The Evolution of Diversity in the Structure and Function of Artificial Organisms

Jose David Fernández Rodríguez
Dept. Lenguajes y Ciencias de la Computación
Universidad de Málaga, Spain

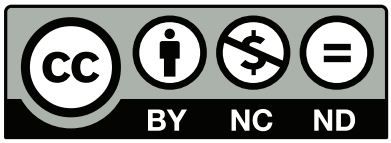
A dissertation submitted for the degree of
Doctor of Philosophy

January 2012

UNIVERSIDAD
DE MÁLAGA

AUTOR: José David Fernández Rodríguez

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



UNIVERSIDAD
DE MÁLAGA

Dr. Francisco J. Vico, Catedrático de Universidad, Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, España.

Dr. René Doursat, Científico Investigador, Grupo de Estudios en Biomimética, Universidad de Málaga, España.

Certificamos que:

Jose David Fernández Rodríguez, Ingeniero Superior en Informática, ha completado su tesis doctoral:

The Evolution of Diversity
in the Structure and Function
of Artificial Organisms

en el Departamento de Lenguajes y Ciencias de la Computación en la Universidad de Málaga, bajo nuestra supervisión. Hemos dirigido esta tesis doctoral, y la aprobamos para ser presentada a la Comisión de Doctorado de la Universidad de Málaga.

Málaga, Enero 2012.

Dr. Francisco J. Vico.
Catedrático de Universidad
Universidad de Málaga
España

Dr. René Doursat.
Científico Investigador
Grupo de Estudios en Biomimética
Universidad de Málaga
España

UNIVERSIDAD
DE MÁLAGA

Dedicated to Isabel,

the newest and brightest thread

in the fabric of my life.

UNIVERSIDAD
DE MÁLAGA

Abstract

In biological systems, a marvelous diversity of form and function emerges from evolutionary processes at all levels, from molecules to organisms and ecosystems. This *evolution of diversity* is one of the major questions in Biology, and it is investigated in many different biological disciplines, as evolutionary developmental biology, molecular biology and evolutionary theory. As the scope of this question is exceedingly vast, no single conceptual framework can be expected to encompass all the aspects of this evolution of diversity.

In recent years, as computational methods have become prevalent in Biology, the evolution of diversity has been studied using computational models and simulations. In this context, this doctoral dissertation employs a computational approach to explore three different aspects in the evolution of diversity: developmental processes, the interaction between the body and its control system, and evolutionary dynamics. Each one of these aspects is studied in a specific and abstract domain: tensegrity structures, models of molecular motor proteins, and virtual plant communities based on L-systems. In each domain, the results show different evolutionary mechanisms giving rise to diversity of form and function: the complexification of developmental processes, the coevolution of form and function, and ecological interactions. From the perspective of evolutionary computation, new evolutionary methods are presented, with several applications: generation of tensegrity structures and automatic agent design.

UNIVERSIDAD
DE MÁLAGA

Acknowledgments

This being a doctoral dissertation, it is only natural to begin by thanking my advisors, Francisco Vico and René Doursat. I am also grateful to David Orden for hosting me during a research internship and helping me to develop different areas of my research. As David Orden put it, research visits provide the tools we need to develop our research interests, and he gave me interesting and useful tools in the mathematical domain. I also wish to thank the FPU grant program from the Spanish government, which has funded my research for four years.

As a graduate student of Francisco Vico, I have been immersed in the dynamic environment provided by the Grupo de Estudios en Biomimética[1]; which is affiliated to the larger research group *Inteligencia Computacional y análisis de imágenes (grupo de excelencia TIC–163)*, led by José Muñoz Pérez, who has provided kind and useful support. This great environment has been made possible by my lab mates. I have had the opportunity to share my enthusiasm for science with my fellow Ph.D. students, Gema Martín, Daniel Lobo, and Carlos Sánchez. Also thanks to the people how have helped me in this path, such as Julio Pérez, for his technical support and vitriolic but witty observations; Miguel Carmona, for his exciting personality and conversations; Ellen Pearce, for her support in language skills and the opportunity to teach popular science; Francisco Osuna, for his administrative support and warm personality; and Vicente Canteli, for his artistic support, illuminating conversations, and rare wisdom (as Daniel Lobo put it, very appropriately). I also would like to thank my other colleages along these years, both old an new: David Albarracín, Jose Serrano, Cristóbal Carnero, Fran Moreno, Juan Travesedo, Alejandro Villegas, Salva Burrezo, Antonio Gómez, Samuel Morales and María Jose Casas.

However, all these people have been here only for the final act of this story. So, it

[1] `http://www.geb.uma.es`

is important to ask: how did it begin?

That question calls for a grandiose and poetic description of the Big Bang...

...but much, much later than that, I happened to my parents, Pepe and Blanca. I cannot possibly express in written words how much I owe to them in every aspect of life, and how much support I have received from them. However, since these are (precisely) just written words, and this being a scientific dissertation, let's settle on expressing that they have nurtured, over decades, the spark that has driven me to pursue a career in science. Just like my brother, Miguel Angel, whose inspiring questions, sharp mind and trust in me have always inspired me to try to do a bit better, get a bit farther, become a bit wiser. And my grandparents José, Carmen and Josefa, who in my early years instilled me with a good dose of common sense[2] and the rest of my family, for their support, especially my aunts Mari Carmen and Carmele. I want also to thank Elisa, for giving me the chance to walk our paths together, and to our daughter Isabel, whose smile is a reward in itself.

Thank you all.

[2] not so common at all, as it happens.

Credits

Some parts of this doctoral dissertation are based on articles written in collaboration with other people, although previously non-published content has been added in each case. Specifically, to give proper credit to each collaborator:

- Chapter 2 is partly based on [53], written in collaboration with my advisors, Francisco Vico (project leader) and René Doursat (lead editor of the paper version).
- Chapter 3 is partly based on [54], written in collaboration with my advisor, Francisco Vico.
- Chapter 4 is based on [55], written in collaboration with Gema Martín (developed early versions of the quantitative measure of diversity and the mutational robustness analysis), Daniel Lobo (provided computational and programming support in the early stages of the model, and also gave critical assessment), my advisor Francisco Vico (project leader), and René Doursat (lead editor of the journal paper version). A research internship with Dr. Doursat at the Institut des Sytèmes Complexes[3] was spent polishing the model and producing the final results toward a journal publication.
- Appendix A is based on [52], written in collaboration with David Orden. This work is the result of a research internship with Dr. Orden at the Universidad de Alcalá de Henares.

[3] `http://www.iscpif.fr/`

While doing the research exposed in this dissertation I have been supported by a FPU grant (AP2007-03704) from the Spanish Government. Additionally, I have been partially supported by two research projects: the BioEmergences project (code 28892) of the Sixth Framework Programme of the European Union, and the GENEX project (P09-TIC-5123) of the Junta de Andalucía.

The image in Figure 2.1 has been taken from Wikimedia Commons, where it was published[4] by the user *Bob Burkhardt* under a Creative Commons License, CC-BY-2.5[5].

The ribbon diagram in left side of Figure 3.1, and the data used to generate the structure shown in the right side, are taken from the main peptide chain in the entry 1KK8 of the Protein Data Bank[6].

The images in Figures A.1, A.2, A.3 and A.4 have been adapted from several presentations by David Orden[7].

This document has been typeset using LyX[8] as an interface to MiKTeX[9], an implementation of TeX with the LaTeX macro language and the BibTeX bibliographic composer. The CTAN repository[10] of LaTeX packages and the community of TeX and LaTeX users[11] have been of inestimable value.

