding product construction I showed above is just one example for building families. Many more could be considered.

In particular, if we are interested in larger mutation operators like k-point mutations instead of just 1-point mutations then it is relatively straightforward to modify the winding landscape construction. As written, Equation 4.3 uses a buffer of 2 bits in $f'(xab)$ to transition from $f(x)$ to its reflection $f(x \oplus x^*)$. In the more general setting, we'd pad the buffer to be $k+1$ bits: define $f'(xy)$ where $|y| = k+1$ with a smooth landscape on the y portion of the input taking us from $f(x)$ to its reflection. Which leaves most of the above arguments unchanged, only modifying Theorem 4.16 to have the landscape to be on kn loci and the recurrence relation at the end of the proof to be $T_{n+1} = 2T_n + k + 1$.

4.4.1 Hard landscapes from random start

Unfortunately, one might not be impressed by a result that requires starting from a specific genotype like 0^m and ask instead for the expected length of the walk starting from a random vertex. Of course, if a genotype on this long walk is chosen as a starting point then the walk will still be long in most cases. However, there are only $2^{n+1} - 2$ vertices in the walk, among 2^{2n} vertices total, so the probability of landing on the walk is exponentially small. Instead, I will rely on direct sums of landscapes and Proposition 4.12 to get long expected walks.

Proposition 4.17. *With probability* $1/4$*, a winding landscape on* $2n$ *loci will take* 2^n *or more fittest mutant steps to reach the fitness peak from a starting genotype sampled uniformly at random.*

Proof. With probability $1/4$, the randomly sampled starting vertex has the form $x00$ (i.e., its last two bits are 0s). By Proposition 4.12, the walk can't leave the $\{0,1\}^{2(n-1)}00$ landscape until reaching its peak at $0^{2(n-2)}1100$. This might happen quickly, or it might even already be at that peak. But after, it has to follow the two steps to $0^{2(n-2)}1111$ and then due to Proposition 4.13 it will have to follow the normal long path, taking $2^n - 2$ more steps. □

Because of the constant probability of an exponentially long walk, we can get a big lower bound on the expected walk time:

Corollary 4.18. *Fittest-mutant SSWM dynamics starting from an initial genotype chosen uniformly at random will have an expected walk length greater than* 2^{n-2} *on a* $2n$*-loci winding landscape.*

Proof. With probability $1/4$, the walk takes 2^n or more steps, and with probability $3/4$ it takes 0 or more steps. Thus the expected walk length is greater than or equal to $(1/4) * 2^n + (3/4) * 0 =$

2^{n-2}. □

However, 75% of the time, we can't make a guarantee of long dynamics. We can overcome this limitation by taking direct sums of landscapes.

Definition 4.19. Given two fitness landscapes, one with fitness f_1 on $\{0,1\}^{n_1}$ and the other with fitness f_2 on $\{0,1\}^{n_2}$, the *direct sum* $(f_1 \oplus f_2)$ is a landscape with fitness f on $\{0,1\}^{n_1+n_2}$ where $f(xy) = f_1(x) + f_2(y)$ for $x \in \{0,1\}^{n_1}$ and $y \in \{0,1\}^{n_2}$.

Now, for any probability of failure $0 < \delta < 1$, let $m_\delta = \lceil \frac{\log \frac{1}{\delta}}{2 - \log 3} \rceil$ (where log is base 2), note that m_δ is linear in $\log \frac{1}{\delta}$.

Theorem 4.20. *There exist semismooth fitness landscapes on $2nm_\delta$ loci that with probability $1-\delta$, will take 2^n or more fittest mutant steps to reach their fitness peak from a starting genotype sampled uniformly at random.*

Proof. Consider a landscape that is the direct sum of m_δ separate $2n$-loci winding landscapes. Since each constituent is semismooth and since sums don't introduce epistasis, the resulting 'tensor sum' landscape is also semismooth. Further, to reach its single peak, the walk has to reach the peak of each of the m_δ independent winding sublandscapes. But as long as at least one sublandscape has a long walk, we are happy. By Proposition 4.17, we know that for each sublandscape, we will have a short-walk starting genotype with probability at most 3/4. The probability that none of them get a long walk then is at most $(3/4)^{m_\delta} \leq \delta$. □

4.5 Lower bound on the gene interaction network of the winding landscape

It is important to note that the winding fitness landscape is not implemented above by a generalized NK-model (as discussed in Section 3.4) but is defined recursively. However, I use this section to prove that if the winding fitness landscape (and also Horn, Goldberg, and Deb [79]'s *Root2path*) was implemented by a gene-interaction network, then that network would need to be very complicated and have a lot of edges. The winding fitness landscape has the dramatic property of having drastic changes in the direction and magnitude of the gradient of the objective function between points which are very close to each other. This is what I will use for the proof. Consider the sub-cube spanned by the first $2(k+1)$-variables. The sub-cube fitness maximum is at $x^*_k = 0^{2k}11$. If $s^- > 0$ (as in Section 4.4) then the sub-cube fitness minimum is only Hamming-distance 2 away at $0^{2(k+1)}$:

so the flows (i.e., differences in the objective function from the current point to its neighbouring points) change from all positive to all negative in just 2 steps. From this we can prove that the total scope size of any constraint graph implementing this fitness landscape must be high. The case in which we do not necessarily have $s^- > 0$ (as in Horn, Goldberg, and Deb [79]) is slightly more complicated since $0^{2(k+1)}$ is no longer a fitness minimum and mostly has negative flows. However, these negative flows have small magnitude compared to the very large magnitude negative flows at x^*_k, so a similar argument can be used. I formalise this argument below.

For convenience, let $||x||_0$ be the number of non-zero entries in x – also known as the Hamming weight or the zero-'norm' of the vector x. As before, I will use $x[i \to b]$ to mean a bit-string that is the same as x at every bit, except the i-th bit is set to b. Or, in symbols: $\forall j \neq i \quad [x[i \to b]]_j = [x]_j$ and $[x[i \to b]]_i = b$. This allows us to define the *gradient* ∇f or flow of a fitness function f entry-wise as $[\nabla f(x)]_i = f(x[i \to 1]) - f(x[i \to 0])$ to state the degree lower-bound lemma:

Lemma 4.21. *Given a fitness function f implemented by a VCSP with constraint graph G and any two distinct variable assignments x and y that differ on a set of variables S, we have that the total degree $d_G(S) = \sum_{i \in S} d_G(i)$ of S in G is lower-bounded by the change in flow: $d_G(S) \geq ||\nabla fx - \nabla fy||_0$.*

Proof. If we look at a variable at position i and compare $\nabla f(x[i \to 1])$ to $\nabla f(x[i \to 0])$ then any differences in the gradient must have been due solely to the change in variable x_i. Thus, given any position j such that $[\nabla f(x[i \to 1])]_j \neq [\nabla f(x[i \to 0])]_j$ there must be a constraint that has both i and j (and maybe others) in its scope. Thus, by looking at the number of non-zero entries in $\nabla f(x[i \to 1]) - \nabla f(x[i \to 0])$, we get a lower bound on the number of other variables with which each variable i co-occurs in a constraint.

This reasoning can be extended over paths between non-adjacent states. Suppose we have two states x_1 and x_t at Hamming distance t from each other. Let $x_1 x_2 ... x_{t-1} x_k$ be any shortest path between them with the bits flipped at each step given by $i_1, i_2, ... i_{t-1}$. Notice the following:

$$\begin{aligned}
||\nabla f x_1 - \nabla f x_k||_0 &= ||(\nabla f x_1 - \nabla f x_2) + (\nabla f x_2 - \nabla f x_3) + \\
&\quad \cdots + (\nabla f x_{t-1} - \nabla f x_t)||_0 && (4.9) \\
&\leq ||\nabla f x_1 - \nabla f x_2||_0 + ||\nabla f x_2 - \nabla f x_3||_0 + \\
&\quad \cdots + ||\nabla f x_{t-1} - \nabla f x_t)||_0 && (4.10)
\end{aligned}$$

In words: given two states x_1 and x_k that differ at a set of variables S, the total number of variables that the variables in S co-occur with is lower-bounded by $||\nabla f x_1 - \nabla f x_k||_0$. □

Now, I can apply this lower bound technique to the winding fitness landscape.

Proposition 4.22. *If the winding fitness landscape f on $2n$ variables from Section 4.4 is implemented by a VCSP with constraint graph G then $d_G(2k+1) + d_G(2k+2) \geq k$ for each $0 \leq k < n$.*

Proof. Let us look at the gradients at the path's starting point:

$$\nabla f(0^{2n}) = [s^+, s^-, \ldots, s^-, s^-] \tag{4.11}$$

and for each $1 \leq k \leq n$, look at the gradients at subcube peaks $\nabla f(0^{2(k-1)}(11)0^{2(n-k)})$: they have a slightly more complicated form, so we define them point-wise for $i \in [1, n]$, $b \in \{0, 1\}$, and $x = 0^{2(k-1)}(11)0^{2(n-k)}$:

$$[\nabla f(x)]_{2i-b} = \begin{cases} -s^+ + b(s^- - s^+) & \text{if } i < k \\ f^k(x^*_k) - s^- & \text{if } i = k \\ s^+ & \text{if } i = k+1 \ \&\ b = 1 \\ s^- & \text{if } i > k + b \end{cases} \tag{4.12}$$

Looking at the odd entries lower than $2k$ (i.e., $i < k, b = 1$ in Equation 4.12), we have:

$$[\nabla f(0^{2(k-1)}(11)0^{2(n-k)}) - \nabla f(0^{2n})]_{2i-1} = -2s^+_i \neq 0. \tag{4.13}$$

Thus, by Equation 4.10, the variables at positions $2k+1$ and $2k+2$ together have degree of at least k. □

Corollary 4.23. *Any VCSP implementing the winding fitness landscape from Section 4.4 must have a constraint graph that is dense and with unbounded treewidth.*

Proof. Summing up over all $1 \leq k \leq n$, we get that any VCSP instance that implements f must have total degree of at least $(n-1)n/2$ (i.e., quadratic in the number $2n$ of variables).

In particular, this means that a constraint graph of bounded treewidth (which would have total degree linear in $2n$) cannot implement the winding fitness landscape f. □

It is important to note that although I expect some parts of the above argument (especially Lemma 4.21) to be useful more generally, the final result (Proposition 4.22 and Corollary 4.23) applies only to the semismooth winding landscape construction from Section 4.4 and Horn, Goldberg, and Deb [79]'s *root2path*. Hence, my results do not establish that every family of fitness landscapes that is hard for fittest-mutant SSWM dynamics must have complex gene-interaction networks. In fact, recently Cohen, Cooper, Kaznatcheev, and Wallace [32] provided a construction of gene-interaction networks of treewidth 7 that produce fitness landscapes that are hard for fittest-mutant SSWM dynamics. But this construction has reciprocal sign epistasis, so it remains an open question whether a hard semismooth landscape can be constructed with a bounded treewidth gene-interaction network.

4.6 Classic NK model with $K \geq 2$ is PLS-complete

The hardness results for semismooth fitness landscapes in the previous section are only for the particular algorithms of random fitter- and fittest-mutant SSWM dynamics. These are useful for developing our intuition about hard landscapes, but are not the limit of what theoretical computer science can offer. The biggest offering from computer science is the ability to abstract over *any* evolutionary dynamic, not just two particular ones. This is done through the study of computational complexity.

For this, it is useful to look at compact representations of rugged fitness landscapes like the classic and generalized NK model that I described in Chapter 3. Weinberger [227] showed that checking if the global optimum in a classic NK model is greater than some input value V is NP-complete for $K \geq 3$. Although this implies that finding a global optimum is difficult, it says nothing about local optima. As such, it has generated little interest among biologists, although it spurred interest as a model in the evolutionary algorithms literature, leading to a refined proof of NP-completeness for $K \geq 2$ in the classic NK model [233].

To understand the difficulty of finding items with some local property like being an equilibrium, Johnson, Papadimitrio & Yannakakis [86] defined the complexity class of polynomial local search (PLS). A problem is in PLS if the problem statement can be specified by three polynomial time algorithms [189]:

1. An algorithm I that accepts an instance (like a description of a fitness landscape) and outputs a first candidate to consider (the initial genotype).

2. An algorithm F that accepts an instance and a candidate and returns a objective function value (i.e., computes the fitness).

3. An algorithm M that accepts an instance and a candidate and returns an output with a

strictly higher objective function value, or says that the candidate is a local maximum.

We consider a PLS problem solved if an algorithm can output a locally optimal solution for every instance. This algorithm does not necessarily have to use I, F, or M or follow adaptive paths. For instance, it can try to uncover hidden structure from the description of the landscape. A classical example would be the ellipsoid method for linear programming. The hardest problems in PLS – the ones for which a polynomial time solution could be converted to a solution for any other PLS problem – are called PLS-complete. It is believed that PLS-complete problems are not solvable in polynomial time (i.e., FP $\neq$ PLS; where FP stands for the set of function problems solvable in polynomial time), but – much like the famous $\mathsf{P} \neq \mathsf{NP}$ question – this conjecture remains open. Note that finding local optima on fitness landscapes is an example of a PLS problem, where I is your method for choosing the initial genotype, F is the fitness function, and M computes an individual adaptive step.

Definition 4.24 (Weighted 2SAT)**.** Consider n variables $x = x_1 ... x_n \in \{0,1\}^n$ and m clauses $C_1, ..., C_m$ and associated positive integer weights $c_1, ... c_m$. Each clause C_k contains two literals (a literal is a variable x_i or its negation $\overline{x_i}$), and contributes c_k to the fitness if at least one of the literals is satisfied, and nothing if neither literal is satisfied. The total fitness $c(x)$ is the sum of the individual contributions of the m clauses. Two assignments x and x' are adjacent if there is exactly one index i such that $x_i \neq x'_i$. We want to maximize fitness.

The Weighted 2SAT problem is PLS-complete [191]. Since the weighted 2SAT problem is a kind of VCSP (and, as I defined in Section 3.4, the generalized NK model is just the VCSP instances) this also means that the *geralized* NK-model of Section 3.4 for $K \geq 1$ is PLS-complete. To show that the *classic* NK-model of Section 3.3 is also PLS-complete, I will show how to reduce any instance of Weighted 2SAT to an instance of the classic NK-model.

Theorem 4.25. *Finding a local optimum in the classic NK fitness landscape with $K \geq 2$ or the generalized NK model with $K \geq 1$ is PLS-complete.*

Note that this is the tightest result possible, since for $K = 0$ both the classic and generalized NK-models are smooth, and for $K = 1$ the classic NK-model is solvable in polynomial time by dynamic programming [233].

The only complication in proving the above theorem is in dealing with the one-gene-one-function assumption built into the classic NK model. But this can be done by representing 2SAT instances via their incidence graph and assigning a gene and corresponding fitness component to handle each vertex of that incidence graph.

Proof. Consider an instance of Weighted 2SAT with variables $x_1, ..., x_n$, clauses $C_1, ..., C_m$ and positive integer weights $c_1, ..., c_m$. We will build a landscape on $m + n$ loci, with the first m genes labeled $b_1, ..., b_m$ and the next n genes labeled $x_1, ..., x_n$. Each b_k will correspond to a clause C_k that uses the variables x_i and x_j (i.e., the first literal is either x_i or $\overline{x_i}$ and the second is x_j or $\overline{x_j}$; set $i < j$ to avoid ambiguity). Define the corresponding fitness effect of the gene as:

$$f_k(0x_ix_j) = \begin{cases} c_k & \text{if } C_k \text{ is satisfied} \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

$$f_k(1x_ix_j) = f_k(0x_ix_j) + 1 \tag{4.15}$$

Link the x_i arbitrarily (say to $x_{(i \mod n)+1}$ and $x_{(i+1 \mod n)+1}$, or to nothing at all) with a fitness effect of zero, regardless of the values.

In any local maximum bx, we have $b = 11..1$ and $f(x) = m + c(x)$. On the subcube with $b = 11...1$, the Weighted 2SAT instance and this classic NK model instance have the same exact fitness graph structure, and so there is a bijection between their adaptive paths and local maxima.

Finally, for the generalized NK model with $K \geq 1$ there is nothing needed to show since every instance of weighted 2SAT is an instance of the generalized NK model with $K = 1$. □

Assuming – as most computer scientists do – that there exists some problem in PLS not solvable in polynomial time (i.e., FP $\neq$ PLS) or even bounded-error randomized polynomial time (i.e., FBPP $\neq$ PLS), then Theorem 4.25 implies that no matter what mechanistic rule evolution follows (even ones we have not discovered, yet), be it as simple as SSWM or as complicated as any polynomial time algorithm, there will be NK landscapes with $K = 2$ such that evolution will not be able to find a fitness peak efficiently. But if we focus only on rules that follow adaptive paths then we can strengthen the result:

Corollary 4.26. *There is a constant $c > 0$ such that, for infinitely many n, there are instances of the classic NK model with $K \geq 2$ or generalized NK model with $K \geq 1$ on $\{0,1\}^n$ and initial genotype v such that any adaptive path from v will have to take at least 2^{cn} steps before finding a fitness peak.*

Proof. If the initial vertex has $b = 11...1$ then there is a bijection between adaptive paths in the fitness landscape and any weight-increasing path for optimizing the weighted 2SAT problem. For the generalized NK-model, weighted 2SAT instances are instances of the generalized model – so the identity function is a bijection. Thus, in both cases, Schaffer and Yannakakis [191]'s Theorem 5.15 applies. □

This result holds independently of any complexity theoretic assumptions about the relationship between polynomial-time and PLS. Hence, there are some landscapes and initial genotypes, such that any rule we use for adaptation that only considers fitter single-gene mutants will take an exponential number of steps to find the local optimum.

If we turn to larger mutational neighbourhoods than singe-gene mutants then – due to the large class of possible adaptive dynamics – a variant of Corollary 4.26 will have to be reproved (by, for example, using a buffer padding argument similar to the end of Section 4.4) but Theorem 4.25 is unaffected:

Corollary 4.27. *For any definition of local equilibrium with respect to a mutation neighbourhood or genetic distance that contains point-mutations as a subset (i.e., if* $\forall x \; \{y \mid ||y - x||_1 = 1\} \subseteq N(x)$*), the classic NK model with* $K \geq 2$ *and generalized NK model with* $K \geq 1$ *is PLS-hard.*

Proof. Any mutation operator that is a superset of point-mutations will only decrease the number of evolutionary equilibria without introducing new ones. Thus, it will only make the task of finding such an equilibrium (just as, or) more difficult. However, since the algorithms studied by PLS do not have to use the mutation operator during their execution, changing it does not give them any more computational resources. □

Finally, it is important to see the NK-model as an example model, albeit a simple and natural one. If we consider more complex models of fitness landscapes – say dynamic fitness landscapes – it is often the case that there is some parameter or limit that produces the special case of a static fitness landscape like the NK-model. In particular, static landscapes are often a sub-model of dynamic fitness landscapes and thus solving dynamic fitness landscapes can only be more difficult that static ones.

4.7 Approximate peaks & selection coefficient time-series

If we want to consider the notion of being 'close' to a peak, or ideas like nearly-neutral networks, then we need to use the whole numeric structure of the fitness function f and not just the rank-ordering that was sufficient until this point. Thus, let us consider relaxations of equilibrium, and being "close" to a peak instead of exactly at one. To measure closeness in fitness, I will treat $f : \{0, 1\}^n \to \mathbb{R}$ as a function from genotype to real-valued fitness. I do this to be consistent with typical approaches to fitness landscapes in the biology literature, but this could have also been expressed in terms of the integer-valued fitness of Chapter 3. In fact, I will do this more carefully in Section 5.1 when I discuss the span arguments that generalize the techniques of this section. The following definitions and proofs are inspired by the combinatorial optimization results of Orlin, Punnen, and Schulz [164].

Definition 4.28. A genotype x is at an s-approximate peak if $\forall y \in N(x)$ $f(y) \leq (1+s)f(x)$.

Equivalently, we can write $s = \frac{f(y)-f(x)}{f(x)}$ to see that s is defined in the same way as selection coefficient for invader type y against wild type x in population genetics [62].

The question becomes how big does s have to be for evolution to find an s-approximate peak. But since there is no absolute unit of fitness, we will need to define $f_{\max} = \max_x f(x)$ and:

$$f_\delta = \min_x \min_{y \in N(x) \text{ s.t. } f(y) > f(x)} (f(y) - f(x)) \tag{4.16}$$

note that f_δ is defined similarly to s^- from Section 4.3.1.

First, it is important to note that all landscapes where f_δ is not small compared to $f_{\max}$ are easy.

Proposition 4.29. *If $f_{max}/f_\delta \in O(n^k)$ for some constant k then an exact peak can be found in a polynomial in n number of mutations by any adaptive dynamic.*

Proof. Since each adaptive step increases fitness by at least f_δ then after t adaptive steps, we have $f(x_t) \geq f_\delta t$. Combine this with $f(x_t) \leq f_{\max}$ to get that $t \leq f_{\max}/f_\delta$. □

So, we need to focus on bigger gaps between f_δ and $f_{\max}$. If the gap is exponential then we can find an approximate peak for moderate sized s on any landscape.

Theorem 4.30. *If $\log(f_{max}/f_\delta) \in O(n^k)$ then fittest mutant SSWM dynamics can find a local s-approximate peak in time polynomial in n and $\frac{1}{s}$.*

Proof. Let x_0 be the initial genotype, if it is an exact peak then we are done. Otherwise, let x_1 be the next adaptive step, by definition of f_δ, we have that $f(x_1) \geq f(x_0) + f_\delta \geq f_\delta$. Now, consider an adaptive path $x_1 ... x_t$ that hasn't encountered an s-approximate peak (i.e., a mutation was always available such that $f(x_{i+1}) > (1+s)f(x_i)$). Thus, we have that $f(x_t) \leq f_{\max}$ and that $f(x_t) \geq (1+s)^t f_1 \geq (1+s)^t f_\delta$. Putting these two together:

$$(1+s)^t f_\delta \leq f_{\max} \tag{4.17}$$

$$t \ln(1+s) \leq \ln \frac{f_{\max}}{f_\delta} \tag{4.18}$$

$$t \leq (\ln \frac{f_{\max}}{f_\delta}) / \ln(1+s) \leq (1 + 1/s) \ln \frac{f_{\max}}{f_\delta} \tag{4.19}$$

Where I used $\ln(1+s) \geq \frac{s}{1+s}$ in the last step. Combining with the conditions on $\log f_{\max}/f_\delta$, we get: $t \in O(\frac{n^k}{s})$. □

But for very small s, finding an approximate peak is as hard as finding an exact peak.

Proposition 4.31. *If $s \leq f_\delta / f_{max}$ then any s-approximate peak is a (exact) local peak.*

Proof. If an s-approximate peak at x is not an exact peak then there exists a $y \in N(x)$ such that $f(y) - f(x) \geq f_\delta$ but $f(y) < (1+s)f(x)$. Combining this with $f(x) \leq f_{\max}$, we get that $s > f_\delta / f_{\max}$. □

Thus, it isn't possible to find an s-approximate peak for very small s on hard fitness landscapes:

Theorem 4.32. *If FP $\neq$ PLS and $\log(f_{max}/f_\delta) \in O(n^k)$ then (for the classic NK-model with $K \geq 2$ and the generalized NK-model with $K \geq 1$) a local s-approximate peak cannot be found in time polynomial in n and $\log \frac{1}{s}$.*

Proof. If such an algorithm existed then we'd run it with $s = f_\delta / f_{\max}$ and – by Proposition 4.31 – the approximate peak it finds would be exact. Further, in this case $\log \frac{1}{s} = \log(f_{\max}/f_\delta) \in O(n^k)$ and thus the runtime would be polynomial in n. This is not possible for the NK-model with $K \geq 2$ by Theorem 4.25 (unless FP = PLS). □

This also means that the selective coefficient of the fittest mutant $s(t) = \max_{y \in N(x_t) \cup \{x_t\}} (f(y) - f(x_t))/f(x_t)$ cannot decay exponentially quickly.

Corollary 4.33. *If FP $\neq$ PLS then there are no evolutionary dynamics such that $s(t) \leq e^{-mt}$ for all instances of the classic NK-model with $K \geq 2$ and the generalized NK-model with $K \leq 1$.*

Contrast this with the always achievable power-law decrease in $s(t)$ from Theorem 4.30.

4.8 Distributions and random fitness landscapes

Given that randomly sampling landscapes can introduce structure like short paths [215], I suspect that the structure of this simple sampling led prior research to miss the possibility of exponentially long walks. The independent sampling of fitness components from the same distributions is especially apt to realize the conditions of Proposition 4.29 since it makes it unlikely to create an exponential gap between the smallest positive fitness gap f_δ and the maximum achievable fitness $f_{\max}$. Future work could provide a more careful analysis of this conjecture.

Although, as I discussed in Section 3.6, there is evidence for simple distributions on small fitness landscapes (on upto 8 genes; see Franke et al. [50] and Szendro et al. [209]), there is little to no data on the distribution of large (i.e., on many loci) fitness landscapes in nature. And given the exponential size of fitness landscapes, it is unlikely that such data could be collected without assumptions on the generating hypothesis class (for example, as I suggest in Section 3.7.1). However, if a single sampling distribution is required then it is tempting to turn to Occam's razor and consider simpler landscapes as more likely. This can be done by sampling landscapes with negative log probability proportional to their minimum description length (i.e., according to the Kolmogorov universal distribution). If landscapes are sampled in this way, then I would expect all the orders of magnitude for hardness results established herein to hold [130]. However, a close examination of this is beyond the scope of this thesis and I leave it as an open question for future work to prove this formally and to contrast the ubiquity of hardness in fitness landscapes sampled under different theoretical distributions. As outlined above, it would be especially interesting to analyze the uniform distribution (that is popular in statistical physics) versus the Kolmogorov universal distribution (that is used in theoretical computer science).

4.9 Summary

In Chapter 2, we saw the deep insights that algorithmic biology can give into evolution. In this chapter, I formalized these insights. In formalizing the theory of hard fitness landscapes, my aim is not only to prove that computational complexity is an ultimate constraint on evolution but to also introduce new methods into theoretical biology. My hope is that both the close analysis of specific algorithms and the classification of fitness landscapes into broad complexity classes can give us new lenses through which to see evolution. In Section 4.4, I used techniques from the analysis of simplex algorithms to formulate hardness results for specific evolutionary algorithms like random fitter- or fittest-mutant strong-selection weak-mutation dynamics. And in Section 4.6, I used complexity theoretic reductions to establish that gene-interaction networks can express fitness landscapes that are hard for any evolutionary dynamic. I expect that techniques like the above can help us engage with many other puzzles in biology. Since my aim in this chapter was to prove the existence of hard landscapes, the overall distinction between easy vs hard that I established was coarse. It is the goal of Chapter 5 to refine our knowledge of this boundary between easy vs hard landscapes.

Chapter 5

Structure of easy vs hard gene-interaction networks

While formalizing the theory of hard fitness landscapes in Chapter 4, I classified the 'hardness' of fitness landscapes according to the coarse epistasis-based criteria defined in Section 3.2. As highlighted in Table 4.1, this coarse-grained criteria allows for only three classes of fitness landscapes (based on their maximum allowed kind of epistasis): smooth (max epistasis: magnitude), semismooth (max epistasis: sign), and rugged (max epistasis: reciprocal sign). In this chapter, I will switch to more fine-grained criteria to refine the separation of easy vs hard fitness landscapes. Instead of classifying fitness landscapes based on a coarse binary distinction (does a certain kind of epistasis occur or not anywhere in a fitness landscape), I will look at the more fine-grained criteria of how the various occurrences of epistasis in the landscape interrelate to each other. Formally, I will base this chapter's more fine-grained separation criteria on Section 3.4: the structure of the gene-interaction networks that encode fitness landscapes.

In Chapter 4, I showed that the presence of either sign or reciprocal sign epistasis is enough to allow for hard landscapes. Thus, according to the coarse-grained three classes, only smooth landscapes are easy. As gene-interaction networks, these smooth landscapes are sign-equivalent to VCSP-instances with only unary constraints (or gene-interaction networks with degree zero). But we would not expect that just a single occurrence of sign epistasis somewhere in the landscape or a single constraint edge in a gene-interaction network would make the fitness landscape hard. In this chapter I will formalize this intuition. Here I will establish more fine-grained criteria on what features of a gene-interaction network will guarantee that the corresponding fitness landscape is easy.

In theoretical computer science, classifying easy instances is like asking for an upper bound: give an algorithm that finds a local fitness peak quickly. But this won't do for evolution: evolution is an algorithm, but it is not *any* algorithm, nor is it one specific algorithm. Since algorithmic biology aims to reason about large classes of potential evolutionary dynamics in order to abstract over multiple realizabilities via unknown microdynamical details and population structure, I want to focus this chapter on a reasonably broad class of algorithms. In particular, the smooth landscapes of Section 4.2 were easy for any *local adaptive dynamics*: i.e., for any process that moved the population 'uphill' by point-mutations. Whereas in Chapter 4, I showed that local fitness optima may not be reachable in a reasonable amount of time, even when allowing progressively more general evolutionary dynamics, in this chapter I will focus on when a local fitness optimum is reachable in a reasonable amount of time under any local adaptive dynamics. In other words, I will be expanding the class of very easy landscapes beyond just smooth fitness landscapes.

Local adaptive dynamics is a large class of potential evolutionary dynamics that is useful not only because it captures many of our intuitions about evolutionary hill climbing and local search but also because it allows me to focus on a structural feature of fitness landscapes: the length of the longest adaptive path in the landscape (see Definition 3.1 for adaptive paths). If a fitness landscape is easy for any local adaptive dynamic then there are no long adaptive paths in the landscape and vice versa. In this chapter, I will use easy (or more appropriately 'very easy') to mean that a family of fitness landscapes only has short adaptive paths (i.e., of length that is polynomially bounded). I will consider the fitness landscapes hard (for some local adaptive dynamic) otherwise.

To extend the boundary of the class of very easy fitness landscapes, I will adapt existing proof techniques and develop new ones. In Section 5.1, I will formalize the technique of minimizing span of distinct fitness values that I used implicitly in the proofs of Proposition 4.29 and Theorem 4.30 (and that have been used in prior work on locally optimal MAX-SAT [180, 153]). This will let me establish, in Section 5.1.1, that biallelic gene-interaction networks of degree ≤ 2 are easy (Theorem 5.6). But, in Section 5.1.2, I will show that span arguments are incapable of extending much beyond this. In particular, even tree-structured gene-interaction networks of max degree 4 can have exponential minimal span even though their fitness landscapes only have short adaptive paths (Example 5.11). To address this limitation, in Section 5.2, I will develop a new proof technique using *encouragement paths* to show that all tree-structured biallelic gene-interaction networks correspond to very easy fitness landscapes (Theorem 5.15). Finally, in Section 5.3, I will show that the encouragement paths technique cannot be pushed further. In particular, there exist exponentially long adaptive paths in tree-structured triallelic gene-interaction networks (Example 5.25) and in biallelic gene-interaction networks of treewidth 2 (Example 5.26). Thus, classes of fitness landscapes that contain either of these two examples are not very easy.

5.1 Minimizing the span of fitness values

In this section, I consider the numerical values of the constraints in a VCSP-instance, and show that a simple function of these provides a bound on the length of the longest directed path in the associated fitness graph, and hence a bound on the number of steps taken by a local search algorithm.

Definition 5.1. Given a VCSP-instance $\mathcal{C} = \{C_{S_1}, \ldots, C_{S_m}\}$ over domain D^n, define $\text{span}(\mathcal{C}) = \sum_{k=1}^{m}(\max_{z \in D^{S_k}} C_{S_k}(z) - \min_{z \in D^{S_k}} C_{S_k}(z))$.

Proposition 5.2. *Given a VCSP-instance $\mathcal{C}$, with associated fitness graph $G_{\mathcal{C}}$, the length of the longest directed path in $G_{\mathcal{C}}$ is less than or equal to* $\text{span}(\mathcal{C})$*.*

Proof. The maximum value of the fitness function f implemented by $\mathcal{C}$ cannot exceed the sum of the largest magnitude in each constraint. Similarly, the minimum value of f cannot be less than the sum of the smallest magnitude in each constraint. The difference between these bounds is precisely $\text{span}(\mathcal{C})$. Since we have defined a VCSP-instance, and hence the associated fitness function f, to be integer-valued, each adaptive step in the fitness graph increases fitness by at least one, so there can be at most $\text{span}(\mathcal{C})$ many such steps in any path. □

Note the similarity in the proof of Proposition 5.2 and the proofs of Proposition 4.29 and Theorem 4.30. In fact, I could have gotten Proposition 5.2 from Proposition 4.29 by noting that for an integer VCSP, $f_\delta \geq 1$ and $f_{\max} \leq \text{span}(\mathcal{C})$. Even more simply, the above results could be restated as: "if a fitness landscape only has a few fitness values and each step increases the fitness value then there can only be a few steps" – and various more complicated evolutionary dynamics are often analyzed in great details on these sort of simple low-span fitness landscapes [35].

The key difference that I want to focus on in this section is that we will express span results (and, later in Section 5.2, encouragement results) in terms of properties of the gene-interaction network (i.e., VCSP-instance) that represents a fitness graph, instead of in terms of the direct properties of fitness landscapes themselves. By doing this, I am aiming to classify fitness landscapes as easy or hard (i.e., to build a hardness dichotomy) based on properties of their representations as gene-interaction networks. The long-term research goal is that we could carry out a process like the following (a fitness landscape version of the game assay that I develop in Chapter 7):

1. measure actual gene-interaction networks from local fitness landscapes (using a procedure like the one in Section 3.7.1) then

2. look at the measured gene-interaction networks and

3. determine from this representation if the corresponding fitness landscapes has short or long adaptive paths – or, more generally, if we should expect evolution to quickly find a fitness peak or not.

Unfortunately, without further refinement, the span of a gene-interaction network corresponding to a fitness landscape is not always informative.

Example 5.3. *Consider the smooth fitness landscape implemented by the following unary components:* $\mathcal{C} = \{C_{\{i\}} \mid C_{\{i\}} = \begin{pmatrix} 0 \\ 2^i \end{pmatrix}\}$*. This fitness landscape is smooth and thus has a longest adaptive path of length* n*, but* $\text{span}(\mathcal{C}) = 2^n$*.*

To avoid a large discrepancy between the longest adaptive path and span and thus obtain a tighter bound, we can consider sign-equivalent VCSP-instances that have a smaller span. In the case of Example 5.3, a sign-equivalent minimal span VCSP would be given by $\mathcal{C}' = \{C_{\{i\}} \mid C_{\{i\}} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}\}$, which has $\text{span}(\mathcal{C}') = n$ – giving us a tight bound on the longest adaptive path. In general, if we do not restrict the arity of the constraints then we can always find a sign-equivalent instance where the length of the longest path in the fitness graph is exactly equal to the span:

Proposition 5.4. *Given a VCSP-instance* $\mathcal{C}$*, on* n *variables, there exists a sign-equivalent VCSP-instance* $\mathcal{C}'$ *of arity* n *such that the length of the longest directed path in the associated fitness graph is exactly equal to* $\text{span}(\mathcal{C}')$*.*

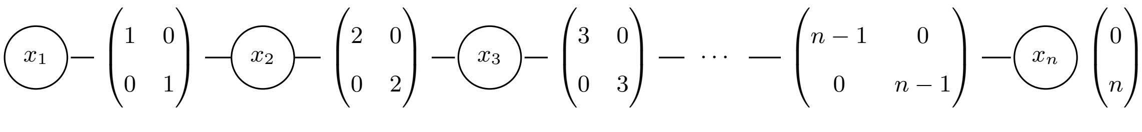
Proof. With an n-ary constraint, we are free to assign arbitrary fitness values to any variable assignment. So just consider the fitness graph $G_{\mathcal{C}}$ as a poset. Let M be the maximum depth of this poset – this is equal to the length of the longest directed path in $G_{\mathcal{C}}$. Let $\mathcal{C}' = \{C_{[n]}\}$, where $C_{[n]}$ is the n-ary constraint where each variable assignment is given the fitness value of M minus its depth in the poset. □

The technique used in the proof of Proposition 5.4 will generate constraints of very high arity. In the above smooth example, however, we had a tight bound while maintaining arity 1. So even if we restrict the arity of the constraints, then we can still obtain useful bounds in some cases by showing that there exists a sign-equivalent VCSP-instance with a small span, as I will show in Section 5.1.1. But these span arguments are also limited and cannot always produce tight bounds with low arity, as I will show in Section 5.1.2.

5.1.1 Quadratic span for gene-interaction networks of degree ≤ 2

In general, there are landscapes other than smooth ones and VCSP instances that contain constraints of arity-n where span arguments can still be useful:

Example 5.5. (Path of length $\binom{n}{2}+n$) *Consider the binary Boolean VCSP-instance $\mathcal{C}$:*


x_1
$\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$
x_2
$\begin{pmatrix}2 & 0\\ 0 & 2\end{pmatrix}$
x_3
$\begin{pmatrix}3 & 0\\ 0 & 3\end{pmatrix}$
$\cdots$
$\begin{pmatrix}n-1 & 0\\ 0 & n-1\end{pmatrix}$
x_n
$\begin{pmatrix}0\\ n\end{pmatrix}$

To obtain a path of length $\binom{n}{2}+n$ in the corresponding fitness graph $G_{\mathcal{C}}$, consider an initial variable assignment of $x = (10)^{\frac{n}{2}}$ if n is even and $x = (10)^{\frac{n-1}{2}}1$ if n is odd, and always select the leftmost variable that is able to flip. This will increase the fitness by 1 at each step, starting from 0 to $\text{span}(\mathcal{C}) = 1 + 2 + \cdot + n = \frac{n(n+1)}{2}$.

For example, when $n = 4$, this gives the following sequence of eleven assignments, each of which increases the value of the fitness function by one:

$$1010 \rightarrow 0010 \rightarrow 0110 \rightarrow 1110 \rightarrow 1100 \rightarrow 1000 \rightarrow 0000 \rightarrow 0001 \rightarrow 0011 \rightarrow 0111 \rightarrow 1111 \tag{5.1}$$

Unfortunately, not all VCSPs will make it as obvious as Example 5.5 that a span argument can be used to bound the length of the longest adaptive path. But there is a span argument that gives us a bound on all cases like Example 5.5 – in the rest of this section, I will prove the following:

Theorem 5.6. *All binary Boolean VCSPs with constraint graphs of degree ≤ 2 have a minimal span of $O(n^2)$ and thus every adaptive path in their fitness graphs has at most $O(n^2)$ steps.*

To find useful span bounds, it is helpful to define a procedure for picking out a sign-equivalent VCSP of minimal span. This can be framed as an optimization problem on a 'lifted' or 'meta' CSP corresponding to a given VCSP instance. To define the lifted CSP, I need to introduce two new kinds of constraints:

Definition 5.7. Given two sets of variables L and R, let $\leq_{+k} [L, R]$ be a *meta-constraint* of arity $|L| + |R|$ that is satisfied when $k + \sum_{x \in L} x \leq \sum_{y \in R} y$. Call L the *left side* of the meta-constraint and R the *right side*. Similarly, define $=_{+k} [L, R]$ as above but with $\leq$ replaced by $=$.

Using the above constraints, we can define a lifted-CSP – a generalization of the linear program that Poljak [180] used to analyze MAX-CUT – as follows:

Definition 5.8. Given a simple binary boolean VCSP-instance $\mathcal{C}$ on n loci (with a constraint graph that has neighbourhood function $N : [n] \to 2^{[n]}$), the corresponding lifted-CSP instance $\mathcal{D}$ has $|\mathcal{C}|$ many meta-variables $\{p_i | C_{\{i\}} \in \mathcal{C}\}$ and $\{p_{ij} | C_{\{i,j\}} \in \mathcal{C}\}$ with domains of $\mathbb{N}$. Divide the meta-variables V into two sets V_+, V_- with $p_i \in V_+$ or $p_{ij} \in V_+$ if $c_i > 0$ or $c_{ij} > 0$ and otherwise $p_i \in V_-$ or $p_{ij} \in V_-$ if $c_i < 0$ or $c_{ij} < 0$.