[4] `http://commons.wikimedia.org/wiki/File:Tensegrity_3-Prism.png`
[5] Text of the license available in `http://creativecommons.org/licenses/by/2.5/deed.en`
[6] `http://www.pdb.org/`
[7] David Orden's professional home page: `http://www2.uah.es/ordend/`
[8] `http://www.lyx.org/`
[9] `http://miktex.org/`
[10] `http://ctan.org/`
[11] Mainly consulted websites:
`http://www.tug.org/`
`http://www.latex-community.org/`
`http://tex.stackexchange.com/`

Table of Contents

List of Figures

Chapter 1

Introduction

In the last decades, the divide between Computer Science and Biology has become increasingly blurred. On the one hand, a diverse array of computational methods (collectively known as *bio-inspired computing*) have been inspired from an equally diverse range of biological systems. On the other hand, most biological disciplines have become (or are in the process of becoming) absolutely dependent on diverse computational methods, as researchers have been driven to integrate them into their workflows. Likewise, computational models of biological systems are now prevalent in Biology. This doctoral dissertation constitutes an interdisciplinary work, sitting at the boundary between Computer Science and Biology.

Arguably, evolutionary theory is one of the disciplines where the divide between both sciences is more permeable. Considering the flow of ideas from Biology to Computer Science, evolutionary dynamics can be easily abstracted into algorithmic patterns, inspiring computer scientists to devise evolutionary-based metaheuristic techniques for optimization, collectively called *evolutionary computation*. In the opposite direction, not only computational methods have been incorporated into evolutionary research, but also purely synthetic computational models are being increasingly used to research questions in evolutionary dynamics.

An intriguing aspect of evolutionary theory is the evolution of diversity: in the last hundreds of millions of years, Earth has witnessed the evolution of an astonishing variety in the size, morphology, function, organization and behavior of living organisms. Currently, the question of why organisms present such a rich diversity is far from satisfactorily answered. This question is mirrored in evolutionary computation, where

the generation and maintenance of diversity in evolutionary algorithms is a major area of research.

Science often faces challenging questions and problems, which may seem intractable if they are tackled directly. Answers and solutions are often obtained by abstracting reality in order to analyze simplified, mathematical models of it. However, in relation to the previously mentioned question of the evolutionary origins of diversity in living organisms, it is reasonable to expect that no single answer, no single conceptual framework is the sole explanatory factor for this question, but a range of them contribute to diversity at different levels. To explore them, this doctoral dissertation presents three simple computational models, each one profoundly different from the others. In this way, each model is suited to study different causes of diversity. Specifically, the models explore the roles in the evolution of diversity of developmental processes (Chapter 2), the coevolution of body and control system (Chapter 3) and evolutionary dynamics (Chapter 4).

The rest of this introductory chapter is organized as follows. Sections 1.1 to 1.5 are devoted to present the main lines of the dissertation, to put in context each one of the three models:

- Section 1.1 describes the rising popularity of agent-based models in computational biology (applicable to all of the dissertation).
- Section 1.2 presents an overview of evolutionary theory from the perspective of evolutionary developmental biology (Chapter 2).
- Section 1.3 provides a brief description of molecular motors (Chapter 3).
- Section 1.4 presents a short introduction to evolutionary computation (Chapters 2 and 3).
- Section 1.5 describes the concept of morphological computation (Chapters 2 and 3).

Finally, Section 1.6 provides an brief outline of the dissertation.

1.1 Agent-based models for computational biology

Evolutionary theory is concerned with understanding in detail the evolutionary processes that have configured all the diverse living organisms along the history of Earth. Of singular interest is the study of evolutionary change at or above the species level, popularly known as *macroevolution*. Evolutionary scientists have traditionally studied it only indirectly, at first with the tools of taxonomy, systematics and paleontology, then with population genetics, and with increasingly sophisticated technologies that have enabled the blossom of new disciplines, like molecular phylogenetics. However, it is difficult to study evolutionary processes in detail, as scarce and incomplete evidence must be patched together from the structure of extant organisms at different scales (from anatomy to genetic data) and paleontological records.

Therefore, much of evolutionary research takes the form of assessment of mathematical models of evolution through computational and/or analytical methods. Traditionally, these models of evolutionary dynamics have been formulated as systems of differential equations [145], whether *well mixed* (ODEs) or spatially extended (PDEs), in which the established goal is to find analytical solutions. These approaches present a number of drawbacks, however, since their formulation and resolution demand expert mathematical knowledge, yet they also have to remain simple enough to be mathematically tractable. For all these reasons, mathematical descriptions at the macroscopic level are generally unrealistic and difficult to extend with new features.

Another approach is to perform direct experimentation: *experimental evolution*, the study of evolutionary dynamics in organisms growing in precisely controlled conditions. However, this approach is somewhat limited in scope, as practical considerations limit it to study simple and fast-reproducing organisms (see for example [199, 217]).

On the other hand, there is an alternative approach to experimental evolution, made possible by the increasing power of computers: *agent-based models*. These models describe communities in a bottom-up fashion through the properties of their individual members, the developmental, functional and behavioral rules they obey, and the interactions with one another and the environment [76]. They can be considered as computational models of experimental evolution.

A great number of agent-based evolutionary computational models have been recently proposed, whether at the abstract level of large interacting populations [26,

221] or at the fine-grained level of multicellular growth [51]. Agents can be modeled in many ways beyond simplified abstractions of actual living organisms; for example, as computer programs [163], structures derived from L-systems [17] or formal automata [127].

Potentially, agent-based models can be used to produce large-scale computational experiments that can condense long evolutionary periods into short computing time frames, while making vast collections of data available for the analysis and extraction of relevant properties [115, 121]. Moreover, increasing computer power and storage capacity provide the means to actually simulate explicative models of biological evolution. The three models presented in this is dissertation are agent-based, albeit each one is profoundly different from the others. Chapter 4 presents a synthetic agent-based model of evolution to study the evolutionary dynamics of diversification, while in Chapters 2 and 3 populations of agents evolve under the rules of evolutionary algorithms; diversity of form and/or function ensues from different aspects of the model rather than proper evolutionary dynamics.

1.2 Evolutionary developmental biology

The now traditional *modern evolutionary synthesis* is still the current mainstream school of thought [130], used by many evolutionary biologists as the conceptual frame for evolutionary theory, stressing population genetics as the main tool to understand evolution. Under this view, diversity of form and function is a byproduct of evolutionary dynamics, gradually and passively increasing as diverse sets of alleles arise among different subpopulations of individuals, and these are expressed in consequently diverse sets of phenotypes, adapting to new environments. Genetic information is at the center of the stage. This leads to the view that phenotypes are *decoded versions* of their corresponding genotypes [129]. However, this point of view, highly abstract, makes it difficult to research the causes of the evolution of diversity at a mechanistic level [112].

An alternative approach is originated from the following observation: the phenotype of an organism is generated by a developmental process, that is to say, the process which transforms a zygote into a full-fledged organism. The developmental process can be described, at its core, as a complex choreography of precisely-timed events, as cells in the developing organism divide, change their physicochemical properties, and arrange

into cell sheets and other configurations, folding into increasingly complex forms [60].

Therefore, the evolution of phenotypes is largely the evolution of the corresponding developmental processes. At first glance, this seems just a complementary observation to the view of evolution as variation, inheritance and selection of the genes in a population of organisms, since the complex choreography that is a developmental process is regulated by the genotype of the organism. In fact, in the last decades, the new discipline of *evolutionary developmental biology* (*evo-devo*) has emerged as the study of developmental processes from an evolutionary perspective. Developmental processes seem to be a key component in the evolution of diversity [14]. From a mainstream point of view, the goal of evo-devo can be roughly described as understanding how genotypic variation translates to phenotypic variation, as changes in the genome modulate the developmental process.