For each locus $i \in [n]$ in the original VCSP, make $2^{|N(i)|}$ many meta-constraints, one for each $Y \subseteq N(i)$ depending on the sign of $s = c_i + \sum_{j \in Y} c_{ij}$:

- If $s < 0$ then add the constraint $\leq_{+1} [(Y \cup \{i\}) \cap V_+, (Y \cup \{i\}) \cap V_-]$, else
- If $s = 0$ then add the constraint $=_{+0} [(Y \cup \{i\}) \cap V_+, (Y \cup \{i\}) \cap V_-]$, else
- If $s > 0$ then add the constraint $\leq_{+1} [(Y \cup \{i\}) \cap V_-, (Y \cup \{i\}) \cap V_+]$.

Note that the lifted-CSP has at least one satisfying assignment given by $p_i = |c_i|$ and $p_{ij} = |c_{ij}|$ (i.e., the absolute value of the magnitude of the original VCSP constraints).

The resultant lifted-CSP is complicated, but we can use various local consistency conditions to make it simpler by pruning the meta-variables corresponding to unary constraints:

Proposition 5.9. *A lifted-CSP instance $\mathcal{D}$ can be pruned of all meta-variables that correspond to unary constraints in the original VCSP instance $\mathcal{C}$ in such a way that:*

- *the pruned lifted-CSP $\mathcal{D}'$ will only have meta-constraints of the types $=_0$ and $\leq_{\{+1,2\}}$.*
- *for any meta-constraint $D \in \mathcal{D}'$ there will be a corresponding locus i in $\mathcal{C}$ such that any meta-variables in the scope of D will be of the form p_{ik} for some $k \in N_{\mathcal{C}}(i)$.*
- *any satisfying assignment to $\mathcal{D}'$ can be converted to a satisfying assignment to $\mathcal{D}$.*

Proof. Let us consider the locus i in the VCSP, the corresponding meta-variable p_i in the lifted-CSP, and its 'neighbouring' meta-variables $V_i = \{p_{ij} \mid j \in N(i)\}$ which we will divide into two sets S and T based on if the constraints that the meta-variables correspond to agree or not on sign with c_i. Specifically:

- if $c_i > 0$ then let $S = (V_i \cap V_+) \cup \{p_i\}$ and $T = V_i \cap V_-$, otherwise
- if $c_i < 0$ then let $S = (V_i \cap V_-) \cup \{p_i\}$ and $T = V_i \cap V_+$.

This is a useful division because the left hand side of every meta-constraint involving p_i will be a subset of one of the sign sets S or T and the right hand side will be a subset of the other sign set.

Order all subsets of T by the sum of their elements, and split this order into three sets:

$$\mathcal{B} = \{X \mid X \subseteq T \text{ and } p_i > \sum_{p_{ij} \in X} p_{ij}\} \tag{5.2}$$

$$\mathcal{E} = \{X \mid X \subseteq T \text{ and } p_i = \sum_{p_{ij} \in X} p_{ij}\} \tag{5.3}$$

$$\mathcal{U} = \{X \mid X \subseteq T \text{ and } p_i < \sum_{p_{ij} \in X} p_{ij}\} \tag{5.4}$$

There are three possibilities:

1. If $|\mathcal{E}| = |\mathcal{U}| = 0$ then all meta-variables in V_i can be eliminated from the lifted-CSP (since if we restored them then they can all be satisfied by setting $p_i \leftarrow 1 + \sum_{p_{ij} \in T} p_{ij}$).
2. If $|\mathcal{E}| > 0$ then
   (a) select any $E \in \mathcal{E}$ as a distinguished member;
   (b) for any other $E' \in \mathcal{E} - \{E\}$, add the meta-constraint $=_0 [E, E']$;
   (c) for every other existing meta-constraint $\leq_{+1} [L, R]$ or $=_0 [L, R]$ corresponding to locus i, we must have $p_i \in L$ or $p_i \in R$, if $p_i \in L \subseteq S$ then replace L by $(L - \{p_i\}) \cup E$ otherwise if $p_i \in R \subseteq S$ then replace R by $(R - \{p_i\}) \cup E$. Note that since $E \subseteq T$, there will be no collisions in the unions.
3. If $|\mathcal{E}| = 0$, $|\mathcal{U}| > 0$ then
   (a) for every pair $B \in \mathcal{B}$ and $U \in \mathcal{U}$, add the meta-constraint $\leq_{+2} [B, U]$,
   (b) for every other existing meta-constraint corresponding to locus i, if it is of the form $\leq_{+1} [L, R]$ then
      - if $p_i \in L \subseteq S$ we will remove the existing meta-constraint and create $|\mathcal{B}|$ new ones: for each $B \in \mathcal{B}$, add the meta-constraint $\leq_{+2} [B \cup (L - \{p_i\}), R]$;
      - if $p_i \in R \subseteq S$ we will remove the existing meta-constraint and create $|\mathcal{U}|$ new ones: for each $U \in \mathcal{U}$, add the meta-constraint $\leq_{+2} [L, U \cup (R - \{p_i\})]$.

   As before, note that since $B, U \subseteq T$, there will be no collisions in the unions. Finally,

   (c) suppose there are m equality constraints: $=_0 [L_j, R_j]$ for $1 \leq j \leq m$ with $p_i \in L_j$. Create
      i. $\frac{m(m-1)}{2}$ new equality constraints for each pair $1 \leq j \leq l \leq m$ add $=_0 [(L_j - \{p_i\}) \cup R_l, (L_l - \{p_i\}) \cup R_j]$ and

ii. $m(|\mathcal{B}| + |\mathcal{U}|)$ new constraints; for each $B \in \mathcal{B}$ add $\leq_{+1}$ $[(L_j - \{p + i\}) \cup B, R_j]$ and for each $U \in \mathcal{U}$ add $\leq_{+1}$ $[R_j, (L_j - \{p_i\}) \cup U\}$.

If any meta-constraint created above has the same meta-variable in both its left and right hand side then that meta-variable can be removed from both sides of that meta-constraint.

This leaves us with meta-variables in the lifted CSP corresponding only to the binary constraints in the original VCSP. Further the meta-constraints are only between meta-variables whose binary constraints share a vertex. □

In particular, Proposition 5.9 means that if our VCSP had degree ≤ 2 then the corresponding pruned lifted-CSP will have binary meta-constraints and degree ≤ 2. We can orient these edges in the meta-constraint graph as follows:

- $\leq_{\{+1,2\}}$ $[\{p_{ij}\}, \{p_{kl}\}]$ means a directed edge $p_{ij} \to p_{kl}$,
- $\leq_{\{+1,2\}}$ $[, \{p_{ij}, p_{kl}\}]$ means no edge between p_{ij} and p_{kl}, and
- $=_0$ $[\{p_{ij}\}, \{p_{kl}\}]$ means an undirected edge p_{ij} — p_{kl}.

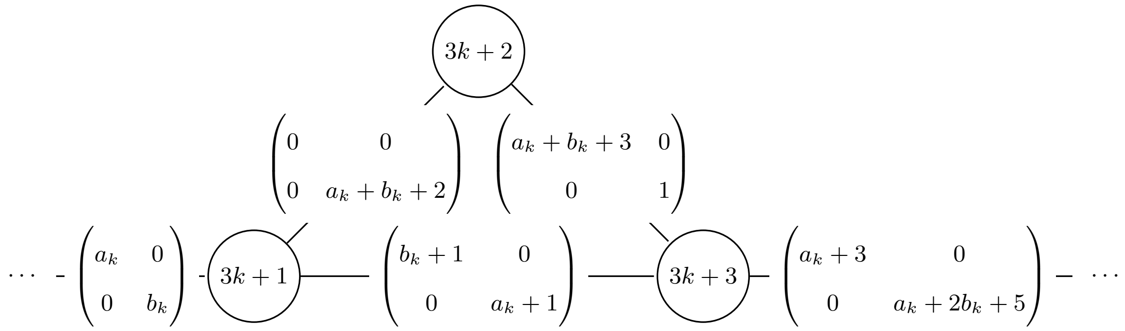
Given that the initial VCSP instance constraint weights satisfy the lifted-CSP, it means there cannot be any directed cycles in the meta-constraint graph. Thus, the meta-constraint graph is a union of undirected cycles and paths that can mix both directed and undirected edges. All cycles of undirected edges can be satisfied by setting their meta-variables to 1. To satisfy the paths, we start by setting the sources to $0, 1$ or 2 and then following the edges to increment the subsequent meta-variable by $0, 1$ or 2 depending on if the meta-constraint leading to it was $=_0$, $\leq_{+1}$ or $\leq_{+2}$, respectively.

Since there are at most n meta-variables, this means that the largest assigned value will be at most $2n$; and the largest sum of the meta-variables will be $< n(n+1)$. If we reconstruct the unary constraints in the VCSP, this means that the span of the corresponding VCSP can be bounded by $2n(n+1)$. Thus, binary Boolean VCSPs with constraint graphs of degree ≤ 2 produce fitness landscapes where every adaptive path is shorter than $2n(n+1)$ steps. This completes the proof of Theorem 5.6.

5.1.2 Limits to the span argument

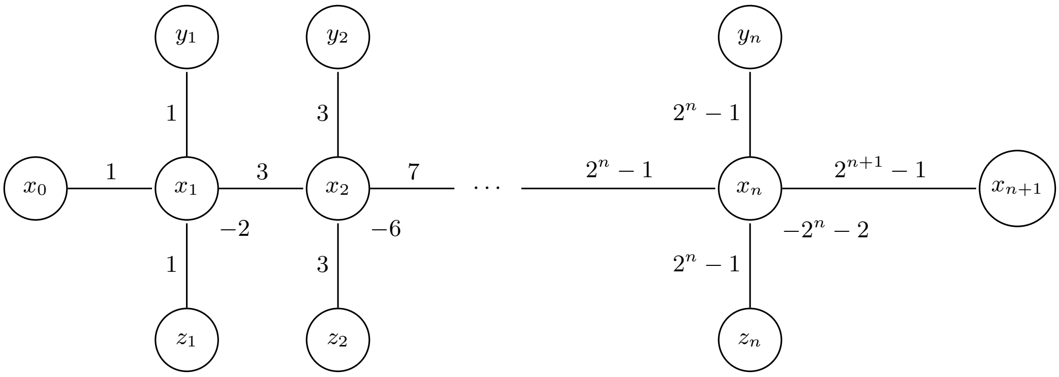
Span arguments like the one above could be extended slightly to trees of degree ≤ 3, but no further. In this section, I will give examples of VCSPs with only short adaptive walks but exponential minimal span.

Example 5.10. (Large span in constraint graph of degree 3**)** *Consider a binary Boolean VCSP-instance $\mathcal{C}$ with $n = 3K + 1$ variables. The constraint graph contains a sequence of disjoint cycles of length three, linked together by a single additional edge joining each consecutive pair of cycles. The sequence of cycles is started by a single node. Hence the constraint graph of $\mathcal{C}$ has maximum degree three, circumference three, and treewidth two. The k-th cycle (for $0 \leq k \leq K-1$) has the following constraints:*


$3k+2$
$\begin{pmatrix} 0 & 0 \\ 0 & a_k + b_k + 2 \end{pmatrix}$
$\begin{pmatrix} a_k + b_k + 3 & 0 \\ 0 & 1 \end{pmatrix}$
$\cdots$
$\begin{pmatrix} a_k & 0 \\ 0 & b_k \end{pmatrix}$
$3k+1$
$\begin{pmatrix} b_k + 1 & 0 \\ 0 & a_k + 1 \end{pmatrix}$
$3k+3$
$\begin{pmatrix} a_k + 3 & 0 \\ 0 & a_k + 2b_k + 5 \end{pmatrix}$
$\cdots$

where a_k and b_k are defined recursively from one cycle to the next with $a_1 = b_1 = 1$ and $a_{k+1} = a_k + 3$ and $b_{k+1} = a_k + 2b_k + 5$.

Example 5.11. (Large span in tree-structured constraint graph of degree 4**)** *Consider the simple binary Boolean VCSP-instance $\mathcal{C}$:*



the constraint graph of $\mathcal{C}$ is tree-structured and has a maximum degree of 4.

It is straightforward to verify that any VCSP that preserves the constraint graphs and corresponding fitness landscapes in these examples must have exponential span. But it is important to assure ourselves that a lower span cannot be achieved by adding superfluous constraints. After all, Proposition 5.4 shows that increasing arity can be used to reduce span, so one might expect that adding extra constraints and thus extra degrees of freedom might help reduce span. In the rest of this section, I establish that this does not help.

The main step to showing that span has to be high by lower bounding minimal span is to link to our concepts of magnitude (Definition 3.14; simple) and sign (Definition 3.20; trim) minimal VCSPs. This well let us reduce the graph structures that we have to consider when showing that no low span VCSP exists for a given fitness graph. More formally:

Theorem 5.12. *Let $\mathcal{C}'$ be the binary Boolean VCSP-instance of minimal span that implements a fitness graph G then there exists a simple trim binary Boolean VCSP-instance $\mathcal{C}$ that implements the same fitness graph with* $\text{span}(\mathcal{C}') \leq \text{span}(\mathcal{C}) \leq 4\ \text{span}(\mathcal{C}')$.

I establish Theorem 5.12 by showing that if we start with $\mathcal{C}'$ and them simplify it and trim it to get a minimal (in the sense of Chapter 3) sign-equivalent VCSP $\mathcal{C}$ then that VCSP will increase the span by at most a factor of 4. This proceeds by two steps, in Proposition 5.13 I establish that simplifying $\mathcal{C}'$ to a minimal magnitude-equivalent VCSP increases span by at most a factor of 4 and in Proposition 5.14 I establish that trimming the simple VCSP does not increase span.

Proposition 5.13. *For a binary Boolean constraint graph $\mathcal{C}'$, the corresponding simple (i.e., minimal magnitude-equivalent) constraint graph $\mathcal{C}$ has* $\text{span}(\mathcal{C}) \leq 4\,\text{span}(\mathcal{C}')$.

Proof. Let us decompose span into the contribution due to unaries (span_1) and binaries (span_2):

$$\text{span}(\mathcal{C}) = \overbrace{\sum_{i \in [n]} |c_i|}^{\text{span}_1(\mathcal{C})} + \overbrace{\sum_{ij \in E(\mathcal{C})} |c_{ij}|}^{\text{span}_2(\mathcal{C})} \tag{5.5}$$

Using Equations 3.6 from the proof of Theorem 3.15, we can express $|c_i|$ in terms of $\mathcal{C}'$ as:

$$\begin{aligned}
|c_i| &= |\ C'_i[1] - C'_i[0] \quad + \sum_{j \mid ij \in E(\mathcal{C}')} C'_{ij}[1,0] - C'_{ij}[0,0]\ | && (5.6)\\
&\leq |\ C'_i[1] - C'_i[0]\ | \quad + \sum_{j \mid ij \in E(\mathcal{C}')} |\ C'_{ij}[1,0] - C'_{ij}[0,0]\ | && (5.7)\\
&\leq \text{span}(C'_i) + \sum_{j \mid ij \in E(\mathcal{C}')} \text{span}(C'_{ij}) && (5.8)\\
&= \text{span}(C'_i) + \text{span}(\{C'_{ij} \mid j \in N_{\mathcal{C}'}(i)\}) && (5.9)
\end{aligned}$$

Notice that the first span in Equation 5.9 is of the unary constraint in $\mathcal{C}'$ that has i as its scope and the second span is of all binary constraints in $\mathcal{C}'$ that have i in scope (or, equivalently: span of all edges incident on i in the constraint graph of $\mathcal{C}'$). This means that if we sum $|c_i|$ over all $i \in [n]$ then we cover the whole graph:

$$\begin{aligned}
\text{span}_1(\mathcal{C}) = \sum_{i=1}^{n} |c_i| \leq \sum_{i=1}^{n} \text{span}(C'_i) + \sum_{i=1}^{n} \text{span}(\{C'_{ij} \mid j \in N_{\mathcal{C}'}(i)\}) && (5.10)\\
= \text{span}_1(\mathcal{C}') + 2\ \text{span}_2(\mathcal{C}') && (5.11)\\
\leq 2\ \text{span}(\mathcal{C}') && (5.12)
\end{aligned}$$

where Equation 5.11 has a double cover in its second summand because each edge in the constraint graph of $\mathcal{C}'$ has two end points (equivalently: all scopes are binary).

Similarly, using Equations 3.7 from the proof of Theorem 3.15, we can also express $|c_{ij}|$ in terms of $\mathcal{C}'$ as:

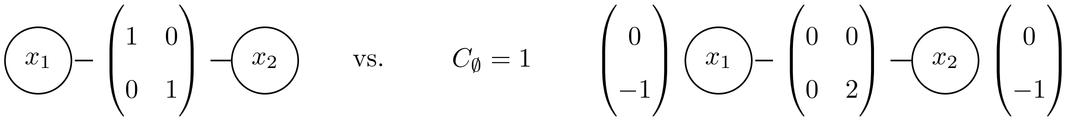
$$\begin{aligned}
|c_{ij}| = |C'_{ij}[0,0] - C'_{ij}[0,1] - C'_{ij}[1,0] + C'_{ij}[1,1]\ | && (5.13)\\
\leq |\ C'_{ij}[1,1] - C'_{ij}[1,0]\ | + |\ C'_{ij}[0,0] - C'_{ij}[0,1]\ | && (5.14)\\
\leq 2\ \text{span}(C'_{ij}) && (5.15)
\end{aligned}$$

As before, if we sum $|c_{ij}|$ over all $ij \in E(\mathcal{C})$ then we cover the whole graph:

$$\text{span}_2(\mathcal{C}) = \sum_{ij \in E(\mathcal{C})} |c_{ij}| \leq 2 \sum_{ij \in E(\mathcal{C}')} \text{span}(C'_{ij}) \leq 2\ \text{span}_2(\mathcal{C}') \qquad (5.16)$$

where for the last inequality we moved from summing over $ij \in E(\mathcal{C})$ to $ij \in E(\mathcal{C}')$ because $E(\mathcal{C}) \subseteq E(\mathcal{C}')$ by Theorem 3.16. Combining Equations 5.12 and 5.16, we get the final result that $\text{span}(\mathcal{C}) = \text{span}_1(\mathcal{C}) + \text{span}_2(\mathcal{C}) \leq 4\ \text{span}(\mathcal{C}')$. □

Note that proposition 5.13 is the best possible since the following two constraint graphs are magnitude equivalent:


x_1
$\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$
x_2
vs.
$C_\emptyset = 1$
$\begin{pmatrix}0\\ -1\end{pmatrix}$
x_1
$\begin{pmatrix}0 & 0\\ 0 & 2\end{pmatrix}$
x_2
$\begin{pmatrix}0\\ -1\end{pmatrix}$

with the constraint graph on the left having a span of 1 and the simplified constraint graph on the right having a span of $1 + 2 + 1 = 4$. Thus, sometimes the simplifying procedure from section 3.7 can increase span by a factor of 4. In contrast, the trimming procedure from section 3.8 can only decrease span:

Proposition 5.14. *For a simple binary boolean constraint graph $\mathcal{C}'$, the corresponding minimal sign-equivalent constraint graph $\mathcal{C}$ made by our trimming procedure has* $\text{span}(\mathcal{C}) \leq \text{span}(\mathcal{C}')$*.*

Proof. The trimming procedure from section 3.8 only removes constraints but doesn't change any remaining ones. □

By combining Propositions 5.13 and 5.14, I show that a minimal span implementation of any given fitness graph has a high span if and only if the simple trim constraint graph has high minimal span. Any more subtle changes to the constraint graph and values can only decrease the lowest span achieved in this more restricted way by a factor of 4. This establishes Theorem 5.12.

5.2 Tree-structured Boolean VCSP-instances

In this section, I will prove the following:

Theorem 5.15. *For a binary Boolean VCSP instance $\mathcal{C}$ on n variables, if the constraint-graph of $\mathcal{C}$ is a tree, then any directed path in the associated fitness graph $G_{\mathcal{C}}$ has length at most $\binom{n}{2} + n$.*

Note that this result bounds the length of *any* adaptive path in $G_{\mathcal{C}}$, not just the path taken by a particular local search algorithm or local adaptive dynamics. Thus, on such landscapes even choosing the worst possible sequence of improving moves results in a local optimum being found in polynomial time.

I will show in Section 5.3 that the conditions of being Boolean and tree-structured are essential to obtain a polynomial bound on the length of all paths. To see that the bound in Theorem 5.15 is the best possible for binary Boolean tree-structured VCSP-instances, consider the path-structured gene-interaction network on n loci from Example 5.5 that has adaptive paths of length $\binom{n}{2} + n$.

For the proof of Theorem 5.15, we introduce some further definitions.

Definition 5.16. Given any directed path $p = x^1 \dots x^t \dots x^T$ in a fitness graph G, define the *flip function* as $m(t) = (i \mapsto b)$ where $x^{t+1} \oplus x^t = e_i$ and $b = x_i^{t+1}$ (i.e., the i-th variable is flipped at time t to value b).

For illustration the above definition, recall the $n = 4$ case of Example 5.5 with an adaptive path of $10 = \binom{4}{2} + 4$ fixations and 11 genotypes from Equation 5.1:

$$1010 \to 0010 \to 0110 \to 1110 \to 1100 \to 1000 \to 0000 \to 0001 \to 0011 \to 0111 \to 1111 \tag{5.1}$$

The above adaptive path corresponds to the following flip function:

t	1	2	3	4	5	6	7	8	9	10
m(t)	$1 \mapsto 0$	$2 \mapsto 1$	$1 \mapsto 1$	$3 \mapsto 0$	$2 \mapsto 0$	$1 \mapsto 0$	$4 \mapsto 1$	$3 \mapsto 1$	$2 \mapsto 1$	$1 \mapsto 1$

To obtain the bound on the length of paths given in Theorem 5.15, I will identify a structure in the flip function to bound the maximum possible value for T. This requires a few more definitions:

Definition 5.17. We say that a flip $m(t') = (j \mapsto c)$ **supports** a flip $m(t) = (i \mapsto b)$ if $t' < t$ and $C_{ij}(b,c) - C_{ij}(\overline{b},c) > C_{ij}(b,\overline{c}) - C_{ij}(\overline{b},\overline{c})$; if $x^t_j = c$, then the support is said to be **strong**.

It is useful to note that the inequality on C_{ij} is symmetric in the sense that:

$$\begin{aligned} & C_{ij}(b,c) - C_{ij}(\overline{b},c) > C_{ij}(b,\overline{c}) - C_{ij}(\overline{b},\overline{c}) \\ \Leftrightarrow\ & C_{ij}(b,c) - C_{ij}(b,\overline{c}) > C_{ij}(\overline{b},c) - C_{ij}(\overline{b},\overline{c}) \\ \Leftrightarrow\ & C_{ji}(c,b) - C_{ji}(\overline{c},b) > C_{ji}(c,\overline{b}) - C_{ji}(\overline{c},\overline{b}) \end{aligned} \tag{5.17}$$

Due in part to this symmetry, the definition of encouragement interacts well with the following notion:

Definition 5.18. Given a binary Boolean VCSP-instance $\mathcal{C}$ implementing fitness function f, the **fitness contribution** of the variable at position i in assignment x, restricted to $S \subseteq [n]$ is defined to be:

$$f^S_i(b|x) = \left\{ \begin{matrix} C_i(b) & \text{if } i \in S \\ 0 & \text{otherwise} \end{matrix} \right\} + \sum_{j \in N_{\mathcal{C}}(i) \cap S} C_{ij}(b, x_j) \tag{5.18}$$

if $S = [n]$ then we just write f_i rather than $f^{[n]}_i$.

Note that for any path p in G, if $m(t) = (i \mapsto b)$ then $f_i(b|x^t) > f_i(\overline{b}|x^t)$.

I now introduce an *encouragement* relation between a flip and its most recent strong supporting flip, if there is one:

Definition 5.19. We say that a flip $m(t) = (i \mapsto b)$ is ***encouraged*** by its most recent strong supporting flip $m(t') = (j \mapsto c)$, and write $(t', j \mapsto c) \leftarrow (t, i \mapsto b)$. If there are no strong supporting flips, then we say that a flip $m(t) = (i \mapsto b)$ is ***courageous***, and write $\perp \leftarrow (t, i \mapsto b)$.

Note that if $(t', j \mapsto c) \leftarrow (t, i \mapsto b)$, then $t' < t$ and $i \in N_{\mathcal{C}}(j)$.

For illustration, consider again the sequence of moves listed in Equation 5.1 of Example 5.5. It corresponds to the following encouragement relation, which I will call the *encouragement graph*:

$$\begin{array}{lllll}
\bot & \leftarrow (1,1 \mapsto 0) & & \bot \;\leftarrow (2,2 \mapsto 1) & \leftarrow (3,1 \mapsto 1) \\
\bot & \leftarrow (4,3 \mapsto 0) & \leftarrow (5,2 \mapsto 0) & \leftarrow (6,1 \mapsto 0) & \\
\bot & \leftarrow (7,4 \mapsto 1) & \leftarrow (8,3 \mapsto 1) & \leftarrow (9,2 \mapsto 1) & \leftarrow (10,1 \mapsto 1)
\end{array}$$

Proposition 5.20. *If $(t_1, j \mapsto c) \leftarrow (t_2, i \mapsto b)$ (or if $\bot \leftarrow (t_2, i \mapsto b)$, set $t_1 = 0$) then for all $t_1 < t' \leq t_2$ we have $f_i(b|x^{t'}) - f_i(\overline{b}|x^{t'}) \geq f_i(b|x^{t_2}) - f_i(\overline{b}|x^{t_2}) > 0$.*

Proof. Define the set of temporary supports S_w as the set of positions of flips after t_1 that supported $(t_2, i \mapsto b)$ but were not strong (i.e., they were flipped back by the time we got to t_2: for supportive $(t'', k \mapsto a)$ with $t'' > t_1$ we have $k \in S_w \Rightarrow a \neq x^{t_2}[k]$).

Consider any flip $m(t') = (k \mapsto a)$ for $t' \in [t_1 + 1, t_2 - 1]$. Since it either didn't support $(t_2, i \mapsto b)$ (and so had $C_{ij}(b, a) - C_{ij}(b, \overline{a}) \leq C_{ij}(\overline{b}, a) - C_{ij}(\overline{b}, \overline{a})$ by Equation 5.17) or was a temporary support, we have that:

$$f_i^{[n]-S_w}(b|x^{t'+1}) - f_i^{[n]-S_w}(\overline{b}|x^{t'+1}) \leq f_i^{[n]-S_w}(b|x^{t'}) - f_i^{[n]-S_w}(\overline{b}|x^{t'}) \tag{5.19}$$

Thus $\delta_i(t') = f_i^{[n]-S_w}(b|x^{t'}) - f_i^{[n]-S_w}(\overline{b}|x^{t'})$ is monotonically non-increasing in t' over the time interval $[t_1 + 1, t_2]$. So:

$$f_i^{[n]-S_w}(b|x^{t'}) - f_i^{[n]-S_w}(\overline{b}|x^{t'}) \geq f_i^{[n]-S_w}(b|x^{t_2}) - f_i^{[n]-S_w}(\overline{b}|x^{t_2}) \tag{5.20}$$

Since every position $k \in S_w$ supported $(t_2, i \mapsto b)$ but is absent in x^{t_2}, we must have $f_i^{S_w}(b|x^{t_2}) - f_i^{S_w}(\overline{b}|x^{t_2}) \leq f_i^{S_w}(b|x^{t'}) - f_i^{S_w}(\overline{b}|x^{t'})$. Noting that $f_i = f_i^{[n]-S_w} + f_i^{S_w}$ then lets us combine this with Equation 5.20 (and the fact that $f_i(b|x^{t_2}) > f_i(\overline{b}|x^{t_2})$) to complete the proposition. □

By Definition 5.19, each flip can only be encouraged by at most one other flip, so each node in the encouragement graph has out-degree at most one. Directed graphs where each vertex has at most one parent are forests, so the encouragement graph is a forest. This forest has a component for each courageous flip, and we will now show that there are at most n of these:

Proposition 5.21. *At each variable position i, only the first flip can be courageous.*

Proof. Consider a courageous flip $\bot \leftarrow (t, i \mapsto b)$, by Proposition 5.20, we know that for all $t' < t$: $f_i(b|x^{t'}) - f_i(\overline{b}|x^{t'}) \geq f_i(b|x^{t}) - f_i(\overline{b}|x^{t}) > 0$. Thus, there is no time $t' \leq t$ such that i could have flipped to $\overline{b}$: hence i was always at $\overline{b}$ for $t' \leq t$. So the courageous flip had to be the first flip at that position. □

I will now prove that an encouragement tree cannot double-back on itself in position (Proposition 5.22), and that every branch is a branch in position (Proposition 5.23). When the constraint graph is itself a tree, this will imply that each tree in the encouragement forest is a sub-tree of the constraint graph.

Proposition 5.22. *If $(t_1, i \mapsto a) \leftarrow (t_2, j \mapsto b) \leftarrow (t_3, k \mapsto c)$ then $i \neq k$.*

Proof. Since $(t_1, i \mapsto a)$ strongly supported $(t_2, j \mapsto b)$, we have $x_i^{t_2} = a$. If, for the sake of contradiction, we assume that $i = k$ then $a = c$ (because if we had $c = \overline{a}$ then the two encouragements would force a contradiction via clashing Equations 5.17) and by Proposition 5.20: $f_i(a|x^{t'}) - f_i(\overline{a}|x^{t'}) \geq f_i(a|x^{t_3}) - f_i(\overline{a}|x^{t_3}) \geq 0$ for all $t_2 < t' \leq t_3$. But this means that i cannot be flipped to $\overline{a}$ and thus $m(t_3) = (i, a)$ is not a legal flip. This is a contradiction and so $i \neq k$. □

Proposition 5.23. *For all i, j and $t_1 < t_2 \leq t_3$: if $(t_1, i \mapsto a) \leftarrow (t_2, j \mapsto b)$ and $(t_1, i \mapsto a) \leftarrow (t_3, j \mapsto c)$, then $t_2 = t_3$.*

Proof. From Proposition 5.20, we can see that for all $t' \in [t_1 + 1, t_3]$, $f_j(c|x^{t'}) - f_j(\overline{c}|x^{t'}) > 0$, so $b = c$ and j couldn't have flipped from c to $\overline{c}$ between t_2 and t_3. Thus, for $(t_2, j \mapsto c)$ to be a legal flip, we must have $t_2 = t_3$. □

Now, if we look along the arrows then each flip in p is the start of a path of encouraged-by links that ends at one of the n courageous flips.

One final case to exclude is that there might be two encouragement paths that go in the opposite direction over the same positions. This cannot happen:

Proposition 5.24. *Having both of the following encouragement paths is impossible:*

$$\bot \leftarrow (t_1, i_1 \mapsto b_1) \leftarrow (t_2, i_2 \mapsto b_2) \qquad \leftarrow \cdots \leftarrow \quad (t_m, i_m \mapsto b_m) \tag{5.21}$$

$$\bot \leftarrow (s_m, i_m \mapsto c_m) \leftarrow (s_{m-1}, i_{m-1} \mapsto c_{m-1}) \qquad \leftarrow \cdots \leftarrow \quad (s_1, i_1 \mapsto c_1) \tag{5.22}$$

Proof. Without loss of generality (by relabeling), we can assume that $t_1 < s_1$. We can extend this with the following claim:

Claim: If $t_k < s_k$ then $t_{k+1} < s_{k+1}$

Since $(t_k, i_k \mapsto b_k) \leftarrow (t_{k+1}, i_{k+1} \mapsto b_{k+1})$, we have, for all $t \in [t_k + 1, t_{k+1}]$, $x^t[i_k] = b_k$. Thus we can't have i_k flipping in that interval, so $s_k > t_{k+1}$.

But now look at $(s_{k+1}, i_{k+1} \mapsto c_{k+1}) \leftarrow (s_k, i_k \mapsto c_k)$. This shows that we also have, for all $t' \in [s_{k+1} + 1, s_k]$, $x^{t'}[i_{k+1}] = c_{k+1}$. So for both flips at i_{k+1} to happen, we need $s_{k+1} > t_{k+1}$.

Applying the claim repeatedly gets us $t_m < s_m$. But this means that i_m flipped before $m(s_m)$, so by Proposition 5.21 $(s_m, i_m \mapsto c_m)$ could not have been courageous. □

This means that it is sufficient to simply count the number of undirected paths in the encouragement trees. I can now pull all the results together to complete the proof.

Proof of ***Theorem 5.15****.* Consider any path p in the fitness graph, and its corresponding flip function m. By the completeness of Definition 5.19, we know that every flip must have been either courageous or encouraged.

Any encouraged flip is the end-point of a unique (non-zero length) encouragement path in the constraint graph starting from some courageous flip (where Proposition 5.22 established that they're encouragement paths, not walks; and Proposition 5.23 established that the encouragement paths are uniquely determined by the variable positions that they pass through.) From Proposition 5.24, we know that there cannot be two encouragement paths that traverse the same positions but in opposite directions. Thus, there can only be as many non-zero-length encouragement paths as undirected paths in our constraint graph. Since our constraint graph is a tree, an undirected non-zero length path is uniquely determined by its pair of endpoints. Thus, there are at most $\binom{n}{2}$ of these paths.

From Proposition 5.21, there are at most n courageous flips (encouragement paths of length 0). Thus, our path p must have length at most $n + \binom{n}{2}$. □

5.3 Long paths in landscapes with simple constraint graphs

In this section, I show that the conditions in Theorem 5.15 are essential. I exhibit binary VCSP-instances with very simple constraint graphs where the associated fitness graphs have exponentially-long directed paths. Thus there is some local adaptive dynamic that will not find a local fitness optimum in a reasonable amount of time. In other words, these fitness landscapes are not easy for all adaptive dynamics: there exist some adaptive dynamics where finding a local optimum takes an unreasonable amount of time. Note that the existence of some long paths (that I give examples of here) is a much weaker condition than the hardness conditions described in Chapter 4 (which are long for clever algorithms or all algorithms) – I visualize the relationship between these different simplicity and complexity measures in Section 5.4.

Example 5.25. (Long adaptive paths in triallelic Domain size 3) *Consider a binary VCSP-instance $\mathcal{C}$, with variables $x_n, x_{n-1,}, \ldots, x_2, x_1, x_0$, and constraints $\{C_{n,n-1}, \ldots, C_{32}, C_{21}, C_{10}\}$ over* *the uniform domain $D = \{0, 1, \rhd\}$, where each constraint C_{ij} is represented by the following matrix:*

$$C_{ij} = 3^{i-1} \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix} \tag{5.23}$$

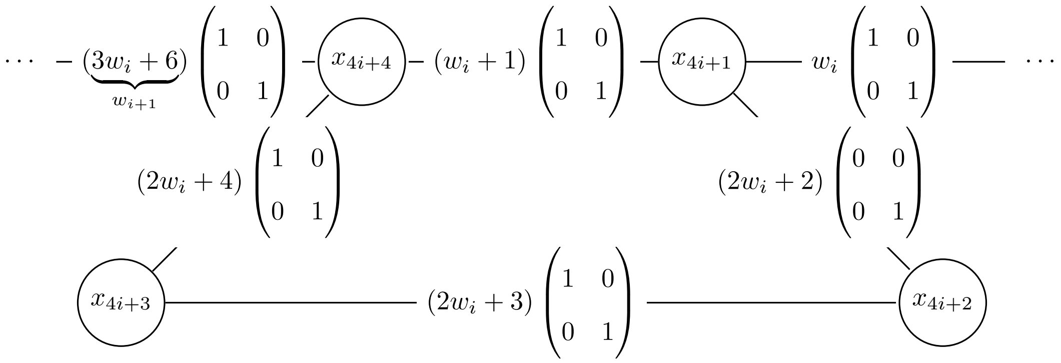
Even though the constraint graph of $\mathcal{C}$ is just a path of length n, we now show the corresponding fitness graph, $G_{\mathcal{C}}$, contains a directed path of exponential length.

Notice that given two natural numbers $M, M' < 2^n$ written in binary as x^M, $x^{M'} \in \{0,1\}^n$ with the least significant digit as x_0, we have that if $M' > M$ then $f(M') > f(M)$. Thus, counting up in binary from 0^{n+1} to 01^n is monotonically increasing in fitness. However, x^{M+1} is often more than a single flip away from x^M (consider the transition from $x^M = 01^n$ for an extreme example). We handle these multi-flip cases with our third domain value, $\rhd$, as follows: (1) given $x^M = y01^k$ where $y \in \{0,1\}^{n-k}$, we proceed to replace the 1s in the right-most block of 1s by $\rhd$, starting from x^M_{k-1} and moving to the right; (2) from $y0\rhd^k$ we can take a 1-flip to $y1\rhd^k$ (regardless of $y_0 = 0$ or 1); (3) from $x' = y1\rhd^k$, we replace the $\rhd$s by 0s, starting from the rightmost $\rhd$ (i.e., x'_0) and moving to the left.

This lets our sequence of moves count in binary from 0^{n+1} to 01^n (passing through 2^n states of just 0s and 1s), while using extra steps with $\rhd$s to make sure all transitions are improving 1-flips; thus, this path in the fitness graph has a length greater than 2^n.

The final example is a binary Boolean VCSP where the constraint graph has tree-width two and maximum degree three, but the associated fitness graph contains an exponentially long directed path. This example is a simplified and corrected version of a similar example for the Max-Cut problem, described by Monien and Tscheuschner [153]. Note, however, that by allowing general valued constraints, instead of just Max-Cut constraints, we are able to reduce the required maximum degree from 4 to 3.

Example 5.26. (Tree-width 2) *Consider a binary Boolean VCSP-instance $\mathcal{C}$ with $n = 4K + 1$ variables. The constraint graph contains a sequence of disjoint cycles of length four, linked together by a single additional edge joining each consecutive pair of cycles. The final cycle is replaced by a single variable x_n with unary constraint $\begin{pmatrix} 0 \\ -w_K \end{pmatrix}$. Hence the constraint graph of $\mathcal{C}$ has maximum degree three and treewidth two. The i-th cycle (for $0 \leq i \leq K-1$) has the following constraints (where the w_i values are defined recursively with $w_0 = 0$):*


$\cdots - \underbrace{(3w_i + 6)}_{w_{i+1}} \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$
x_{4i+4}
$(w_i + 1) \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$
x_{4i+1}
$w_i \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} \cdots$
$(2w_i + 4) \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$
$(2w_i + 2) \begin{pmatrix}0 & 0\\0 & 1\end{pmatrix}$
x_{4i+3}
$(2w_i + 3) \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$
x_{4i+2}

To begin the long path all variables are assigned 0, except $x_n = 1$. The path will proceed by always flipping variables in the smallest 4-cycle block possible.

Within each 4-cycle block, let us write the 4 variables by decreasing index as $x_{4i+4}x_{4i+3}x_{4i+2}x_{4i+1}$. We will make the following transitions within each cycle: if $x_{4(i+1)+1} = 1$ then we'll transition $0000 \to 1000 \to 1001 \to 1101$; if $x_{4(i+1)+1} = 0$ then we'll transition $1101 \to 0101 \to 0100 \to 0110 \to 0010 \to 0011 \to 0001 \to 0000$. Every time that x_{4i+1} is flipped from 0 to 1 or vice versa, we'll recurse to the $(i-1)$th cycle. Because x_{4i+1} ends up flipping from 1 to 0 twice as often as $x_{4(i+1)+1}$, this means that we double the number of flips in each cycle. Variable x_n will flip once, from 1 to 0, due to the unary constraint, which will cause $x_{4(K-1)+1}$ to flip twice from 1 to 0, which will cause $x_{4(K-2)+1}$ to flip four times from 1 to 0, and so on, until eventually this will cause x_1 to flip 2^K times from 1 to 0. Hence we have an improving path of length greater than 2^K.

Note that Examples 5.25 and 5.26 have long paths, but these are not the paths that would be followed by fittest-mutant SSWM dynamics (nor random fitter-mutant SSWM). In the above examples, the fittest-mutant path is still short and so the examples are easy for fittest-mutant SSWM. However, with careful padding, Example 5.25 can be converted to a Boolean VCSP of treewidth 7 that is hard for fittest-mutant SSWM. This is an involved construction that is presented in Cohen et al. [32].

5.4 Visualizing boundary between easy vs. hard landscapes

In Chapter 4, I introduced the distinction between easy vs hard fitness landscapes and showed the most extreme examples of the two kinds. In this chapter, the goal of both the span arguments of Section 5.1 and the encouragement paths of Section 5.2 is to better understand where provably easy fitness landscapes end and hard landscapes begin. So to finish this chapter, I want to visualize in Figure 5.1 this boundary that separates easy vs hard. For simplicity, I focus on just biallelic fitness landscapes since the only higher-allelic landscape I explicitly discussed was in Example 5.25.

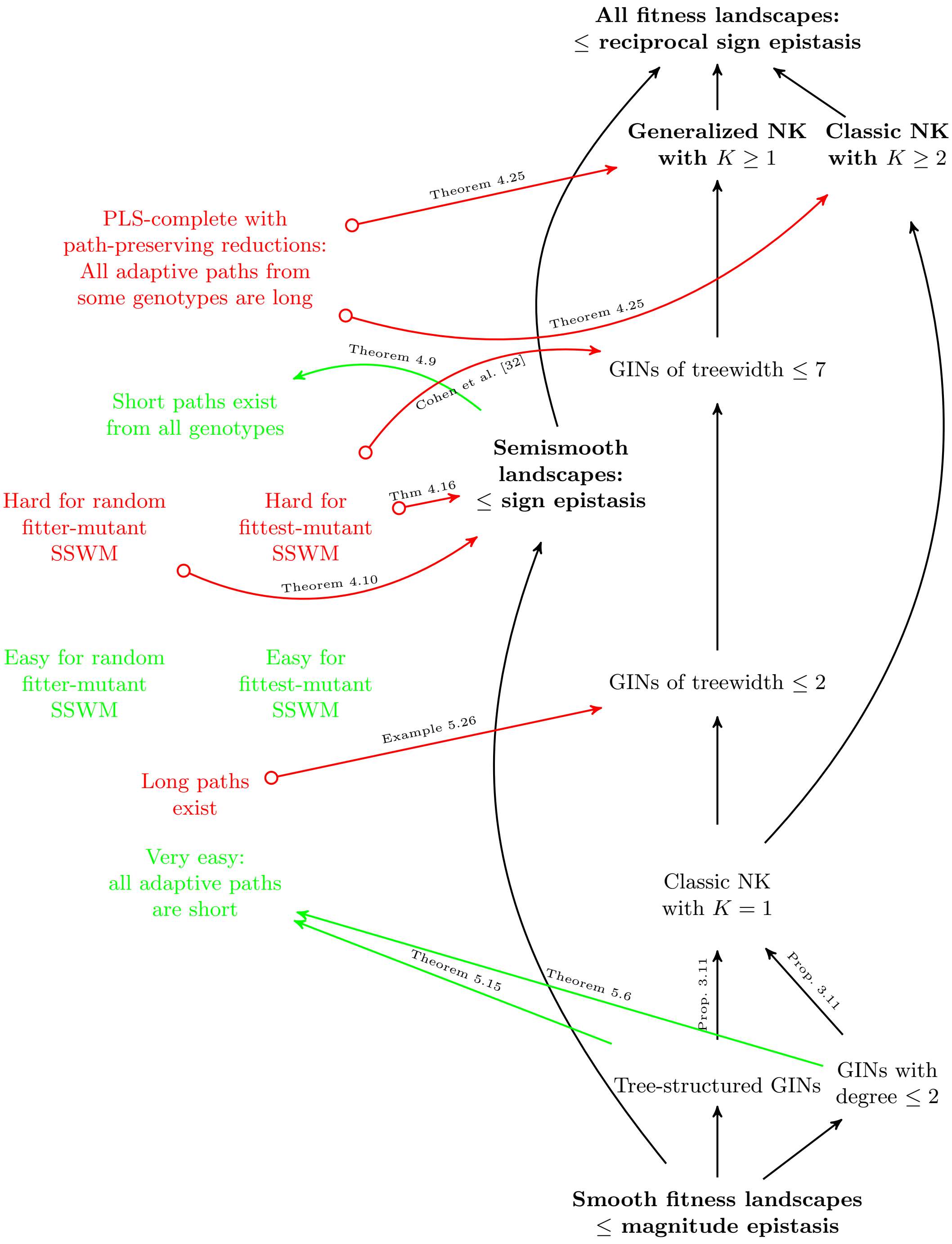

All fitness landscapes:
≤ reciprocal sign epistasis
Generalized NK with $K \geq 1$
Classic NK with $K \geq 2$
Theorem 4.25
PLS-complete with path-preserving reductions: All adaptive paths from some genotypes are long
Theorem 4.25
GINs of treewidth ≤ 7
Theorem 4.9
Cohen et al. [32]
Short paths exist from all genotypes
Semismooth landscapes: ≤ sign epistasis
Hard for random fitter-mutant SSWM
Hard for fittest-mutant SSWM
Thm 4.16
Theorem 4.10
Easy for random fitter-mutant SSWM
Easy for fittest-mutant SSWM
GINs of treewidth ≤ 2
Example 5.26
Long paths exist
Very easy: all adaptive paths are short
Classic NK with $K = 1$
Theorem 5.15
Theorem 5.6
Prop. 3.11
Prop. 3.11
Tree-structured GINs
GINs with degree ≤ 2
Smooth fitness landscapes
≤ magnitude epistasis

Figure 5.1: **Structure of easy vs hard biallelic gene-interaction networks:** Each node is a class of fitness landscapes: i.e., a set of families of fitness landscapes. On the left are simplicity (green) and complexity (red) classes and on the right are expressiveness classes. Complexity proceeds from hardest on top and easiest on the bottom. Black edges between expressiveness classes correspond to set inclusion: class at the tail of the arrow is a subset of the class at the head. Green edges between expressiveness classes and simplicity classes correspond to set inclusion: expressiveness class at the tail is a subset of the simplicity class at the head – this is interpreted as 'easier than'. Red edges between complexity classes and expressiveness classes correspond to membership: there exists families of fitness landscapes in the complexity class at the tail that are also expressible by (i.e., members of) the class at the head – this is interpreted as 'harder than'.

The most difficult part of the visualization is that I am dealing with two different variables at once. On the one hand (the left side of Figure 5.1) are the different notions of easiness (green) and hardness (red). On the other hand (the right side of Figure 5.1) are the different kinds of fitness landscapes. To attach these hands to the same torso, it helps to consider both as classes of landscapes, i.e., as sets that contain families of fitness landscapes where a family is itself a set of fitness landscapes indexed by a size parameter n. In Figure 5.1, I draw a node (i.e., a block of text) for each class. On the left I have the different notions of hardness (complexity classes – in red) and notions of easiness (what I call simplicity classes – in green) and on the right side I have the different classes of landscapes (which I call expressiveness classes, since I mostly describe them in terms of fitness graphs expressible by certain restricted kinds of gene-interaction networks).

For example, consider the node for the complexity class 'Long paths exist': a family of fitness landscapes belongs to this class, if as we increase the index n of the family, the longest adaptive path in the fitness landscape on n loci scales super-polynomially in n. As another example, consider the node for the simplicity class 'Easy for fittest SSWM': a family of fitness landscapes belongs to this class, if as we increase the index n of the family, the fittest mutant strong-selection weak mutation dynamics on the landscape on n loci takes a number of adaptive steps that scales polynomially with n. As a final example, consider the expressiveness class 'GINs of treewidth ≤ 2': a family of fitness landscapes belongs to this class if every fitness landscape in that family is sign-equivalent to a gene-interaction network that has a constraint graph of treewidth 2.

Between these nodes, I draw edges to represent three kinds of relationships, and I omit two other kinds of edges. Figure 5.1 is organized with harder classes on top and easier classes lower down, this lets me omit two kinds of edges: those between pairs of red complexity classes and those between pairs of green simplicity classes. In particular, every complexity class is a subset of every other complexity class below it in the hierarchy. For example, if a family of fitness landscapes is PLS-complete then it must have long paths thus the class of 'PLS-complete...' is a subclass of 'Long paths exist'. Similarly, every simplicity class is a subset of every simplicity class *above* it in Figure 5.1. Hence, the edges corresponding to the above two kinds of relationships are omitted in the figure. For the expressiveness classes, however, the subset relationship is not as straightforward so I mark it with black edges. In particular, the class X at the tail end of a black edge is a subset of the class Y at the head. In terms of expressiveness, this means that every fitness landscape that is expressible by a gene-interaction network that has the restrictions given by X is also expressible by a gene-interaction network that has the restrictions given by Y. Most of these relationships are self-evident, although in the case of the classic NK-model with $K = 1$, the relationship was proved in Proposition 3.11.

The most important edges are the relationships that tie the two hands, that connect the simplicity and complexity classes on the left to the expressiveness classes on the right of Figure 5.1. Each of these relationships was shown in the previous two chapters and so each edge is labeled by the corresponding Theorem, Example, or – in the case the hardness of gene-interaction networks of treewidth ≤ 7 – reference to my other work. As with the prior edges, the green edges connecting expressiveness classes to higher simplicity classes correspond to subsets. What is important, however, is their interpretation in terms of hardness. If there is a green edge from X to Y then that means that the fitness landscapes in X are at least as easy as the fitness landscapes in Y. What departs from the prior convention is the red edges from a complexity class to a higher expressiveness class. Instead of the subset relationship, it corresponds to a membership-like relationship. In particular, if there is a red edge from a complexity class X to an expressiveness class Y then there is a family of fitness landscapes in X that can be expressed by the gene-interaction networks that specify Y. I interpret this as a hardness relationship: Y is at least as hard as X in the worst case.

Finally, I place some expressiveness classes at suggestive heights in Figure 5.1 to capture some conjectures that are beyond the scope of this thesis. For example, I place gene-interactions networks of treewidth ≤ 2 at the same level of hardness as 'easy for fittest SSWM'. This reflects my belief that Example 5.26 having long paths but still being easy for fittest-mutant dynamics is not a pathological feature but something true of all treewidth ≤ 2 networks – I suspect they are all easy for fittest-mutant SSWM. Due to Cohen et al. [32], we know that this is not true for treewidth 7, so it remains an open question to find at what treewidth binary Boolean VCSPs transition from being easy for fittest-mutant SSWM to being hard. The biggest open question, however, is the status of the classic NK model with $K = 1$. For $K = 0$ it is a smooth landscape and, for $K = 2$ it is PLS-complete (Theorem 4.25). Since the global optimum of the classic NK model can be found with dynamic programming for $K = 1$ [233], we can conclude that it is not PLS-complete. In fact, given that it differs by a single Boolean constraint from tree-structured gene-interaction networks, I suspect it is a very easy fitness landscape – i.e., I suspect that all adaptive paths are short in the biallelic classic NK-model with $K = 1$. That is why I placed it slightly below the 'very easy' simplicity class in Figure 5.1. But it remains for future work to confirm these conjectures and draw in the missing edges.

5.5 Summary

As I mentioned in Chapter 2, prior to my theory of hard fitness landscapes, if a biologist found a population to be at a local fitness peak then that in itself did not motivate further questions. But that approach took easy landscapes for granted. Now finding a population at a local fitness peak can start to motivate a new question: is this fitness landscape easy, and if so why? To approach such a question, we need techniques for formally proving that certain landscapes are easy vs hard. In this chapter, I reviewed the existing technique of span arguments (Section 5.1) and showed its limits (Section 5.1.2). To overcome these limits, I had to develop a new proof technique based on encouragement paths (Section 5.2). I used this to establish that all fitness landscapes implemented by binary Boolean tree-structured gene-interaction networks on n loci are easy and have adaptive paths of length at most $\binom{n+1}{2}$ (Theorem 5.15). Along the way, I started a taxonomy of fitness landscapes into complexity and simplicity classes that I summarized in Figure 5.1 and Section 5.4. Many aspects of this taxonomy could be refined and further populated in future work. But it is also important to question the idealization of fitness as a scalar that made fitness landscapes so easy to work with, as I do in Part II of this thesis.

Part II

Empirical Abstraction: Ecology from Evolutionary Games

Nothing in evolution makes sense except in the light of ecology

Grant and Grant [69]

Chapter 6

Reductive vs effective evolutionary games

Throughout the first part of the thesis, I have taken fitness as given without much critical scrutiny. In this chapter, I want to remedy this oversight by asking: what category or kind of thing is fitness? By exploring this question, we will move from fitness landscapes to evolutionary game theory (EGT) and discover that there are two different conceptions of evolutionary games: reductive vs. effective. Reductive vs effective games is a subtle distinction that will take me the whole chapter to illustrate before I can move on, in Chapter 7, to directly measuring the effective games played by non-small cell lung cancer.

Fitness landscapes conceptualize fitness as a single scalar value – a number. This scaler view makes fitness into a feature of a single organism (or genotype), independent of the rich biotic ecology of other organisms. In other words, the scalar view ignores ecology. But the fitness of most real organisms is entwined with the distribution of other organisms. To represent this rich ecology, we have to transform fitness from a scalar to a function.

Although the above is a general point for all of biology, for simplicity, I will often use the language of the microscopic biology of cells, especially of cancer cells, as an example throughout this part of the thesis. In this setting, a scalar fitness can only express cell-autonomous effects, where fitness is inherent to the properties of a single cell. Real cells, however, can display important non-cell-autonomous effects that allow fitness to depend on a cell's micro-environmental context, including frequency of other cell types. To accommodate this, evolutionary game theory views these cell types as strategies, and models fitness as a function which depends on the abundance of each of the strategies in the population. On the surface, the games perspective is more expressive, since scalars can be represented as constant functions.

But we pay for greater expressiveness with a loss of techniques for analyzing the dynamics. For example, when dealing with fitness landscapes, we can often consider the strong-selection weak-mutation limit – as I did in Section 4.3 – which allows us to replace a population by a single point in the landscape. In the case of evolutionary games, such an approximation is unreasonable since it would eliminate the very interactions that EGT aims to study. Instead, more complicated algorithms like replicator dynamics need to be studied. In practice, this means that the strategy space that can be analysed in an evolutionary game is usually much smaller than the genotype/phenotype space considered in a fitness landscape. Typical EGT studies consider just a handful of strategies (most often just two [17, 115, 174], three [18, 118], or four [19]), while fitness landscapes start at dozens of genotypes and go up to tens of thousands – or even hyper-astronomical numbers of potential genotypes as I did in the theoretical work in the first part of this thesis.

One of the reasons for my interest in EGT models is due to their successful use in mathematical oncology. Game theory models have so far had more direct impact in oncology than fitness landscape models [214, 213, 17, 10, 188]. Even limiting to just my own theoretical work, I've used EGT to establish the significance of tissue edges on cancer cell motility [115]; and the importance of treatment timing due to social dilemmas of tumour acidity and vasculature [118] and due to the bone-remodelling cycle [226].

Up to now, the standard approach has been to develop a game theory model from the bottom up, starting from a reasonable reductive grounding and adding micro-dynamic details. This is in keeping with the reductionist tactics common in cell biology and molecular biology. For example, Basanta, Hatzikirou, and Deutsch [17] studied motility in cancer by defining two intuitive strategies: Go vs Grow. The first model included no spatial aspects; later work built on this by adding minimal spatial effects and considering the heterogeneity of spatial structure in a tumour [115]. This progression to more complicated and detailed models is a common pattern among EGT models in oncology. The other common aspect is that the games rely on biological or clinical intuition; the exact game parameters are seldom measured. This is what I call the *reductive* game perspective. This reductive EGT perspective has helped oncologists to express a number of interesting theoretical consequences of frequency-dependent fitness.

But to achieve direct experimental relevance of EGT in oncology, we need to break out of this reductive and intuitive mindset. To break us out of this mindset, I will develop a notion of *effective* games as a way to empirically abstract over aspects that we cannot measure. In this chapter, I will introduce into evolutionary game theory the notion of reductive and effective games and the explicit distinction between them. In Section 6.1, I will motivate replicator dynamics as the central algorithm in evolutionary game theory. In Section 6.2, I will discuss two different kinds of fitness: token vs type. I use these two notions of fitness to give rise to two different kinds of evolutionary games (reductive vs effective) in Sections 6.3 and 6.4 (respectively). Since effective games are the more surprising and unfamiliar concept, I discuss a potential pitfall with over-interpreting them in Sections 6.5 and 6.6. Finally, we need to actually operationalize and then directly measure these effective games in real cancers: this is what I will do in Chapter 7 with non-small cell lung cancer. In Chapter 8, I will expand on the dangers of over-interpreting games, and how effective games can change the way we approach spatially structured populations.

6.1 From fitness to replicator dynamics

Fitness is probably the most central concept in evolutionary biology. This makes fitness something, that in the words of Stearns [205], “everyone understands but no one can define”. But if we stick to just mathematics then there is a popular formal definition for fitness w as:

$$w = \frac{1}{N}\frac{dN}{dt} \tag{6.1}$$

where N is the size of the population for which we want to define the fitness. This is often called the ‘per capita’ growth rate to allude to the simplest exponential growth model where N is measured as head count and w is a constant.

A single fitness by itself is seldom useful, instead it makes more sense to have several different kinds of organisms, with an index x for each, and define a fitness for each x as:

$$w_x = \frac{1}{N_x}\frac{dN_x}{dt} \tag{6.2}$$

In this setting, we can think of the fitness functions from the first part of the thesis as a mapping $f(x) = w_x$ where each w_x is assumed to be a constant, and we can think of the fitness landscape as defining an appropriate notion of ‘nearby’ so that quantities like $\frac{\partial w_x}{\partial x}$ can be defined.

But if we want to peer into the details of the dynamics of the various types of organisms, x, together in a common population then we might want to take a definition of evolution as the change in allele frequencies over time. For this, we will suppose that the possible x come from a finite set A and the total population is $N = \sum_{x \in A} N_x$, Now, we can define the frequency or proportion of x in the population as $p_x = N_x/N$ and look at its time dynamics:

$$\begin{aligned}\dot{p}_x &= \frac{d}{dt}(\frac{N_x}{N}) && (6.3)\\ &= \frac{\dot{N}_x}{N} - \frac{N_x\dot{N}}{N^2} && (6.4)\\ &= \frac{w_x N_x}{N} - \frac{N_x}{N}\frac{\sum_{x\in A} w_x N_x}{N} && (6.5)\\ &= \frac{N_x}{N}(w_x - \sum_{x\in A} w_x\frac{N_x}{N}) && (6.6)\end{aligned}$$

$$\dot{p}_x = p_x(w_x - \langle w\rangle) \qquad (6.7)$$

where I used the definition of w_x from Equation 6.2 and defined $\langle w\rangle$ as the average fitness over the whole population. The final Equation 6.7 is the replicator dynamics, which is the central object of study for evolutionary game theory [211, 146, 78]. Since nothing demands that w_x is a constant, and since the proportions are of such interest, it is natural to consider each w_x as its own function of p where $p \in \Delta_A$ is a vector of the p_x values for each $x \in A$. This makes fitness into a mapping $w_x : \Delta_A \to \mathbb{R}$ from the simplex over the set of all types A to a fitness value. This is what EGT does, converting fitness from a scalar to a function and calling that mapping from type to fitness function a game. In the special case of w_x that are linear in p – that is when $w_x(p) = \sum_{y\in A} G_{xy}p_y$ – we summarize the mapping of type to fitness function by the game's payoff matrix $G = [G_{xy}]$.

By moving from fitness landscapes to games, I have had to extend the mathematical category of fitness from a scalar ($w_x : \mathbb{R}$) to a function of proportion ($w_x : \Delta_A \to \mathbb{R}$). The questions remains: what is the conceptual category of fitness? To answer this question, we need to turn to the philosophy of biology. And from the two answers this question, we will also see two different ways to answer central questions about evolutionary games, like: what are the players? what are the strategies?

6.2 Token fitness vs type fitness

In the philosophy of biology, there are at least two competing philosophical interpretations for what is the conceptual category (or ontology) of fitness: token fitness vs. type fitness [1]. Explaining the difference between them requires a brief philosophical diversion.

There are only twenty-six letters in the English alphabet, and yet there are more than twenty-six letters in this sentence. How do we make sense of this?

Peirce [172] introduced the type-token distinction to make sense of sentences like the above. Types are abstract descriptive concepts, while tokens are objects that instantiate concepts. This is a bit of an imprecise definition, so an example might be more useful: the are twenty-six letter types in English but more than twenty-six letter tokens occurred in this sentence. To give a less linguistic example: I – the writer of this thesis – am a token of the writer type. At the time of writing, I am also a token of the DPhil student type. More relevant to biology: I am a token of the human type; and I am a token of the brown-eyed type; and – at the time of writing – I am a token of the bearded type.

A bit more generally, specific cells or organisms are tokens, while a property shared by one or more cells or organisms defines a type. This distinction between types and tokens translates to two conceptions of fitness. Here, I want to focus on the top level of the Abrams [1] taxonomy of fitness concepts: the distinction of token fitness vs type fitness.

- Token fitness concepts attribute fitness as a property of a particular individual organism: "token fitnesses reflect an individual's complete set of genes, heritable and non-heritable phenotypic properties, and any details of surrounding environmental variations that can affect eventual reproductive success or success of descendants" [1]. It is the way an agent-based modeler might conceptualize fitness: an attribute of each individual agent, something that might be shaped by interactions with other agents and the environment but that 'resides' in the individual agent. Some examples include number of offspring, reproduction time (like time from mitosis to mitosis), or probability to reproduce.

- Type fitness concepts attribute fitness as a property of a type. This is closer to how a population geneticist conceptualizes fitness: an attribute of a genotype or phenotype which might be instantiated in many individual organisms. Type fitness is the quantity that describes the measured changes of the type in the population. A classic example in microscopic systems would be population doubling time or growth rate.

Token fitnessess are sometimes also called individual fitnesses [203], or organismic fitnesses [175]. I prefer the token terminology to avoid potential confusions of the token vs type fitness distinction as an individual vs group selection distinction. The two distinctions are independent of each other. Type fitnesses are more commonly known as trait fitnesses [203, 175]. I prefer the terminology of 'type' versus 'trait' because I feel that traits have a mixed metaphor as both a 'container' of tokens, and with tokens as a 'container' of traits. I think that this ambiguity is considered by others as a feature, not a bug of traits. But with types such ambiguity is absent. Although a token can be of many types, it is less natural to think of a token as a 'container' of types. Unfortunately, some common uses of 'genotype' and 'phenotype' have swayed this thinking of a token as "having" a genotype or phenotype, so some confusion will remain.

6.2.1 Type fitness as an abstract statistic over tokens

It may be helpful to highlight the difference between the token vs type interpretations through an analogy with physics. The setting of statistical mechanics defines properties like kinetic energy for individual molecules and thus it mirrors what I am calling the reductive view based on token fitness. Thermodynamics defines properties like temperature for ensembles of molecules and thus it mirrors what I am calling the effective view based on type fitness. It simply doesn't make sense to talk of the temperature of an individual molecule. Of course, in simple models like the ideal gas, there is a simple correspondence between the reductive and effective views (or token and type): temperature is just mean kinetic energy. But this is not always the case for more complex models. In biology, the analogy of an ideal gas would be an unstructured (inviscid) population. Here, the effective fitness of a type is just the average token fitness of the individuals of that type. Think of an idealized microbial experiment where the population doubling time (type fitness) is the average reproduction time – mitosis to mitosis – on the individual cells (token fitness). But this ideal case seldom happens in nature. In general, there are many ways, like recombination systems, spatial subdivision, and admixture, in which structured populations depart from the idealized panmixis or inviscid population. This can be very relevant to how we interpret games, as I discuss in Chapter 8 in the context of spatial structure.

Of course, type fitness is generally a consequence of the interactions of the various tokens. As such, we can think of a microscopic experiment as a physical implementation of 'some statistic' on tokens. However, this statistic might not necessarily be on just token-fitness but on tokens and their interactions more generally. For example – and as I discuss in more detail in Chapter 8 – it might take into account the distance between and location of tokens in some spatial structure. More importantly, this statistic might be difficult to reverse engineer and replace by a simple formula.

Computer science can help us make sense of this.

For a computer scientist, an abstraction is a way to hide the complexity of computer systems. It is a way to make programs that can be used and re-used without having to re-write all the code for each new application on every different computer. It is in this sense that an algorithm is an abstraction of the actual sequence of bit flips that carry out the physical processes that is computation. To turn it around: the physical process carried out by your computer is then an implementation of some abstract algorithm. Abstraction and implementation are in some sense dual to each other.

Abstract objects or processes are multiply-realizable by a number of concrete objects or processes. The concrete objects might differ from each other in various ways but if the implementations are 'correct' then the ways in which they differ are irrelevant to the abstraction. The abstraction is less detailed than the implementation. To many researchers, less detailed means less particular details about the experimental system being modelled. From this perspective, it might seem like connecting to experiment must always make a model less abstract. In this chapter and the next, I will show that this is not always the case.

The act of measurement itself can be a way to abstract [92]. This is what is achieved with phenomenological or effective theories. For the experimenter, type fitness abstracts over the complexities of population structures (i.e. evolutionary algorithm) that we do not know how to model or measure explicitly. It is nature that figures out the particular computation that transforms token fitness into type fitness and we do not need to know it once we are working at the level of abstract type fitness. If our questions can be expressed at the level of types and the error generated by this abstraction is sufficiently low, then this approach never needs to explicitly reference tokens at all. In such cases, the abstract measurement can be enough for prediction.

It is important to note that I am not advocating for either the token or type conception of fitness as the "correct" or "true" view. Rather, I think that the two views correspond to what Maynard Smith [147] described as "two kinds of mathematical or formal theory that one can make in science." When we think about evolutionary games in terms of tokens, we get reductive games. And when we think about evolutionary games in terms of type, we get effective games. These are two different modes of thinking. For Maynard Smith [147], "most scientists think in one of those modes, but not both." Many phenomena in science can be profitably described from either (or both) perspectives. However, in my experience, the type view of effective games is underrepresented in the current literature on evolutionary game theory. Hence, in this chapter I will aim to convince evolutionary game theorists to spend more time thinking about and – perhaps more importantly – measuring effective games using the techniques I develop in Chapter 7. I also think that the type-based effective view of games matches the ontology of the basic terms of economic game theory a bit more closely, as I will now discuss.

6.2.2 Two analogies to economic game theory

In economic game theory, the concepts of player, strategy, and game are intertwined but relatively straightforward. Players use a rational decision process to select strategies which are then mapped by the rules of the game to payoffs – the utility given to the players. Or, as I say more concisely in the first column of Table 6.1: utility is given to a player based on its strategy, which results from a rational decision process carried out by the player. All of this is summarized as the game. But,

Economic game theory	Evolutionary game theory	
Utility	**Fitness**	
is given to a	is a property of a	is a statistical summary property of a
Player	**Token** organism	Type-structured **Population** of organisms
based on its	based on its	based on the population's
Strategy	**Phenotype** (pure strategy)	**Distribution** of phenotypes (mixed strategy)
which results from	which is	which results from
Rational choice	Fixed from birth & heritable	Replicator dynamics
all summarized as a	all summarized as a	all summarized as an
Game	**Reductive Game**	**Effective Game**

Table 6.1: Defining evolutionary game theory by two different analogies to economic game theory.

how does this classic picture translate to evolutionary games?

In this chapter, I will provide two different interpretation of terms like players, strategies, decision process and games in the evolutionary setting. These are the reductive and effective views of evolutionary games. By starting with these two different readings of fitness in evolutionary theory as analogs for the economic game theoretic concept of utility, we end up with different biological interpretations for the key game theoretic terms. If we start with token fitness then the summary of these interpretations is the reductive game. The reductive view is more useful to computational modelers, and I summarize it in Section 6.3. If we start with type fitness then it is the effective game. The effective view is more useful to experimental biologists and I summarize it in Section 6.4. Both of these views are summarized and contrasted in columns two and three of Table 6.1. It is important to note that the two kinds of games provide different answers to questions like: who or what is the players? what are the strategies? These questions will be answers in Section 6.3 for reductive games and Section 6.4 for effective games.

To make concrete the distinction between the reductive and effective views of games, I will give several examples of how replicator dynamics can be realized and interpreted from each perspective. In each case, I will give an example in expanding populations (Sections 6.3.2,6.4.1) and non-expanding populations (Sections 6.3.1,6.4.2). The reductive implementations of replicator dynamics that I highlight in Sections 6.3.1 and 6.3.2 are well known and the effective implementations in Sections 6.4.1 and 6.4.2 are new but obvious. My primary contribution is in showing these implementations together to show the multiple realizability of evolutionary games and to highlight how much the implementations differ in their ontological grounding. Since the grounding of the effective games perspective is unusual and new to the EGT literature, I will use Sections 6.5 and 6.6 to discuss some tricks and pitfalls in making sense of effective games. Finally, in Chapter 8, I will discuss how to translate between the reductive and effective interpretations in the special case of spatial structure.

6.3 From token fitness to reductive games

Maynard Smith [147] viewed himself as a “microscopic man” and felt more at home with reductive models or “microscopic theory” where “you try to explain the behaviour of something in terms of its components [i.e tokens] and the way they interact.” Many evolutionary game theorists followed in his footsteps, especially computational modelers who like to think in terms of simulations and agent-based models and often take fitness as a property of an individual organism [2]. In that case, we can define a player as an organism that receives a payoff from local interactions that happen between pairs of organisms (or more for multi-player games). The summary of this local interaction is what I would call the reductive game.

In the most common EGT setting, what the organisms do in the game is fixed by their genes. Under this reductive interpretation players do not alter their strategies. This makes it easy to present classic vs evolutionary game theory as two extremes on the spectrum of decision making. In classic game theory, players are unbounded rational decision-makers. In evolutionary game theory, players are the most bounded possible: they make no decisions at all; their behaviour is genetically fixed. Or, as I say more awkwardly in the second column of table 6.1: fitness is a property of an individual token organism based in its phenotype, which is fixed from birth and heritable. All of this is summarized as the reductive game. The proportion of agents in the population is then updated according to an evolutionary process like replicator dynamics. Of course, it is possible to consider models of minimal cognition between these two extremes of fixed-from-birth and unbounded-rationality [20, 149]. I have previously done so by extending the genotype-to-behaviour map with behaviours conditioned on arbitrary observed tags [200, 107, 109, 116, 72] or based on subjective perceptions of game payoffs [114]. But a general discussion of this is outside the scope of this thesis.

In this section I will discuss how two different realizations of reductive fitness can both implement replicator dynamics. In Section 6.3.1, I will focus on fixed population sizes with fitness as probability to reproduce. In Section 6.3.2, I will consider exponentially growing populations with fitness as number of offspring. By providing these two different reductive realizations of replicator dynamics, I hope to make the concept of reductive games clearer by example.

6.3.1 Moran: fitness as probability to reproduce

In a Moran process [154, 210], we imagine that a population is made up of a fixed number N of individuals. An agent is selected to reproduce in proportion to their game payoff, and their offspring replaces another agent in the population, chosen uniformly at random. This gives us a very clear individual account of fitness as a measure of the probability to place a replicate into the population.

Traulsen, Claussen, and Hauert [216] wrote down the Fokker-Planck equation for the above Moran process, and then used Ito-calculus to derive a Langevin equation for the evolution of the proportions of each strategy p_k. The fluctuations in this stochastic equation scale with $1/\sqrt{N}$ and so vanish in the limit of large N. This reduces them to a deterministic limit of the replicator equation in Maynard Smith form [146], with the fitness functions as the payoff functions:

$$\dot{p}_k = p_k \frac{w_k - \langle w \rangle}{\langle w \rangle} \tag{6.8}$$

where $\langle w \rangle$ is the average fitness and the extra condition of $\langle w \rangle > 0$ is introduced.

Alternatively, we might be interested in directly getting the Taylor form [211] of replicator dynamics:

$$\dot{p}_k = p_k (w_k - \langle w \rangle) \tag{6.9}$$

The Maynard Smith and Taylor systems of equations differ only by dynamic time rescaling and thus have the same fixed points, orbits, and paths. If we only care about this in our analysis then we can use the equations interchangeably. But Traulsen, Claussen, and Hauert [216] also show how to achieve the Taylor form directly, too. Instead of birth-death, they consider an imitation process:

1. Two agents are selected individually uniformly at random.

2. If the payoff of the first individual is w_1 and the second is w_2 then the first copies the second with probability $p = \frac{1}{2} + \frac{s}{2}\frac{w_1 - w_2}{\Delta w}$ where Δw is the maximum possible gap in the payoff of two agents in the model.

With this version of the Moran process, Traulsen, Claussen, and Hauert [216] get the Taylor form, with the fitness as payoff. They are not the first to derive the Taylor form replicator equation from imitation processes. In fact, Schlag [192] went further by showing that with the proportional imitation rule (only copy those that have higher payoffs, in proportion to how much higher the payoff is), you not only get the Taylor form replicator equation (in a large population limit), but also that this local update rule is optimal from the individual agent's perspective in certain social learning settings.

6.3.2 Exponential: fitness as number of offspring

One of the biggest difference between ecological modeling in micro- vs. macro- organisms is that macro-organisms seldom have the opportunity to undergo exponential growth; they are almost always at carrying capacity. That is one of the reasons that the Moran process model is so popular (although it is not the only choice of model for fixed population sizes). But what if we want to model this ecological difference – the fact that the total population size grows or that the cell density in the Petri dish changes?

Population size does not have to be constant for evolution to implement replicator dynamics. Consider – as Taylor and Jonker [211] did – m types of cells with $N_1, ..., N_m$ individuals with each individual of type k leaving w_k offspring for the next generation. As before, these offspring numbers w_k could be functions of various other parameters. The population dynamics are then described by the set of m differential equations: $\dot{N}_k = w_k N_k$ for $1 \leq k \leq m$. Now, with $N = N_1 + ... + N_m$, as in Section 6.1, we can look at the dynamics of $p_k = N_k/N$:

$$\begin{aligned}
\dot{p}_k &= \frac{\dot{N}_k}{N} - \frac{N_k \dot{N}}{N^2} && (6.10)\\
&= \frac{w_k N_k}{N} - \frac{N_k}{N}\frac{\sum_{i=1}^m w_i N_i}{N} && (6.11)\\
&= \frac{N_k}{N}(w_k - \sum_{i=1}^m w_i \frac{N_i}{N}) && (6.12)\\
&= p_k(w_k - \langle w \rangle) && (6.13)
\end{aligned}$$

which is just the replicator dynamics. If the w_k are functions of proportions then replicator dynamics can perfectly describe an exponentially growing population. It is important to note that although the mathematics here and in Section 6.1 are identical, the ontologies are not. Here I provided a specific interpretation of w_k as the number of offspring left by each agent of type k, while in Section 6.1 w_x was an abstract quantity that could be implemented in any fashion. The only reason that w_x in Section 6.1 is often called “per capita growth rate” is due to this particular implementation, via the number of offspring, being popular.

6.4 From type fitness to effective games

In contrast to Section 6.3, fitness does not have to be defined as a property of individual organisms. An alternative perspective is to see fitness as defined only as a summary statistic or emergent property of types. This is the perspective that makes the most sense when operationalizing fitness in microscopic systems; especially when using typical fitness measures like growth-rates. In that case, the player is the type-structured population that receives the payoff of fitness. The game then becomes the macroscopic coupling between types implemented by microscopic agents. It can even be misleading to call this coupling an "interaction" since that suggests something too active and direct; as I show in Section 6.6, the coupling could be as indirect as two types feeding on a single resource in batch culture. This population-level description is what I call an effective game. Given its roots in operationalization of microscopic systems, the effective games can be measured directly and we recently developed a game assay for this purpose [119] that I discuss in Chapter 7. As with other effective theories [230], effective games can serve as a bridge between theory and experiment.

The effective games perspective has some curious consequences:

1. Since the players are type-structured populations, the types of organisms – i.e., behaviourally identical classes of organisms – are the strategies. The distribution of phenotypes in the population is then interpreted as a mixed strategy. This is in sharp contrast to the reductive games approach, where a mixed strategy would be a stochastic 'choice' by individual organisms (either during their life or, in the case of something like bet-hedging, at birth). From the effective games perspective, it is the population as a whole that is the player. And the population is neither all type A nor type B but a mixture. Thus, in the inviscid (i.e., not spatially structured) case, for example, when two agents meet they might be type A and they might be type B, and we can interpret their frequencies in the population as probabilities of each pure strategy (although see Chapter 8 for limits to this particular interaction-based interpretation).

2. The player is not static but carries out a 'decision process' specified by the rules of the evolutionary dynamics. This is usually described by the replicator equation. In other words, the mixed strategy encoded the proportion of types is (deterministically) updated according to the replicator equation. Make what you will of the correspondence between replicator dynamics and Bayesian inference, reinforcement learning and other forms of rational decision making [22, 12].

To me, this effective games perspective seems like both a closer correspondence to the aspirations of economic game theory and easier to link to experiment than the reductive perspective of Section 6.3.

To summarize, as in the third column of table 6.1, this perspective is that: fitness is a statistical summary property of a type-structured population based in its distribution of phenotypes, which is updated according to replicator dynamics. All of this is summarized as the effective game.

What does this mean for “cells are players”? For the reductive game, each individual cell is a player that follow a fixed strategy. For the effective game, the population of cells is a player and each cell type is a strategy. As with the reductive games in Section 6.3, in the effective case there are many ways to implement replicator dynamics from different type definitions of fitness. I go over two examples in this case, one for growing populations where changes in population size can be measured (Section 6.4.1) and one for more static populations where only changes in frequency can be measured (Section 6.4.2).

6.4.1 Replating: fitness as fold change

Consider the following idealized experimental protocol that is loosely inspired by Archetti, Ferraro, and Christofori [9] and the *E. coli* Long-Term Evolution Experiment [126, 232, 185, 102]. We will follow these steps:

E1: take a new petri dish or plate;

E2: fill it with a fixed mix of nutritional medium like fetal bovine serum;

E3: put a known number $N_I = N_1^I + ...N_m^I$ of m different cell on the medium (on the first plate we will also know the proportion of types in the mixture);

E4: let them grow for a fixed amount of time Δt which will be on the order of a couple of cell cycles;

E5: remove the cells off the medium and measure the final numbers $N_F := N_1^F + ...N_m^F$;

E6: return to step (E1) while selecting N_I cells at random from the ones we got in step (E5) to seed step (E3).

From comparing steps E3 and E5, we can get the experimental population growth rates (or fold change) as:

$$w_k := \frac{N_k^F - N_k^I}{N_k^I \Delta t} \tag{6.14}$$

this can be rotated into a mapping $N_I \mapsto N_F$ given by $N_k^F = N_k^I(1 + w_k \Delta t)$.

From defining the initial and final population sizes $N^{\{I,F\}} = \sum_{k=1}^m N_k^{\{I,F\}}$, we can compare the initial and final proportions of each cell type $p_k^I = \frac{N_k^I}{N^I}$ and

$$p_k^F = \frac{N_k^F}{N^F} = p_k^I \frac{1 + w_k \Delta t}{1 + \langle w \rangle \Delta t} \tag{6.15}$$

where $\langle w \rangle = \sum_{k=1}^m p_k^I w_k$ is the average fitness.

So far we were looking at a discrete process. But we can approximate it with a continuous one. In that case, we can define $p_k(t) = p_k^I, \;\; p_k(t + \Delta t) = p_k^F$ and look at the limit as Δt gets very small:

$$\begin{aligned}
\dot{p} &= \lim_{\Delta t \to 0} \frac{p_k(t + \Delta t) - p_k(t)}{\Delta t} && (6.16)\\
&= \lim_{\Delta t \to 0} \frac{p_k^I}{\Delta t}\Big(\frac{1 + w_k \Delta t}{1 + \langle w \rangle \Delta t} - 1\Big) && (6.17)\\
&= \lim_{\Delta t \to 0} p_k \frac{w_k - \langle w \rangle}{1 + \langle w \rangle \Delta t} && (6.18)\\
&= p_k(w_k - \langle w \rangle) && (6.19)
\end{aligned}$$

We recover replicator dynamics as an explicit experimental interpretation for all of our theoretical terms.

Note that I did not make any assumptions about whether things are inviscid or spatial; whether I am talking about individual or inclusive fitness; or, whether we have growing populations in log phase or static populations with replacement. All of these microdynamical details are simply buried in the definition of experimental fitness. More importantly, I provided a precise description of how we will measure this quantity. This allows me to hide the details of microdynamics inside of how we measure.