However, an alternative point of view is the following: the developmental process is the result of a very convoluted interplay between the genome and laws of geometric and chemico-physical nature [60]. Thus, the key is not just in the genome, but in that interplay as a whole. Under this interpretation, the genome is displaced from the center of the stage: to understand how diversity arises through evolutionary dynamics, genes are not sufficient; physics are at least equally important, since the way in which physical processes are influenced by the genome become the key to understand diversity, and developmental processes can inherently impose a bias to evolutionary dynamics [5].

It is even debatable whether the genome has to be the main source of regulatory control in development: non-genetic features of chemico-physical nature may be even more determinant in some contexts, especially in the early stages of multicellular evolution [144] and in some unicellular organisms [71]. Chapter 2 presents a model where diversity ensues from convoluted developmental processes with minimal genetic control; while in the other chapters there is not a strong component of development.

1.3 Molecular motors

Biological molecular motors are nanoscale devices capable of transforming chemical energy into mechanical work, associated to many vital functions at all scales [173]. While these molecular motors are primarily studied in molecular biology, they are of interest to many other disciplines, from medicine [69] and cladistics [202] to

nanotechnology [205].

From a computational point of view, the characteristics and dynamics of these motors are studied at multiple time scales, ranging from very detailed and complex molecular dynamics simulations spanning a very small amount of time [99, 104, 119], to extremely simple and coarse-grained theoretical models of their working cycles [77, 215]. They have been extensively studied in order to understand their biochemical and structural features [203], and their *processivity*, i.e., the way some of them are able to walk along filaments taking many steps in a row [68, 209, 225].

However, this research is performed only in the (relatively few) instances known from molecular biology. In Chapter 3, heuristics inspired by elastic network analysis [161] and behavior-finding methods [120] are applied to explore a subset of the configuration space of template molecular structures that are able to transform chemical energy into directed movement. Thus, this particular and very specific domain (the study of molecular motors) is used as a testbed for the study of the evolution of diversity under very specific and constrained conditions (walking over a filament with a fixed cycle of operation), with the interesting result that, in fact, from the coevolution of the structural configuration (form) and the induced gait pattern (function), molecular motion can be attained in many different ways.

1.4 Evolutionary computation

Nature has always been a source of inspiration for engineers and scientists. In this regard, computer engineers are no exception: many algorithmic techniques have been inspired in nature; to name the most relevant: artificial neural networks, artificial immune systems, ant colony optimization and evolutionary algorithms [67].

Evolutionary algorithms are metaheuristic techniques used to find admissible solutions to optimization problems. Most evolutionary algorithms follow a common pattern: a changing set of candidate solutions (the population) undergoes a repeated cycle of evaluation, selection and reproduction with variation. The first step in the algorithm is to generate the candidate solutions of the initial set, usually in a more or less random way. Each candidate is then evaluated using the *fitness* function, i.e., an heuristic rule to measure its quality as a solution to the problem[1]. After the evaluation

[1]This is the meaning of fitness in evolutionary computation, and is the meaning commonly used in this dissertation. This is inspired in (but should not be confused with) the concept of *fitness*

phase, selection is performed: a new set of candidate solutions is generated from the old one from copies of the candidate solutions; each candidate solution is copied a number of times probabilistically proportional to its fitness. This step decreases the diversity of the population, which is restored by applying to a fraction of the candidate solutions some operator designed to increase the variation (for example, mutation or recombination operators). These steps (the previously mentioned evaluation, selection and reproduction with variation) are applied iteratively; the effect is a slow, biased change in the population; the mean (and best) fitness gradually increases as superior solutions are promoted through selection and new variations are created.

While this algorithmic pattern is common to all evolutionary algorithms, a wide range of different algorithms use different sets of selection rules, variation operators and solution encoding. The most relevant varieties are genetic algorithms [84], evolutionary strategies [8], evolutionary programming [59] and genetic programming [107]. Among these variants, Holland's proposal (genetic algorithms) is one of the most popular.

In a genetic algorithm, the encoded form of a candidate solution is named the genotype, while the phenotype is the translation of that coded form into a solution. Holland's original formulation of genetic algorithms is strongly associated with a plain and direct encoding of genotypes as binary strings, but this is not necessarily the case in all instances. In fact, the models presented in Chapters 2 and 3 use what can be described as genetic algorithms (though with far more elaborated encodings) to evolve diverse structures. However, to avoid any potential misunderstanding, these will named with a more generic label, as evolutionary algorithms.

A recent trend in evolutionary computation, named *artificial embryogeny*, is the use of artificial developmental processes as *indirect encodings* for candidate solutions [183]; that is to say, instead of encoding solution in a more or less direct way, they are encoded as some specification to direct or modulate some kind of abstract developmental process. The point is to enable evolutionary algorithms to tackle bigger and more complex problems, as well-engineered indirect encodings reduce the complexity, size and dimensionality (with all the associated issues of optimization in high-dimensional spaces) of the genetic spaces where the evolutionary search is performed [47]. Chapter 2

in evolutionary biology: a measure of the ability of an individual to propagate its genes (often incorrectly assumed to be just the reproductive success). In general, the word *fitness* will be used in the evolutionary computation sense in Chapters 2 and 3, while the biological sense will be used in Chapter 4.

provides an extreme example of indirect encoding through developmental process, while in Chapter 4 agents are also indirectly encoded, but in a less convoluted way, using a developmental process based on rewriting rules from formal language theory.

1.5 Morphological computation and tensegrity

The term morphology refers to the form, shape or structure of an object. The concept of *morphological computation* has emerged in recent years in the field of Robotics, to name the idea that the behavior of a robot does not only depend on the nature of the control system and its wiring to the body of the robot, but also on the morphology of the body itself [154, 157]. This means that, if the morphology of the robot is taken into account in the design, the control system can be simpler than expected, by exploiting the dynamics of interaction between the body and the control system. Paul [154] provided a proof of concept for this idea by designing a robot with a XOR-type behavior, while its control system was based on simple one-layer perceptrons (which cannot solve the XOR function); the XOR-type behavior was the result of the interplay between the control system and the structure of the robot.

This concept has been applied to design bipedal robots with minimal control systems [128] and robots with open-loop controllers and minimal numbers of degrees of freedom which self-stabilize fast gait patterns and generate diverse sets of behaviors, all of it through the interaction between the body and the control system [157]. The key is that the way the robot is built (not only the morphology; but the characteristics of the constituent materials) can non-linearly modulate the activity of the control system.

While this concept has been primarily used in the context of Robotics, it represents a useful paradigm in other fields. An example of morphological computation in agent-based models is Lobo's model of path followers [120]: relatively simple agents evolved to follow curved paths, whose control systems are implicitly their own structures. As an example in the context of Biology, molecular motors [173] can be considered as nanoscale devices with a significant degree of morphological computation, because of their inherent nature: these motors are enzymes that walk over cytoskeletal filaments inside biological cells; and their locomotion is the self-organized outcome of two factors: their morphology and the biochemical details (which can be collectively deemed as the control system) of the cycle of interaction with the filament. In Chapter 3, an

evolutionary algorithm is used to evolve diverse structures inspired in molecular motors; their equally diverse gait patterns arise from the complex interaction between their structures and their cycle of operation, by morphological computation.

Tensegrity structures show potential to enable morphological computations: they are stable structures composed of compression-bearing rigid elements, balanced by a network of tensioned cables, such that the whole structure is in equilibrium [138]. Because some elements of a tensegrity structure bear compression while others bear tension (the so-called self-stress of the structure, which is analyzed from a mathematical perspective in Appendix A), tensegrity structures store potential energy; as a side effect, they also have highly non-linear dynamics. These facts have been used to design tensegrity-based walker robots [153]; whose gait patterns are modulated by the non-linear interaction between the control system and the characteristics of the structure. In developmental processes, the genome can be considered to play the role of control system. Chapter 2 presents an evolutionary algorithm using a model based on tensegrity structures to generate diverse developmental processes (in turn generating diverse final morphologies). The primary role as control system of the developmental process is not provided by a genome, as usual, but by the non-linear dynamics of the tensegrity structures themselves.