If we are able to peek inside the system more, for example, as I do with time-lapsed microscopy in Chapter 7, then we can also replace the fold-change of Equation 6.14 by more specific measurements of fitness like inferred growth rates. An advantage is that the goodness-of-fit of exponential models can provide a good estimate of the error associated with these measurements. But the cost is a slightly more specific set of assumptions on the microdynamics of our system. In particular, I should modify E4 to include a check that Δt is short enough to keep the population in growth phase. However, since many experimental models can lead to transient exponential growth curves for various microdynamic implementations, these assumptions still do not have to be as stringent as the definitions in Section 6.3.2. I discuss this in more detail in Section 7.2.1.

6.4.2 Measuring the gain function directly

We do not necessarily need to measure separate fitness functions for each cell type, or consider populations that change in total size. It is more important to know the fitness differences, which we can measure directly instead.

Suppose the proportion of cell line A in a mixture is p, with cell line B making up $1-p$, and the fitnesses of the cells are $w_A(p)$ and $w_B(p)$, respectively. Then the replicator dynamics are given by:

$$\frac{dp}{dt} = p(1-p)\underbrace{(w_A - w_B)}_{\text{gain function}} \tag{6.20}$$

To summarize all of the evolutionary dynamics, we just need to measure the **gain function**. Conceptually, the gain function is the increase in growth rate from 'switching' from type A to B (or more precisely: the type-structured population switching between the strategies that those types implement) with p held constant. Mathematically, the gain function between types A and B is defined as the difference between the fitness functions of the two types $(w_A - w_B)$. For the theoretical importance of gain functions, see Peña, Lehmann, and Nöldeke [174] and Kaznatcheev et al. [118].

To directly measure these functions, I will use a simple calculus trick. Consider the log-odds $s = \ln\frac{p}{1-p}$, then:

$$\frac{ds}{dt} = \frac{dp}{dt}(\frac{1}{p} + \frac{1}{1-p}) = \frac{dp/dt}{p(1-p)} = w_A - w_B \tag{6.21}$$

By looking at the log-odds of p instead of just p, we have 'factored away' the logistic growth part of the equation. Now, to measure the gain function, we just have to measure the derivative of s. Unfortunately, experimentalists do not have a derivative detector in the lab, so we have to approximate the derivative by looking at the change in s over a short period of time.

In the case of the basic replating experiments we have a natural discretization of time: p_{in} can be the proportion of type-A cells at the start of our experiment, and p_{out} can be the proportion of type-A cells when we replate. We can then run the experiments for several initial values p_{in} and plot the results as $\Delta s = \ln\frac{p_{out}}{1-p_{out}} - \ln\frac{p_{in}}{1-p_{in}}$ versus p_{in}. This graph is the gain function.

Here we run into the important question of "how short is short enough?" that I discuss in a concrete case in Section 7.2.1. If we run the experiment for too short of a time then the change in p will be overwhelmed by the measurement noise, but if we run for too long before measuring p_{out} then it does not make sense to say we are measuring the derivative at a particular time.

Since we are considering experimental data, it is important to look at the errors associated with our measurements. I do not mean the variance between different runs in different Petri dishes, although that is also important, but the accuracy of the proportions of our initial seeds and the precision of our measurements. For example, if I measure length with a ruler that has millimeter markings, I cannot say that I have measured the length x to better than $x \pm 0.5$ mm. The case is similar for these studies. Each experimental set up will serve as a different ruler, and we will need to do the metrology for each. This will be a big focus in Chapter 7, but I also want to touch briefly on it here.

The reason I want to discuss error explicitly is that Δs amplifies the error in $p_{\text{in}}, p_{\text{out}}$ and does so in a nonlinear fashion. There are several ways we could propagate the errors from p to s, but as an estimate:

$$\sigma_{\Delta s} = \sqrt{(\frac{\partial}{\partial p_{\text{in}}}\Delta s)^2 \sigma_{p_{\text{in}}}^2 + (\frac{\partial}{\partial p_{\text{out}}}\Delta s)^2 \sigma_{p_{\text{out}}}^2} \tag{6.22}$$

$$= \sigma_p \sqrt{\frac{1}{p_{\text{in}}^2(1-p_{\text{in}})^2} + \frac{1}{p_{\text{out}}^2(1-p_{\text{out}})^2}} \tag{6.23}$$

where σ_p is the error on p and $\sigma_{\delta s}$ is the error on our gain function.

So the error is amplified by $4\sqrt{2}$ near $p_{\text{in}} = p_{\text{out}} = 0.5$ and the amplification increases as the proportions approach 0 or 1. For example, for $p_{\text{in}} = p_{\text{out}} = 1/m$ and large m, it becomes approximately a factor of $m\sqrt{2}$. Of course, each experimental set up will serve as a different ruler, and we will need to do the metrology more carefully for each.

6.5 Choosing units of size for populations

It is important to note that populations cannot be defined arbitrarily. To have biologically relevant type-structured populations, we need to have some natural or experimental boundary (sometimes spatial, sometimes conceptual, sometimes historic or energetic) to keep the population together. So here I follow Millstein [152] and take a population as a collection of types held together by a struggle for existence.

Once a population is defined, it is also important to reflect on the units in which the sizes of it and its types will be measured. So far, we have probably imagined populations as numbers of discrete agents: individuals, organisms, or cells. But what is so special about discrete agents? Or to restate in the context of microscopic systems like cancer: what is so special about the number of cells? In this section, I want to question the reasons that microscopic biology tends to focus on individual cells (at the expense of other choices) as basic atoms.

Let us look at what we could mean by 'size of population'. The obvious definition is number of cells, and if all we did was *in silico* simulations and token fitness then it is the definition we could stick to. Especially for agent-based models, it is very tempting to have cells as your agents and building everything up around them. But consider two populations that have the same number of cells and everything else is equal, *except ...*

1. *... the cells in the first population are metabolically twice as active as cells in the second population.* In this case, the more active cells can easily strain their environment more, as they use more resources to fuel themselves. If your limiting resource in the petri dish is growth medium then the more metabolically active cells will consume more of it. The extreme case of this is cells that are completely inactive or even dead. This does not come up as much in simulation, since we can just cleanly remove dead agents but it can matter in experiments. With slower metabolic activity, the cell becomes less of an effect not only on its own future, but also on other cells it interacts with – for example, by moving around less or releasing fewer cytokines and thus interacting with fewer other cells. In this case, the more natural set of physical units might be the power consumption in terms of watts-used or ATP-used. This might be more compatible with the metabolic theory of ecology [23].

2. *... the cells in the first population are twice as big as cells in the second population.* In the case of cancer, the tumor corresponding to the more voluminous population would be much more burdensome to the patient. In fact, tumor burden is often measured and reported as volume in x-ray or other imaging. The number of cells in the tumor is then inferred from these volumetric measures by assuming (or measuring outside the body) the size of a typical cancer cell. From the point of view of games mediated by things like diffusive factors or cell-cell contact, the bigger cells will have more area to absorb/release factors or to contact other cells. If we are working *in vitro*, larger cells also exhaust the limiting factor of free space quicker than small cells. For example, the importance of area has come up in thinking about prostate cancer metastases to the bone [6]. Osteoclasts and osteoblasts take up drastically different amounts of area on the bone, and they are only of significant consequence to the model if they are in contact with the bone (else they are not remodeling it). Area-On-Bone becomes the important variable here. On top of this, size feeds back into the first point, with larger cells usually doing more things metabolically and in terms of other activity. In this case, the more natural set of physical units might be area-covered.

Of course, we could try to express the above in terms of individual cells by converting back and forth between numbers of cells and watts-used or area-covered. Practically, this would mean finding a conversion factor which amounts to a measure of how much power or area a typical cell uses. But in doing so, we have swept some amount of heterogeneity under the rug – after all, each cell takes up a different amount of space or uses a different amount of energy, especially when facing new circumstances like chemotherapy – and it is not clear what useful thing we got in return. But without individual cells to ground us, reductionist story telling becomes more difficult; something that can be both a plus or a minus:

On the plus side, it is hard to imagine how 10 watts-used by cancer interacts with 10 watts-used by fibroblasts, instead we are forced to make these measurements experimentally. Since these measurements are almost always at the level of populations, we do not feel a need to make sense of them in reductionist terms of how a single watt-used interacts with another watt-used. You might have noticed that even the word 'interacts' felt awkward in the last two sentences. Watt- or area-use invite us to recognize the importance of both the size of the other population and the environment more generally. This makes it easier to notice evolutionary games with only indirect interactions like those I discuss in Section 6.6. On the minus side, these effective games, more abstract units, and operationalist perspective can hinder the imagination, and it can often become more difficult to explain the work or to design new experiments.

6.6 Effective games without direct interactions

It is important to not over-interpret effective games. The effective game is an (ecological) interaction in an abstract sense, but this abstract sense might not correspond to our intuitive ideas of what constitutes interaction. This is what I want to explore in this section based on the idealized Petri dish model from Section 6.4.1. In particular, I will explore how what our measurement abstracts over (and thus hides) can determine the effective game.

Let p be the proportion of type-A in the population, and y be the nutritional content of the medium – normalized so that the most nutrient rich mix possible has $y = 1$ and distilled water has $y = 0$. For each cell we will have some (analytic) feeding function $f_A(y)$ and $f_B(y)$ which translates between the nutritional content of the medium and the organism's fitness, such that:

$$\dot{p} = p(1-p)(f_A(y) - f_B(y)) \tag{6.24}$$

It is important to note that f_A and f_B are functions of y and completely independent from p. At this point, we might be tempted to stop by saying that since experimental step (E2) uses a fixed mix of nutritional medium, we can just treat $f_A(y) - f_B(y)$ as a constant and thus (excluding the neutral case) we will always have the population converge to all-A or all-B depending on the sign of the gain function. So we should expect to see no (non-trivial) evolutionary game dynamics. This is the standard intuition behind Gause' exclusion principle: two species cannot co-exist on a single abiotic resource [71].

But stopping here would be a bit premature. The reason that we have to renew the medium on each cycle of this kind of experiment is because it gets consumed between the replating (as opposed to say a continuous culturing in a chemostat where the media is actively maintained at a constant level). Further, the rate of consumption might differ between the two cell types. Suppose that each cell type consumes the nutrients at rate $2k_A$ and $2k_B$, such that if y_{in} was our initial level of nutrients in step (E2) then our final level it step (E5) is $y_{\text{out}} = y_{\text{in}}(1 - 2k_A p - 2k_B(1-p))$.

For simplicity, let me assume that the cell cycle is significantly slower than the metabolic cycle, so that I can start working with the average consumption: $\langle y \rangle_p = \frac{y_{\text{out}}+y_{\text{in}}}{2} = y(1-k_B+p(k_B-k_A))$ (where I relabeled y_{in} by just y in the last equality).

Now, the dynamics become:

$$\dot{p} = p(1-p)(f_A(\langle y \rangle_p) - f_B(\langle y \rangle_p)) \tag{6.25}$$

and suddenly our gain function is no longer independent of p. For this, it might be helpful to switch to our prior notation by noticing that in this case $w_{\{A,B\}}(p) = f_{\{A,B\}}(\langle y \rangle_p)$. Equation 6.25 is the main sleight-of-hand, but let's take the trick to its conclusion.

Let us expand the gain function, noting that I assumed analytic feeding functions, so $f_A(y) = \sum_{n=0}^{\infty} A_n y^n$ for some sequence $\{A_n\}$ and similar for $f_B(y)$ but with $\{B_n\}$:

$$\begin{align}
\Gamma(p) &= f_A(\langle y \rangle_p) - f_B(\langle y \rangle_p) \tag{6.26}\\
&= \sum_{n=0}^{\infty}(A_n - B_n)\langle y \rangle^n \tag{6.27}\\
&= \sum_{n=0}^{\infty}(A_n - B_n)y^n(1-k_B+p(k_B-k_A))^n \tag{6.28}\\
&= \sum_{n=0}^{\infty}\frac{(A_n - B_n)y^n}{(k_A-k_B)^n}(p - \frac{1-k_B}{k_A-k_B})^n \tag{6.29}
\end{align}$$

The last line is the power series of some analytic function $\Gamma(p)$ with coefficients $\{\Gamma_n = \frac{(A_n-B_n)y^n}{(k_A-k_B)^n}\}$ around the point $x_0 = \frac{1-k_B}{k_A-k_B}$. In particular, given any desired (analytic) gain function $\Gamma(p)$, there is some choice of feeding functions f_A and f_B (thus, their corresponding coefficients $\{A_n\}$ and $\{B_n\}$) and k_A and k_B such that the population follows identical dynamics. In other words, in this experimental set up, we can recreate any evolutionary game dynamics without having the cells interacting directly but just based on how they turn nutrition into reproduction. In particular, we can implement games like HAWK-DOVE to have co-existence of A and B on a single abiotic resource, thus violating the competitive exclusion principle.

As an example, suppose you want to recreate an arbitrary cooperate-defect game [107, 108] (also see Section 8.1.1):

$$\begin{pmatrix} 1 & U \\ V & 0 \end{pmatrix}. \tag{6.30}$$

In that case, you need to create the gain function of $\Gamma(p) = U + p(U + V - 1)$ where U, V are the game parameters. To achieve this, just pick any k_A and k_B such that $1 - k_B = k_B - k_A$; A_0 and B_0 such that $A_0 - B_0 = U$; and A_1, B_1, and your initial nutrient concentration y such that $\frac{A_1 - B_1}{1 - k_B} y = U + V - 1$; for all higher A_n, $_n$ just have them equal to each other (for example by setting them all to zero, giving you linear feeding functions).

Thus, simple effective games can hide not-so-simple and not-that-game-like evolutionary dynamics. In Chapter 8, I will discuss more drastic examples of the multiple-realizability of effective games due to the effects of spatial structure. But first, I want to use Chapter 7 to expand on the measurement process for effective games that I sketched partially in Section 6.4.1 and actually measure a game in a real experimental cancer system. We just need to remember to not over-interpret the results.

6.7 Summary

By critically scrutinizing the concept of fitness in this chapter, I shifted our focus from fitness as a scalar to fitness as a function. This fitness-as-a-function view is the foundation of evolutionary game theory, and the main algorithm of EGT is replicator dynamics. As with many of the most interesting algorithms in nature, replicator dynamics is multiply realizable. I gave examples of four such realizations in Sections 6.3.1, 6.3.2, 6.4.1, and 6.4.2 with the main distinction between them being whether they are based on token fitness (Section 6.3) or type fitness (Section 6.4). The former give us reductive games and the latter give us effective games. Reductive games have already proven to be a useful theoretical tool in cancer research [214, 213, 17, 115, 118, 226, 10, 188]. But to give an empirical grounding to evolutionary games in cancer, we have to shift to thinking in terms of effective games. So in the next chapter, I will develop a game assay for directly measuring effective games and use it to measure the games played by non-small cell lung cancer.

Chapter 7

Game assay: measuring the ecology of cancer

Tumours are heterogeneous, evolving ecosystems [151, 74], comprised of different types of neoplastic cells that follow distinct strategies for survival and propagation [82]. The success of a strategy employed by any single type of neoplastic cell is dependent on the distribution of other strategies, and on various components of the tumour microenvironment, and the tumour's population structure. As we saw in Chapter 6, an evolutionary game is the rule mapping the population's strategy distribution to the fitness of individual strategies. Previous work has considered games like SNOWDRIFT [66], STAG HUNT [131], PRISONER'S DILEMMA [220, 219], ROCK-PAPER-SCISSORS [121], and PUBLIC GOODS [138, 8] alongside experiments. But many of the details of the complex biological interactions that implement evolutionary games are experimentally inaccessible at the resolution of cell-cell interactions required to specify reductive games. So prior work has focused on a two-track approach. In the two-track approach, theory and experiment are done side-by-side and success is judged in (an often informal) hypothesis-testing or model-selection perspective by looking at agreement between the macroscopic output of a reductive theory and the experiment. If thought of as a measurement then I would consider the two-track approach as an *indirect measurement*.

Here, for the first time, I combine these two parallel tracks into a single track by experimentally operationalizing the effective game as an assayable hidden variable of a population and its environment. Following Chapter 6, I define the effective game as the game played by an idealized population that shows the same frequency dynamics as the experimental population under consideration. Unlike the reductive games of the two-track approach, this effective game is defined as a *direct measurement* of the experimental system by the game assay that I develop in this chapter.

As such, I am not aiming to test EGT as an explanation. Instead, in line with the approach of this thesis, I am defining a game assay to quantitatively describe the ecology of an empirical system in the language of EGT.

In order to combine the two tracks of theory and experiment, it is important that both are designed together. Since I developed the concept of effective games in Section 6.4 with microscopic measurements in mind, it remains for this chapter to design the actual experiments to measure the game or corresponding gain function. In both Sections 6.4.1 and 6.4.2, I introduced potential experiments as a kind of "derivative detector". More precisely: we measure a type's fitness value over a short enough time interval Δt such that we can treat the measured value as if it was measured instantaneously. This allows me to get experimental point-estimates for the value of the fitness function. Since I am looking at fitness as a function ($w : \Delta_A \to \mathbb{R}$) from proportions (over a set of types A) to a fitness value, the point-estimates we need to reconstruct the functions are the outputs from different initial proportions.

Although not currently common in cancer biology, competitive fitness assays are a gold standard for studying bacteria. In a competitive fitness assay, two cell types are seeded in a petri dish at a known ratio (usually 1:1) and then the fitness of one or both types is measured. Typically, such a competitive fitness assay is conducted with a single initial ratio of two competing cell types. I define the **experimental part of the game assay** as the extension of the competitive fitness assay to a series of different initial seeding ratios. For example, in the experiments of this chapter, we seeded the wells with 8 different ratios of the R and P types (where R and P stand for resistant and parental, and are described in Section 7.1): all-R, 9R:1P, 4R:1P, 3R:2P, 2R:3P, 1R:4P, 1R:9P and all-P. All 8 of these initial conditions are then run in parallel for $\Delta t \approx 5$ days to get estimates of fitness of both types for each ratio.

For the **analysis part of the game assay**, I plot the measured fitness values for each initial proportion. In this simplest case of a linear game assay (that is the primary focus of this chapter, with the exception of Section 7.12), I fit two linear functions to these measured values – one function for each cell type. These are then fitness functions and they can be written in the form:

$$w_P(p) = Ap + B(1 - p) \tag{7.1}$$

$$w_R(p) = Cp + D(1 - p) \tag{7.2}$$

to get us the measured game payoff matrix $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$. Finally, this game can be represented as a point in a two dimensional game space spanned by the axes $C - A$ and $B - D$ (see Figure 7.5b).

The experimental procedure sketched above along with the mapping from its results to the final game point is the **game assay**.

Although this gives the general idea of the game assay for measuring effective games, a number of the steps require fleshing out for any specific experimental system. Hence, I use the rest of this chapter to go through a concrete example of the first ever use of the game assay to measure the Leader and Deadlock games between Alectinib-sensitive and Alectinib-resistant cell types in non-small cell lung cancer. I have already published in the biology literature as Kaznatcheev et al. [119], where further biological details can be found. The goal of this case study is to provide a prototypical example for other scientists interested in applying the game assay. I go through all the main steps of the game assay in the following sections:

7.1: Selecting the experimental model. In this case: two types of non-small cell lung cancer cells, one that is Alectinib-sensitive (parental) and one that is resistant, alongside four other environmental variables: the presence of absence of Alectinib and the presence or absence of cancer associated fibroblast (CAFs).

7.2: Converting microscopy images – the raw output of the experiment – into measures of fitness and also measuring the initial proportion of parental type (Section 7.2.2).

7.3: Summarizing the whole first step of the game assay from experiment to fitness measures in a series of competitive fitness assays and visualizing all the data this step produces in Figure 7.3.

7.6: Converting the series of fitness value point-estimates from Figure 7.3 into best-fit fitness functions in Figure 7.4.

7.8: Converting the eight fitness functions into four games and plotting those four games in the game space of Figure 7.5b.

7.11: Analyzing the fixed-point that is present in one of the four measured ecologies.

7.12: Generalizing the above procedure to non-linear fitness functions and checking the robustness of the linear measurements.

As I go through the game assay for this non-small cell lung cancer case study, I also pause occasionally to mention consequences or observations of interest to oncologists. I discuss these notable consequences as early as possible in the analysis to show just how much or little of the game assay is required to state them. The consequences and observations are:

7.4: If we relied only on the monoculture experiments that are prevalent in cancer research then they would have produced very misleading results, especially in thinking about the cost of resistance.

7.5: The classic model of drug resistance, assumes that the resistant phenotype is neutral or costly in environments where the drug is not present (outside of drug). This is contradicted by the results of the competitive fitness assays in Figure 7.3, challenging a common assumption in cancer modelling.

7.7: The fitness functions in Figure 7.4 suggests that there is commensalism between parental and resistant cells (i.e., resistant cells benefit from the interaction with the parental cells, without exerting positive or negative impact on them) that is switched by the presence of CAFs (i.e., parental benefit from resistant without positive or negative impact on resistant cells).

7.9: The games played by non-small cell lung cancer are Leader and Deadlock. Neither of these games is studied in the existing mathematical oncology literature, but as I show later in Chapter 8, they might be related to well-studied games like Hawk-Dove and Prisoner's Dilemma by inverting the population's spatial structure (see Section 8.4).

7.10: The Leader and Deadlock game are qualitatively different, and the addition of Alectinib or removal of CAFs switches the ecology from one game to the other. This confirms the previously theoretical postulate of EGT in oncology: it is possible to treat not just the player but also the game.

Thus, the non-small cell lung cancer system provides not only a good prototype for applying the game assay but also gives us surprising new insights into oncology.

7.1 Experimental model: Alectinib resistance in non-small cell lung cancer

In some sense, the choice of experimental system is arbitrary, and I could have developed the game assay in any experimental system where accurate measurements of the population size over time of two or more types is possible. Good initial candidates would be other microscopic experimental systems in which frequency dependent fitness effects have been considered before, like: *Escherichia coli* [121, 139], yeast [138, 66], bacterial symbionts of hydra [131], viruses [220, 219], breast cancer [142] and pancreatic cancer [8]. But given the interests and expertise of my experimental collaborators at the Moffitt Cancer Center and Cleveland Clinic, we chose to focus on non-small cell lung cancer (NSCLC), cancer associated fibroblasts (CAFs), and Alectinib.

The EML4-ALK fusion, found in approximately 5% of non-small cell lung cancer (NSCLC) patients, leads to constitutive activation of oncogenic tyrosine kinase activity of ALK, thereby "driving" the disease. Inhibitors of tyrosine kinase activity of ALK (ALK TKI) – such as the drug Alectinib – have proven to be clinically effective, inducing tumour regression and prolonging patient survival [198, 177]. Unfortunately, virtually all of the tumours that respond to ALK TKIs eventually relapse [197] – an outcome typical of inhibitors of other oncogenic tyrosine kinases [63]. Resistance to ALK TKI, like most targeted therapies, remains a major unresolved clinical challenge. Despite significant advances in deciphering the resultant molecular mechanisms of resistance [88], the evolutionary dynamics of ALK TKI resistance remains poorly understood. The inability of TKI therapies to completely eliminate tumour cells has been shown to be at least partially attributable to protection by aspects of the tumour microenvironment [143]. CAFs are one of the main non-malignant components of tumour microenvironment and the interplay between them and tumour cells is a major contributor to microenvironmental resistance, including cytokine mediated protection against ALK inhibitors [237].

To study the eco-evolutionary dynamics of these various factors, we studied the competition between treatment-naive cells of ALK mutant NSCLC cell line H3122 – a "workhorse" for studies of ALK+ lung cancer – and a derivative cell line in which we developed resistance to Alectinib – a highly effective clinical ALK TKI [170] – by selection in progressively increasing concentrations of the drug [43]. Throughout the Chapter, I will refer to the treatment-naive type/strategy as 'parental' and the Alectinib resistant type/strategy as 'resistant'. We aimed to come to a quantitative understanding of how the evolutionary dynamics of parental and resistant strategies were affected by clinically relevant concentrations of Alectinib (0.5μM; see [195]) in the presence or absence of CAFs isolated from a lung cancer.

Cell lines: H3122 cell lines and primary lung cancer associated fibroblasts (CAFs) were obtained from the Moffitt Cancer Center.CAFs were isolated as previously described in Mediavilla-Varela et al. [150] and expanded for 3-10 passages prior to the experiments. The alectinib resistant derivative cell line was obtained through escalating inhibitor concentration protocol, as described in Dhawan et al. [43]. Alectinib sensitive parental H3122 cells were cultured in DMSO for the same length of time, as the alectinib resistant derivate. In this sense, the 'parental' nomenclature is a bit misleading since the parental and resistant strategies are cousins with a common H3122 ancestor. But given that with respect to Alectinib-resistance. we can expect the parental line to better reflect that common ancestor, I think it is a reasonable nomeclature. As we read, we should just remember that 'parental' is shorthand for "parental-like with respect to Alectinib-resistance". I will discuss this more in Section 7.5.1. In order to distinguish the two morphologically similar types under the microscope, we added the genetic code for fluorescent marker proteins (i.e. GFP and mCherry) to their genomes. We cultured both H3122 cells and CAFs in RPMI media, supplemented with 10% fetal bovine serum.

Experimental set-up: Mixtures of parental and resistant H3122 cells were prepared at 8 different ratios: all-resistant, 9:1 resistant to parental, 4:1, 3:2, 2:3, 1:4, 1:9, and all-parental. For each of the competitive fitness assays, 2,000 H3122 cells from the 8 mixtures were seeded with or without 500 CAF cells in 50 μL RPMI media per well into 384 well plates. 6 wells used for each resistant:parental ratio in each of the 4 conditions. 20 hours after seeding, Alectinib or DMSO vehicle control (i.e. no drug), diluted in 20 μL RPMI was added to each well to achieve a clinically relevant final Alectinib concentration of 500 nM/L [195]. Time-lapse microscopy measurements were performed every 4 hours in phase-contrast white light, as well as green and red fluorescent channels.

7.2 Measuring population sizes and fitnesses

Consider a well that is seeded with an initial population size N_P^I of parental and N_R^I of resistant cells; total population size $N^I = N_P^I + N_R^I$. Let $N_{\{P,R\}}^F$ be the population size of {parental,resistant} cells after being grown for an amount of time Δt. It is important to note, as I did in Section 6.5, that it does not matter what units population size is measured in, as long as the measurement is consistent between initial and final time, experimental condition, and interpretation. For this experimental system, I used fluorescent area as a unit of population size measured from time-lapse images via python code using the OpenCV package [97]. See Figures 7.3a,b,d,e for examples of the image analysis.

Given a measure of population size, I can give an experimental definition of replicator dynamics in the spirit of Section 6.4. Experimental growth rate can be defined in the standard way based on fold change as:

$$w_{\{P,R\}} := \frac{N_{\{P,R\}}^F - N_{\{P,R\}}^I}{N_{\{P,R\}}^I \Delta t} \tag{7.3}$$

this can be rotated into a mapping $N^I \mapsto N^F$ given by $N_{\{P,R\}}^F = N_{\{P,R\}}^I (1 + w_{\{P,R\}} \Delta t)$.

By defining the initial and final proportion of parental cells as $p^{\{I,F\}} = N_P^{\{I,F\}} / N^{\{I,F\}}$, we can find the mapping:

$$p^F = \frac{N_P^F}{N^F} = p^I \frac{1 + w_P \Delta t}{1 + \langle w \rangle \Delta t} \tag{7.4}$$

where $\langle w \rangle = p^I w_P + (1 - p^I) w_R$. This is the discrete-time replicator equation.

We can approximate this discrete process with a continuous one by defining $p(t) = p^I$, $p(t + \Delta t) = p^F$ and looking at the limit as Δt gets very small:

$$\begin{aligned}
\dot{p} &= \lim_{\Delta t \to 0} \frac{p(t + \Delta t) - p(t)}{\Delta t} && (7.5)\\
&= \lim_{\Delta t \to 0} \frac{p^I}{\Delta t} (\frac{1 + w_P \Delta t}{1 + \langle w \rangle \Delta t} - 1) && (7.6)\\
&= \lim_{\Delta t \to 0} p \frac{w_P - \langle w \rangle}{1 + \langle w \rangle \Delta t} && (7.7)\\
&= p(w_P - \langle w \rangle) && (7.8)\\
&= p(1 - p) \underbrace{(w_P - w_R)}_{\text{gain f'n for } p} && (7.9)
\end{aligned}$$

Thus, we recover replicator dynamics as an explicit experimental interpretation for all of our theoretical terms. Note that we did not make any assumptions about whether things are inviscid or spatial; whether we are talking about individual or inclusive fitness; or, whether we have growing populations in log phase or static populations with replacement. All of these microdynamical details are buried in the above definition of experimental fitness. This allows us to focus on effective games [111] and avoid potential confusions over aspects like spatial structure [100].

7.2.1 Better estimates of w: growth rate as fitness

The problem with the definition of w in Equation 7.3 is that it depends on just two time points, and thus is not good for quantifying error. In our experimental system, we are able to peek inside the system with time-lapse microscopy. This allows us to get more than just the initial and final population sizes and replace fold-change by the more specific measurements of inferred growth rates for $w_{\{P,R\}}$. An advantage of this approach is that the goodness-of-fit of the exponential growth model provides a good estimate of the error associated with each measurement of w. Thus, we are able to quantify error within each well and not just between experimental replicates in different wells with similar initial conditions.

We use the exponential growth rate from Equation 6.2 as our measure of fitness. Note that this is a property of populations of cells, not of individual cells, hence it is a type-fitness. In order to minimise the impact of growth inhibition by confluence, we analyzed the competitive dynamics during the first 5 days of culture, when the cell population was expanding exponentially. We learned growth rate – along with a confidence interval – from the time-series of population size in each well using the Theil-Sen estimator [212, 194]. Since the Theil-Sen estimator is a rank-based median method (unlike least-squares, which is a numeric-based mean method), it is more robust to noise and does not need to choose between a linear or log representation for computing the error-term (since log transforms do not change rank orders). The robustness to rare but large-magnitude noise is useful for our purposes because such errors do not reflect biological function or noise but are more likely to be due to errors in image processing, for example in response to sudden condensation on the well plate. See Figures 7.3c,f for examples of fitting.

Accounting for finite Δt: The small time-step definition of the derivative can be thought of as a way to approximate a function by local linearizations. It is why, for simulations, modelers often use the discrete-time replicator dynamics to represent continuous-time replicator dynamics: effectively using the discretization as a simple ODE solver/plotter. In the limit of Δt going to 0, this linearization recapitulates the function. Unfortunately in practice, our experimental system cannot take the limit as Δt goes to 0 because of a precision-accuracy trade-off. Accuracy increases as Δt decreases because the continuous dynamics is approximated by more and more, shorter and shorter straight lines. But – from an experimentalist's perspective – the precision decreases because any measurement is noisy: if we measure growth rate over a shorter period of time then we are less certain whether our measurement reflects reality or noise. For very short measurements, we might get higher accuracy (assuming biological factors like time from seeding to adherence could be ignored) but would have incredibly low precision (due to only one, two or three time points from which to calculate growth rate). As we increase the time of the experiment, the accuracy might decrease but the precision will tend to increase. This is a classic trade-off between random noise (low precision) and systematic noise (linearization being progressively less accurate over larger Δt). Since each of our growth rate measurements has an associated error term (see section 7.2), we quantify the random and systematic noise together and propagate it throughout our analysis. Given the biological constraints of our system, we judged that 5 days was a good trade-off point. This will most likely be different for other experimental systems.

Given that the $w_{\{P,R\}}$ are defined over a finite range of time, we need to pick a particular time-point to associate each measurement with. As is common for a discrete time process, we attribute the value of the growth rate to the initial point. In particular, this means that when we consider $w_{\{P,R\}}$ as a function of p, then the values of growth rate are attributed to the initial

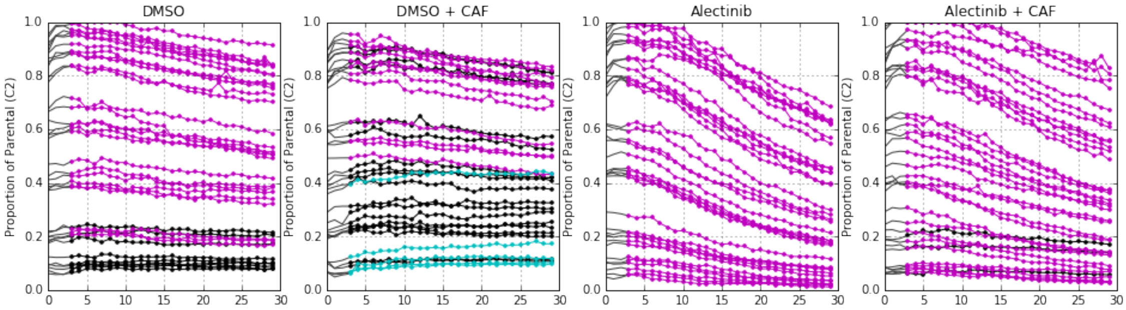


Figure 7.1: **Evolutionary dynamics of proportion of parental cells versus time for competition of parental vs. resistant NSCLC.** Each line corresponds to the time dynamics of a separate well. A line is coloured magenta if proportion of resistant cells increased from start (time step 3 to 8) to end (time step 24 to 29); cyan if proportion of parental cells increased; black if statistically indistinguishable proportions at start and end.

proportion of parental cells and not the final one. This customary choice is further reinforced by the fact that we have a less noisy estimate of initial proportions of cells than of the final, and so other definitions would lead to less precise measurements. Finally, this procedure can be viewed as a series of standard competitive fitness assays, but with the initial ratio of the two types as a varied experimental parameter. Thus, for consistency with both theoretical and experimental literature, we associated the growth rates with the initial – more controlled – seeding proportion.

7.2.2 Proportions

There are two ways to approach seeding proportions, we can get them from the mixing ratios of the experimental preparation or from the first images of microscopy. The former should be treated as more reliable, since I have more confidence in the experimenter's ability to mix two beakers of liquid at specified ratios than in the complicated pipeline of microscopy and image processing. However, the second approach is required if we want to look at the temporal dynamics of proportion during the experiment.

Since raw population sizes have different units (GFP Fluorescent Area (GFA) vs mCherry Fluorescent Area (RFA)), we converted them to common cell-number-units (CNU) by learning the linear transform that scales GFA and RFA into CNU. Specifically, I take the initial seeding numbers based on experimental mixing ratio as given and the GFA and RFA measurements as noisy estimates of them. We can then learn the two parameters of systemic noise corresponding to the slope and intercept of the linear regression of GFA or RFA data on experimental seeding count. We can then interpret the slopes as the typical amount of GFP or mCherry fluorescence per cell and the intercept as the media's auto-fluorescence and zeroth-order errors in image processing. We defined proportions based on this common CNU as $p = N_P/(N_P + N_R)$ where $N_{\{P,R\}}$ is the CNU size of parental and resistant populations. The transform of GFA and RFA into CNU is associated with an error that is propagated to measures of p as σ_p. Thus, although we used 8 different ratios of resistant to parental cells with 6 wells per condition seeded at each of the ratios, we do not average over these 6 wells but associated each with its own proportion $p \pm \sigma_p$ from the initial image. This helps us control for systemic noise from field of view and our image processing algorithm. The time dynamics of p can be seen in the insets of Figure 7.5b for DMSO and DMSO+CAF or in Figure 7.1 for all conditions.

7.3 Fitness as growth rate vs experimental control parameters

Our overall dataset can be viewed as pairs of growth rates (one for the parental type/strategy and one for the resistant type/strategy) measured across a nested set of experimental control parameters. The outer nesting is by the four conditions corresponding to the presence or absence of Alectinib or CAFs. The inner nesting is by initial proportion of parental cells. The rest of the chapter is then organized around a series of figures that visualize and summarize this dataset in various ways. More ambitiously, I call these visualizations and summaries the game assay.

The learned parental growth rate and resistant growth rate of each well are used as the y coordinates in the monoculture experiments of Figure 7.2 and mixed culture in Figure 7.4 (along with errors on the growth rate) and as the x and y coordinates of the main part of Figure 7.3. Due to too much information content, the errors on the growth rates are omitted in Figure 7.3, but they are shown explicitly as error-bars in Figure 7.4. Note that this means that each point in Figure 7.2 and the main part of Figure 7.3 and each pair of points in Figure 7.4 (one magenta and one cyan at the same x-position) correspond to one biological replicate, with the error term coming from the confidence interval on the growth-rate estimate from the 30 time-series points that we recorded for each biological replicate (see Section 7.2 on how this relates to the accuracy-precision trade-off). Thus, each of the 6 wells corresponding to a given resistant:parental ratio (in each of the 4 conditions) has its own independent growth rate with associated error. The wells are not averaged together: each acts as its own data point (with noise) for later analysis (that propagates the noise).

I will interpret the scientific content of these figures throughout the rest of the Chapter. In particular, in Section 7.4, I will motivate Figure 7.3 by explaining why the monotypic data of Figure 7.2 is insufficient for our purposes. And in Section 7.6 I will motivate Figure 7.4 as a summary of the features of Figure 7.3 that are relevant to the game assay. But for now, let us focus a bit more on the form and not the content of the figures.

7.3.1 Figure 7.3 as map of analysis flow

Along with showing all the data, Figure 7.3 serves as a map to the analysis pipeline described in Section 7.2. The subfigures can be understood in the following order:

[a,b,c,d]: Within each image from the series generated by time-lapse microscopy: identify the fluorescent regions for GFP and mCherry and calculate their areas to serve as units of population size (GFA and RFA).

[e,f]: For parental (mCherry measured in RFA) and resistant (GFP measured in GFA) plot the population sizes from each image in the series on a semilog grid as population vs. time. Find the slope of the two lines to serve as parental and resistant fitness.

[g]: Use the parental fitness as x value and resistant fitness as y value to plot each well as a data-point according to the above process, and colour the point according to its experimental condition (with opacity for initial parental proportion; see Section 7.6). For ease of viewing: put a convex hull binding polygon around each well data-point dependent on their experimental condition.

Given the complexity of Figure 7.3, it is tempting to ask for a simple summary statistic of the data in the main figure. But it is not reasonable to ask for the "average" growth rate in Figure 7.3 because each point differs not only along the four experimental conditions of the environment, but also along the micro-environmental conditions of the initial parental proportion (represented by the opacity as explained in Section 7.6). Averaging over this information would be akin to assuming that the growth rates are cell-autonomous (discussed in more detail in Section 7.4). It would be attributing the variance in growth rates to noise instead of the independent variable or initial parental proportion. As such, the game assay developed in the rest of the chapter can be viewed as a method for summarizing Figure 7.3 when the underlying process is non-cell-autonomous. And the games derived through Figure 7.4 and presented in Figure 7.5b can be interpreted as the summary of the data in the main part of Figure 7.3.