1.6 Outline

The rest of this dissertation is organized as follows. Each one of the next three chapters introduces a different topic, then describes a model in the context of that topic, and uses it to study different aspects of the topic; among these aspects is present the evolution of diversity, mostly shown by examples of diverse evolved agents.

Chapter 2 introduces an artificial model of development with minimal genetic control; using evolutionary algorithms to study how diversity arises from the complexification of developmental processes. Chapter 3 changes the domain to molecular biology, studying the characteristics of artificial models of molecular motors designed with an evolutionary algorithm, and how diversity can emerge even in very constrained settings. Chapter 4 introduces a fairly different model where agents evolve through ecological interactions (not properly an evolutionary algorithm, but an artificial model of evolution); interesting evolutionary dynamics emerge as a result. These chapters

constitute the main part of the dissertation. After them, a summary with the main conclusions is presented in Chapter 5. Appendix A presents a novel mathematical analysis of tensegrity structures, which are used in Chapter 2; the analysis is too long to be included as a section of that chapter. Appendix B provides a visual representation of three simulations analyzed in Chapter 4. Finally, Appendix C details some practical aspects of the implementation of the models and simulations, and Appendix D is a Spanish version of the summary and conclusions of the dissertation.

Chapter 2

Diversity by complex developmental process

The thesis of this chapter revolves around the proposal of tensegrity structures as excitable media able to undergo complex developmental processes with minimal genetic control. While development plays a critical role in the emergence of diversity, its mechanical and chemical actions are usually considered to be inextricably correlated with genetic control. Since in most extant species the complex growth from a zygote to the adult organism is orchestrated by a complex gene regulatory network, the prevalent view is that the evolution of diverse morphologies must result from the evolution of diverse complex gene networks.

However, in this chapter we propose an abstract model of self-regulated structure without genetic regulation, but only modulation of initial conditions. Morphologies are generated by a simple evolutionary algorithm searching for complex instances of unfolding dynamics based on tensegrity structures. The usual regulatory function of the genome is taken over by physical constraints in the structures, making morphological diversity a pure product of structural complexification, thus providing an example for the hypothesis that developmental processes leading to morphological diversity are the result of a complex interplay of physical forces, and genes play a modulating and stabilizing role in developmental processes, rather than guiding them [87]. From

Part of the results presented in this chapter have been accepted to be published in [53].

the perspective of evolutionary computation and artificial development, the model presented in this chapter represents a working example of how small the genetic component of the model can become, while still enabling an evolutionary search to solve a problem.

The evolutionary approach used in this chapter is similar to the one used in Chapter 3, though there is not a straightforward environmental goal to drive the generation of diversity, but it arises as a byproduct of the complexification of developmental processes.

This chapter is organized as follows. Section 2.1 provides an historical background and an introduction on the potential of complex developmental processes to drive the evolution of morphological diversity, along with a brief exposition of the perspective from evolutionary computation, and an introduction to the concept of tensegrity. An abstract and computational model of complex developmental process is described in Section 2.2. Then, experimental results using this model are presented and analyzed in Section 2.3, and finally in Section 2.4 the significance of the model is discussed. See also Appendix A for a mathematical study of tensegrity structures related to this work, but not directly concerned with the issue of diversity.

2.1 Introduction

In this chapter, we discuss about diversity of form, i.e. diversity of morphology. The aim is to design computational models inspired in biological development capable of generating diversity. The phenotype of a multicellular organism is generated by a complex developmental process, which transforms a single fertilized cell, the zygote, into a full-fledged living architecture composed of millions to trillions of cells. At its core, this process is a fascinating self-made choreography of precisely timed events, during which cells in the growing organism spontaneously divide, modify their physicochemical properties, and arrange into layers and tissues, themselves folding into increasingly complicated shapes to form organs and appendages [60]. The recent discipline of *evolutionary developmental biology*, or *evo-devo*, studies development from an evolutionary perspective (and vice-versa). It examines how phenotypic variation arises from the interplay between the physics of growth and both genetic and environmental variations. Evo-devo takes the viewpoint that morphological trends in

evolution are biased to a large extent by developmental mechanisms [5], re-interpreting the traditional but simpler concept of phenotypic plasticity [211]. It also applies to some unicellular organisms, albeit at the molecular scale [70].

In this context, *genetic regulatory networks* (GRNs) are emerging as a unifying operating concept [37]. A GRN model postulates that the genome can be described as a vast and complex network of interacting genes, promoting and repressing each other's expression through the intermediation of produced proteins that bind to specific regulatory sites on the DNA. Abstracting out the highly convoluted pathways and feedback loops between genes, RNA, proteins and metabolites, the state of a GRN can be formalized as a set of transcription levels, one for each gene; in the simplest case, a on/off state for each gene. From there on, it is conceptualized as a dynamical system that follows different trajectories in state space, typically landing on fixed-point or limit-cycle attractors. Each attractor can then be interpreted as a cell type with precise behavior and properties [98]. Added to the discovery of the *homeobox* genetic toolkit, development is construed as a highly complex process tightly modulated by an underlying GRN model: large-scale, coordinated changes in the phenotype are a more or less direct consequence of small, localized changes in the regulation of the developmental process [211], be they genetic [185] or environmental [206] changes.

2.1.1 An historical perspective

In a dissertation with a strong biological inspiration, it is not possible to discuss about diversity of form without taking into account a biological perspective. To put evo-devo in an historical context, we will summarize two of the main schools of thought in evolutionary theory[1].

In the first school of thought, traditionally identified with the *modern evolutionary synthesis* (but also known as *Neo-Darwinism*), genes are at the center of the evolutionary stage [179]. In this paradigm, the genome is essentially considered as a blueprint, and the process of decoding the programmed information in this blueprint is not viewed as a primary area of interest in evolutionary theory [129].

[1]While the theory of evolution is universally accepted in the scientific community, many details remain to be worked out, and controversies over the subtle (and sometimes not-so-subtle) details linger for decades. Consequently, the history of the relevant evolutionary thought in the last decades is so rich that a comprehensive analysis would fill several books [18, 72, 113]. However, since this is not the place to discuss these issues in depth, only a summarized account of the controversies will be provided in this introduction.

Evolution is considered the result of differential changes in gene frequencies among populations, resulting from genetic variation which is directly expressed into phenotypic variation and thus subject to natural selection. In this view, changes are always extremely subtle, and macroevolution is explained as essentially microevolution over very long stretches of time. Consequently, diversity is understood to be a byproduct of evolutionary dynamics, gradually building up as diverse sets of alleles arise among different populations of individuals, and these are expressed in consequently diverse sets of phenotypes, adapting to new environments.

In the second school of thought, which may be called *structuralism* [87], evolution is thought to operate at the level of organisms, which are considered as complex self-organizing entities. While natural selection is still given a key (or at least significant) role, organisms are considered as complex systems and this fact must be taken into account when considering them from any point of view (especially for their evolutionary history). In this framework, scientists have suggested that diversity arises from the differences in the self-organization of the developmental processes of different organisms, and morphological evolutionary trends are biased to a large extent by development [5].

It is important to note that, however bitter the debate may become [184], it is not a black and white one: there are many differing points of view ranging from these two extreme positions, and even staunch supporters of gene-centric evolution (as Richard Dawkins [38]) will generally concede that biases in developmental genotype-phenotype mappings may constrain natural selection in some ways, while it would be fairly difficult to find a structuralist (even a radical one, as Brian Goodwin [70]) ready to dismiss the genome and/or population genetics as a totally irrelevant part of the equation. The controversy is better understood as different scientists coming from fairly different scientific traditions, with different research interests and methodologies, stressing different parts of the subject, and more often than not misunderstanding the statements from people in the other side, and raging about the subject and their opponents.

In this context, it is interesting to frame the controversy in terms of Mayr's distinction between proximate and ultimate biological causes [129]. A proximate cause is a direct factor (physiological, developmental or environmental) to explain a biological trait, while ultimate causes are specific events and general laws providing historical explanations for the traits, interpreted as evolutionary dynamics. In Mayr's conceptual

map, proximate and ultimate causes roughly correspond to *how* and *why* questions, clearly separated in their scope and explanatory power. However, structuralists promote a more holistic approach [112], were traditionally *proximate* causes (like evidently, developmental processes, but not only them) are also used to understand evolutionary dynamics.

Traditionally, gene-centric evolution has been the mainstream discipline, while structuralists tend to be either outcasts (however brilliant and inspiring their work may be; D'Arcy Thompson being a classical example [194]) or polemic individuals, as Kauffman [98] or Gould [72]. In the last decades, structuralism is steadily gaining support through evolutionary developmental biology (evo-devo).

2.1.2 Evo-devo: a new synthesis

While the traditional Neo-Darwinian view focuses on population genetics to explain how biological diversity of form is shaped by evolutionary dynamics driven by natural selection, it is debatable whether it is able to provide a mechanism to explain how or why this diversity *arises* in the first place. A detailed knowledge of the genome enables scientists to pinpoint where and when different genes are expressed in the embryo, but it is insufficient to understand how the organism builds itself [140]. To cover this gap, evo-devo researchers describe developmental processes as a *coupling* between genetic regulatory networks and cellular biomechanics and chemical signaling [60, 211]. In this chapter, focus is put on the latter part of that interplay: the physical properties enabling biological diversity of form; specifically, the potential of tensegrity structures (see Section 2.1.4) to generate a wide diversity of forms.

Highly complex morphologies can be generated by developmental processes. Mechanical cues (strain, compressive forces, bending forces, and so on) play a key role in development by modulating cell differentiation, influencing the direction of growth, and deforming tissues [80]. Even late-stage (post-embryogenesis) events, such as the folding of the mammalian gut, are primarily determined by mechanical dynamics [171]. Geometrically, the embryogenesis of a typical triploblastic animal consists of a set of complex and carefully timed developmental steps, which dynamically change the physical properties of the cells, the surrounding extracellular matrix and the differential adhesiveness among cells and between cells and other substrates [60].

At a mechanical level, several types of morphogenetic processes can be distin-

guished, such as cell sheets growing in a given direction, or cells detaching from these sheets and migrating to create new groups. Some of the most common morphogenetic processes include *invagination*, where a sheet of cells folds inward at a point or line, possibly resulting in the folded cells detaching to form another sheet (e.g., the neural tube), and *branching*, in which a tube recursively branches off into a fractal-like pattern (e.g., lung organogenesis [91] or vasculogenesis [106]).

2.1.3 Artificial development

Moving now to an engineering perspective, the model presented in this chapter can be framed in the context of *artificial development*: an emerging field at the crossroads of computational biology, artificial life and evolutionary computation that studies computational models of developmental processes toward various scientific and engineering applications [183]. Typically, the use of evolutionary methods to conduct a parameter search or stochastic optimization becomes especially difficult when the individuals are complex systems, i.e. made of a large number of interacting components, because of their high dimensionality (number of degrees of freedom). Therefore, the application of evolutionary algorithms to very complex problems is limited by the ability to efficiently encode solutions to these problems.

To solve this issue, complex solutions to complex problems are proposed to be encoded indirectly through developmental processes (*via* a developmental or generative stage), in a compact way, reducing the size of the genome and thus enhancing the efficiency of evolutionary algorithms (despite the added developmental calculation). However, with the introduction of a developmental stage, a new layer of complexity is added, since in most models of artificial development published up to date, indirect encoding is implemented through a complex interaction between the genotype (often a genetic regulatory network) and the developing phenotype. Examples abound: Lipson's robots with genetic encoding based on L-systems [85], later versions of Framstick's genomes based on genetic programming techniques [102], or Lobo's path followers with artificial genomes [120].

Some researchers have investigated more complex and biologically realistic models of developing phenotypes with the intention of minimizing the size and complexity of the genotype even further to make it more amenable to evolutionary methods [47]. This represents, in essence, an application of morphological computation [154, 157] to

developmental processes: a methodology to exploit the dynamics of interaction between (in this case) the developing phenotype and the genetic control (see Section 3.1.1 for more details). In this respect, the model presented in this chapter represents a test on how small can be the genetic machinery of a model of artificial development while still enabling an evolutionary search to solve a problem (in this specific case, of complex developmental processes giving rise to a diversity of morphologies). To this aim, tensegrity structures are used, as explained in the next section.

2.1.4 Tensegrity

Once the biological and engineering perspectives have been presented, this section presents the concept of tensegrity, which plays a key role in the model used in this chapter.

Historically, many definitions of tensegrity structure[2] have been proposed [96]. Most of them are variations on the original definition of Snelson [180] and Fuller [65], respectively the creator and the namer of the concept (from a contraction of *tensional integrity*). The most generic definition of the concept can be phrased in the following way: a tensegrity structure is a stable object composed of a set of compression-bearing rigid elements connected by a network of tensioned cables, such that the sum of forces acting on each vertex is null [138]; that is to say, there is a balance of forces maintaining the structure in equilibrium. The compressive and stretching forces sustained by the elements of the tensegrity structure are defined collectively as the *self-stress* of the structure. This self-stressing characterizes the high structural resilience of tensegrity structures, known as *tensile integrity* [138]. Combined with an exceptionally high rigidity-to-mass ratio, all these properties make tensegrity structures an object of active study with numerous applications in architecture, civil and mechanical engineering [138].

The last decade has also seen applications of tensegrity structures in Robotics, evolving control systems for walking tensegrity robots by analytical [73] and evolutionary [153] methods, optimizing the dynamical response of wing frameworks [10], exploiting large-scale couplings and nonlinear dynamics of tensegrity structures to

[2]While the word *tensegrity* has been commonly used as a noun to name structures of a certain class, this usage results in awkward wording (particularly derived terms like *tensegrital*, *tensegritic*, etc., or the lengthy periphrases used to avoid them) when describing the properties of these structures. To avoid this issue, the term *tensegrity structure* will be used in the following.

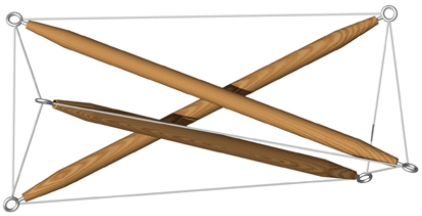


Figure 2.1: ***An example of tensegrity structure.*** **Tensegrity structures are composed of compression-bearing components (called *bars* or *struts*) tied together into a structure by stretching-bearing components, called *cables*. The whole structure is in a stable equilibrium.**

achieve locomotion with minimal control [166], and designing ballistic structures able to withstand landing impacts while minimizing the impact shock [121].

In these applications, the tensegrity structures are either selected beforehand or designed for the task to be solved. In both cases, the design of stable tensegrity structures is an extremely difficult task: if the potential energy of the structure (considered as the energy stored as compression and stress forces in the elements of the tensegrity, the self-stress) is considered as a function of the relative positions of the vertices (geometric configurations), the structure will be stable in some geometric configuration only if the potential energy function has a local minimum for that geometric configuration [32].