7.4 Monotypic vs mixed cultures

To establish baseline characteristics, we performed assays in monotypic cultures of parental (Alectinib-sensitive) and resistant cell lines with and without Alectinib and CAFs. From the time series data, we inferred the growth rate with confidence intervals for each one of 6 experimental replicates

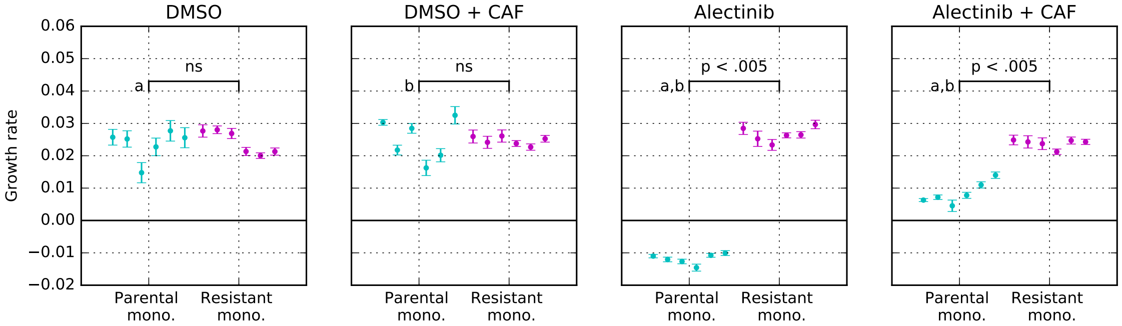


Figure 7.2: **Monotypic culture exponential growth rates** for parental (cyan) and resistant (magenta) cells in indicated experimental conditions. Confidence intervals on each experimental replicate is from confidence on the estimate of growth rate for that single replicate according to the Theil-Sen estimator. Comparisons between experimental conditions (of 6 replicates each) are made using Wilcoxon rank-sum. In addition to conditions linked by lines with reported p-values, conditions labeled by 'a' and 'b' are pairwise distinguishable with $p < .005$. In other words, parental in DMSO is statistically different from parental in Alectinib, which is different from parental in Alectinib + CAF and vice-versa (the 'a' label); and similar for DMSO + CAF vs Alectinib vs Alectinib + CAF (the 'b' label).

in four different experimental conditions (total of 24 data points, each with confidence intervals), as seen in Figure 7.2. As expected, alectinib inhibited growth rates of parental cells (DMSO vs Alectinib: $p < .005$; DMSO + CAF vs Alectinib + CAF: $p < .005$), whereas the growth rate of the resistant cells was not affected. And, as previously reported [237], CAFs partially rescued growth inhibition of parental cells by Alectinib (Alectinib vs Alectinib + CAF: $p < .005$; Alectinib + CAF vs DMSO: $p < .005$), without impacting growth rates of resistant cells.

But we did not limit ourselves to monotypic assays. Our experience observing non-cell-autonomous biological interactions [142] and modeling eco-evolutionary interactions [18, 115, 118] in cancer led us to suspect that the heterotypic growth rates would differ from monotypic culture. Cell-autonomous fitness effects are ones where the benefits/costs to growth rate are inherent to the cell: the presence of other cells are an irrelevant feature of the micro-environment and the growth rates from monotypic cultures provide all the necessary information. Non-cell-autonomous effects [142] allow fitness to depend on a cell's micro-environmental context, including the frequency of other cell types: growth rates need to be measured in competitive fitness assays over a range of seeding frequencies. Hence, we continued our experiments over a range of initial proportions of resistant and parental cells in mixed cultures for each of the four experimental conditions.

Figure 7.3 shows the resulting growth rates of each cell type in the co-culture experiments for all experimental (colour, shape) and initial conditions (opacity is parental cell proportion). In the heterotypic culture – unlike monotypic – CAFs slightly improved the growth rates of the parental cells, even in DMSO. More strikingly, even in the absence of drug, resistant cells tend

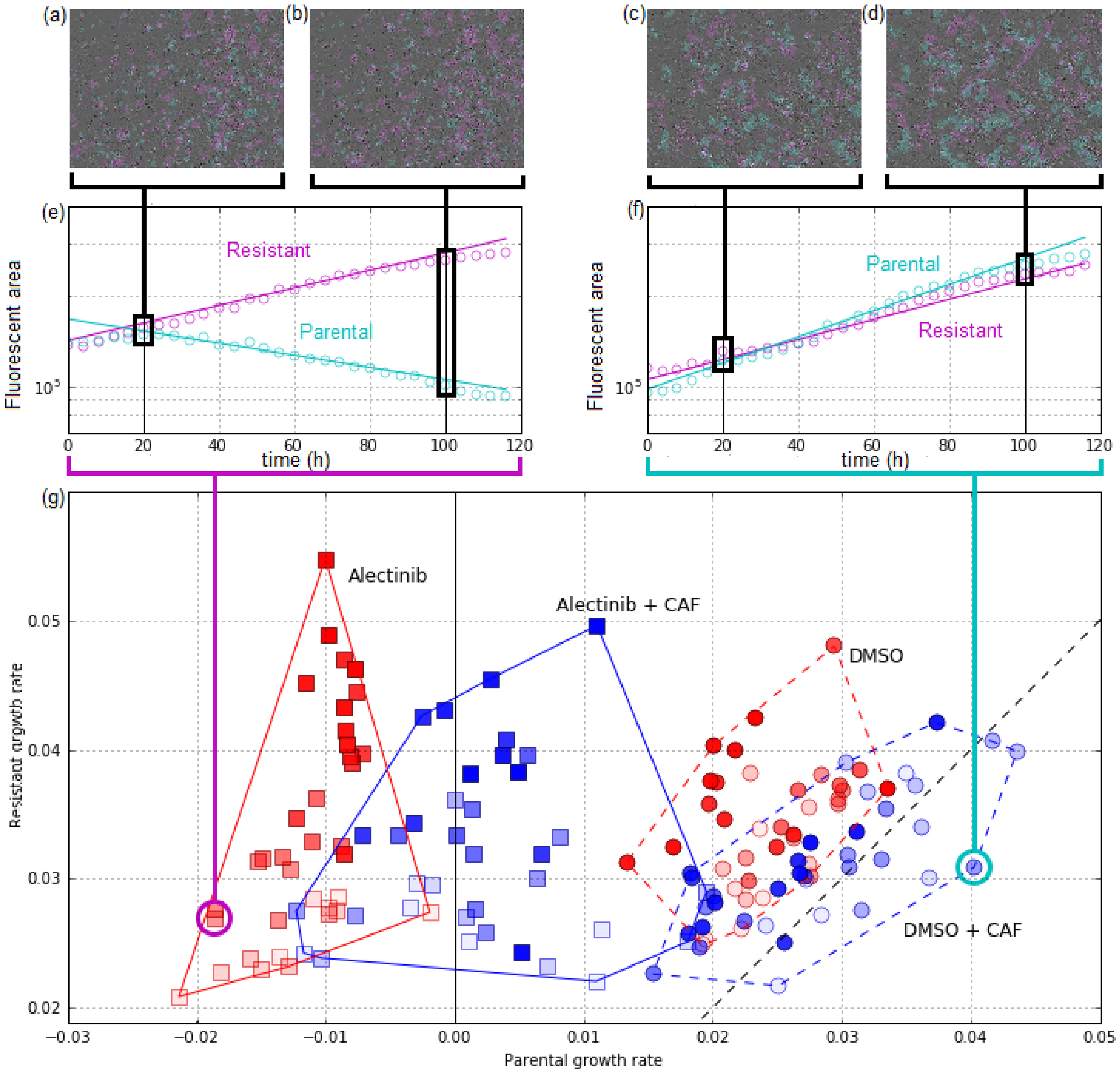


Figure 7.3: **Coculture growth rates across four experimental conditions.** (a-f) serve as a sketch of the analysis procedure to produce the main subfigure (g); for more detailed discussion, see Section 7.3.1

(a,b,c,d): In each experimental replicate at each time step, I quantify population size by fluorescent area of each cell type (shown: two different time points per well, from two different wells). Together, 30 time-lapse microscopy images (one every 4 hours) from each replicate create subfigures (e) and (f).

(e,f): time-series of parental and resistant population size (shown: two example wells). With x-axis is time, y-axis is log of population size. Exponential growth rates (and confidence intervals; omitted) were estimated for each well using the Theil-Sen estimator. These exponential models are shown as solid lines and their slopes serve as the coordinates in (g). See Figure 7.4 for growth rate confidence intervals and Section 7.2 for detailed discussion of growth-rate measurement.

(g): Each point is a separate replicate of a competitive fitness assay with initial proportion of parental cells represented by opacity and experimental condition represented by shape (DMSO: circle; Alectinib: square) and colour (no CAF: red; + CAF: blue). Each replicate's x-position corresponds to the measured parental growth rate and y-position for resistant growth rate; the dotted black line corresponds to the line of equal fitness between the two types at $x = y$.

to have a higher growth rate than parental cells in the same environment (i.e., with the same proportion of parental cells in co-culture). This is evident from most DMSO points being above the dotted diagonal line ($y = x$) corresponding to equal growth rate of the two types and quantified in Figure 7.5b and further discussed in Section 7.9.

7.5 Cost of resistance

The classic model of resistance posits that the resistant phenotype receives a benefit in environments where it is exposed to the drug (in our case: Alectinib or Alectinib + CAF) but is neutral, or even carries an inherent cost, in the absence of drug (in our case: DMSO or DMSO + CAF). For example, experimentalists frequently regard resistance-granting mutations as selectively neutral in the absence of drug, and the modeling community often goes further by considering explicit costs like up-regulating drug efflux pumps, investing in other defensive strategies, or lowering growth rate by switching to sub-optimal growth pathways [4, 82]. If we limited ourselves to the monotypic assays of Figure 7.2, then our observations would be consistent with this classic model of resistance. But in co-culture, we observed that resistant cells have higher fitness than parental cells in the same environment, even in the absence of drug. This is not consistent with the classic model of resistance. This higher fitness of resistant cells might not surprise clinicians as much as the biologists: in clinical experience, tumours that have acquired resistance are often more aggressive than before they were treated, even in the absence of drug.

7.5.1 Reductive vs effective definitions of resistance

As I described above, we observed that even in the absence of drug, resistant cells tend to have a higher growth rate than parental cells in the same environment (i.e. proportion of parental cells in the co-culture). A reductionist could rationalize our observations by saying that we actually selected for two different qualities in our resistant line: (i) a general growth advantage, and (ii) resistance to Alectinib.

This is a reasonable hypothesis, but it faces a few challenges. First, both parental and resistant cells were evolved for the same length of time, with escalating dosages of DMSO for the former and Alectinib for the latter (see Mediavilla-Varela et al. [150] and Section 7.1). Thus, (i) cannot be due to just subculturing, but is somehow linked to drug. Second, there is no growth rate advantage of resistant cells in monoculture (see Figure 7.2); the advantage is only revealed when parental and resistant cells are cultured with a common proportion of parental cells. Finally, to even make the distinction between (i) and (ii), one has to implicitly assume that resistance has to be neutral or

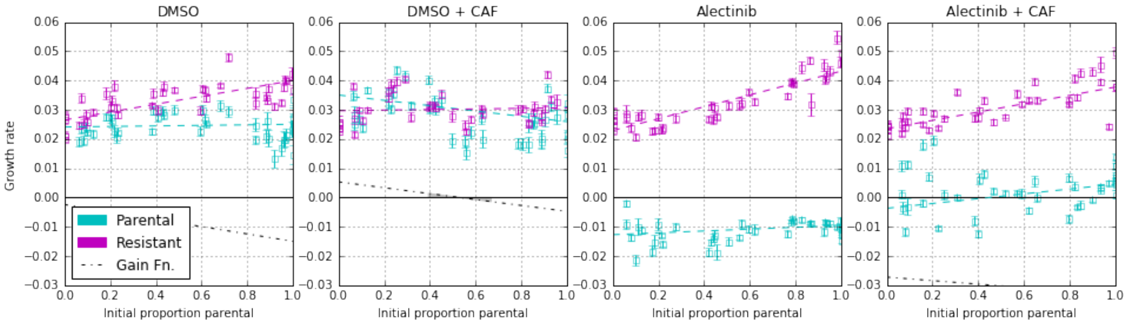


Figure 7.4: **Fitness functions for competition of parental vs. resistant NSCLC.** For each plot: growth rate with confidence intervals versus initial proportion of parental cells. This is the same data, measured in the same way, as Figure 7.3. Cyan data points are growth rates of parental cells, and magenta for resistant cells. Dotted lines represent the linear fitness function of the least-squares best fit; fit error is visualised in Figure 7.5b. The black dotted line is the gain function for parental (see Figure 7.5a), it is well below the $y = 0$ line in the Alectinib conditions (indicating the strong advantage of resistance) and thus cut out of the figure. See Section 7.6 for more discussion and equations for lines of best fit, and Section 7.12 for alternative fits with non-linear fitness functions.

costly by definition. This is putting the cart before the horse: it is assuming resistance is neutral or costly before trying to measure if it is. For an oncologist, however, both (i) and (ii) would constitute clinical resistance if they led to a tumour escaping therapeutic control, and thus both should be a possibility in a reasonable definition of (clinical) resistance. By using a definition of clinical resistance that is broad enough to capture both aspects, we can actually measure if resistance is neutral, costly, or neither. In the case of our cancer system, we observe resistance that is neither neutral nor costly in DMSO co-culture. This sort of 'beneficial' or 'negative cost' resistance should be studied more closely in the theoretical models of mathematical oncology because I think that it corresponds more closely to the sort of ruthless resistance that clinicians usually encounter.

7.6 Frequency dependence in NSCLC fitness functions

As I discussed at the start of the chapter, although not common in cancer biology, competitive fitness assays are a gold standard for studying bacteria. But they are typically conducted with a single initial ratio of the two competing cell types. However, in Figure 7.3, if we view the initial proportion of parental to resistant cells as a variable parameter represented by opacity then we can see a hint of frequency dependence in both parental and resistant growth rates. The goal of this section is to show how to summarize this mass of competitive fitness as the fitness function of evolutionary game theory. This is shown more clearly as a plot of fitness versus proportion of parental cells in Figure 7.4.

7.6.1 Lines of best fit as fitness functions

To measure the fitness functions we plotted fitness of each cell-type in each well vs seeding proportion (p) of parental cells in Figure 7.4. The x-axis proportion of parental cells (p) was computed from the first time-point: see Section 7.2 for an interpretation of this as a measurement of dp/dt or as a series of competitive fitness assays. We estimated the line of best-fit and error on parameters for this data using least-squares weighted by the inverse of the error on each data point (i.e. $\text{weight}_{p,w} = 1/\sqrt{\sigma_p^2 + \sigma_w^2}$). This provides the error estimates on the line's parameters that we use later. The lines of best fit (with coefficients rounded to the thousandths for presentation) from weighted least-squares are:

$$\hat{w}_P^{\text{DMSO}} = 0.025 - 0.001(1-p) = 0.025p + 0.024(1-p) \tag{7.10}$$

$$\hat{w}_R^{\text{DMSO}} = 0.027 + 0.013p = 0.04p + 0.027(1-p) \tag{7.11}$$

$$\hat{w}_P^{\text{DMSO + CAF}} = 0.026 + 0.009(1-p) = 0.026p + 0.035(1-p) \tag{7.12}$$

$$\hat{w}_R^{\text{DMSO + CAF}} = 0.03 + 0.001p = 0.031p + 0.03(1-p) \tag{7.13}$$

$$\hat{w}_P^{\text{Alectinib}} = -0.01 - 0.002(1-p) = -0.01p - 0.013(1-p) \tag{7.14}$$

$$\hat{w}_R^{\text{Alectinib}} = 0.023 + 0.02p = 0.043p + 0.023(1-p) \tag{7.15}$$

$$\hat{w}_P^{\text{Alectinib + CAF}} = 0.005 - 0.009(1-p) = 0.005p - 0.004(1-p) \tag{7.16}$$

$$\hat{w}_R^{\text{Alectinib + CAF}} = 0.024 + 0.014p = 0.038p + 0.024(1-p) \tag{7.17}$$

7.6.2 Interpretable fitness functions

The above fitness functions are a bit difficult to interpret, so I will regularize them further by restricting beyond linear fitness functions to focus on conceptually simple ones. In particular, let us favour cell-autonomous functions over frequency-dependent ones (i.e. l_0 regularization on the fitness function coefficients) and let us favour coefficients that are shared between different S and C in each w_S^C. This results in the following regularized fitness functions:

$$w_P^{\text{DMSO}} = 0.025 \tag{7.18}$$

$$w_R^{\text{DMSO}} = 0.025 + 0.015p \tag{7.19}$$

$$w_P^{\text{DMSO + CAF}} = 0.025 + 0.01(1-p) \text{ (or } 0.03 - 0.01(\frac{1}{2} - p)) \tag{7.20}$$

$$w_R^{\text{DMSO + CAF}} = 0.03 \tag{7.21}$$

$$w_P^{\text{Alectinib}} = -0.010 \tag{7.22}$$

$$w_R^{\text{Alectinib}} = 0.025 + 0.018p \tag{7.23}$$

$$w_P^{\text{Alectinib + CAF}} = 0.005 - 0.009(1-p) \text{ (or } 0.009(\frac{1}{2} - p)) \tag{7.24}$$

$$w_R^{\text{Alectinib + CAF}} = 0.025 + 0.013p \tag{7.25}$$

Note that for both P and R strategies, we used the proportion of the other strategy $(1-p,\, p)$ as the parameter that captures the non-cell-autonomous contribution. In equations 7.20,7.24, we also consider the parameter $\frac{1}{2} - p$ because of the elegant form it provides.

We can compare these regularized fitness functions w_S^C to the non-regularized $\hat{w}_S^C$ in equations 7.10-7.17. As can be seen, all w_S^C are close to their respective $\hat{w}_S^C$ and are actually within the error estimates on $\hat{w}_S^C$. We can see the regularization in action with a push towards a constant base fitness of 0.025 shared by $w_{\{P,R\}}^{\text{DMSO}}$,$w_P^{\text{DMSO + CAF}}$, and $w_R^{\{\text{Aletinib,Alectinib + CAF}\}}$. The absence of frequency-dependent perturbation terms for $w_P^{\{\text{DMSO,Alectinib}\}}$ and $w_R^{\text{DMSO + CAF}}$ suggests that these fitnesses can be explained in terms of cell-autonomous processes and treated like scalars. However, the other fitnesses in the other contexts ask for a non-cell-autonomous explanation and demand that we take the view of fitness-as-a-function. I will expand more on this in the next section.

7.7 Switching the direction of commensalism in NSCLC fitness functions

In all four conditions, we see that the growth rate of the resistant and parental cell lines depends on the initial proportion of parental cells. As discussed in Section 7.6, to capture the principal first-order part of this dependence, we consider a line of best fit between initial proportion of parental cells and the growth rates. In three of the conditions, resistant cell growth rates increase with increased seeding proportion of parental cells, while parental growth rates remain relatively constant (in the case of no CAFs) or slightly increase (for Alectinib + CAFs). In DMSO, this suggests that parental cells' fitness is independent of resistant cells: $w_P^{\text{DMSO}} = 0.025$. Parental fitness in DMSO could be well characterized as cell-autonomous. However, resistant cells in monotypic culture have approximately the same fitness as parental cells (Figure 7.3a), but they benefit from the parental cells in co-culture: $w_R^{\text{DMSO}} = 0.025 + 0.015p$ (where p is the proportion of parental cells). Their fitness has a non-cell-autonomous component. The positive coefficient in front of p suggests commensalism between resistant and parental cells, i.e. resistant cells benefit from the interaction with the parental cells, without exerting positive or negative impact on them.

The DMSO + CAF case differs from the other three in that we see a constant – although elevated $w_R^{\text{DMSO + CAF}} = 0.03$ – growth rate in resistant cells; but a linearly decreasing (in p) growth rate of parental cells: $w_P^{\text{DMSO + CAF}} = 0.025 + 0.01(1-p)$ (or, equivalently: $w_P^{\text{DMSO + CAF}} = 0.03 - 0.01(\frac{1}{2} - p)$). This could be interpreted as CAFs switching the direction of commensalism between parental and resistant cells.

7.8 From fitness functions to games

To measure the effective game that describes the non-cell-autonomous interactions in NSCLC, we can either summarize the fitness functions directly as a payoff matrix, or as an object known as the gain function (see [174, 118] for a theoretical perspective) if we just want the location of the game in an abstract game space.

7.8.1 Summarizing fitness functions as payoff matrices

Let us start by converting our inferred fitness functions from Figure 7.4 into a payoff matrix. That is why, for the final column of our presentation of $\hat{w}_S^C$ in equations 7.10-7.17, I rewrote the fitness functions in a suggestive form of $\hat{w}_P^C = Ap + B(1-p)$ and $\hat{w}_R^C = Cp + D(1-p)$. This is done to show at a glance where the matrix entries in Figure 7.5b come from. This is because the $p = 0$ and $p = 1$ intercepts of the fitness functions serve as the entries of the game matrices. In other words, each row in the matrix correspond to a strategy's fitness function with the column entries as the $p = 1$ and $p = 0$ intersects of this line of best fit. Note that in Figure 7.5b, we multiplied the entries by 100 for easier presentation.

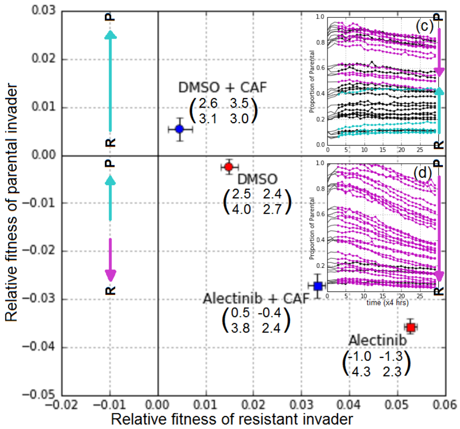
In an idealized inviscid population one can interpret the payoff matrix $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ as the fitness effect on a type given by the row strategy from interacting with a type given by the column strategy. So, for example – in an idealized population – a cell of parental type (i.e. strategy of row 1) will receive a fitness effect of A from interacting with another parental type (i.e. strategy of column

$$\begin{array}{c|cc} & P & R \\ \hline P & A & B \\ R & C & D \end{array} \Rightarrow \begin{cases} \dfrac{d}{dt}N_P = N_P \overbrace{\left(A\dfrac{N_P}{N_T} + B\dfrac{N_R}{N_T}\right)}^{\hat{w}_P\text{: parental growth rate}} \\ \dfrac{d}{dt}N_R = N_R \underbrace{\left(C\dfrac{N_P}{N_T} + D\dfrac{N_R}{N_T}\right)}_{\hat{w}_R\text{: resistant growth rate}} \end{cases}$$

$$\Rightarrow \frac{dp}{dt} = p(1-p)\overbrace{\Big(\underbrace{(B-D)}_{\substack{\text{relative fitness of}\\\text{parental invader}}}(1-p) - \underbrace{(C-A)}_{\substack{\text{relative fitness of}\\\text{resistant invader}}}p\Big)}^{\text{gain function for } p}$$

where $N_T = N_P + N_R$ and $p = \frac{N_P}{N_T}$.

(a) Replicator dynamics of parental-resistant NSCLC



(b) Two dimensional game space.

Figure 7.5: **Measured games: (a) Replicator dynamics.** Consider an idealized population of two strategies in a competitive co-culture: parental (P) and resistant (R). When the type P interacts with P the type experiences a fitness effect A; when P interacts with R then P experience fitness effect B and R a fitness effect C; two Rs interact with fitness effects D, summarized in the matrix. This can be interpreted as an idealized exponential growth model from Section 6.3.2 for the number of parental (N_P) and resistant (N_R) cells. The dynamics of the proportion of parental cells $p = \frac{N_P}{N_P + N_R}$ over time is described by the replicator equation (bottom). Of course, as I showed in Chapter 6 and Section 7.2. these games and replicator dynamics can also be interpreted experimentally.
(b) Mapping of the four measured *in vitro* games into game space. The x-axis is relative fitness of a resistant focal in a parental monotypic culture: $C - A$; y-axis is relative fitness of a parental focal in a resistant monotypic culture: $B - D$. Games measured in our experimental system are given as specific points with error bars based on goodness of fit of linear fitness functions in Figure 7.4. The games corresponding to our conditions are given as matrices (with entries multiplied by a factor of 100) by their label. See Section 7.9 for more details on the games. The game space is composed of four possible dynamical regimes, one for each quadrant. The typical dynamics of each dynamic regime are represented as qualitative flow diagram between P and R: an upward cyan arrow corresponds to an increase in the parental proportion, and a downward magenta arrow correspond to an increase in the resistant proportion. In the case of the two dynamic regimes observed in the NSCLC experimental system, I also include insets of measured dynamics (c,d).
Experimental time-series of proportion of parental cells for DMSO + CAF (c) and Alectinib + CAF (d). Each line corresponds to the time dynamics of a separate well. A line is coloured magenta if proportion of resistant cells increased from start to end; cyan if proportion of parental cells increased; black if statistically indistinguishable proportions at start and end (where start/end are defined as the first/last 5 time-pints (20 hours)). See Figure 7.1 for proportion dynamics of all four games.

1) but a fitness effect of B from interacting with a resistant type (i.e. strategy of column 2). In a real population, however, these payoff matrix entries are abstract phenomenological quantities that could be implemented by various biological or physical processes [111].

7.8.2 Gain functions and game space

Two-strategy matrix games have a convenient representation in a two dimensional game-space. This is the output of the game assay. We plot the inferred games in a game-space spanned by the theoretical fitness advantage a single resistant invader would have if introduced into a parental monotypic culture versus the fitness advantage of a parental invader in a resistant monotypic culture; as shown in Figure 7.5b. In this representation, the game points can be calculated from the matrices as $x := C - A$ and $y := B - D$, and the error is propagated from the error estimates on the fitness function's parameters. Alternatively, a particularly important equation for studying two-strategy games is the gain function. This represents the relative fitness difference between two strategies. Thus, it is a measure of selection strength and a proxy for the rate of evolution. The parental gain function (i.e. gain function for p in Figure 7.5a and equation 7.9) is given by $\hat{g}^C_P(p) = \hat{w}^C_P(p) - \hat{w}^C_R(p)$; and the resistant gain function (i.e. gain function for $q = 1 - p$) is $\hat{g}^C_R(q) = \hat{w}^C_R(1-p) - \hat{w}^C_P(1-p) = -\hat{g}^C_P(1-p)$. The end-points of this gain function determine the game coordinates in the game space of Figure 7.5b with $(x, y) := (\hat{g}^C_R(0), \hat{g}^C_P(0)) = (-\hat{g}^C_P(1), \hat{g}^C_P(0))$. These points can be interpreted as the idealized quantities of relative fitness of a resistant invader in parental monoculture ($\hat{g}^C_P(0)$) and relative fitness of a parental invader in resistant monoculture ($\hat{g}^C_R(0) = -\hat{g}^C_P(1)$).

It is important to note that defining $x := C - A$ and $y := B - D$ makes the game space coordinates a linear function of the measured fitnesses. This prevents error from blowing up unreasonably. This is the reason for using this linear representation of games over the more common non-linear representation [190, 107, 108] of $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ (with $A > D$) as $\begin{pmatrix} 1 & U \\ V & 0 \end{pmatrix}$ that I discuss in Section 8.1.1.

The $x = 0$ and $y = 0$ lines, divide the game space into four quadrants, with each corresponding to a different dynamic regime that I illustrate with a sample dynamic flow in Figure 7.5b and discuss in more detail in Section 7.11.

7.8.3 Games from interpretable fitness functions

Above, we used the raw fitness functions from Section 7.6 and the corresponding gain functions. I think this is the appropriate thing to do in general. But we might also be interested in starting from

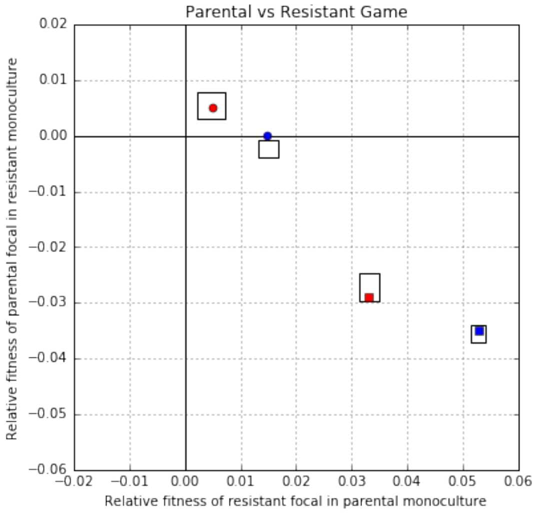


Figure 7.6: **Mapping of the regularized fitness functions for the four conditions into game space.** The x-axis is the relative fitness of a resistant focal in a parental monotypic culture: $C - A$. The y-axis is the relative fitness of a parental focal in a resistant monotypic culture: $B - D$. Games measured in our experimental system are specified by the bounding boxes corresponding to the range of their errors. The games corresponding to the regularized fitness functions in equations 7.18-7.25 are given as points. Experimental condition is represented by shape (DMSO: circle; Alectinib: square) and colour (no CAF: red; + CAF: blue).

the regularized fitness functions of Section 7.6.2. For a visual confirmation that the regularization of w_S^C in equations 7.18-7.25 are reasonable, we can transform them into regularized games. We do this in the same way as we did for transforming the non-regularized $\hat{w}_S^C$ in equations 7.10-7.17 into the game-points of figure 7.5b. The results are in Figure 7.6. The regularized games (points) are within the confidence rectangles of the measured games (boxes), with the exception of DMSO which is just outside its box. This is reasonable given that the boxes correspond to error: i.e. around 2/3rds confidence.

7.9 Leader and Deadlock games in NSCLC

Looking at the games that we measured in Figure 7.5b, we can see that the game corresponding to DMSO + CAF – although quantitatively similar to DMSO – is of a qualitatively different type compared to all three of the other combinations.

If we look at our empirical measurements for DMSO + CAF (upper-right quadrant Figure 7.5b) then we see the Leader game, and Deadlock in the other three cases (we will use DMSO to illustrate the Deadlock game). These are both two-strategy, two-player symmetric games that differ in the values of the payoff matrix $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ that specifies them. If we want an intuitive story for these games, we can think of player 1's strategy as specifying which row she chooses and player

2's strategy as specifying which column he chooses. The matrix entry corresponding to the pair's choice is then the payoff for player 1, and – since the game is symmetric – the transpose entry is the payoff for player 2. This story is a fiction which I will use to give intuitions about the LEADER and DEADLOCK games below, but as we saw in Chapter 6 the actual effective interpretation of these games is ontologically much more involved but mathematically equivalent to this convenient fiction.

The DEADLOCK game observed in DMSO is in some ways the opposite of the popular PRISONER'S DILEMMA (PD) game (in fact, Robinson and Goforth [187] call it the anti-PD). If we interpret parental as cooperate and resistant as defect then, similar to PD, each player wants to defect regardless of what the other player does (because $4.0 > 2.5$ and $2.7 > 2.4$; payoff numbers used in these examples are from the matrix entries we measured in Figure 7.5) but hopes that the other player will cooperate (because $4.0 > 2.7$). However, unlike PD, mutual cooperation does not Pareto dominate mutual defection (because $2.5 < 2.7$) but is instead strictly dominated by it. Thus, the players are locked into defection. In our system, this corresponds to resistant cells having an advantage over parental in DMSO as I discussed in Section 7.5.

The LEADER game observed in DMSO + CAF is one of Rapoport [184]'s four archetypal 2×2 games (with the other 3 being the EXPLOITER, HERO, and MARTYR games) and a social dilemma related to the popular game known as HAWK-DOVE, CHICKEN, or SNOWDRIFT (in fact, Robinson and Goforth [187] call it BENEVOLENT CHICKEN). This means that, unlike DEADLOCK, LEADER is already a game of significant interest to the wider (evolutionary) game theory community. If we interpret parental as 'lead' (for SNOWDRIFT: wait) and resistant as 'work' (for SNOWDRIFT: shovel) then similar to SNOWDRIFT, mutual work is better than both leading (because $3.0 > 2.6$) and thus no work being done (for SNOWDRIFT: both waiting and thus not getting out of the snowdrift) but each player would want to lead while the other works (because $3.5 > 3.0$). However, unlike SNOWDRIFT, mutual work is not better than the "sucker's payoff" of working while the other player leads (because $3.1 > 3.0$). Rapoport [184] sees this as a tension with a player switching from a "natural" point of mutual work to lead and thus benefit both players ($3.5 > 3.0, 3.1 > 3.0$), but if the second player also does the same and becomes a leader then all benefit disappears (because 2.6 is the smallest payoff). In our system, this corresponds to cells in the tumour experiencing selective pressure to lose some but not all of its resistance in DMSO + CAF.

Note that the above intuitive stories are meant as heuristics, and the effective games that we measure are summaries of population level properties [111, 100] as discussed in Chapter 6: the population is the player and the two types of cancer cells are the strategies. This means that the matrix entries should not be interpreted as direct interactions between cells, but as general couplings between subpopulations corresponding to different strategies. The coupling term includes not only direct interactions, but also indirect effects due to spatial structure, diffusible goods, contact inhibition, etc.. But this does not mean that an effective game is not interpretable. For example, the DEADLOCK game captures the phenomenon of the resistant population always being fitter than parental (for example, in DMSO). We noted this effect intuitively in Figure 7.3 (also see Section 7.5) from replicates being above the $y = x$ diagonal. Measuring a DEADLOCK game for DMSO with confidence intervals that do not extend outside the bottom right quadrant of the game space in figure 7.5b allows us to show the statistical significance of our prior intuitive understanding. In other words, effective games allow us to quantify frequency-dependent differences in growth rates.

7.10 Treating the game

So far, measuring a linear gain function has enabled us to develop an assay that represents the inter-dependence between parental and resistant cells as a matrix game. Experimentally cataloging these games allows us to support existing theoretical work in mathematical oncology that considers treatment (or other environmental differences) as changes between qualitatively different game regimes [7, 18, 115, 118]. In this framework, treatment has the goal not to directly target cells in the tumour, but instead to perturb the parameters of the game they are playing to allow evolution to steer the tumour towards a more desirable result (for examples, see [7, 18, 115, 118, 157, 19]). Empirically, this principle has inspired or built support for interventions like buffer therapy [54], vascular renormalization therapy [83], and adaptive therapy [238] that target the micro-environment and interactions instead of just attacking the cancer cell population. The success of the Zhang et al. [238] trial suggests that therapeutic strategies based on modulating competition dynamics are feasible. This highlights the need for a formal experimental method like our game assay that directly measures the games that cancer plays and tracks if and how they change due to treatment.

In our system, we can view an untreated tumour as similar to DMSO + CAF and thus following the LEADER game. Treating with Alectinib (move to Alectinib + CAF) or eliminating CAFs through a stromal directed therapy (move to DMSO), moves the game into the lower-right quadrant of Figure 7.5b, and the game becomes a DEADLOCK game. Not only are these games quantitatively different among the four environmental conditions – see Figure 7.5b – but they are also of two qualitatively different types. To my knowledge, neither of the LEADER and DEADLOCK games are considered in the prior EGT literature in oncology. Given that the DEADLOCK of drug-resistant over drug-sensitive cells is a challenge for classic models of resistance, I would be particularly interested in theoretical models of resistance that produce the Deadlock game. In addition to challenging theorists by adding two new entries to the catalogue of games that cancers play, this switch allows us to show that the theoretical construct of EGT – that treatment can qualitatively change the type of game – has a direct experimental realization. Unfortunately, neither of our *in vitro* games would lead to a therapeutically desirable outcome if they occurred in a patient.

7.11 Fixed points, heterogeneity and latent resistance

A particularly important difference between Leader and Deadlock dynamics is the existence of an internal fixed point in Leader but not in Deadlock. This can be seen from the Leader game having both coordinates positive, while the Deadlock games have $y < 0 < x$. As we saw in Section 7.8.2, if these two coordinates have the same sign then the gain function has to cross 0 in getting from $p = 0$ to $p = 1$ and thus the dynamics have a fixed point. In general, in this representation, there are four qualitatively different types of games corresponding to the four quadrants, each of which I illustrative with a sample dynamic flow. If the game point has opposite signs (either $y < 0 < x$ or $x < 0 < y$) then the dynamics flow from the unstable strategy (strategy 1 if $x < 0$; strategy 2 if $y < 0$) to the stable strategy (strategy 1 if $x > 0$; strategy 2 if $y > 0$). If the two coordinates are both positive (top right quadrant of Figure 7.5b) then the fixed point is stable, if both are negative (bottom left quadrant of Figure 7.5b) then the fixed point is unstable. In our experimental system, only the DMSO + CAF condition has a fixed point at 0.53 ± 0.14 (rounded to the nearest percent).

7.11.1 Width and height of fixed regions

Since we propagate the errors on our measurement from the image all the way to the game, we find it more helpful to think of an experimental fixed point not as a point but as a fixed region $p \in (0.39, 0.67)$ of finite width. This can provide an alternative explanation for the apparent slowness of convergence to the fixed point in Figure 7.1. Some of the fixed region's width is noise in measurement, but some could be due to true variance between wells: in particular, even if the reductive game is the same, the spatial structure will be slightly different in each well and thus there will be a slightly different effective game. As such, apparent slowness in Figure 7.1 might be from different lines being very close to slightly different fixed points that are all within the fixed region's width. An alternative view to width is in terms of height: a fixed region corresponds not just to the point where $\hat{g}_P^{\text{DMSO + CAF}}$ crosses 0 but to the region where the gain function crosses 0 ± 0.0014 (rounded to the nearest thousandth). I call this the fixed region's height (and use it in Section 7.12). This height is due to propagation of error and can be interpreted as our measurement not being able to distinguish relative growth rates in $(-0.0014, 0.0014)$ from zero. In the case of the other three conditions (DMSO, Alectinib, and Alectinib + CAF), in going from $p = 0$ to $p = 1$, the gain functions do not pass within their fixed region height of zero, and thus no fixed regions exist. Of course, fixed points and regions are a property of equilibrium dynamics: in the most general case, even on very long timescales these fixed points might not be realized due to the evolutionary constraints of population size [58] or computation [33, 96] (as discussed in Chapter 2). Thus, it is important to check to what extent this qualitative difference can translate to a quantitative difference in finite time horizons. Although based on the proportion dynamics in Figure 7.1, it is reasonable to believe that our experimental system equilibrates relatively quickly. But for certainty, it would be useful for future work to consider replating experiments (for example, with replating as described in Section 6.4.1) to extend the time-scales past what we considered.

7.11.2 Coexistence in DMSO + CAF as latest resistance

In our system, we can see a quantitative difference in the convergence towards the fixed point in the DMSO + CAF condition of Figure 7.5c, and no such convergence in the other three cases (Figure 7.5d for Alectinib + CAF; Figure 7.1). Since the strength of selection (magnitude of the gain function) is small near a fixed point, the change in p also slows in the DMSO + CAF condition.

Since the DMSO + CAF condition is our closest to an untreated patient, it might have important consequences for latent resistance. As discussed in Section 7.5, many classical models of resistance assume a rare preexistent mutant taking over the population after the introduction of drug. In our experimental system, however, if the resistant strategy is preexistent then negative frequency dependent selection will push the population towards a stable polyclonal tumour of resistant and sensitive cells before the introduction of drug. This allows for much higher levels of preexisting heterogeneity in resistance than predicted by the classical picture. As such, we urge theorists to reconsider the assumption of the rare pre-existing resistant clone.

Of course, our results are for a single *in vitro* system. But if similar games occur *in vivo* and/or for other cancers, then such preexisting heterogeneity could be a possible *evolutionary mechanism* behind the speed and robustness of treatment resistance to targeted therapies in patients. This could help explain the ubiquity and speed of resistance that undermines our abilities to cure patients or control their disease in the long term. We will not know this unless we set out to quantify the non-cell autonomous processes in cancer. Building a catalogue of the games cancers play – by adopting our game assay in other cancers, and other experimental contexts – can help

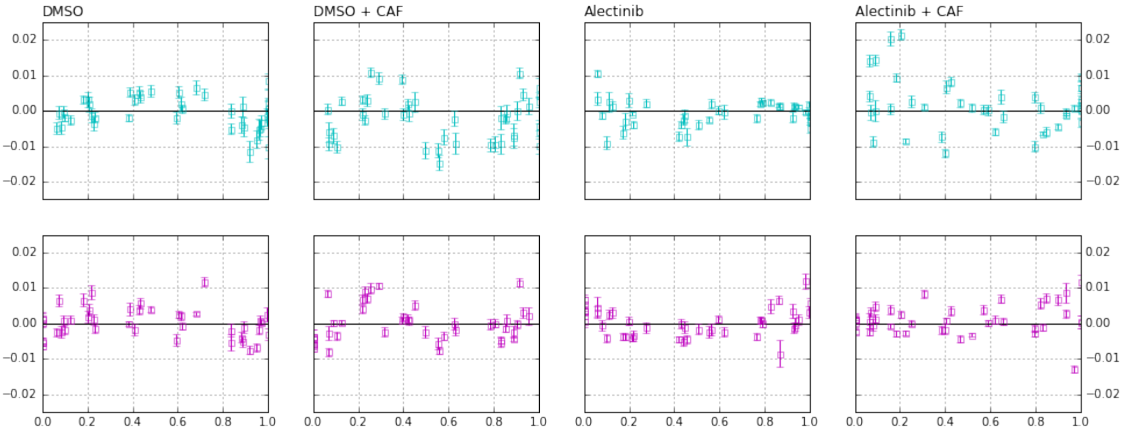


Figure 7.7: **Residuals for the fitness functions.** The x-axis is proportion and y-axis is residuals of the lines of best fit from Figure 7.4 for parental (Cyan,Top) and resistant (Magenta,Bottom).

resolve this and other questions.