Several methods have been proposed to find new designs for tensegrity structures (see [195] and [96] for a review), but the problem of automating the discovery of new and complex tensegrity structures remains open, in spite of several new methods based in evolutionary algorithms, using either direct encoding schemes [155], L-systems [165] or developmental processes [120]. As explained in the following sections, the approach used here is more unusual, in that the topology of the structure is given beforehand, and new geometric configurations emerge through dynamical developmental processes. However, a systematic framework to analyze tensegrity structures and synthesize new designs is still lacking. Towards this end, Appendix A provides a mathematical analysis of tensegrity structures, based on a new way to decompose tensegrity structures into a set of constitutive atoms [78].

Getting back to the applications of tensegrity, for the purposes of this work it is interesting to note that it has also been used in Biology, being applied from the nanostructure (for example in the capsid of viruses [176]) to whole organisms (the musculoskeletal system can be considered as a tensegrity structure [181]). A subdiscipline in cellular microbiology and biophysics has been developed, working on Ingber's hypothesis that the cellular cytoskeleton is a dynamic tensegrity structure [89].

In this view, many properties of cells and tissues are hypothesized to arise from their nature as tensegrity structures:

- The extraordinary resilience and ability to bear high loads of the cytoskeleton [21], because in tensegrity structures, loads tend to be more or less uniformly distributed over a large portion of the structure.
- The mechanical properties of the cell [34, 89], that is to say, the mechanical response to cell deformation and substrate-dependent shape change. Since tensegrity structures store a considerable amount of potential energy in the self-stresses of their constitutive elements (compressive and stretching forces), they have potential for dynamically changing their shapes between metastable states.
- The conversion of mechanical signals into biochemical responses, or signal *mechanotransduction*, relying on force distribution throughout the cytoskeleton [90]. For example, cell fate, including mitosis, differentiation and apoptosis is governed in some cases by mechanical cues which are sensed through the cytoskeleton [28].

Interestingly, all of these properties are also significant from the point of view of developmental processes: as sheets of cells grow and cells assemble in specific locations, they are subject to forces that provide mechanical cues which in turn eventually affect the timing of the different steps of the developmental process. The tensegrity properties of the cytoskeleton are critical in supporting this whole process [126]. For example, in a recursive branching process, the growth of new branches is guided by local changes in the tension of the extracellular matrix [91], and in invagination processes, cells in a localized patch in an embryonic sheet undergo the contraction of their apical surfaces, inducing the invagination, just by mechanical cues [139].

2.1.5 Development with minimal genomic control

In this section, building upon the previously laid down concepts, the model and the main thesis of this chapter is proposed. The first concept to consider is that of developmental steps: as previously said, the developmental process of an organism consists of a series of complex developmental steps which have to happen in a very specific spatio-temporal pattern in order to build the organism without any

malformation. Evo-devo advocates have advanced the view that the morphology of the organism is that of a series of physical processes running during development, while the role of the genome is that of a regulatory network modulating cascades of morphogenetic processes at multiple levels [140].

The following hypothesis is the result of extending this view back in evolutionary time: in the beginning of multicellular life, the morphologies of organisms were generated through developmental processes guided by physical processes and their environmental conditions to a large extent, and only later genetic regulatory networks stepped in the role of stabilizing and canalizing the developmental process [144], making morphology subject to natural selection, though in an indirect way. Put in simple terms, morphologies emerged purely as the contingent but deterministic result of physicochemical constraints (not subject to genetic variation and natural selection in a direct way) of morphogenetic processes, and, later in phylogenetic history, new layers of genetic regulatory mechanisms helped to stabilize the pre-existing developmental process and expose it to natural selection. This hypothesis applies just to the very beginning of multicellular life, at a time older than the oldest reliable paleontological record of modern multicellular organisms.

In this context, it is interesting to note that the elements of a tensegrity structure store potential energy, and the structure can be unstable, meaning that if the balance of forces is perturbed beyond a certain point (be it by deformation of the shape or changing the characteristics or the stress of the constitutive elements), it will snap and settle into a possibly very different shape to accommodate a new balance of forces, releasing potential energy in the process. The work presented in this chapter is inspired on a model of developmental process which is based on this property of tensegrity structures. It is inspired by a simple 2D model of invagination [139, 148], where a circular sheet of cells is modeled as a closed chain of cellular cytoskeletons. The cytoskeleton of each cell is modeled as six overdamped elastic links: four external ones arranged as a quadrilateral and two more ones as the two diagonals of the quadrilateral. The quadrilaterals are attached together in a mass-spring model of a closed, circular sheet of cells (Figure 2.2). The whole sheet is a self-stressed tensegrity structure, with several possible configurations of compressed and stretched elements. To induce an invagination process in a region of the cell sheet, it is enough to shorten the natural length of the outer elastic links in that region. Thus, an invagination process can be

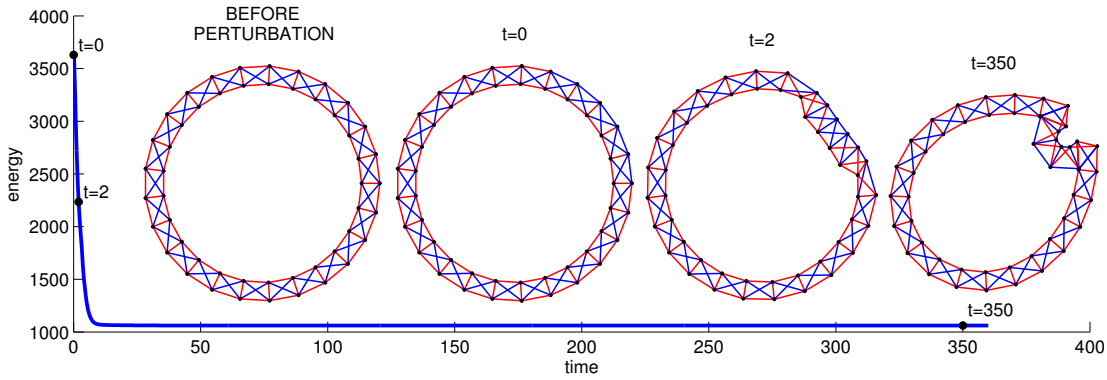
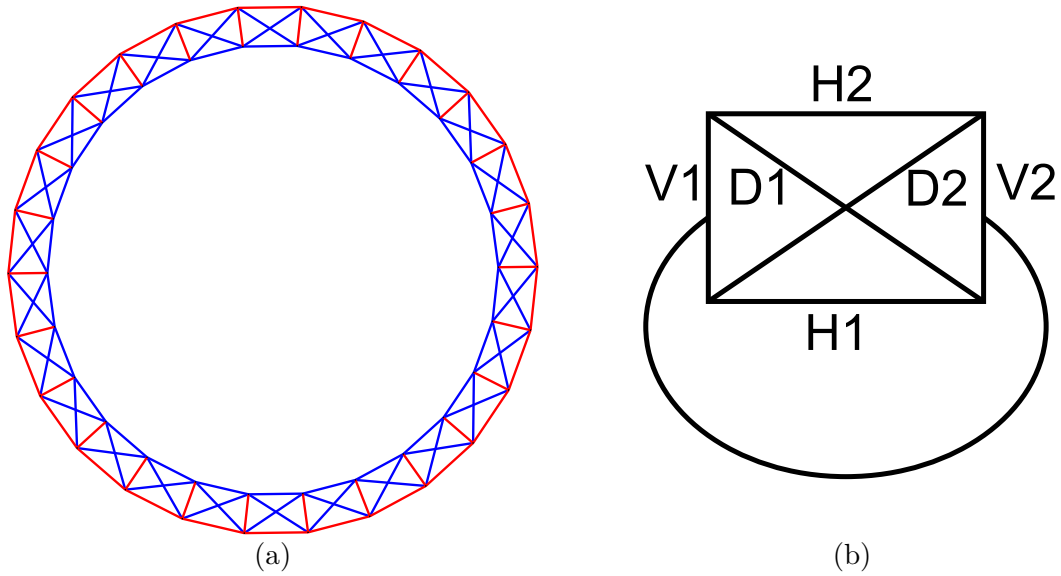