7.12 Generalizing game assay to non-linear fitness functions

Regularization is a machine learning technique for reducing over-fitting by biasing towards more succinct models. It is the use of *a priori* knowledge on what constitutes a simpler or more likely model to anchor our inference. A classic example of this is preferring lower-order over higher-order polynomials for describing data unless there is overwhelming evidence otherwise. Of course, what constitutes overwhelming evidence depends on the goals of the scientists. If the only goal is prediction then cross-validation is a good way to test how heavily inference should be regularized. But if the goal is explanation then accordance with extant theory is an important factor to consider.

As such, our choice of focusing on fitness function (i.e., mappings from the strategy distribution in a population to the fitness of a strategy) that are linear in Sections 7.6 and 7.8 and Figure 7.4 can be seen as a form of regularization to accord with the extensive existing theoretical work on matrix games. In particular, we can see our inference procedure as either restricted to the hypothesis class of linear functions, or as considering the hypothesis class of all polynomials but with prohibitively high costs for non-zero components (l_0 regularization) on orders beyond linear. But we prefer to think of it in terms of operationalization. By introducing a game assay, we are defining the hidden variable of (matrix) games in terms of the measurement procedure that we described in Sections 7.2, 7.6, and 7.8.

Although slight deviations from a linear fit – that might not be attributable to noise alone – might be present in the data (see Figure 7.7), I do not think that they justify considering higher-order fitness functions (although I discuss higher-order functions briefly here for completeness).

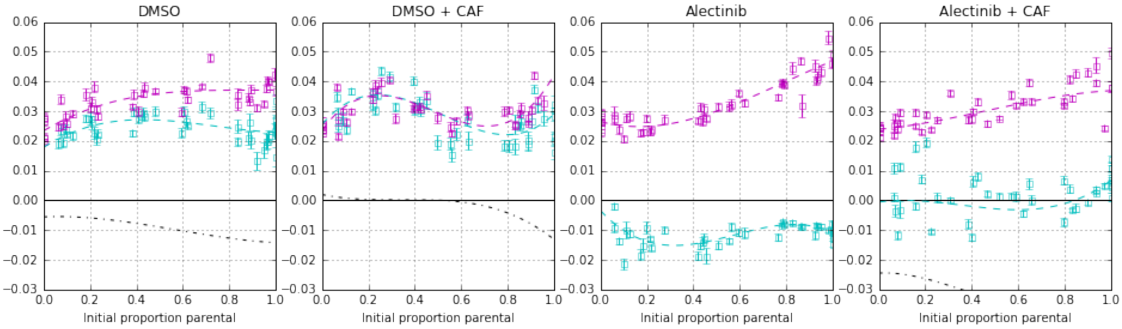


Figure 7.8: **Cubic Fitness functions for competition of parental vs. resistant NSCLC.** For each plot: growth rate with confidence intervals versus initial proportion of parental cells. This is the same data as Figure 7.4. Cyan data points are growth rates of parental cells, and magenta for resistant cells. Dotted lines represent the 3rd-order (cubic) fitness function of the least-squares best fit. The black dotted line is the gain function for parental (see Figure 7.5a), it is well below the $y = 0$ line in the Alectinib conditions (indicating the strong advantage of resistance) and thus cut out of the figure.

This is due to the higher explanatory value of linear models and my hope to influence the well-established study of matrix games in microscopic systems. Note that this linearity is not guaranteed to be a good description for arbitrary experimental systems. For example, the game between the two Betaproteobacteria *Curvibacter* sp. AEP1.3 and *Duganella* sp. C1.2 was described by a quadratic gain function [131, 106]. And although some good EGT work has recently been done on non-linear games [7, 8, 131], this is very little compared to the immense literature on matrix games. More importantly, I think that our focus on matrix games is better viewed not from the perspective of model selection but rather as an operational definition of effective games. I am not aiming to provide the best or most predictive account of non-small-cell lung cancer in the petri dish, but rather a method for measuring (matrix) games. If the error of the measured (matrix) games ends up very high – which is not the case from the error bars in Figure 7.5b – then we know that this first order approximation of interactions is not sufficient and higher orders should be pursued. However, we will not know this unless we first have a robust method for measuring the lower order terms.

Alternatively, if one views this work from the perspective of model selection then in this Chapter, I proceed from the assumption of linearity. Here, I relax this assumption, extend the game assay to non-linear games, and compare linear and non-linear models with information criteria. The qualitative results are unchanged, although the exact quantitative results for non-linear models differ slightly.

In particular, whereas Sections 7.6 and 7.8 used linear functions as the hypothesis class for fitting the growth-rate vs. proportion, one could use any other class of functions. An obvious candidate is polynomial fitness functions of orders higher than 1. We provide an example in

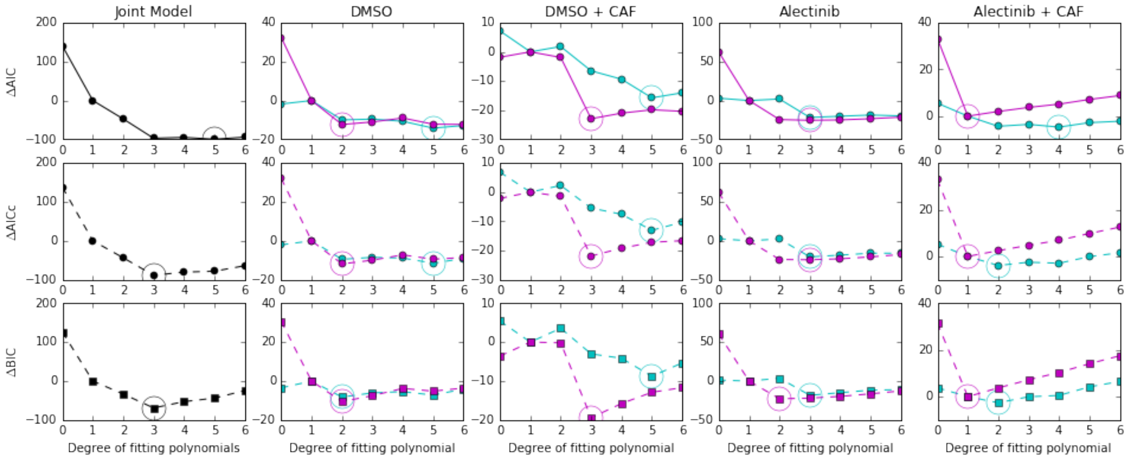


Figure 7.9: **AICs and BICs for best polynomial fits of a given degree (up to additive offset).** In cyan are AIC and BIC values for models parental cell fitness functions, while in magenta are AIC and BIC values for models of resistant cell fitness functions. Circles surround the minimum of AIC and BIC.

Figure 7.8 of a 3rd-order (cubic) fit. Visually, the cubic provides a better fit than the linear one in Figure 7.4, which is to be expected from the extra degrees of freedom. But qualitatively it provides the same interpretation as the linear fitness functions, including the same number of fixed points. In particular, DMSO + CAF has a single fixed point at $p = 0.52$ and (using the fixed region height from Section 7.11.1) a single fixed region for $p \in (0.04, 0.668)$. This is much like the linear fit, but the fixed point region is expanded. The other three conditions (DMSO, Alectinib, and Alectinib + CAF) still have no fixed points and no fixed regions.

7.12.1 Information criteria for non-linear fits

If we treat the game assay not as a measurement and definition of games but as a model selection problem for parameter fitting then it becomes important to quantify the trade-off between the goodness of fit and model simplicity. For this, we can use techniques like the Akaike information criterion (AIC), its small-sample size correction (AICc), or the Bayesian information criterion (BIC) – or any other statistical model selection procedure. Given that (i) a polynomial of degree d has $k = d + 2$ degrees of freedom as a statistical model (+1 for zeroth order term, +1 for noise term); the eight models (4 conditions, 2 fitness functions per condition) are trained on $n = 42$ data points each; and AIC/BIC only works reasonably when $n >> d$. For example, given our relatively small dataset for each model, Burnham and Anderson [26] would advocate to always prefer AICc over AIC (they suggest $n/k < 40$ as the cut off). Hence, we show the results of all three of AIC, AICc and BIC for polynomial fitness functions for degree $d \leq 6$ in Figure 7.9. In this figure, a better model corresponds to a lower AIC, AICc or BIC value (lower on the y-axis). Since

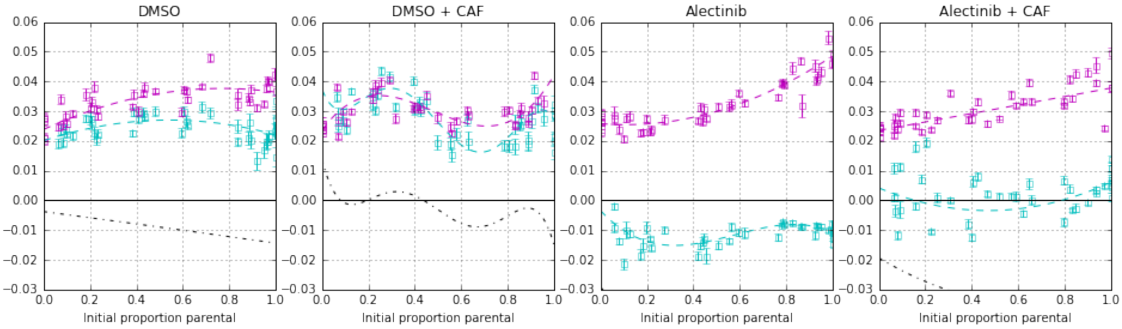


Figure 7.10: **Fitness functions for competition of parental vs. resistant NSCLC as selected by BIC.** For each plot: growth rate with confidence intervals versus initial proportion of parental cells. This is the same data as Figure 7.4. Cyan data points are growth rates of parental cells, and magenta for resistant cells. Dotted lines represent the fitness function of the least-squares best fit for models selected by BIC. These are a linear model for resistant fitness function in Alectinib + CAF; quadratic models for parental fitness function in Alectinib + CAF, resistant fitness function in Alectinib, and both fitness functions in DMSO; cubic for resistant in DMSO + CAF, and parental in Alectinib; and quintic for parental in DMSO + CAF. The black dotted line is the gain function for parental (see Figure 7.5a), it is well below the $y = 0$ in the Alectinib conditions (indicating the strong advantage of resistance) and thus not visible in the figure.

constant offsets in the information criteria do not matter for model selection, the axes are set so that the linear model has $\Delta\{\text{AIC}, \text{AICc}, \text{BIC}\} = 0$. The leftmost column of Figure 7.9 considers the joint product model where each fitness function has the same degree – for the $d = 1$ model, this would correspond to the linear game assay as presented in Section 7.6 and 7.8. Both AICc and BIC select the 3rd-degree polynomial model that we discussed above. AIC doesn't differentiate strongly between the 3rd, 4th, 5th and 6th degree, but prefers slightly the 5th degree. Too much emphasis should not be placed on AIC however, given the number of parameters compared to sample size [26]. The four right columns of Figure 7.9 consider independent models for each of the fitness functions across the 4 different conditions – so a total of 8 models. At the cost of extra researcher degrees of freedom, it is possible to look at the fits where the model for each of the 8 fitness functions is selected independently. Such a fit, as selected by BIC, is shown in Figure 7.10. Note the two extra crossings of zero by the gain function in the DMSO + CAF case.

7.12.2 Plotting nonlinear games

Just like with the linear games, it is possible to plot nonlinear games in our 2D game space based on the $p = 0$ and $p = 1$ endpoints of the gain function. We do this in Figure 7.11 with each point labeled by the degree of the corresponding polynomial fitness functions. Unsurprisingly, at a brief glance there is broad qualitative agreement – all (but one) points are in the same quadrant as the linear model – although little quantitative agreement with the linear game assay – most points

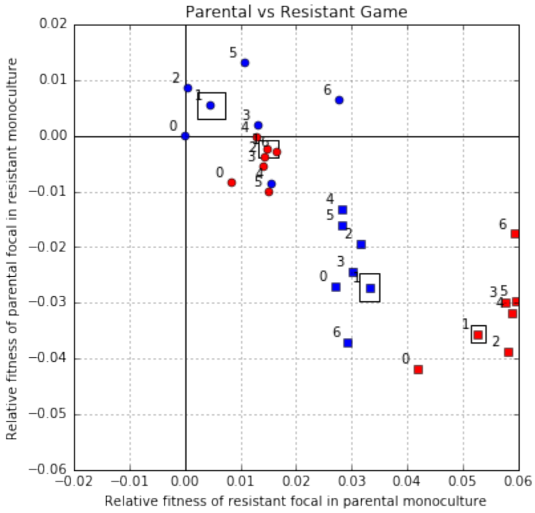


Figure 7.11: **Mapping of the AIC and BIC selected fitness functions for the four conditions into game space.** The x-axis is the relative fitness of a resistant focal in a parental monotypic culture. The y-axis is the relative fitness of a parental focal in a resistant monotypic culture. Games measured in our experimental system are specified by the bounding boxes corresponding to the range of their errors. The games corresponding to joint degree models are given as points, with joint degree labeled nearby. Experimental condition is represented by shape (DMSO: circle; Alectinib: square) and colour (no CAF: red; + CAF: blue).

are outside of the error-box corresponding to the linear game. However for a general nonlinear game, unlike with linear games, two points in the same quadrant might not correspond to the same qualitative kind of dynamic. In particular, for a general nonlinear game, a quadrant only tells us the parity of the number of roots in $(0, 1)$ – where roots are counted by their multiplicity – and the order of alternations on the flow. For more on discrete flow alternation representation of gain functions, see Peña, Lehmann, and Nöldeke [174].

Fortunately, for our particular experimental system the above generality is not realized. In particular, for all degrees of the DMSO, Alectinib, and Alectinib + CAF games the gain functions have no fixed points – just like the linear case. For DMSO + CAF, degrees $d = \{1, 2, 3\}$ have one fixed point and $d = \{5, 6\}$ have 3 fixed points (although only two fixed regions: $p \in \{(0.07, 0.25), (0.38, 0.46)\}$ for $d = 5$ and $p \in \{(0.09, 0.31), (0.44, 0.50)\}$ for $d = 6$. For $d = 0$ it is impossible for any model to be in the top right or bottom left quadrant – since no constant line can be both negative and positive – and there is no fixed point, but the fitness difference for DMSO is so tiny that there is a single fixed region for the whole space $p \in (0, 1)$. The real outlier for DMSO is $d = 4$ since it has two fixed points (and is thus in the bottom right quadrant) and two fixed regions at $p \in \{(0.07, 0.11), (0.29, 040)\}$. Thus, the existence of fixed point(s) in DMSO + CAF and absence of fixed points in the other conditions is robust across the nonlinear models. The exact position of the fixed point(s) in DMSO + CAF, however, is not as robust to model choice.

7.13 Summary

In Chapter 6, I introduced replicator dynamics with frequency-dependent fitness as a representation for the ecology of an evolving system and defined the idea of empirical abstraction. In this chapter, we got to see empirical abstraction in action by measuring the ecology of non-small cell lung cancer. I achieved this by defining the game assay as an experimental technique for measuring ecology in microscopic systems. In the process, we saw that Alectinib-sensitive and Alectinib-resistant cell types play two qualitatively different kinds of games (Leader vs Deadlock; see Section 7.9) based on the absence vs presence of Alectinib and cancer-associated fibroblasts. This empirically confirmed the previously theoretical postulate of evolutionary oncology: we can treat not just the player, but also the game (Section 7.10). In the process, I also challenged other theoretical postulates like costly-resistance (Section 7.5). I expect that this game assay will prove useful in systems beyond cancer. But even in cancer, I hope that in the near future we will be able to extend the four games measured in this chapter into a more comprehensive catalogue of the different games that different cancers play.

Chapter 8

Spatial structure and the multiple realizability of effective games

In this chapter, we finally reach an explicit discussion of spatial structure. This is important to study because, after all, with the possible exception of leukemia and lymphoma, cancers exist in space, adjacent to cells and boundaries, with daughter cells taking over the space of (or near) their mothers. And the importance of space is not limited to cancer. Spatial structure in evolutionary game theory is so important that a particular approach to it even has its own sub-field name: evolutionary graph theory [132, 207, 196, 137]. Durrett and Levin [47] and Shnerb et al. [199] have provided a particularly good demonstration of how much spatial structure and stochasticity can matter as they built from mean-field approaches (i.e., *inviscid* model – meaning that the probability to interact with a strategy is the same as the proportion of the strategy in the whole population; of which the inviscid replicator dynamics with which we started in Chapter 6 is an example), to patch models of discrete individuals, to reaction-diffusion equations, to full-fledged interacting particle systems. It has been shown that spatial structure can promote cooperation [159, 163, 116, 72], or inhibit it [73], or complicate the whole discussion around it [122, 176, 107]. Since we do not expect such complications in the mean-field, we can see from these results that spatial structure can drastically change the effect of the reductive game such that the mean-field analysis is completely inapplicable.

With all of these arguments for the importance of space, why does anybody even bother to think about inviscid replicator dynamics? Why does not everybody build spatial models? Even the argument of analytic intractability is no defence when we have pair-approximation techniques (and other alternatives) that can be applied with few arbitrary model commitments [145, 222, 162].

But there is more than one approach to dealing with spatial structure. In Section 8.1, I will introduce the traditional approach of working from the bottom up: transforming a reductive game through an incorporation of spatial structure. This mirrors the discussion of reductive games from Section 6.3 and it is the most common approach to incorporating spatial structure into EGT. I will highlight this bottom-up approach with a discussion of the classic Ohtsuki-Nowak transform for random k-regular graphs [162] alongside my novel interpretation of the transform as 1st-order approximation of any spatial structure [93, 115]. In Section 8.2, I will discuss an example of this bottom-up approach in cancer by briefly reviewing Kaznatcheev, Scott, and Basanta [115]'s work on spatializing the Go-vs-Grow game. Unfortunately, as with reductive games in Chapter 6, this bottom-up approach to space is not compatible with how direct measurements can be done in microbiology.

So to better align with experiment, I use Section 8.3 to advocate for starting at the effective games from Section 6.4 and working our way down towards the reductive game. In Section 8.4, I justify effective games as an abstract object through their multiple realizability by the combination of different reductive games and spatial structures. This can be especially important because spatial structure can both create (Section 8.4.1) and hide (Section 8.4.2) non-cell-autonomous interactions. Finally, in Section 8.5, I consider a proposal for how to experimentally extract the contribution of space from an effective game that would normally swallow-up or abstract the spatial structure into measurement. This aims to push the effective games down a level of abstraction. Unfortunately – unlike Chapter 7 for top-level effective games – I leave it for future work to actually realize these sorts of experimental measurements.

As with Chapter 6, the novelty of this chapter is in the way that I frame the distinction between reductive games vs effective games and the view of spatial structure as a transformation between them. By providing this new framing, I make clear the implicit direction from reductive to effective of much of the existing work on spatial structure. And I open the door for work in the opposite direction: from directly measured effective games to inferred reductive games. This allows me to use this chapter to conclude the empirical abstraction theme of this second part of the thesis.

8.1 Approximating space with Ohtsuki-Nowak transform

A typical study of space in evolutionary game theory will start with the reductive game and then simulate that interaction over a model of space to show a surprising difference in dynamics between the spatial model and the mean-field inviscid (i.e. non-spatial, zeroth-order) model with the same payoff matrix (i.e. reductive game). Particularly strong works like that of Ohtsuki and Nowak [162] (more recently, Nanda and Durrett [156]), provide a general method for combining a reductive game with spatial structure. Although in its original presentation, Ohtsuki and Nowak [162] focused on dynamics on k-regular random graphs, I think it can be useful to frame their work as a general first-order approximation of an arbitrary spatial structure.

First, let me present the replicator equation for reductive games as a zeroth-order or mean-field approximation of spatial structure. Without knowing anything about spatial structure, the roughest guess we can make of the probability of interacting with another agent is just to say that we sample agents from a distribution given by their proportion in the population – a mean-field approximation. This gives us a utility function for agents of type i as $[Gp]_i$ where G_{ij} is the payoff of an agent of type i interacting with an agent of type j and p is a vector of proportions of agents of each type. From here on in, as in Section 6.3, our hands are tied and the math forces us to replicator dynamics: $\dot{p}_i = p_i([Gp]_i - p^T Gp)$. But the perfect sampling used in the mathematics of fitness effectively makes interactions global: what might be considered as a 0th-order approximation of space.

To avoid assuming this global interaction, we can say that instead of the fitness being the mean-field $[Gx]_i$, we instead sample M interaction partners from the distribution given by x and use these local interaction groups for our fitness calculation. This would be our 0.5th-order approximation. In this case, Hilbe [76] showed that the result is still replicator dynamics (although with a different time-scale that is irrelevant to the functional form) but with a modified payoff matrix $G' = G + \frac{1}{M-2}(G - G^T)$, where G^T is the matrix transpose of G. This is a great way to reintroduce some local effects, but the groups of M agents are constantly re-sampled and fitness-competition still happens at a global level; in other words, there is no spatial structure. Hence the 0.5 and not 1.

To get things completely localized, we will assume a fixed population size, and make our replication procedure more explicit to make a first-order approximation of spatial structure. Since the population size is fixed, we can only get a new agent if an old one dies. This can be a severe limitation in general, but when this replacement idealization *is* acceptable, it gives us a great way to localize. Once a focal agent dies, there is some spatial neighborhood of the focal agent with k agents that compete for the focal spot (this is often represented as a graph, but a graph-theoretical model is not always necessary, as I discuss in Section 8.5). Now, we can have some extra information, instead of just keeping track of the proportion of agents of type i given by p_i, we can keep track of neighbors. More explicitly, we will use pair-approximation [145, 222] to keep track of proportion p_{ij}: the probability of seeing an agent of type i in the neighborhood of an agent of type j.

This tells us who is competing for the vacated spot, but these neighbouring agents can interact with agents at distance two from the vacated spot. Thus, to calculate the neighbouring agents fitness, we would need to know the probability p_{ijk} of seeing an agent of type i near an agent of type j and k. To update that probability, by similar logic, we would need to know more long range effects like p_{ijkl}, etc. Hence, for a first-order approximation, we truncate the series here and approximate the further effects by saying that $p_{ijk} = p_{ij}$. Since we assumed that the neighbors of the perished agent i are drawn from the same sort of distribution as the neighbors' neighbors, we have ignored extra correlations that might arise from looking out to distance two or more, hence the first order nature of the approximation. The approximation is only exact for infinite Bethe-lattices and relatively good for k-regular random graphs. If we were looking at other graph structures like grids then higher order terms would dominate.

The fun part of my presentation of the Ohtsuki-Nowak transform is that it shows us that we do not *need* to assume our spatial structure is a graph. We just need to think in terms of sampling k times from distributions of agents' neighbors in space with some locality assumptions on these distributions (i.e., the distribution becomes effectively biased in a way that makes it more likely to return as a focal's neighbour an agent that is similar to the focal – this is also known as local-child placement). Since Ohtsuki and Nowak [162] have a separation of time-scales in their analysis, effects of the graph beyond this random sampling are not preserved anyway. Thus, this approximation technique is generally applicable, even to strangely structured settings like a solid tumour. Of course, like any first-order approximation, if higher-order effects are important then the model will not agree with experimental data. However, if we just look at data without a first-order theory then we wouldn't even know that higher-order terms are important. Thus, the first-order approximation is always a good first step; if empirical results contradict it then at least we know where to look, second-order and higher correlations in the distributions of neighbors.

Working under these assumptions, Ohtsuki and Nowak [162] showed that we still get replicator dynamics but with a modified payoff matrix $G'' = \mathrm{ON}_k(G)$ given by:

$$\mathrm{ON}_k(G) = G + \underbrace{\frac{1}{k-2}(\vec{\Delta}\vec{1}^T - \vec{1}\vec{\Delta}^T)}_{\text{assortativeness from local dispersal}} + \underbrace{\frac{1}{(k+1)(k-2)}(G - G^T)}_{\substack{\text{finite sampling from} \\ \text{death-birth updating}}} \tag{8.1}$$

where $\vec{\Delta}$ is a vector of the diagonal elements of the game matrix G, i.e. $\vec{\Delta}_i = G_{ii}$ and $\vec{1}$ is the all ones vector; thus $\Delta\vec{1}^T$ $(\vec{1}\Delta^T)$ is a matrix with diagonal elements repeated to fill each row (column); and I have reordered the terms from Ohtsuki and Nowak [162]'s presentation to highlight the transform's logical structure. Note that we have k agents competing for a spot, and each one samples $k-1$ other agents (since one spot they are neighboring was just vacated) so $M = k(k-1)$. Thus, the last term is the Hilbe [76] finite sampling effect and the first term is the spatial structure.

To restate: the effective game $G'' = \mathrm{ON}_k(G)$ then has the same mean-field replicator dynamics as the reductive game G carried out on (a first-order approximation of) the spatial structure. In other words, Ohtsuki and Nowak [162] provide a transform from reductive to effective game. That effective game then has the same nonspatial dynamics as the reductive game played out on the spatial structure. This allows us to use our tools of evolutionary game theory to analyze the transformed game and thus learn something about the system implemented by the reductive game and spatial structure. We can think of this approach as bottom up: start with the reductive game, find the corresponding effective game (or other description of macroscopic population-level dynamics) and then use these as a prediction to compare against observed phenomena.

8.1.1 General Ohtsuki-Nowak transform as game-space transformation

If the Ohtsuki-Nowak transform is interpreted literally as applied to k-regular random graphs then it is important to note that it is sensitive not just to the degree k but also to the particular update rule used. I will use ν to index this second dimension for the microdynamic parameters. Above, I presented death-birth updating ($\nu = \frac{1}{k+1}$) – the most local rule: a focal agent is selected uniformly at random to be removed and its neighbours compete to replace it, with each neighbour's replacement probability proportional to its fitness. But Ohtsuki and Nowak [162] also consider other update rules like birth-death and imitation from Section 6.3.1. For imitation updating ($\nu = \frac{3}{k+3}$), the focal agent is not removed but instead copies one of its neighbours or itself proportional to fitness. For birth-death updating ($\nu = 1$), a focal agent is sampled according to its fitness and then selects one of its k neighbours uniformly at random (i.e. with probability $1/k$) as an alter with the focal agent's cloning itself to replace the alter. This lets me present the Ohtsuki-Nowak transform in a new general form in two parameters:

$$\mathrm{ON}_{k,\frac{1}{\nu}}(G) = G + \frac{1}{k-2}(\Delta\vec{1}^T - \vec{1}\Delta^T) + \frac{\nu}{k-2}(G - G^T) \tag{8.2}$$

Now, I want to apply the Ohtsuki-Nowak transformation to a general two strategy game:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{8.3}$$

Unlike the game space of Section 7.8 and Figure 7.5, I want to focus on the case when $A > D$ to build a UV-game space of cooperate-defect games [190, 107, 108] through the following

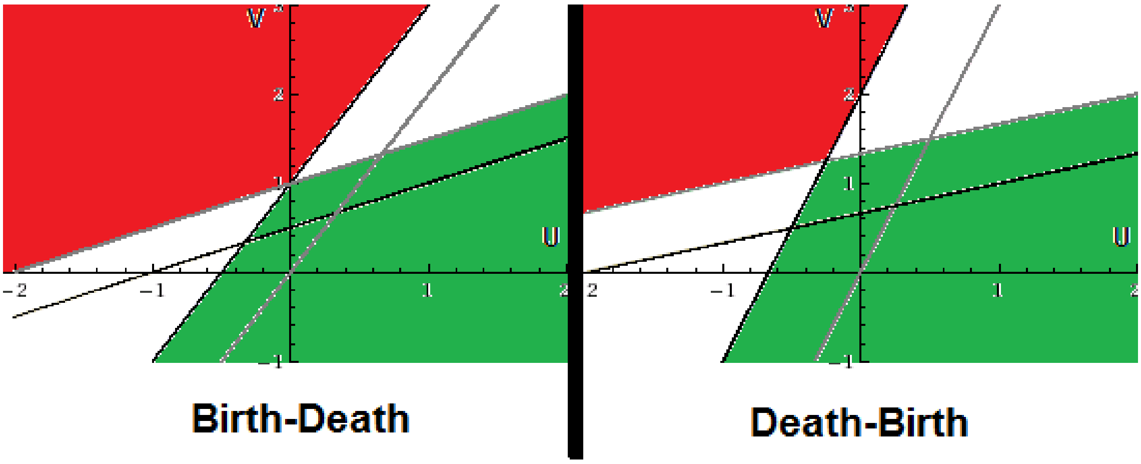


Figure 8.1: Ohtsuki-Nowak transform for birth-death and death-birth with $k = 3$ (so $\nu = \frac{1}{4}$) applied to a general reductive cooperate-defect game $\begin{pmatrix} 1 & U \\ V & 0 \end{pmatrix}$. The game space has the axes of U and V. The green region corresponds to when the effective fitness of strategy one is higher than that of strategy two. The red region corresponds to when the effective fitness of strategy two is higher than that of strategy one.

transformation:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \Rightarrow \begin{pmatrix} A-D & B-D \\ C-D & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & \frac{B-D}{A-D} \\ \frac{C-D}{A-D} & 0 \end{pmatrix} =: \begin{pmatrix} 1 & U \\ V & 0 \end{pmatrix} \tag{8.4}$$

where the first step notes that only relative fitness matters, and second step follows by changing the units in which fitness is measured. I can now apply the ON-transform from Equation 8.2 to all such cooperate-defect games to give us:

$$\mathrm{ON}_{k,\nu}(\begin{pmatrix} 1 & U \\ V & 0 \end{pmatrix}) = \begin{pmatrix} 1 & U \\ V & 0 \end{pmatrix} + \frac{1}{k-2}\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} + \frac{\nu}{k-2}\begin{pmatrix} 0 & U-V \\ V-U & 0 \end{pmatrix} \tag{8.5}$$

$$= \begin{pmatrix} 1 & X \\ Y & 0 \end{pmatrix} \tag{8.6}$$

where k is the degree and $\nu = 1$ for Birth-Death updating or $\nu = \frac{1}{k+1}$ for Death-Birth updating.

We can see equation 8.6 as a transformation from the U-V reductive game coordinates of Equation 8.2 to the X-Y effective game coordinate system given by the affine transform:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \frac{1}{k-2}\begin{pmatrix} k-2+\nu & -\nu \\ -\nu & k-2+\nu \end{pmatrix}\begin{pmatrix} U \\ V \end{pmatrix} + \frac{1}{k-2}\begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{8.7}$$

The effects of this coordinate transform on UV-space can be seen in Figure 8.1 for $k = 3$. The green region is where cooperation is the only evolutionary stable strategy, the red is where defection is the only ESS. The uncoloured regions have a fixed point of C-D dynamics; in the top right region of game space the point is an attractor, in the bottom left it is a point of bifurcation.

8.2 Typical bottom-up study: spatializing Go-vs-Grow game

As an example of the typical study of spatial structure in EGT, consider Kaznatcheev, Scott, and Basanta [115] – whose main results are presented in Figure 8.2 and that I will recap in this section. In this study, we started with the reductive Go-vs-Grow game [17] between the two strategies of invasive (INV; Go) and autonomously growing (AG; Grow) cells with the payoff matrix given in Equation 8.8 (as before, first player selects row and the second column, the payoff is for the first player). For the biological details and justifications of the Go-vs-Grow game, see Basanta, Hatzikirou, and Deutsch [17] and Kaznatcheev, Scott, and Basanta [115].

We wanted to know how the game was affected by different spatial structure in the bulk versus a static boundary of a tumour, so we transformed it according to the Ohtsuki and Nowak [162] transform:

$$\begin{array}{c c} & \begin{array}{cc} \text{INV} & \text{AG} \end{array} \\ \begin{array}{c} \text{INV} \\ \text{AG} \end{array} & \begin{pmatrix} \frac{1}{2}b + \frac{1}{2}(b - c) & b - c \\ b & \frac{1}{2}b \end{pmatrix} \end{array} \tag{8.8}$$

$$\downarrow$$

$$ON_k(\circ)$$

$$\downarrow$$

$$\begin{pmatrix} \frac{1}{2}b + \frac{1}{2}(b - c) & b\frac{2k-3}{2(k-2)} - c\frac{2k^2-k-1}{2(M-2)} \\ b\frac{2k-5}{2(k-2)} + c\frac{k+3}{2(M-2)} & \frac{1}{2}b \end{pmatrix} \tag{8.9}$$

where $M = k(k - 1)$ and $b \geq c \geq 0$.

Here, the reductive game from Equation 8.8 – if it was without spatial structure – has one of two possible dynamics:

- if $\frac{c}{b} \geq \frac{1}{2}$ then the AG cells are always fitter and we would expect a benign (i.e. non-invasive) tumour of autonomous growth cells, but

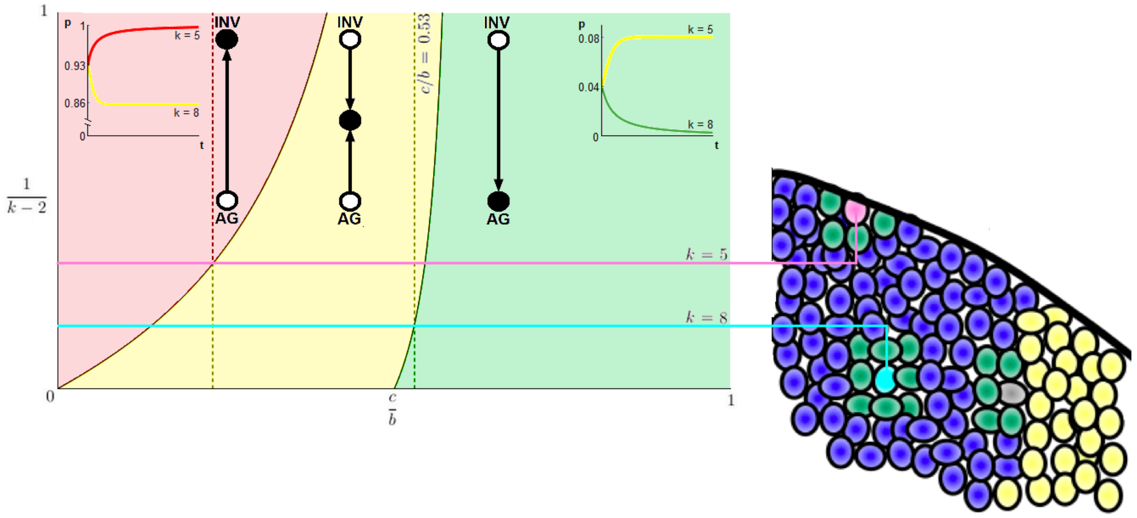


Figure 8.2: **Evolutionary game dynamics as a function of changing neighborhood size and relative cost of motility.** This figure is adapted from Kaznatcheev, Scott, and Basanta [115] and discussed in detail in Section 8.2. Here, I plot level of viscosity $\frac{1}{k-2}$ (where k is the degree of the random regular graph for the ON-transform) versus relative cost of motility $\frac{c}{b}$. The parameter space is divided into three regions that correspond to qualitatively different dynamics. In the red, the population evolves toward all INV; in the yellow – toward a polyclonal tumour of INV and AG cells; and in the green the tumour remains all AG.
When $\frac{1}{k-2} = 0$ (i.e., $k \to \infty$) we recover the standard inviscid replicator dynamics of previous work [17]: in this case the reductive and effective game are the same. For $\frac{1}{k-2} = 1$ (i.e., $k = 3$, the top edge of the plot), we have the environment with the smallest local neighbourhood to which the ON-transform applies. The top horizontal coloured line marks $k = 5$ (pink cell in the cartoon at right corresponding to the static edge of a tumour) and the bottom line is $k = 8$ (teal cell in the cartoon corresponding to the bulk of the tumour). The left vertical dotted line is at $c/b = 0.23$, and shows that it is possible to go from a polyclonal tumour in the bulk to a completely invasive population at a static edge. The right vertical dotted lines shows that is possible to see a qualitative shift from all AG to a polyclonal tumour in dynamics with the game fixed at $c/b = 0.53$ by decreasing k from 8 and 5 (increasing $\frac{1}{k-2}$ from 1/6 to 1/3) at the tumour boundary.
Example dynamics from a numerical solution of the replicator equation of the transformed game are shown in the insets, where the proportion of INV (p) is plotted versus time (t). The equation specifying the dynamics is $\dot{p} = p((ON_k(G)\vec{p})_1 - \vec{p}^T ON_k(G)\vec{p})$ where $\vec{p}^T = (p \;,\;\; 1-p)$, G is the game in Equation 8.9, and ON_k is the transform from Equation 8.1. The left inset corresponds to $c/b = 0.23$, an initial proportion of invasive agents of $p_0 = 0.93$, $k = 5$ (tumour edge) for the red line, and $k = 8$ (tumour bulk) for the yellow. The right inset corresponds to $c/b = 0.53$, an initial proportion of invasive agents of $p_0 = 0.04$, $k = 5$ (tumour edge) for the yellow line, and $k = 8$ (tumour bulk) for the green. The cartoon at right shows an example tumour, with a static boundary (like an organ capsule) in black, cancer cells in blue and healthy cells in yellow.

- if $\frac{1}{2} > \frac{c}{b}$ then the AG to INV cells can co-exist and we would expect a heterogeneous tumour.

This is the sort of conclusions that Basanta, Hatzikirou, and Deutsch [17] draw when analyzing the non-spatial game. But if we allow space then the effective game in Equation 8.9:

(a) shifts up the $\frac{1}{2}$ threshold by $\frac{1}{2(2k+1)}$ to $\frac{k+1}{2k+1}$ thus making co-existing of go and grow cells slightly easier; and – more importantly –

(b) creates a new dynamic regime of a fully invasive tumour when $\frac{k+1}{k^2+1} \geq \frac{c}{b}$.

This is summarized in Figure 8.2.

From (a) we drew the conclusion that spatial structure can make tumours more dangerous; and from (b) we drew the surprising conclusion that a tumour can have a much more invasive phenotype at the boundary – where k is lower – than the bulk – where k is higher [115].

8.3 Effective games and the confusion over spatial structure

But when we apply the typical pipeline: how do we know that the local interactions of the reductive game are the right ones to start with? In the example above, how do we know that the reductive Go-vs-Grow game is given by Equation 8.8? For macroscopic systems like human or other large animals, we might be able to directly observe or maybe even design the reductive game. In microscopic systems like cancer, however, we tend to guess these games from intuitions acquired by looking at population-level experiments. Unfortunately, these experiments – even when spatially structured like the images in Chapter 7 – seldom explicitly account for the effect of their spatial structure. Hence, intuitions based on these experiments are actually intuitions about the effective game that we then feed into our models as the reductive game. This is the common confusion about spatial structure in microscopic systems. The common procedure that we followed in Kaznatcheev, Scott, and Basanta [115] is taking a game from a top-level view, feeding it into the bottom level, getting a different result at the top-level and then publishing that surprising conclusion.

This is backwards. At best, it is just telling us that our intuitions about the game were wrong – since correct intuitions about the reductive game should yield the observed effective games. At worst, this is a category mistake and thus incoherent: we are putting an effective game where we should be putting a reductive game. To make the approximating spatial structure by techniques like the Ohtsuki-Nowak transform useful to microscopic systems like cancer, we have to invert the typical pipeline. We must start at the top with a carefully designed game assay to measure the effective game played by the population, or else design new assays that measure both the game and space together. This measured effective game encodes the combined effect of the reductive game, spatial structure, and other aspects of the experimental system. We can then push down by measuring spatial structure and for example inverting the transforms of Ohtsuki and Nowak [162]. In the case of two-strategy games, this can be done easily by inverting the affine transform from reductive to effective games in Equation 8.7 into one from effective to reductive games:

$$\begin{pmatrix} U \\ V \end{pmatrix} = \frac{1}{k - 2(1-\nu)} \begin{pmatrix} k-2+\nu & \nu \\ \nu & k-2+\nu \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} + \frac{1}{k-2(1-\nu)} \begin{pmatrix} -1 \\ 1 \end{pmatrix} \tag{8.10}$$

This removes the contribution of this kind of space, allowing us to arrive at a more reductive game. When this reductive game ends up contrary to our intuitions, then we are entitled to claim that we genuinely have a surprising conclusion.