Figure 2.2: ***A model of invagination.*** **A 2D tensegrity structure modeling a sheet of cells (*before perturbation*). Each cell is modeled with six elastic links representing its cytoskeleton, chained together in a circular, closed sheet. Red elastic links are compressed, blue ones are stretched. The morphogenetic process of invagination can be triggered by just shortening the lengths of the exterior elastic links of a group of neighboring cells ($t = 0$), simulating a local contraction in their cytoskeletons [148]. As a result, the structure relaxes to a new equilibrium position, and the curvature of the resulting invagination depends on the parameters of the tensegrity structure (intensity of the self-stress, stiffness of the elastic elements). Three snapshots of the relaxation process are shown: one just as the simulation starts ($t = 0$), other after the invagination has started ($t = 2$), and other when equilibrium is reached ($t = 350$). A plot of the energy (potential energy plus kinetic energy) of the structure as a function of time during the relaxation process has been included. The structure dissipates energy to reach a new equilibrium state, and the resulting energy profile follows an (approximately) exponential decay law, typical of most perturbed tensegrity systems as they reach a new equilibrium.**

specified by a very simple and localized genetic signal controlling the length of some of the cytoskeletal elements in a region of the sheet of cells.

In this chapter, we adopt a similar 2D structure (see Figure 2.3.a) and drop its typical, direct interpretation as the dynamics of a physical sheet of cells. Instead, it is considered as an abstract model for morphogenetic processes, where the shape of the structure undergoes a complex and well-ordered set of morphological transformations, even if these are not topologically possible in an actual 3D structure. Tensegrity mass-spring structures are shown to be able to arrange a complex and well timed set of morphological transformations, with minimal genetic control, resulting in a diverse set of final shapes, thus exploring their potential as a physical substrate for complex developmental processes.



(a) (b)

Figure 2.3: ***Initial tensegrity structure for morphogenetic processes.*** **(a) Circular tensegrity structure of 26 connected cells used as the common initial state of all morphogenetic processes. Red elastic links are compressed, blue ones are stretched. (b) The six elastic links composing each quadrilateral cell, named H_1, H_2, V_1, V_2, D_1, and D_2, as shown in the figure. Each elastic link has a current length d, a natural length n, and a stiffness k. See Figure 2.4 for the numerical values of the parameters of the structure. The natural length of H_1 is *smaller* than its length in the circle (it is *stretched*), while for H_2 it is *longer* (it is *compressed*). Note also that for each pair of consecutive cells, the elastic link V_1 of one cell is superposed with the elastic link V_2 of the next one.**

2.2 A model with minimal genetic control

To explore the space of morphogenetic processes, an evolutionary algorithm has been used. In the algorithm, the individuals are sets of local and global perturbations which change the balance of forces and/or add potential energy to an initial tensegrity structure which is common for all individuals. To evaluate an individual, the initial structure is perturbed as specified by the individual, and the structure is simulated, undergoing a physical relaxation process until it stabilizes into a new shape. Then, the quality of each individual's morphogenetic process is measured. In the following subsections, the details of these methods are specified.

2.2.1 Individuals

As previously said, an individual is a set of local and global perturbations to an initial tensegrity structure (see Figure 2.3.a), which is the same for all individuals: a closed sheet of 26 quadrilateral cells. We denote by H_1, H_2, V_1, V_2, D_1, and D_2 the six types

link type	d	n	k
H_1	14.8285948251829	13.975	20
H_2	17.3933707403416	20.15	20
D_1, D_2	19.2641442449179	18.1264008079376	20
V_1, V_2	10.6389852023717	11.0406139435269	20

Figure 2.4: ***Numerical parameters of the initial tensegrity structure.*** **The initial tensegrity structure (Figure 2.3.a) is composed of two concentric rings of 26 equally spaced vertices of mass $m = 1$. The radius of the inner ring is** 61.5107152186897**, and the outer ring** 72.1497004210614**. For each type of elastic link, all links have the same parameters d (current length), n (natural length) and k (stiffness), as specified in the table.**

of elastic links of a single quadrilateral cell (see Figure 2.3.b), and by l_{ij} an instance of (undirected) link between two vertices i and j. At any time, the force exerted by l_{ij} on vertex i is defined by the equation $\mathbf{F}^e_{ij} = -k(n-d)\mathbf{u}_{ij}$, where k_{ij} is the stiffness constant of the elastic link, n_{ij} is its natural length (the length at which no force is exerted), d_{ij} is its current length (absolute values), and $\mathbf{u}_{ij}$ is a unit vector oriented from i to j. The initial structure is in equilibrium, i.e., the sum of the forces exerted by the incident links in each vertex is null: $\mathbf{F}^e_i = \sum_j \mathbf{F}^e_{ij} = \mathbf{0}$ for all i. See Figure 2.4 for the geometric specifications of the structure.

Each cell in the sheet is indexed by an integer $c \in [1, 26]$ number from 1 to 26. As an individual (a genome) is a list of perturbations, each perturbation can be deemed as a gene. A gene is specified as a quintuple $(c_a, c_b, S, t, x,)$, where:

- c_a and c_b are integer indices in the range $[-5 \dots 30]$ such that the perturbation is applied only to the cells that verify $c \in [\min(c_a, c_b) \dots \max(c_a, c_b)] \cap [1, 26]$.
- S is a subset of $\{H_1, H_2, V_1, V_2, D_1, D_2\}$ (the types of links in a cell, 2.3.b), such that, inside each cell c, the perturbation is applied only to the categories of links listed in S (, right).
- t is a symbol from the set $\{K, R, K_f, R_f\}$, and x is a real-valued coefficient $[0.5 \dots 2]$.
  - If $t = K$, the link's stiffness k is multiplied by x.
  - If $t = R$, its natural length n is multiplied by$x$.
  - If $t = K_f$, same effect as K; in addition, n is adjusted such that the magnitude of the force exerted by the elastic link $F^e = -k(n-d)$ remains

the same, under the link's current length d. If the adjusted value of n is negative, the changes are undone, to prevent unphysical parameters.

– If $t = R_f$, same effect as R using coefficient x' instead of x such that: if $x > 1$ then $x' = x$ for a compressed link ($n > d$) and $x' = 1/x$ for a stretched link ($n < d$), and conversely for $x < 1$; in addition, k is adjusted such that F^e remains unchanged. This rule is designed to avoid negative values in the adjusted value of k, because if the link is compressed, it becomes more compressed, and if is stretched, it becomes more stretched.

All perturbations coded in the genome are applied to the initial circular sheet of cells in the order they appear in the list, then the structure is left free to rearrange itself. Note that, while perturbations of types K_f and R_f leave the balance of forces seemingly unchanged, they potentially alter the mechanical response of affected links to subsequent perturbations, thus greatly influencing the developmental process. Finally, after all perturbations have been applied, links of stiffness k lower than a given threshold are discarded because they play no significant role in the dynamics. Likewise, the values of k and n are limited to given maximum values to prevent the system from becoming unstable.