Of course, this top-down pipeline should be carried out with a model of space that is more appropriate to the experimental system in question. For this purpose, I suggest a more easily parameterizable invertible model of space in Section 8.5. More importantly it should be used when learning a more reductive game is useful. In many cases – especially ones where space cannot be experimentally manipulated – this reductive game is not particularly useful and the effective game suffices. It is just important to not over-interpret this effective game because it could implemented by many different combinations of reductive games and spatial structure, as I now discuss.

8.4 Multiple realizability of effective games

To see the multiple-realizability of effective games, we can look at an experimentally measured effective game. Consider the Leader game from Chapter 7 that we measured in the DMSO + CAF case of the non-small cell lung cancer system:

$$\begin{pmatrix} 2.6 & 3.5 \\ 3.1 & 3.0 \end{pmatrix} \tag{8.11}$$

where parental is the strategy corresponding to the first row and column and resistant is the strategy for the second row and column.

There are several reductive games that could implement the effective game in Equation 8.11.

For a first example, if we thought that every cell interacted with every other cell and updating was done by imitation then the same exact matrix as above would be the reductive game.

For a second example, if we thought that our population lived on a 3-regular random graph and updated itself with death-birth dynamics then we could invert the ON-transform to calculate

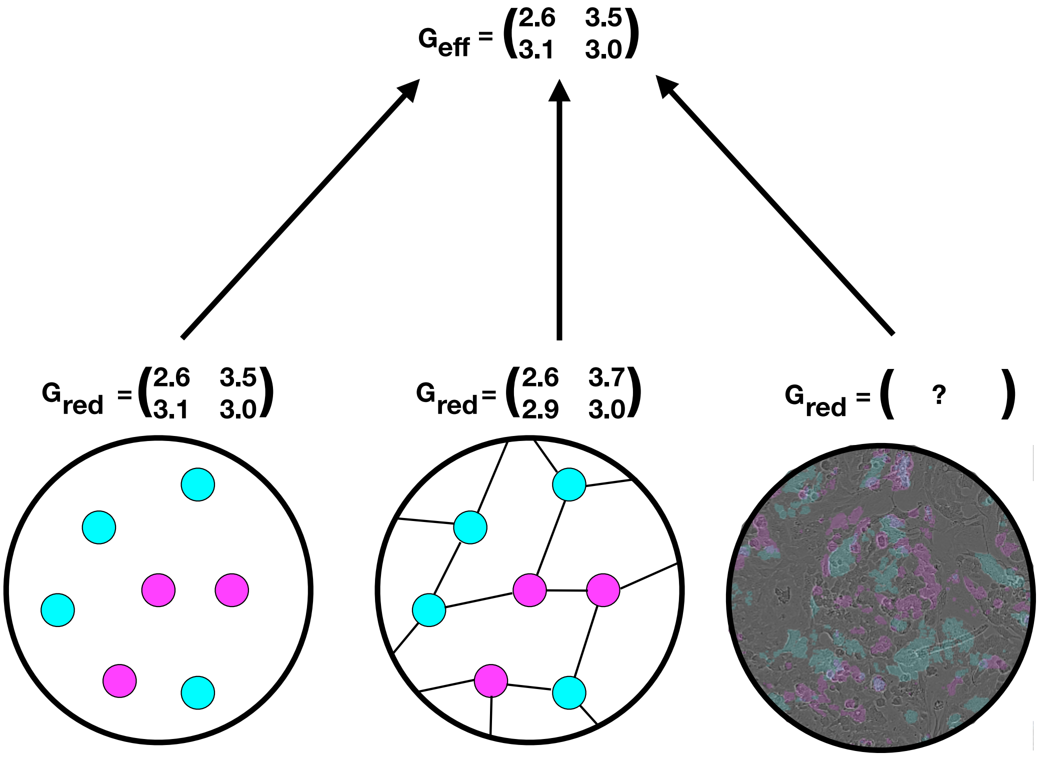

$G_{eff} = \begin{pmatrix} 2.6 & 3.5 \\ 3.1 & 3.0 \end{pmatrix}$
$G_{red} = \begin{pmatrix} 2.6 & 3.5 \\ 3.1 & 3.0 \end{pmatrix}$
$G_{red} = \begin{pmatrix} 2.6 & 3.7 \\ 2.9 & 3.0 \end{pmatrix}$
$G_{red} = (\; ? \;)$

Figure 8.3: **The same effective game implemented by three different population structures and reductive games.** From left to right: inviscid population, random 3-regular graph, experimental *in vitro* non-small-cell lung cancer.

the reductive game according to Equation 8.10. In this case, the reductive game corresponding to the above effective game would be:

$$\begin{pmatrix} 2.6 & 3.7 \\ 2.9 & 3.0 \end{pmatrix} \tag{8.12}$$

This might not seem like a huge change numerically, but it transforms a Leader game ($B > C > D > A$) into a Hawk-Dove game ($B > D > C > A$). In other words, if the effective Leader game that Kaznatcheev et al. [119] measured had been implemented in a well-mixed population then the corresponding reductive game would also be Leader but if it had been implemented in a slightly spatially structured population then the corresponding reductive game would be Hawk-Dove. Thus, two qualitatively different reductive games can implement the same abstract effective game depending on the spatial structure that we abstract over.

Of course, there is no reason to believe that either of these two spatial structures are a good description of our actual experimental system. Thus, we cannot currently take our measured

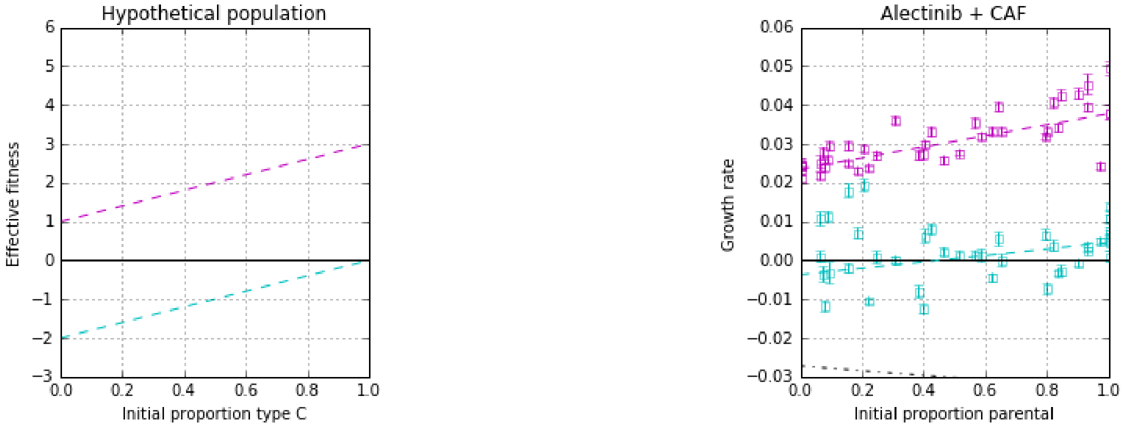


Figure 8.4: **A hypothetical (left) and empirical (right) measurement of effective fitness functions.** The x-axis is the proportion of type C (cyan) in the left graph and parental (cyan) in the right graph and the y-axis is the fitness of cyan and magenta types where magenta is type D for the left graph and resistant for the right graph.

effective game and "push it down" to a particular reductive game implementation. Instead, we can view nature as implementing some unknown transformation from reductive to effective game, as described at the end of Section 6.2. Thus, as summarized in Figure 8.3, we have at least three different potential reductive implementation of the same effective game – and many more are possible.

This multiple realizability is why effective games are a useful abstraction – the detailed implementation might not matter. But this rich multiple realizability is also a hazard: it means we need to be careful in our conclusions from and interpretations of reductive games. In Section 6.6, I already showed that the "interactions" of an effective game can be implemented by very indirect methods like differences in feeding rate and yield on a single common resource. In the following two subsections, I will show how space can create (Section 8.4.1) effective non-cell-autonomous fitness from reductive cell-autonomous fitness; and how space can hide (Section 8.4.2) non-cell-autonomous token fitness in a cell-autonomous type fitness.

8.4.1 Space can create effective games

Suppose that a scientist was using the game assay [119] from Chapter 7 to measure an effective game, and got the left graph of Figure 8.4 for the fitness functions of her two types. On the x-axis, she has seeding proportion of type C and on the y-axis she has fitness. In cyan (lower line) she has the measured fitness function for type C and in magenta (upper line), she has the fitness function for type D. The particular fitness scale of the y-axis is not important, not even the x-intercept – I have chosen them purely for convenience. The only important aspect is that the cyan and magenta lines are parallel, with a positive slope, and the magenta above the cyan.

This is not a crazy result to get, compare it to the fitness functions for the Alectinib + CAF condition measured in Figure 7.4 of Chapter 7 and which I show again at the right of Figure 8.4. There, cyan is parental and magenta is resistant non-small cell lung cancer. The two lines of best fit are not exactly parallel, but they are not that far off (they are within the experimental error of being parallel).

How should the scientist interpret Figure 8.4? Is there a game-like (i.e. non-cell autonomous) interaction happening there? The answer will depend on if she is asking about effective or reductive games, and what she knows about the population structure.

In Figure 8.4, there is clearly an effective game happening (i.e., the type fitness is not frequency-independent). As the proportion of type C increases, both type C and type D benefit. In fact, if we look at just the fitness functions then the story of a Public Good or a benefit-cost Prisoner's Dilemma game would be consistent with this graph. We can think of type C as paying a constant cost – the cost is the distance between the parallel lines – to produce a public good that benefits both type C and D cells equally – the benefit is the slope of the parallel lines.

If our scientist was to write down the corresponding game matrix, she would get:

$$\begin{pmatrix} 1 & 3 \\ -2 & 0 \end{pmatrix} \tag{8.13}$$

where type D corresponds to the first row and column, and type C corresponds to the second row and column. From this, she would see that this is not a traditional benefit-cost Prisoner's Dilemma since we need benefit $b = 2$ and cost $c = 3$, while we usually think of traditional PDs as having $b > c$. But this would still be a textbook example of a non-cell-autonomous process; even if not the most standard one. In fact, for the similar Alectinib + CAF condition shown at right, Kaznatcheev et al. [119] make this conclusion. We say that the fitness functions in this condition (as well as several other ones that are not close to conforming to parallel lines) are non-cell-autonomous.

Unfortunately, this is a conclusion about the effective games. And it does not need to hold for the reductive game.

Suppose that the scientist knows that the effective game she measured is actually implemented by a reductive game played on a random 3-regular graph with birth-death updating. In this case, consider taking the token fitness of type D cells to be 1 and the token fitness of type C cells to be 0 (i.e. each type has constant token fitness independent of who they interact with). This is a cell-autonomous fitness that most people would not describe as a game-like dynamic. If we insist on writing down this cell-autonomous fitness as a payoff matrix then we get the reductive payoff matrix:

$$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \tag{8.14}$$

Plugging this reductive game into Equation 8.7 for a 3-regular random graph with Birth-Death updating, results in an effective game given by:

$$\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 3 \\ -2 \end{pmatrix} \tag{8.15}$$

$$\downarrow$$

$$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \rightarrow ON_{3,1}(\circ) \rightarrow \begin{pmatrix} 1 & 3 \\ -2 & 0 \end{pmatrix} \tag{8.16}$$

where the final matrix is the same as Equation 8.13.

In words: two kinds of cells that each have cell-autonomous token fitness at the reductive level can have non-cell-autonomous type fitness at the effective level if their population is spatially structured. Spatial structure can create an effective game where there is no reductive game.

When the spatial structure is simple – as it is for the ON-transform – then the resulting effective game is also relatively simple. I would also expect this general 'effective game creation' result to hold for more complicated and realistic population structures. Except instead of creating a simple linear effective game of two parallel fitness functions, it could create a much more complicated non-linear game that depends on the details of the spatial structure itself. In other words, space on its own can create effective games – maybe even complicated ones. But my general result needs to be tested in future work that is beyond the scope of this thesis.

8.4.2 Space can hide reductive games

To make things even more difficult: if space can create an effective game from a constant fitness reductive 'game' then space can also hide a reductive game in a constant fitness effective 'game'.

Consider effective observations where the cell-autonomous type-fitness of type C is 0 and of type D is 1. This would correspond to the effective 'game' of $\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ or $f_C = 1$, $f_D = 0$. If we plug this into equation 8.10 with a 3-regular graph with Birth-Death updating then the resulting reductive game is non-cell-autonomous:

$$\frac{1}{3}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{3}\begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad = \quad \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix} \tag{8.17}$$

$$\downarrow$$

$$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \rightarrow \quad ON_{3,1}^{-1}(\circ) \quad \rightarrow \begin{pmatrix} 1 & 1/3 \\ 2/3 & 0 \end{pmatrix} \tag{8.18}$$

If our scientist was judging just from the effective fitness function then she might conclude that the system has a cell-autonomous fitness. But at the reductive level, it would actually have frequency dependent fitness that is masked by the spatial structure.

These examples show that if we are trying to make reductive statements from effective measurements of spatially-structured populations then we have another complication to consider in addition to the feeding-functions of Section 6.6. As before, this complication is important even if we are only interested in broad qualitative conclusions like distinguishing cell-autonomous vs. non-cell-autonomous processes.

8.5 Operationalizing spatial structure

Of course, I used the 3-regular graphs of the Ohtsuki-Nowak above only for illustrative purposes. I do not expect them to be useful models for the spatial structure of real experimental systems. Graphs, in general, seem to be an overly static, restrictive, and – most importantly – difficult to measure representation for the spatial structure of microscopic systems. Instead, we should think about what sort of spatial information is relevant and easy to measure and then use that as a starting point for operationalizing spatial structure. I propose that in this case we let experiment, rather than convenient existing theory lead. I propose starting with our measurement of effective games and pushing it down with an operationalization of spatial structure. We can achieve this by giving an experimental definition of the local environment of cells. This local perspective might be very different from the perspective that an experimenter has of the system as a whole.

Let us suppose that we are considering a system with two possible strategies A and B. Then we can define the following functions:

- Let $\rho_A : \mathbb{N} \times \mathbb{N} \to [0, 1]$ be the distribution over number of type A and type B partners that a cell with strategy A encounters; i.e. $\rho(k_A, k_B)$ is the probability of encountering k_A many cells of type A and k_B many cells of type B during the timescale relevant to the calculation of local fitness. (This is similar to p_{ij} in our discussion of the Ohtsuki-Nowak transform in Section 8.1)

- Let $\hat{\omega}_A : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ be the local fitness function for a cell of type A; i.e. $\hat{\omega}_A(k_A, k_B)$ is the local fitness of an agent of type A that encountered k_A many players of type A and k_B many players of type B during the timescale relevant to the calculation of local fitness.

Define $\rho_B, \hat{\omega}_B$ analogously for strategy B. This allows us to write down the replicator dynamics at the level of the whole population as $\dot{p} = p(1-p)\Gamma(p)$ where the gain function $\Gamma(p)$ is given by:

$$\Gamma(p) = \mathbb{E}_{(k_A,k_B) \sim \rho_A}[\hat{\omega}_A(k_A, k_B)] - \mathbb{E}_{(k_A,k_B) \sim \rho_B}[\hat{\omega}_B(k_A, k_B)] \tag{8.19}$$

Since $\rho_{\{A,B\}}$ depends on p, let us name this mapping $S(p;\vec{s}) \mapsto (\rho_A, \rho_B)$ and introduce an extra state vector $\vec{s}$ which might also change with time according to some general relationship: $\dot{\vec{s}} = T(\vec{s}; p)$. This relationship between p,$\vec{s}$ and ρ is meant to capture the functional role of space (or any other discrepancy between the local and global perspectives). The hope is that in practical settings, $\vec{s}$ is simple or non-existent or the dynamics of $\vec{s}$ can be decoupled from the dynamics of p by something like a separation of timescales, similar to what happens when using weak selection in the derivation of Ohtsuki and Nowak [162].

Suppose that we were able to find a system where the equations for $\vec{s}$ can be decoupled from p. Now, instead of having games on graphs, we will have games on experimentally measurable correlated frequencies. With this formalism, we can start to operationalize space. The first step is to measure the gain function $\Gamma(p)$ as described in Section 6.4.2. But measuring S is more difficult because it is encoding much more information about the microdynamical structure. A good first guess might be to take something like a structured core biopsy or time-lapse microscopy and define an interaction radius r. Then go through taking each cell as a focal agent and count how many cells of type A and B are within distance r of the focal agent. The result is an empirical estimate for ρ_A, ρ_B. Repeat for different initial p to get as many points of function S as desired. Clearly, if r is taken as the diameter of the slide then S will be an identity map since $p = \frac{\rho_A}{\rho_A + \rho_B}$ in that case. At the other extreme, taking r as less than a cell radius will make S into a constant map with $\rho_A = \rho_B = 1$. For intermediate values of r, we will potentially have a variance in different local densities for each focal agent, and picking a good r will depend on trade-offs between the level of error introduced by this variance versus the level of error that's introduced in the propagation from raw gain function to game.

Unfortunately, just like with other effective concepts, this operationalized spatial structure might not always have clear microdynamic interpretations. However, it does allow us to go one step closer to experimentally understanding the effects of spatial structure on populations without confusing effective and reductive games. In future work, it would be interesting to apply this method – or one like it – to spatially structured empirical populations like our experiments from Chapter 7, but such an application is beyond the scope of this thesis.

8.6 Summary

In Chapter 6, I showed the multiple realizability of replicator dynamics – the central algorithm of evolutionary game theory. In this chapter, I showed the multiple realizability of games due to spatial structure. In Chapter 6, I defined the notion of reductive games vs. effective games. In this chapter, we saw that spatial structure is one of the non-trivial transformations that can change a reductive game into a different effective game. Usually, this transformation is done implicitly from the bottom-up by starting with an assumed reductive game. I highlighted Kaznatcheev, Scott, and Basanta [115] as a cancer example of this in Section 8.2. But we can also work in the opposite direction. We can work top-down by starting with a careful measurement of an effective game (by, for example, using the game assay from Chapter 7) and pushing it ‘down’ through the spatial structure to infer a more reductive game. My hope is that this perspective opens paths for new kinds of empirical work measuring evolutionary games in cancer and other microscopic systems.

Chapter 9

Conclusion: abstracting the Darwinian engine

> Nothing in evolution or ecology makes sense except in the light of the other
>
> Pelletier, Garant, and Hendry [173]

Let us return to the Darwinian engine that powers evolution. The Darwinian engine in Figure 9.1a is made of two cycles that together change the distribution of genotypes. On the top, we have the genesis of new variants via the mutation cycle. On the bottom, we have the struggle for existence via the development-ecology-selection cycle. This figure is already a simplification of the kinds of feedbacks that evolutionary biologists care about. But even in this simplification, a lot of details are hidden in each edge. So in this thesis, I idealized the Darwinian engine further to focus on specific aspects of its operation. I represent that as the two projections of the Darwinian engine to the right and down. Figure 9.1b corresponds to Part I and Figure 9.1c corresponds to Part II of the thesis. In each case, I followed the mathematical imagination of Section 1.3: I idealized one part of the engine (dashed line) so that I could abstract over the other part. And in each case, we can divide the projection into a part that correspond to a 'problem' (the space on or according-to which things happen) and another that correspond to an 'algorithm' (the dynamics that happen) as I described in Section 3.5. This idealization, abstraction, and division allowed me to engage the techniques of theoretical computer science in analyzing the Darwinian engine. In this final chapter, I want to review and synthesize what we learned from algorithmic biology in this thesis while also setting forth future directions for this nascent field.

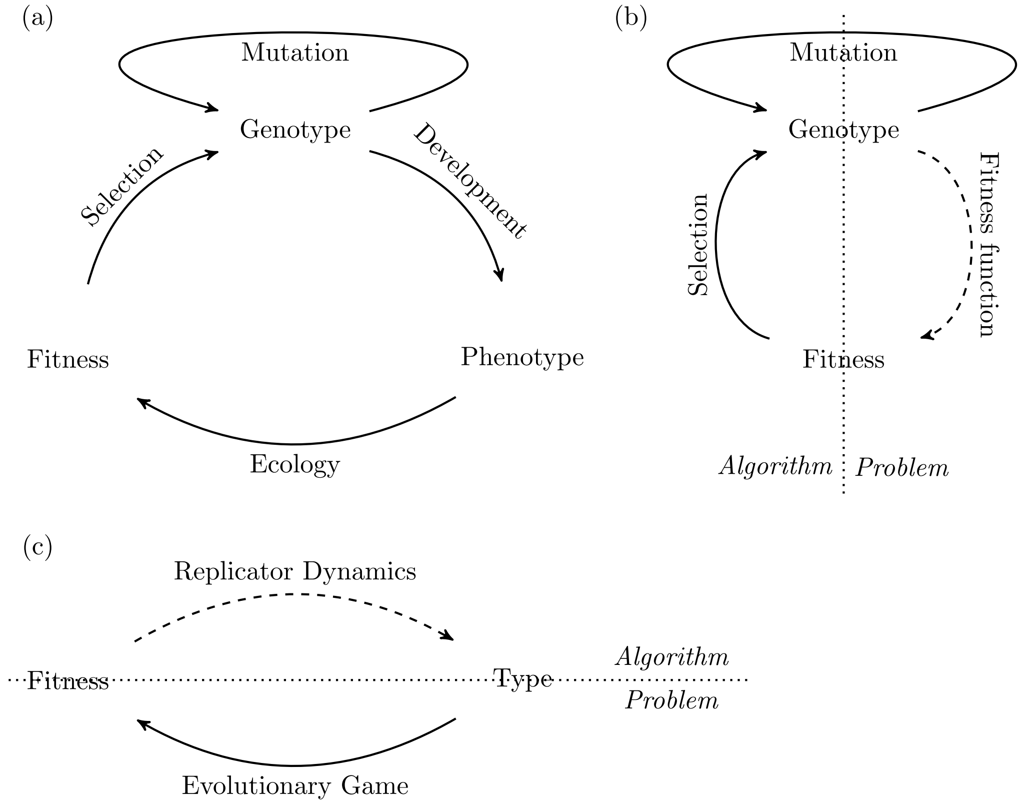


Figure 9.1: **Darwinian engine powering (a) eco-evolutionary dynamics alongside two projections focused on (b) evolution and (c) ecology.** The bottom cycle of (a) captures the struggle for existence (it is inspired by similar figures from presentations by Joachim Krug and Amitabh Joshi), and the top cycle of (a) captures the genesis of new variants. Panel (b) is the evolution on fitness landscape projection of (a) that I use for Part I of the thesis. Panel (c) is the ecology from evolutionary game projection of (a) that I use for Part II of the thesis. The idealizations that allow the projections are represented by dashed arrows. Dotted lines show how each projection can be divided into algorithm vs problem. This is the same figure as Figure 1.3.

9.1 Evolution on fitness landscapes

As Dobzhansky [45] famously quipped in 1964: "Nothing in biology makes sense, except in the light of evolution." Given that I want to build an algorithmic biology, it makes sense to first look at evolution. So for Part I of the thesis (Chapters 2, 3, 4 & 5), I summarized the contribution of both development and ecology as a fitness function from genotype to scalar fitness. This summary was made as an idealization, since it replaced the multifaceted idea of fitness by a one-dimensional scalar (which I discussed the limits of in the next part at the start of Chapter 7). But this idealization gave certain benefits that opened the door for theoretical abstraction.

1. It allowed me to divide the two cycles of the Darwinian engine into two parts familiar to computer scientists: problem and algorithm. The edges leading out of the genotype node serve as the 'problem': they are the mutation graph and fitness function that together specify the fitness landscape. The edges leading into the genotype node serve as the algorithm: they determine which mutations are considered and selected and thus how the population moves through the fitness landscape.

2. The view of fitness as scalar, made the set of problems – i.e., the space of all fitness landscapes – into a manageable and analyzable space. In Chapters 2, 4 and Section 3.2 I divided fitness landscapes into three coarse classes of smooth, semismooth, and rugged landscapes based on the maximum allowed epistasis. In the rest of Chapters 3 and 5, I focused on the more fine-grained epistatic structure by introducing the idea of gene-interaction networks (or valued constraint satisfaction problem instances) as a representation of fitness landscapes.

3. By defining families of fitness landscapes as 'problems', I could now analyze the performance of arbitrary evolutionary dynamics as algorithms. In Chapter 2, I introduced my theory of hard landscapes by noting that even local fitness peaks can take exponentially long to reach in some families of fitness landscapes. This has important consequences for open-ended evolution and various biological puzzles around adaptationism. In Chapter 4, I formalized my theory and showed that hard landscapes exist for progressively more general algorithms; starting from fittest-mutant and random fitter-mutant strong-selection weak-mutation dynamics, to any adaptive dynamic, to – finally – any evolutionary dynamic.

4. Since hard landscapes exist, we can no longer take the easiness of landscapes as a given. Instead, we need to formally prove that any given family of landscapes is easy or not for some set of evolutionary dynamics. So in Chapter 5, I developed a new proof technique of

encouragement paths to show that any fitness landscape representable by a binary Boolean tree-structured gene-interaction network is easy: such a landscape cannot contain an adaptive path longer than $\binom{n+1}{2}$ fixations (Theorem 5.15). I also showed that this new proof technique is strictly more powerful than prior techniques based on span arguments (Section 5.1).

5. The existence of provably hard and provably easy families of fitness landscapes, pushed me to define complexity and simplicity classes of fitness landscapes in Section 5.4. This allowed me to refine the easy-to-hard spectrum and start to populate a taxonomy of easiness and hardness for families of fitness landscapes based on the structure of their gene-interaction networks in Figure 5.1.

Refining this taxonomy is the most natural next step for computer scientists passionate about theoretical abstraction. For example, in Section 4.4, I gave a recursive construction for winding semismooth fitness landscapes that are hard for fittest-mutant strong-selection weak-mutation. But I also showed in Section 4.5 that this landscape cannot be represented by a gene-interaction networks of bounded treewidth. In contrast, Cohen et al. [32] gave an explicit construction for landscapes with gene-interaction networks of treewidth 7 that are hard for fittest-mutant SSWM but are not semismooth. This raises the question: are there hard semismooth fitness landscapes with gene-interaction networks of bounded treewidth?

The most prominent open question in the taxonomy, however, is the exact hardness of the classic NK-model with $K = 1$. For $K = 0$ it is a smooth landscape and, for $K = 2$ it is PLS-complete (Theorem 4.25). With $K = 1$, the classic NK-model differs by a single Boolean constraint from tree-structured gene-interaction networks (Proposition 3.11), so I suspect it is a very easy fitness landscape. Specifically, I conjecture that all adaptive paths are short in the biallelic classic NK-model with $K = 1$, but this cannot be established by the encouragement path arguments from Section 5.2. Future work would need to extend the encouragement path arguments, maybe combining them with the span arguments, to prove this conjecture.

Unfortunately, such questions might not be immediately exciting for a biologist. For biologists, a more pressing issue is to push toward empirical abstraction and learn what kind of gene-interaction networks occur in nature. In Section 3.7.1, I provided a simple algorithm for inferring gene-interaction networks from local fitness landscapes. Future works needs to extend this algorithm to handle noisy measurements and apply it to experimental systems. Once we have measurements of some empirical gene-interaction networks, we can see where they fit into the taxonomy of easiness and hardness. This will allow us to learn if the ultimate constraint of computational complexity is also a major (i.e., ubiquitous) constraint on evolution.

9.2 Ecology from evolutionary games

A half-century after Dobzhansky [45], Grant and Grant [69] responded: “Nothing in evolution makes sense, except in the light of ecology.” So for Part II of the thesis (Chapters 6, 7 & 8), I critically examined the idealization of fitness as a scalar from Part I that had eliminated biotic interactions. To represent the frequency-dependent fitness that fitness landscapes had idealized away, we combined genotypes and phenotypes as just types, summarized ecology as an evolution game and idealized the selection-mutation process as replicator dynamics. This framing of evolution opened the door for empirical abstraction and direct measurement.

1. This framing let me combine the two cycles of the Darwinian engine into one and then, as in Part I, I divided that one cycle into two parts familiar to computer scientists: problem and algorithm. But in this case, it was the algorithm of evolution that I idealized as replicator dynamics. Although, as I showed in Chapter 6, this idealized algorithm of replicator dynamics has many implementations, both reductive ones that are usually linked to computer simulations of agent-based models (Section 6.3) and effective ones that can linked to experiment (Section 6.4).

2. By asking “what is the problem ‘solved’ by this idealized algorithm”, I was able to abstract over ecology – summarizing the biotic ecology as an evolutionary game. This allowed me to define the game assay in Chapter 7 for summarizing ecology by directly measuring effective evolutionary games in microscopic systems. In Chapter 8, I showed that these abstract effective games are multiply realizable by different combinations of reductive games and spatial structure.

3. By having a game assay, I could directly measure the ecology between drug-sensitive and drug-resistant types of non-small cell lung cancer in Chapter 7. The result across four different conditions was four quantitatively distinct effective games of two qualitatively distinct kinds: Leader and Deadlock (Section 7.9). Since the qualitative game shift corresponded to the introduction of drug or the removal of cancer-associated fibroblasts, this observation provided the first empirical realization for a previously theoretical postulate of EGT in oncology: we can treat not just the player but also the game (Section 7.10). But the measurements also challenged a common theoretical postulate that drug-resistance is a neutral or costly adaptation outside of drug, since in our system, it carried a ‘negative cost’ (Section 7.5).

The Leader and Deadlock games have not previously been studied by the EGT in oncology community. So the observation of them in an experimental system will hopefully motivate new theory. The reason that Leader and Deadlock might not have appeared before is due to the focus of prior work being on reductive games, and the measurements being of effective games. It is important to not over-interpret effective games as reductive games. In particular, one should be careful given that the abstract effective games are multiply realizable by different combinations of reductive games and spatial structure. To connect better to the existing reductive literature, future work must push the abstraction of effective games 'down' through spatial structure (as I discussed briefly in Section 8.5).

So far, we have only measured four games of two qualitative types. Going forward, it is important to expand our catalog of evolutionary games, both in cancer and other microscopic systems. I am already working on this with two different teams from the Moffitt Cancer Center and Cleveland Clinic. The real test for the usefulness of effective games will be the outcome of these further measurements. In particular, once we know the games corresponding to many different cancers in many different conditions, will we see useful patterns or will we have to abandon games for another representation of the biotic ecology? Of particular importance to the 'don't treat the player, treat the game' approach of EGT in oncology, is for future work to gather enough game measurements to start looking at drugs as game transforms (i.e., as functions that take one effective game as input and give another effective game as output). This will require both theoretical and empirical developments.

Finally, although the experimental part of the game assay is straightforward to carry out for *in vitro* systems, it is not clear how to translate this work to the patient. Future work will need to develop analogs to the game assay for systems where a series of different initial proportions of types cannot be seeded. In particular, it is important to build techniques for learning evolutionary games from the sort of sparse historic data that might be available in a patient history. And since EGT is not limited to just microscopic systems, I hope that such techniques can also be applied to macroscopic systems with sparse historic data like human evolution. It would be nice to be able to measure the evolutionary games that might have shaped the evolution of human brains and behavior instead of just assuming that everything is a Prisoner's Dilemma [117].

9.3 Unifying evolution and ecology

Beyond the opportunities for further development described above, a key remaining challenge is to thoroughly unify evolution and ecology because – as Pelletier, Garant, and Hendry [173] noted – "nothing in evolution or ecology makes sense, except in light of the other." Future work can aim to look at the Darwinian engine through the algorithmic lens in a way that captures both the rich combinatorial structure of evolution on fitness landscapes (Part I) and the ecological feedback of evolutionary games (Part II). In Kaznatcheev [101], I have started working in this direction by defining *game landscapes* as an ecological extension of Valiant [221]'s algorithmic Darwinism. The major practical difficulty is to find effective ways to combine the fitness-as-function approach of evolutionary game theory with the vast size and rich combinatorial structure of fitness landscapes. Extending evolutionary games from a small number of types to an exponentially large landscape, makes it impossible to carry out a single exhaustive measurement like the game assay from Chapter 7. So akin to the local fitness landscapes, the next empirical step on this path is to find ways to operationalize and measure local game landscapes.

The next theoretical step on this path is to find a compact representation for game landscapes akin to the representation of fitness landscapes by gene-interaction networks from Chapter 3. The goal would be to find a representation that is both theoretically rich and still learnable from local game landscapes in the same way that gene-interaction networks are learnable from local fitness landscapes (Section 3.7.1). If this could be achieved, then future research could combine both evolution and ecology, and also theoretical and empirical abstraction. That way, we can hope to one day say that nothing in biology makes sense, except in the light of evolution and ecology focused by the algorithmic lens.

References

[1] M. Abrams. “Measured, modeled, and causal conceptions of fitness”. In: *Frontiers in Genetics* 3 (2012), p. 196.

[2] C. Adami, J. Schossau, and A. Hintze. “Evolutionary game theory using agent-based methods”. In: *Physics of life reviews* 19 (2016), pp. 1–26.

[3] N. Amenta and G. M. Ziegler. “Shadows and slices of polytopes”. In: *Proceedings of the twelfth annual symposium on computational geometry.* 1996, pp. 10–19.

[4] A. R. Anderson, A. M. Weaver, P. T. Cummings, and V. Quaranta. “Tumor morphology and phenotypic evolution driven by selective pressure from the microenvironment”. In: *Cell* 127.5 (2006), pp. 905–915.

[5] S. W. Angrist. “Perpetual motion machines”. In: *Scientific American* 218.1 (1968), pp. 114–123.

[6] A. Araujo, L. M. Cook, C. C. Lynch, and D. Basanta. “An integrated computational model of the bone microenvironment in bone-metastatic prostate cancer”. In: *Cancer research* 74.9 (2014), pp. 2391–2401.

[7] M. Archetti. “Evolutionary game theory of growth factor production: implications for tumour heterogeneity and resistance to therapies”. In: *British Journal of Cancer* 109.4 (2013), pp. 1056–1062.

[8] M. Archetti, D. A. Ferraro, and G. Christofori. “Heterogeneity for IGF-II production maintained by public goods dynamics in neuroendocrine pancreatic cancer”. In: *Proceedings of the National Academy of Sciences* 112.6 (2015), pp. 1833–1838.

[9] M. Archetti, D. A. Ferraro, and G. Christofori. “Heterogeneity for IGF-II production maintained by public goods dynamics in neuroendocrine pancreatic cancer”. In: *Proceedings of the National Academy of Sciences* 112.6 (2015), pp. 1833–1838.

[10] M. Archetti and K. J. Pienta. “Cooperation among cancer cells: applying game theory to cancer”. In: *Nature Reviews Cancer* 19.2 (2019), pp. 110–117.

[11] A. Ariew. “Ernst Mayr’s ‘ultimate/proximate’ distinction reconsidered and reconstructed”. In: *Biology and Philosophy* 18.4 (2003), pp. 553–565.

[12] S. Arora, E. Hazan, and S. Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications.” In: *Theory of Computing* 8.1 (2012), pp. 121–164.

[13] J. M. Baldwin. “A new factor in evolution”. In: *The american naturalist* 30.354 (1896), pp. 441–451.

[14] C. Bank, S. Matuszewski, R. T. Hietpas, and J. D. Jensen. “On the (un)predictability of a large intragenic fitness landscape”. In: *Proceedings of the National Academy of Sciences* 113.49 (2016), pp. 14085–14090.

[15] N. Barlow. *The Autobiography of Charles Darwin 1809-1882.* Collins London, 1958.

[16] N. Barton and L. Partridge. “Limits to natural selection”. In: *BioEssays* 22.12 (2000), pp. 1075–1084.

[17] D Basanta, H Hatzikirou, and A Deutsch. “Studying the emergence of invasiveness in tumours using game theory”. In: *The European Physical Journal B* 63.3 (June 2008), pp. 393–397.

[18] D. Basanta, J. G. Scott, M. N. Fishman, G. Ayala, S. W. Hayward, and A. R. Anderson. “Investigating prostate cancer tumour–stroma interactions: clinical and biological insights from an evolutionary game”. In: *British Journal of Cancer* 106.1 (2012), pp. 174–181.

[19] D. Basanta, J. G. Scott, R. Rockne, K. R. Swanson, and A. R. Anderson. “The role of IDH1 mutated tumour cells in secondary glioblastomas: an evolutionary game theoretical view”. In: *Physical biology* 8.1 (2011), p. 015016.

[20] E. Birmele. “Tree-width and circumference of graphs”. In: *Journal of Graph Theory* 43.1 (2003), pp. 24–25.

[21] L. Boltzmann. “Entgegnung auf die wärmetheoretischen Betrachtungen des Hrn. E. Zermelo”. In: *Annalen der physik* 293.4 (1896), pp. 773–784.

[22] T. Börgers and R. Sarin. “Learning through reinforcement and replicator dynamics”. In: *Journal of Economic Theory* 77.1 (1997), pp. 1–14.

[23] J. H. Brown, J. F. Gillooly, A. P. Allen, V. M. Savage, and G. B. West. “Toward a metabolic theory of ecology”. In: *Ecology* 85.7 (2004), pp. 1771–1789.

[24] S. Bullock. “The Invention of an Algorithmic Biology”. In: *Richard Dawkins: how a scientist changed the way we think: reflections by scientists, writers, and philosophers.* Ed. by A. Grafen and M. Ridley. Oxford University Press, USA, 2007.

[25] F. H. Burkhardt and S. Smith, eds. *The Correspondence of Charles Darwin vol. 6: 1856-1857.* Cambridge University Press, 1990.

[26] K. P. Burnham and D. R. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach.* Springer Science & Business Media, 2003.

[27] S. Carnot, E. Clapeyron, and R. Clausius. *Reflections on the Motive Power of Fire: And Others Papers on the Second Law of Thermodynamics by E. Clapeyron and R. Clausius.* Ed. by E. Mendoza. New York: Dover Publications, 1960.

[28] P. Chapdelaine and N. Creignou. “The complexity of Boolean constraint satisfaction local search problems”. In: *Annals of Mathematics and Artificial Intelligence* 43.1 (2005), pp. 51–63.

[29] K. Chatterjee, A. Pavlogiannis, B. Adlam, and M. A. Nowak. “The time scale of evolutionary innovation”. In: *PLoS computational biology* 10.9 (2014), e1003818.

[30] H. H. Chou, H. C. Chiu, N. F. Delaney, D. Segre, and C. J. Marx. “Diminishing returns epistasis among beneficial mutations decelerates adaptation.” In: *Science* 6034 (332 2011), pp. 1190–1192.

[31] A. Church. “An unsolvable problem of elementary number theory”. In: *American Journal of Mathematics* 58.2 (1936), pp. 345–363.

[32] D. A. Cohen, M. C. Cooper, A. Kaznatcheev, and M. Wallace. “Steepest ascent can be exponential in bounded treewidth problems”. In: *Operations Research Letters* (2020).

[33] V. Conitzer. “The exact computational complexity of evolutionarily stable strategies”. In: *International Conference on Web and Internet Economics.* Springer. 2013, pp. 96–108.

[34] M. C. Cooper, S. De Givry, and T. Schiex. “Optimal Soft Arc Consistency”. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence.* IJCAI’07. 2007, pp. 68–73.

[35] D. Corus, D.-C. Dang, A. V. Eremeev, and P. K. Lehre. “Level-based analysis of genetic algorithms and other search processes”. In: *IEEE Transactions on Evolutionary Computation* 22.5 (2017), pp. 707–719.

[36] A. Couce and O. A. Tenaillon. “The rule of declining adaptability in microbial evolution experiments”. In: *Frontiers in Genetics* 6 (2015), p. 99.

[37] Y. Crama and P. L. Hammer. *Boolean functions: Theory, algorithms, and applications.* Cambridge University Press, 2011.

[38] B. J. Crespi. “The evolution of maladaptation”. In: *Heredity* 84.6 (2000), p. 623.

[39] K. Crona, D. Greene, and M. Barlow. “The peaks and geometry of fitness landscapes.” In: *Journal of Theoretical Biology* 317 (2013), pp. 1–10.

[40] C. Darwin. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life.* John Murray, 1859.

[41] C. Darwin and A. R. Wallace. “On the tendency of species to form varieties: and on the perpetuation of varieties and species by natural means of selection”. In: *Journal of the Proceedings of the Linnean Society of London* 3 (1858), pp. 45–62.

[42] J. de Visser, S. Park, and J. Krug. “Exploring the effect of sex on empirical fitness landscapes.” In: *The American Naturalist* (2009).

[43] A. Dhawan, D. Nichol, F. Kinose, M. E. Abazeed, A. Marusyk, E. B. Haura, and J. G. Scott. “Collateral sensitivity networks reveal evolutionary instability and novel treatment strategies in ALK mutated non-small cell lung cancer”. In: *Scientific Reports* 7 (2017).

[44] H. Dircks. *Perpetuum Mobile: Or, A History of the Search for Self-motive Power from the 13th to the 19th Century.* E. & FN Spon, 1870.

[45] T. Dobzhansky. “Biology, molecular and organismic”. In: *American Zoologist* (1964), pp. 443–452.

[46] J. Domingo, G. Diss, and B. Lehner. “Pairwise and higher-order genetic interactions during the evolution of a tRNA”. In: *Nature* 558.7708 (2018), p. 117.

[47] R. Durrett and S. Levin. “The importance of being discrete (and spatial)”. In: *Theoretical population biology* 46.3 (1994), pp. 363–394.

[48] J. Fearnley, S. Gordon, R. Mehta, and R. Savani. “Unique end of potential line”. In: *Journal of Computer and System Sciences* (2020).

[49] R. A. Fisher. *The genetical theory of natural selection.* Oxford University Press, 1930.

[50] J. Franke, A. Klozer, J. de Visser, and J. Krug. “Evolutionary accessibility of mutation pathways.” In: *PLoS Comp. Biol.* 7 (8 2011), e1002134.

[51] R. Gandy. “Church’s thesis and principles for mechanisms”. In: *Studies in Logic and the Foundations of Mathematics.* Vol. 101. Elsevier, 1980, pp. 123–148.

[52] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco, CA.: Freeman, 1979.

[53] C. F. von Gärtner. *Versuche und Beobachtungen über die Bastarderzeugung im Pflanzenreich: mit Hinweisung auf die ähnlichen Erscheinungen im Thierreich.* Gedruckt bei KF Hering, 1849.

[54] R. A. Gatenby, E. T. Gawlinski, A. F. Gmitro, B. Kaylor, and R. J. Gillies. "Acid-mediated tumor invasion: a multidisciplinary study". In: *Cancer research* 66.10 (2006), pp. 5216–5223.

[55] S. Gavrilets. "Evolution and speciation in a hyperspace: the roles of neutrality, selection, mutation, and random drift". In: *Evolutionary Dynamics: Exploring the Interplay of Selection, Accident, Neutrality, and Function.* Ed. by J. P. Crutchfield and P. Schuster. Oxford University Press, 2003, pp. 135–162.

[56] S. Gavrilets and J. Gravner. "Percolation on the fitness hypercube and the evolution of reproductive isolation". In: *Journal of theoretical biology* 184.1 (1997), pp. 51–64.

[57] P. Gerlee. "Directional variation in evolution: consequences for the fitness landscape metaphor". In: *bioRxiv* (2015), p. 015529.

[58] P. Gerlee and P. M. Altrock. "Extinction rates in tumour public goods games". In: *Journal of The Royal Society Interface* 14.134 (2017), p. 20170342.

[59] J. Gillespie. "A simple stochastic gene substitution model." In: *Theor. Pop. Biol.* 23 (1983), p. 202.

[60] J. Gillespie. "Molecular evolution over the mutational landscape." In: *Evolution* 38 (1984), p. 1116.

[61] J. Gillespie. *The causes of molecular evolution.* Oxford University Press, 1991.

[62] J. H. Gillespie. *Population genetics: a concise guide.* John Hopkins University Press, 2010.

[63] R. J. Gillies, D. Verduzco, and R. A. Gatenby. "Evolutionary dynamics of carcinogenesis and why targeted therapy does not work". In: *Nature Reviews Cancer* 12.7 (2012), pp. 487–493.

[64] P. W. Goldberg. "A survey of PPAD-completeness for computing Nash equilibria". In: *arXiv preprint arXiv:1103.2709* (2011).

[65] D Goldfarb. "On the complexity of the simplex algorithm". In: *Advances in optimization and numerical analysis.* 1994, pp. 25–38.

[66] J. Gore, H. Youk, and A. Van Oudenaarden. "Snowdrift game dynamics and facultative cheating in yeast". In: *Nature* 459.7244 (2009), pp. 253–256.

[67] C. Goulart, M. Mentar, K. Crona, S. Jacobs, M. Kallmann, B. Hall, D. Greene, and M. Barlow. “Designing antibiotic cycling strategies by determining and understanding local adaptive landscapes.” In: *PLoS One* 8 (2 2013), e56040.

[68] S. J. Gould. “Allometry and size in ontogeny and phylogeny”. In: *Biological Reviews* 41.4 (1966), pp. 587–638.

[69] P. R. Grant and B. R. Grant. *How and why species multiply: the radiation of Darwin’s finches*. Princeton University Press, 2011.

[70] S. P. Hammarlund, B. D. Connelly, K. J. Dickinson, and B. Kerr. “The evolution of cooperation by the Hankshaw effect”. In: *Evolution* 70.6 (2016), pp. 1376–1385.

[71] G. Hardin. “The competitive exclusion principle”. In: *science* 131.3409 (1960), pp. 1292–1297.

[72] M. Hartshorn, A. Kaznatcheev, and T. Shultz. “The evolutionary dominance of ethnocentric cooperation”. In: *Journal of Artificial Societies and Social Simulation* 16.3 (2013), p. 7.

[73] C. Hauert and M. Doebeli. “Spatial structure often inhibits the evolution of cooperation in the snowdrift game”. In: *Nature* 428.6983 (2004), pp. 643–646.

[74] G. H. Heppner. “Tumor heterogeneity”. In: *Cancer Research* 44.6 (1984), pp. 2259–2265.

[75] J. P. Heredia, B. Trubenová, D. Sudholt, and T. Paixão. “Selection limits to adaptive walks on correlated landscapes”. In: *Genetics* 205.2 (2017), pp. 803–825.

[76] C. Hilbe. “Local replicator dynamics: a simple link between deterministic and stochastic models of evolutionary game theory”. In: *Bulletin of mathematical biology* 73.9 (2011), pp. 2068–2087.

[77] D Hilbert and W Ackerman. *Grundzuge der Theoretische Logik*. Springer, 1928.

[78] J. Hofbauer and K. Sigmund. *Evolutionary games and population dynamics*. Cambridge university press, 1998.

[79] J. Horn, D. E. Goldberg, and K. Deb. “Long path problems”. In: *International Conference on Parallel Problem Solving from Nature*. Springer. 1994, pp. 149–158.

[80] J. Hunt. *Agricultural Memoirs; Or History of the Dishley System in Answer To Sir John Saunders Sebright, Bart. M.P.* London: Longman, Hurst, Rees, Orme and Brown, 1812.

[81] S. Hwang, B. Schmiegelt, L. Ferretti, and J. Krug. “Universality classes of interaction structures for NK fitness landscapes”. In: *Journal of Statistical Physics* 172.1 (2018), pp. 226–278.

[82] A. Ibrahim-Hashim, M. Robertson-Tessi, P. Enrizues-Navas, M. Damaghi, Y. Balagurunathan, J. W. Wojtkowiak, S. Russell, K. Yoonseok, M. C. Lloyd, M. M. Bui, et al. "Defining cancer subpopulations by adaptive strategies rather than molecular properties provides novel insights into intratumoral evolution". In: *Cancer Research* (2017), p. 2844.

[83] R. K. Jain. "Normalizing tumor microenvironment to treat cancer: bench to bedside to biomarkers". In: *Journal of Clinical Oncology* 31.17 (2013), p. 2205.

[84] P. Jeavons, A. Krokhin, and S. Živný. "The complexity of valued constraint satisfaction." In: *Bulletin of the EATCS.* 113 (2014), pp. 21–55.

[85] R. G. Jeroslow. "The simplex algorithm with the pivot rule of maximizing criterion improvement". In: *Discrete Mathematics* 4.4 (1973), pp. 367–377.

[86] D. Johnson, C. Papadimitriou, and M. Yannakakis. "How easy is local search?" In: *Journal of Computer and System Sciences* 37 (1 1988), pp. 79–100.

[87] V. Kanade. "Computational questions in evolution". PhD thesis. Harvard University, 2012.

[88] R. Katayama, C. M. Lovly, and A. T. Shaw. "Therapeutic targeting of anaplastic lymphoma kinase in lung cancer: a paradigm for precision cancer medicine". In: *Clinical Cancer Research* 21 (10 2015), pp. 2227–2235.

[89] S Kauffman. *The origins of order: Self organization and selection in evolution.* Oxford University Press, 1993.

[90] S. Kauffman and S. Levin. "Towards a general theory of adaptive walks on rugged landscapes." In: *Journal of Theoretical Biology* 128 (1 1987), pp. 11–45.

[91] S. Kauffman and E. Weinberger. "The NK model of rugged fitness landscapes and its application to maturation of the immune response." In: *Journal of Theoretical Biology* 141 (2 1989), pp. 211–245.

[92] A. Kaznatcheev. "Abstract is not the opposite of empirical: case of the game assay". In: *Theory, Evolution, and Games Group* June 2 (2018). At: `https://egtheory.wordpress.com/2018/06/02/abstract-vs-empirical/`.

[93] A. Kaznatcheev. "Approximating spatial structure with the Ohtsuki-Nowak transform". In: *Theory, Evolution, and Games Group* February 26 (2014). At: `https://egtheory.wordpress.com/2014/02/26/approximating-spatial-structure/`.

[94] A. Kaznatcheev. "British agricultural revolution gave us evolution by natural selection". In: *Theory, Evolution, and Games Group* May 25 (2019). At: `https://egtheory.wordpress.com/2019/05/25/agriculture-to-evolution/`.

[95] A. Kaznatcheev. "Complexity of evolutionary equilibria in static fitness landscapes". In: *arXiv preprint:1308.5094* (2013).

[96] A. Kaznatcheev. "Computational Complexity as an Ultimate Constraint on Evolution". In: *Genetics* 212.1 (2019), pp. 245–265. ISSN: 0016-6731. DOI: 10.1534/genetics.119.302000.

[97] A. Kaznatcheev. "Counting cancer cells with computer vision for time-lapse microscopy". In: *Theory, Evolution, and Games Group* April 21 (2016). At: https://egtheory.wordpress.com/2016/04/21/automicroscopy/.

[98] A. Kaznatcheev. "Darwin as an early algorithmic biologist". In: *Theory, Evolution, and Games Group* August 4 (2018). At: https://egtheory.wordpress.com/2018/08/04/darwin-algorithm/.

[99] A. Kaznatcheev. "Double-entry bookkeeping and Galileo: abstraction vs idealization". In: *Theory, Evolution, and Games Group* June 15 (2018). At: https://egtheory.wordpress.com/2018/06/15/bookkeeping-abstraction-vs-idealization/.

[100] A. Kaznatcheev. "Effective games and the confusion over spatial structure". In: *Proceedings of the National Academy of Sciences* (2018), p. 201719031.

[101] A. Kaznatcheev. "Evolution is exponentially more powerful with frequency-dependent selection". In: *bioRxiv* (2020).

[102] A. Kaznatcheev. "Evolutionary game theory without interactions". In: *Theory, Evolution, and Games Group* February 13 (2015). At: https://egtheory.wordpress.com/2015/02/13/evolutionrary-game-theory-without-interactions/.

[103] A. Kaznatcheev. "Falsifiability and Gandy's variant of the Church-Turing thesis". In: *Theory, Evolution, and Games Group* September 1 (2014). At: https://egtheory.wordpress.com/2014/09/01/falsifiability-and-gandys-variant-of-the-church-turing-thesis/.

[104] A. Kaznatcheev. "From perpetual motion machines to the Entscheidungsproblem". In: *Theory, Evolution, and Games Group* March 9 (2019). At: https://egtheory.wordpress.com/2019/03/09/halting-perpetual-motion/.

[105] A. Kaznatcheev. "Idealization vs abstraction for mathematical models of evolution". In: *Theory, Evolution, and Games Group* August 24 (2019). At: https://egtheory.wordpress.com/2019/08/24/eseb-idealization-vs-abstraction/.

[106] A. Kaznatcheev. “Lotka-Volterra, replicator dynamics, and stag hunting bacterian”. In: *Theory, Evolution, and Games Group* February 9 (2016). At: `https://egtheory.wordpress.com/2016/02/09/stag-hunting-bacteria/`.

[107] A. Kaznatcheev. “Robustness of ethnocentrism to changes in interpersonal interactions”. In: *2010 AAAI Fall Symposium Series*. 2010.

[108] A. Kaznatcheev. “Space of cooperate-defect games”. In: *Theory, Evolution, and Games Group* March 14 (2012). At: `https://egtheory.wordpress.com/2012/03/14/uv-space/`.

[109] A. Kaznatcheev. “The cognitive cost of ethnocentrism”. In: *Proceedings of the Cognitive Science Society*. Vol. 32. 32. 2010.

[110] A. Kaznatcheev. “Transcendental idealism and Post’s variant of the Church-Turing thesis”. In: *Theory, Evolution, and Games Group* September 11 (2014). At: `https://egtheory.wordpress.com/2014/09/11/transcendental-idealism-and-posts-variant-of-the-church-turing-thesis/`.

[111] A. Kaznatcheev. “Two conceptions of evolutionary games: reductive vs effective”. In: *bioRxiv* (2017), p. 231993.

[112] A. Kaznatcheev. “Web of C-lief: conjectures vs. model assumptions vs. scientific beliefs”. In: *Theory, Evolution, and Games Group* August 31 (2019). At: `https://egtheory.wordpress.com/2019/08/31/web-of-c-lief/`.

[113] A. Kaznatcheev, D. A. Cohen, and P. G. Jeavons. “Representing fitness landscapes by valued constraints to understand the complexity of local search”. In: *International Conference on Principles and Practice of Constraint Programming*. Springer. 2019, pp. 300–316.

[114] A. Kaznatcheev, M. Montrey, and T. R. Shultz. “Evolving useful delusions: Subjectively rational selfishness leads to objectively irrational cooperation”. In: *Proceedings of the 36th annual conference of the cognitive science society* (2014).

[115] A. Kaznatcheev, J. G. Scott, and D. Basanta. “Edge effects in game-theoretic dynamics of spatially structured tumours”. In: *Journal of The Royal Society Interface* 12.108 (2015), p. 20150154.

[116] A. Kaznatcheev and T. R. Shultz. “Ethnocentrism Maintains Cooperation, but Keeping One’s Children Close Fuels It”. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 33. 33. 2011.

[117] A. Kaznatcheev and T. R. Shultz. “Moral externalization may precede, not follow, subjective preferences”. In: *Behavioral and Brain Sciences* 41 (2018).

[118] A. Kaznatcheev, R. Vander Velde, J. G. Scott, and D. Basanta. "Cancer treatment scheduling and dynamic heterogeneity in social dilemmas of tumour acidity and vasculature". In: *British Journal of Cancer* (2017).

[119] A. Kaznatcheev, J. Peacock, D. Basanta, A. Marusyk, and J. G. Scott. "Fibroblasts and alectinib switch the evolutionary games played by non-small cell lung cancer". In: *Nature Ecology & Evolution* 3.3 (2019), p. 450.

[120] C. P. Kempes, G. B. West, and M. Koehl. "The scales that limit: the physical boundaries of evolution". In: *Frontiers in Ecology and Evolution* 7 (2019), p. 242.

[121] B. Kerr, M. A. Riley, M. W. Feldman, and B. J. Bohannan. "Local dispersal promotes biodiversity in a real-life game of rock–paper–scissors". In: *Nature* 418.6894 (2002), pp. 171–174.

[122] T. Killingback and M. Doebeli. "Spatial evolutionary game theory: Hawks and Doves revisited". In: *Proceedings of the Royal Society of London B: Biological Sciences* 263.1374 (1996), pp. 1135–1144.

[123] V. Klee and G. J. Minty. *How good is the simplex algorithm*. Tech. rep. DTIC Document, 1970.

[124] K. Kouvaris, J. Clune, L. Kounios, M. Brede, and R. A. Watson. "How evolution learns to generalise: Using the principles of learning theory to understand the evolution of developmental organisation". In: *PLoS computational biology* 13.4 (2017).

[125] S. Kryazhimskiy, G. Tkacik, and J. Plotkin. "The dynamics of adaptation on correlated fitness landscapes." In: *Proc. Natl. Acad. Sci. USA* 106 (44 2009), pp. 18638–18643.

[126] R. E. Lenski, M. R. Rose, S. C. Simpson, and S. C. Tadler. "Long-term experimental evolution in Escherichia coli. I. Adaptation and divergence during 2,000 generations". In: *The American Naturalist* 138.6 (1991), pp. 1315–1341.

[127] R. E. Lenski, M. J. Wiser, N. Ribeck, Z. D. Blount, J. R. Nahum, J. J. Morris, L. Zaman, C. B. Turner, B. D. Wade, R. Maddamsetti, et al. "Sustained fitness gains and variability in fitness trajectories in the long-term evolution experiment with Escherichia coli". In: *Proc. R. Soc. B* 282.1821 (2015), p. 20152292.

[128] D. A. Levinthal. "Adaptation on rugged landscapes". In: *Management science* 43.7 (1997), pp. 934–950.

[129] C. Li, W. Qian, C. J. Maclean, and J. Zhang. "The fitness landscape of a tRNA gene". In: *Science* 352.6287 (2016), pp. 837–840.

[130] M. Li and P. M. Vitányi. “Average case complexity under the universal distribution equals worst-case complexity”. In: *Information Processing Letters* 42.3 (1992), pp. 145–149.

[131] X.-Y. Li, C. Pietschke, S. Fraune, P. M. Altrock, T. C. Bosch, and A. Traulsen. “Which games are growing bacterial populations playing?” In: *Journal of The Royal Society Interface* 12.108 (2015), p. 20150121.

[132] E. Lieberman, C. Hauert, and M. A. Nowak. “Evolutionary dynamics on graphs”. In: *Nature* 433.7023 (2005), pp. 312–316.

[133] A. Livnat, C. Papadimitriou, J. Dushoff, and M. W. Feldman. “A mixability theory for the role of sex in evolution.” In: *Proc. Natl. Acad. Sci. USA* 105 (50 2008), pp. 19803–19808.

[134] A. Livnat and C. Papadimitriou. “Sex as an algorithm: the theory of evolution under the lens of computation”. In: *Communications of the ACM* 59.11 (2016), pp. 84–93.

[135] E. R. Lozovsky, T. Chookajorn, K. M. Brown, M. Imwong, P. J. Shaw, S. Kamchonwong-paisan, D. Neafsey, D. Weinreich, and D. L. Hartl. “Stepwise acquisition of pyrimethamine resistance in the malaria parasite.” In: *Proc. Natl. Acad. Sci. USA* 106 (29 2009), pp. 12025–12030.

[136] M. Lynch, R. Bürger, D Butcher, and W. Gabriel. “The mutational meltdown in asexual populations”. In: *Journal of Heredity* 84.5 (1993), pp. 339–344.

[137] W. Maciejewski and G. J. Puleo. “Environmental evolutionary graph theory”. In: *Journal of theoretical biology* 360 (2014), pp. 117–128.

[138] R. C. MacLean and I. Gudelj. “Resource competition and social conflict in experimental populations of yeast”. In: *Nature* 441.7092 (2006), p. 498.

[139] R. Maddamsetti, R. E. Lenski, and J. E. Barrick. “Adaptation, clonal interference, and frequency-dependent interactions in a long-term evolution experiment with Escherichia coli”. In: *Genetics* 200.2 (2015), pp. 619–631.

[140] T. R. Malthus. *An Essay on the Principle of Population.* J. Johnson, London, 1798.

[141] F. Markowetz. “All biology is computational biology”. In: *PLoS Biology* 15.3 (2017), e2002050.

[142] A. Marusyk, D. P. Tabassum, P. M. Altrock, V. Almendro, F. Michor, and K. Polyak. “Non-cell autonomous tumor-growth driving supports sub-clonal heterogeneity”. In: *Nature* 514.7520 (2014), p. 54.

[143] A. Marusyk, D. P. Tabassum, M. Janiszewska, A. E. Place, A. Trinh, A. I. Rozhok, S. Pyne, J. L. Guerriero, S. Shu, M. Ekram, et al. “Spatial proximity to fibroblasts impacts molecular features and therapeutic sensitivity of breast cancer cells influencing clinical outcomes”. In: *Cancer Research* 76.22 (2016), pp. 6495–6506.

[144] J. Matousek and T. Szabo. “RANDOM EDGE can be exponential on abstract cubes.” In: *Advances in Mathematics* 204 (1 2006), pp. 262–277.

[145] H Matsuda, N Tamachi, A Sasaki, and N Ogita. “A lattice model for population biology”. In: *Mathematical topics in population biology, morphogenesis and neurosciences*. Springer, 1987, pp. 154–161.

[146] J. Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.

[147] J. Maynard Smith. *George Price’s theorem and how scientists think*. Web of Stories: `https://www.webofstories.com/play/john.maynard.smith/45`. 1997.

[148] E. Mayr. “Cause and effect in biology”. In: *Science* 134.3489 (1961), pp. 1501–1506.

[149] J. McNamara. “Towards a richer evolutionary game theory”. In: *J, R. Soc., Interface* 10 (88 2013).

[150] M. Mediavilla-Varela, K. Boateng, D. Noyes, and S. J. Antonia. “The anti-fibrotic agent pirfenidone synergizes with cisplatin in killing tumor cells and cancer-associated fibroblasts”. In: *BMC Cancer* 16.1 (2016), p. 176.

[151] L. M. Merlo, J. W. Pepper, B. J. Reid, and C. C. Maley. “Cancer as an evolutionary and ecological process”. In: *Nature Reviews Cancer* 6.12 (2006), pp. 924–935.

[152] R. L. Millstein. “Populations as individuals”. In: *Biological Theory* 4.3 (2009), pp. 267–273.

[153] B. Monien and T. Tscheuschner. “On the Power of Nodes of Degree Four in the Local Max-Cut Problem”. In: *Algorithms and Complexity*. Ed. by T. Calamoneri and J. Diaz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 264–275. ISBN: 978-3-642-13073-1.

[154] P. A. P. Moran. “Random processes in genetics”. In: *Proceedings of the Cambridge Philosophical Society*. Vol. 54. 1958, p. 60.

[155] E. P. Murchison, D. C. Wedge, L. B. Alexandrov, B. Fu, I. Martincorena, Z. Ning, J. M. Tubio, E. I. Werner, J. Allen, A. B. De Nardi, et al. “Transmissible dog cancer genome reveals the origin and history of an ancient cell lineage”. In: *Science* 343.6169 (2014), pp. 437–440.

[156] M. Nanda and R. Durrett. “Spatial evolutionary games with weak selection”. In: *Proceedings of the National Academy of Sciences* (2017), p. 201620852.

[157] D. Nichol, P. Jeavons, A. G. Fletcher, R. A. Bonomo, P. K. Maini, J. L. Paul, R. A. Gatenby, A. R. Anderson, and J. G. Scott. "Steering evolution with sequential therapy to prevent the emergence of bacterial antibiotic resistance". In: *PLoS computational biology* 11.9 (2015), e1004493.

[158] A. E. Noether. "Invariante Variationsprobleme". In: *Nachr. D. König. Gesellsch. D. Wiss.* 918.3 (1918), pp. 235–257.

[159] M. A. Nowak and R. M. May. "Evolutionary games and spatial chaos". In: *Nature* 359.6398 (1992), pp. 826–829.

[160] U. Obolski, Y. Ram, and L. Hadany. "Key issues review: evolution on rugged adaptive landscapes". In: *Reports on Progress in Physics* 81.1 (2017), p. 012602.

[161] T. Ohta. "The nearly neutral theory of molecular evolution". In: *Annual Review of Ecology and Systematics* 23 (1992), pp. 263–286.

[162] H. Ohtsuki and M. A. Nowak. "The replicator equation on graphs". In: *Journal of theoretical biology* 243.1 (2006), pp. 86–97.

[163] H. Ohtsuki, C. Hauert, E. Lieberman, and M. A. Nowak. "A simple rule for the evolution of cooperation on graphs and social networks". In: *Nature* 441.7092 (2006), pp. 502–505.

[164] J. Orlin, A. Punnen, and A. Schulz. "Approximate local search in combinatorial optimization". In: *SIAM J. Comput.* 33 (5 2004), pp. 1201–1214.

[165] H. A. Orr. "The distribution of fitness effects among beneficial mutations". In: *Genetics* 163.4 (2003), pp. 1519–1526.

[166] H. A. Orr. "The genetic theory of adaptation: a brief history." In: *Nature Reviews. Genetics* 6 (2005), pp. 119–127.

[167] H. Orr. "The population genetics of adaptation: the adaptation of DNA sequences." In: *Evolution* (56 2002), pp. 1317–1330.

[168] S. H. Orzack and E. Sober. *Adaptationism and optimality.* Cambridge University Press, 2001.

[169] J. Otwinowski, D. M. McCandlish, and J. B. Plotkin. "Inferring the shape of global epistasis". In: *Proceedings of the National Academy of Sciences* 115.32 (2018), E7550–E7558.

[170] S.-H. I. Ou, J. S. Ahn, L. De Petris, R. Govindan, J. C.-H. Yang, B. Hughes, H. Lena, D. Moro-Sibilot, A. Bearz, S. V. Ramirez, et al. "Alectinib in crizotinib-refractory ALK-rearranged non–small-cell lung cancer: a phase II global study". In: *Journal of Clinical Oncology* 34.7 (2015), pp. 661–668.

[171] A. Pavlogiannis, J. Tkadlec, K. Chatterjee, and M. A. Nowak. “Construction of arbitrarily strong amplifiers of natural selection using evolutionary graph theory”. In: *Communications Biology* 1.1 (2018), p. 71.

[172] C. S. Peirce. “Prolegomena to an apology for pragmaticism”. In: *The Monist* 16.4 (1906), pp. 492–546.

[173] F. Pelletier, D. Garant, and A. Hendry. *Eco-evolutionary dynamics*. 2009.

[174] J. Peña, L. Lehmann, and G. Nöldeke. “Gains from switching and evolutionary stability in multi-player matrix games”. In: *Journal of Theoretical Biology* 346 (2014), pp. 23–33.

[175] C. H. Pence and G. Ramsey. “Is Organismic Fitness at the Basis of Evolutionary Theory?” In: *Philosophy of Science* 82.5 (2015), pp. 1081–1091.

[176] A. S. Penn, T. C. Conibear, R. A. Watson, A. R. Kraaijeveld, and J. S. Webb. “Can Simpson’s paradox explain co-operation in Pseudomonas aeruginosa biofilms?” In: *FEMS Immunology & Medical Microbiology* 65.2 (2012), pp. 226–235.

[177] S. Peters, D. R. Camidge, A. T. Shaw, S. Gadgeel, J. S. Ahn, D.-W. Kim, S.-H. I. Ou, M. Pérol, R. Dziadziuszko, R. Rosell, et al. “Alectinib versus crizotinib in untreated ALK-positive non–small-cell lung cancer”. In: *New England Journal of Medicine* 377.9 (2017), pp. 829–838.

[178] F. Poelwijk, D. Kiviet, D. Weinreich, and S. Tans. “Empirical fitness landscapes reveal accessible evolutionary paths.” In: *Nature* 445 (2007), pp. 383–386.

[179] F. Poelwijk, T.-N. Sorin, D. Kiviet, and S. Tans. “Reciprocal sign epistasis is a necessary condition for multi-peaked fitness landscapes.” In: *Journal of Theoretical Biology* 272 (2011), pp. 141 –144.

[180] S. Poljak. “Integer linear programs and local search for max-cut”. In: *SIAM Journal on Computing* 24.4 (1995), pp. 822–839.

[181] E. L. Post. “Finite combinatory processes—formulation 1”. In: *The Journal of Symbolic Logic* 1.3 (1936), pp. 103–105.

[182] R. E. Prothero. *English Farming, Past and Present.* London: Longmans, Green and co., 1912.

[183] O. Puchta, B. Cseke, H. Czaja, D. Tollervey, G. Sanguinetti, and G. Kudla. “Network of epistatic interactions within a yeast snoRNA”. In: *Science* 352.6287 (2016), pp. 840–844.

[184] A. Rapoport. “Exploiter, Leader, Hero, and Martyr: the four archetypes of the 2× 2 game”. In: *Systems Research and Behavioral Science* 12.2 (1967), pp. 81–84.

[185] N. Ribeck and R. E. Lenski. “Modeling and quantifying frequency-dependent fitness in microbial populations with cross-feeding interactions”. In: *Evolution* 69.5 (2015), pp. 1313–1320.

[186] J. W. Rivkin and N. Siggelkow. “Patterned interactions in complex systems: Implications for exploration”. In: *Management Science* 53.7 (2007), pp. 1068–1085.

[187] D. Robinson and D. Goforth. *The topology of the 2x2 games: a new periodic table.* Vol. 3. Psychology Press, 2005.

[188] R. C. Rockne, A. Hawkins-Daarud, K. R. Swanson, J. P. Sluka, J. A. Glazier, P. Macklin, D. A. Hormuth, A. M. Jarrett, E. A.B. F. Lima, J. T. Oden, G. Biros, T. E. Yankeelov, K. Curtius, I. A. Bakir, D. Wodarz, N. Komarova, L. Aparicio, M. Bordyuh, R. Rabadan, S. D. Finley, H. Enderling, J. Caudell, E. G. Moros, A. R. A. Anderson, R. A. Gatenby, A. Kaznatcheev, P. Jeavons, et al. “The 2019 Mathematical Oncology Roadmap”. In: *Physical Biology* 16.4 (2019), p. 041005.

[189] T Roughgarden. “Computing equilibria: A computational complexity perspective.” In: *Economic Theory* 42 (1 2010), pp. 193–236.

[190] F. C. Santos, J. M. Pacheco, and T. Lenaerts. “Evolutionary dynamics of social dilemmas in structured heterogeneous populations”. In: *Proceedings of the National Academy of Sciences* 103.9 (2006), pp. 3490–3494.

[191] A. Schaffer and M. Yannakakis. “Simple local search problems that are hard to solve.” In: *SIAM Journal on Computing* 20.1 (1991), pp. 56–87.

[192] K. H. Schlag. “Why imitate, and if so, how?: A boundedly rational approach to multi-armed bandits”. In: *Journal of economic theory* 78.1 (1998), pp. 130–156.

[193] I. Schurr and T. Szabó. “Jumping doesn’t help in abstract cubes”. In: *International Conference on Integer Programming and Combinatorial Optimization.* Springer. 2005, pp. 225–235.

[194] P. K. Sen. “Estimates of the regression coefficient based on Kendall’s tau”. In: *Journal of the American statistical association* 63.324 (1968), pp. 1379–1389.

[195] T. Seto, K. Kiura, M. Nishio, K. Nakagawa, M. Maemondo, A. Inoue, T. Hida, N. Yamamoto, H. Yoshioka, M. Harada, et al. “CH5424802 (RO5424802) for patients with ALK-rearranged advanced non-small-cell lung cancer (AF-001JP study): a single-arm, open-label, phase 1–2 study”. In: *The Lancet Oncology* 14.7 (2013), pp. 590–598.

[196] P. Shakarian, P. Roos, and A. Johnson. “A review of evolutionary graph theory with applications to game theory”. In: *Biosystems* 107.2 (2012), pp. 66–80.

[197] A. T. Shaw and J. A. Engelman. “ALK in lung cancer: past, present, and future”. In: *Journal of Clinical Oncology* 31.8 (2013), pp. 1105–1111.

[198] A. T. Shaw, D.-W. Kim, K. Nakagawa, T. Seto, L. Crinó, M.-J. Ahn, T. De Pas, B. Besse, B. J. Solomon, F. Blackhall, et al. “Crizotinib versus chemotherapy in advanced ALK-positive lung cancer”. In: *New England Journal of Medicine* 368.25 (2013), pp. 2385–2394.

[199] N. M. Shnerb, Y. Louzoun, E. Bettelheim, and S. Solomon. “The importance of being discrete: Life always wins on the surface”. In: *Proceedings of the National Academy of Sciences* 97.19 (2000), pp. 10322–10324.

[200] T. R. Shultz, M. Hartshorn, and A. Kaznatcheev. “Why is ethnocentrism more common than humanitarianism”. In: *Proceedings of the 31st annual conference of the cognitive science society.* Austin, TX: Cognitive Science Society. 2009, pp. 2100–2105.

[201] J. Silvertown, P. Poulton, E. Johnston, G. Edwards, M. Heard, and P. M. Biss. “The Park Grass Experiment 1856–2006: its contribution to ecology”. In: *Journal of Ecology* 94.4 (2006), pp. 801–814.

[202] G. G. Simpson. “The Baldwin effect”. In: *Evolution* 7.2 (1953), pp. 110–117.

[203] E. Sober. “Trait fitness is not a propensity, but fitness variation is”. In: *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences* 44.3 (2013), pp. 336–341.

[204] P. Stanley et al. *Robert Bakewell and the Longhorn breed of cattle.* Farming Press Ltd., 1995.

[205] S. C. Stearns. “Life-history tactics: a review of the ideas”. In: *The Quarterly review of biology* 51.1 (1976), pp. 3–47.

[206] A. Strimbu. “Simulating Evolution on Fitness Landscapes represented by Valued Constraint Satisfaction Problems”. In: *arXiv:1912.02134* (2019).

[207] G. Szabó and G. Fath. “Evolutionary games on graphs”. In: *Physics reports* 446.4 (2007), pp. 97–216.

[208] T. Szabó and E. Welzl. “Unique sink orientations of cubes”. In: *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on.* IEEE. 2001, pp. 547–555.

[209] I. Szendro, M. Schenk, J. Franke, J. Krug, and J. de Visser. “Quantitative analyses of empirical fitness landscapes.” In: *J. Stat. Mech.* (2013), p01005.

[210] C. Taylor, D. Fudenberg, A. Sasaki, and M. A. Nowak. "Evolutionary game dynamics in finite populations". In: *Bulletin of Mathematical Biology* 66.6 (2004), pp. 1621–1644.

[211] P. D. Taylor and L. B. Jonker. "Evolutionary stable strategies and game dynamics". In: *Mathematical Biosciences* 40.1 (1978), pp. 145–156.

[212] H Thiel. "A rank-invariant method of linear and polynomial regression analysis, I, II, III". In: *Proceedings of Koninalijke Nederlandse Akademie van Weinenschatpen A*. Vol. 53. 1950, 386—392, 521—525, 1397—1412.

[213] I. P. Tomlinson. "Game-theory models of interactions between tumour cells". In: *European Journal of Cancer* 33.9 (1997), pp. 1495–1500.

[214] I. P. Tomlinson and W. F. Bodmer. "Modelling the consequences of interactions between tumour cells". In: *British Journal of Cancer* 75.2 (1997), pp. 157–160.

[215] C. A. Tovey. "Low order polynomial bounds on the expected performance of local improvement algorithms". In: *Mathematical Programming* 35.2 (1986), pp. 193–224.

[216] A. Traulsen, J. C. Claussen, and C. Hauert. "Coevolutionary dynamics in large, but finite populations". In: *Physical Review E* 74.1 (2006), p. 011901.

[217] A. M. Turing. "On computable numbers, with an application to the Entscheidungsproblem". In: *Proceedings of the London Mathematical Society* 2.1 (1937), pp. 230–265.

[218] A. M. Turing. "The Chemical Basis of Morphogenesis". In: *Philosophical Transactions of the Royal Society of London* 237.641 (1952), pp. 37–72.

[219] P. E. Turner and L. Chao. "Escape from prisoner's dilemma in RNA phage $\Phi 6$". In: *The American Naturalist* 161.3 (2003), pp. 497–505.

[220] P. E. Turner and L. Chao. "Prisoner's dilemma in an RNA virus". In: *Nature* 398.6726 (1999), p. 441.

[221] L. Valiant. "Evolvability". In: *Journal of the ACM* 56 (1 2009), p. 3.

[222] M. Van Baalen. "Pair approximations for different spatial geometries". In: *The geometry of ecological interactions: simplifying spatial complexity* 742 (2000), pp. 359–387.

[223] J. A. G. de Visser and R. E. Lenski. "Long-term experimental evolution in Escherichia coli. XI. Rejection of non-transitive interactions as cause of declining rate of adaptation". In: *BMC Evolutionary Biology* 2.1 (2002), p. 19.

[224] S. Živný. *The Complexity of Valued Constraint Satisfaction Problems.* Cognitive Technologies. Springer, 2012, pp. 3–152. ISBN: 978-3-642-33973-8.

[225] D. L. Waltz. “AI’s 10 to Watch”. In: *IEEE Intelligent Systems* 21.3 (2006), pp. 5–14.

[226] P. I. Warman, A. Kaznatcheev, A. Araujo, C. C. Lynch, and D. Basanta. “Fractionated follow-up chemotherapy delays the onset of resistance in bone metastatic prostate cancer”. In: *Games* 9.2 (2018), p. 19.

[227] E. Weinberger. “NP completeness of Kauffman’s N-k model, a tunably rugged fitness landscape.” In: *Santa Fe Institute working paper* (1996), pp. 1996–02–003.

[228] D. Weinreich, R. Watson, and L. Chan. “Sign epistasis and genetic constraint on evolutionary trajectories.” In: *Evolution* 59 (2005), pp. 1165–1174.

[229] H. S. Wilf and W. J. Ewens. “There’s plenty of time for evolution”. In: *Proceedings of the National Academy of Sciences* 107.52 (2010), pp. 22454–22456.

[230] R. G. Winther, R. Giordano, M. D. Edge, and R. Nielsen. “The mind, the lab, and the field: Three kinds of populations in scientific practice”. In: *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences* 52 (2015), pp. 12–21.

[231] M. J. Wiser, N. Ribeck, and R. E. Lenski. “Long-term dynamics of adaptation in asexual populations”. In: *Science* 342.6164 (2013), pp. 1364–1367.

[232] M. J. Wiser, N. Ribeck, and R. E. Lenski. “Long-term dynamics of adaptation in asexual populations”. In: *Science* 342.6164 (2013), pp. 1364–1367.

[233] A. Wright, R. Thompson, and J. Zhang. “The computational complexity of N-K fitness functions.” In: *Evolutionary Computation, IEEE Transactions on* 4 (4 2000), pp. 373–379.

[234] S Wright. “The roles of mutation, inbreeding, crossbreeding, and selection in evolution.” In: *Proceedings of the Sixth International Congress of Genetics* (1932), pp. 356–366.

[235] J. Z. Xue, A. Costopoulos, and F. Guichard. “A Trait-based framework for mutation bias as a driver of long-term evolutionary trends”. In: *Complexity* 21.5 (2016), pp. 331–345.

[236] J. Z. Xue, A. Kaznatcheev, A. Costopoulos, and F. Guichard. “Fidelity drive: A mechanism for chaperone proteins to maintain stable mutation rates in prokaryotes over evolutionary time”. In: *Journal of theoretical biology* 364 (2015), pp. 162–167.

[237] T. Yamada, S. Takeuchi, J. Nakade, K. Kita, T. Nakagawa, S. Nanjo, T. Nakamura, K. Matsumoto, M. Soda, H. Mano, et al. “Paracrine receptor activation by microenvironment triggers bypass survival signals and ALK inhibitor resistance in EML4-ALK lung cancer cells”. In: *Clinical Cancer Research* 18.13 (2012), pp. 3592–3602.

[238] J. Zhang, J. J. Cunningham, J. S. Brown, and R. A. Gatenby. “Integrating evolutionary dynamics into treatment of metastatic castrate-resistant prostate cancer”. In: *Nature communications* 8.1 (2017), p. 1816.