2.2.2 Evaluation

Denoting the position of vertex i by $\mathbf{P}_i$, we can write $d_{ij} = \|\mathbf{P}_j - \mathbf{P}_i\|$ and $\mathbf{u}_{ij} = (\mathbf{P}_j - \mathbf{P}_i)/d_{ij}$. All vertices have unit mass ($m_i = 1$ for all i), and the stiffness is $k_{ij} = 20$ for every link l_{ij}. The equation of motion of i then reads $\ddot{\mathbf{P}}_i = \mathbf{F}_i^e + \mathbf{F}_i^v$, where $\mathbf{F}_i^e$ is the sum of elastic forces:

$$\mathbf{F}_i^e = -m_i \sum_{j=1}^{N} A_{ij} \cdot k_{ij} \cdot (n_{ij} - \|\mathbf{P}_j - \mathbf{P}_i\|) \cdot \frac{\mathbf{P}_j - \mathbf{P}_i}{\|\mathbf{P}_j - \mathbf{P}_i\|}$$

($A_{ij} = 1$ or 0 depending on the existence or absence of l_{ij}) and $\mathbf{F}_i^v = -\mu \cdot \dot{\mathbf{P}}_i$ is a linear damping force on vertex i, with a damping constant $\mu = 0.1$.

After perturbing the circular sheet according to the genome of an individual, the resulting structure is generally dynamically unstable, and it is simulated applying a standard method to integrate the equations of motion[3], and recording the simulation

[3]Specifically, `ode45`, the standard implementation in MATLAB® of the Dormand-Prince method (a

with a fixed time step $\Delta t = 0.01$. The total energy of the system is $E = E_k + E_p$, where $E_k = \frac{1}{2} m \sum_i \left\| \dot{\mathbf{P}}_i \right\|^2$ is the kinetic energy of the vertices and

$$E_p = \frac{1}{2} \sum_i \sum_j A_{ij} \cdot k_{ij} \cdot (n_{ij} - \|\mathbf{P}_j - \mathbf{P}_i\|)^2$$

is the potential energy stored in the elastic links. In energetic terms, the simulations are fueled by the conversion of E_p into E_k. In typical simulations, because of the damping force $\mathbf{F}_i^v$, most of the kinetic energy E_k is rapidly dissipated (especially after the initial steps of the relaxation, when the elastic links operate in the overdamped regime), hence E monotonically decreases over time. In rough quantitative terms, E_k is of the order of 10% of E in the very first steps of relaxation, then at best a fraction of 1% of E on the long run (see Figures 2.5, 2.6 and 2.7). The simulation is performed in chunks of 10 time units (1000 recorded steps at time steps of $\Delta t = 0.01$). When the difference in E between the beginning and the end of the chunk is lower than 10^{-5}, the simulation is stopped, because a low rate of energy loss suggests (heuristically) that the structure is close to its final stable configuration.

In a simple metastable system such as the invagination process of Figure 2.2, the relaxation from the initial to the final state follows an exponential-looking energy curve, characterized by a sharp drop before a slow decrease (see energy profile in Figure 2.2). Figuratively speaking, the potential energy is spent in just one shot. However, tensegrity structures have the potential to undergo much longer and more complex transformations, consisting of a sequence of developmental stages and qualitative transitions (Figure 2.5). We propose to heuristically detect the transitions through the conspicuous *accelerations* that they produce in the energy loss rate. Then, the number of steps of the energy function (and their size and distribution) can be measured, as an heuristic to measure the quality (fitness) of the developmental process.

Staircase steps in the energy curve $E(t)$ appear as peaks and valleys in $\left|\dot{E}(t)\right|$[†]. Let a_q be the time of peak q, b_{q-1} and b_q the times of the valley flats before and after a_q, and $d_q = E(b_{q-1}) - E(b_q)$ the energy drop corresponding to peak q. Non-significant peaks, defined as having a value lower than double the value of any of their corresponding

Runge-Kutta integrator with variable step).

[†]And also in $E_k(t)$, because energy is lost exclusively by friction, so the shapes of $E_k(t)$ and $\left|\dot{E}(t)\right|$ approximately match. However, in very strong oscillations of the vertices (especially, in the initial steps of the simulation), significant fractions of energy can be repeatedly converted back and forth between kinetic and potential, distorting the role of $E_k(t)$ as a signal of the staircase steps in $E(t)$.

valley flats: $\left|\dot{E}(a_q)\right| < 2 \cdot \max\left(\left|\dot{E}(b_{q-1})\right|, \left|\dot{E}(b_q)\right|\right)$, are discarded, merging them with the neighboring highest peak (so, if the peak a_q is merged into the peak a_{q+1}, the valley before a_{q+1} becomes b_{q-1} instead of b_q).

The initial steps of the relaxation are typically characterized by a very sharp drop in $E(t)$ at very high but often irregular loss rates. This shows in the curve of $\left|\dot{E}(t)\right|$ as an initial, short and crowded sequence of very high peaks and also high valleys. To discard this region from the analysis, we consider the first valley b' whose corresponding value of energy loss rate $\left|\dot{E}(b')\right|$ is below a given threshold, heuristically set to 50: $b' = \min\left\{b_q \,|\, \left|\dot{E}(b_q)\right| < 50\right\}$. Then, we define the sequence D of all the energy drops whose peaks are after that valley: $D = (d_1, \ldots d_n \,|\, \forall d_k : a_k > b')$. Finally, we rescale the values d_k in D with respect to its maximum value and define the fitness f as the sum of these values:

$$f = \sum_{k=1}^{n} \frac{d'_k}{\max(D)}$$

Therefore, by attempting to maximize this function, the goal is to favor processes characterized by an energy profile made of multiple steps of similar size, under the supposition that this is an heuristic signature for long and complex developmental processes.

2.2.3 Evolutionary algorithm

Randomly generated individuals are unlikely to produce complex developmental processes and diverse final morphologies. For this reason, the following evolutionary algorithm is used:

1. First, 100 individuals are randomly generated and evaluated. This set constitutes the initial population.

2. Then, the following steps are repeated 500 times:

   2.1 Individuals from the current population are selected through a size-4 tournament, with elitism, to generate a new population (i.e., the best individual is automatically copied, while the other individuals compete in randomly chosen groups of 4).

   2.2 Some individuals of the new population are mutated and re-evaluated.

As an individual is a list of genes, each gene specifying a perturbation $(c_a, c_b, S, t, x,)$ to the initial structure. Several types of mutation are possible:

- Delete a randomly chosen gene.
- Change the value of a randomly chosen component of a randomly chosen gene.
- Insert a randomly created gene at a randomly chosen location in the gene sequence.
- Duplicate a randomly chosen gene.

Each mutation is applied to each individual with a probability of 0.05. The initial 100 individuals contain exactly two randomly created genes, in such a way that the first perturbation gene is always global (i.e., the interval defined by c_a and c_b includes $[1, 26]$ completely), while the second perturbation gene is always localized to 5 cells or less (i.e., $|c_a - c_b| \leq 5$).

2.3 Experimental results

Through several runs of the above evolutionary algorithm, many examples of long processes containing multiple developmental steps were found. However, a sizable fraction of these were discarded: because the heuristics only relies on an indirect criterion (maximizing f, which is related to the number and relative size of developmental steps), in many instances the processes did not resemble a sequence of developmental steps, for various reasons. For example, in many cases, the structure collapsed into a small area, making it difficult to visualize a shape, and no meaningful diversity of morphologies could be inferred from these cases.

In other examples, however, the structure did not collapse upon itself and even exhibited interesting bilateral symmetry. In symmetric structures, shape creation could be followed visually as different segments of the cell sheet folded in various ways and gave rise to a wide variety of transient and final morphologies. Three examples are shown: the developmental processes in Figures 2.5, 2.6 and 2.7, whose genomes are in Figure 2.8. While these developmental processes are difficult to depict in static figures (because of their essentially dynamic nature), the figures are illustrative of two typical properties of a successful developmental process:

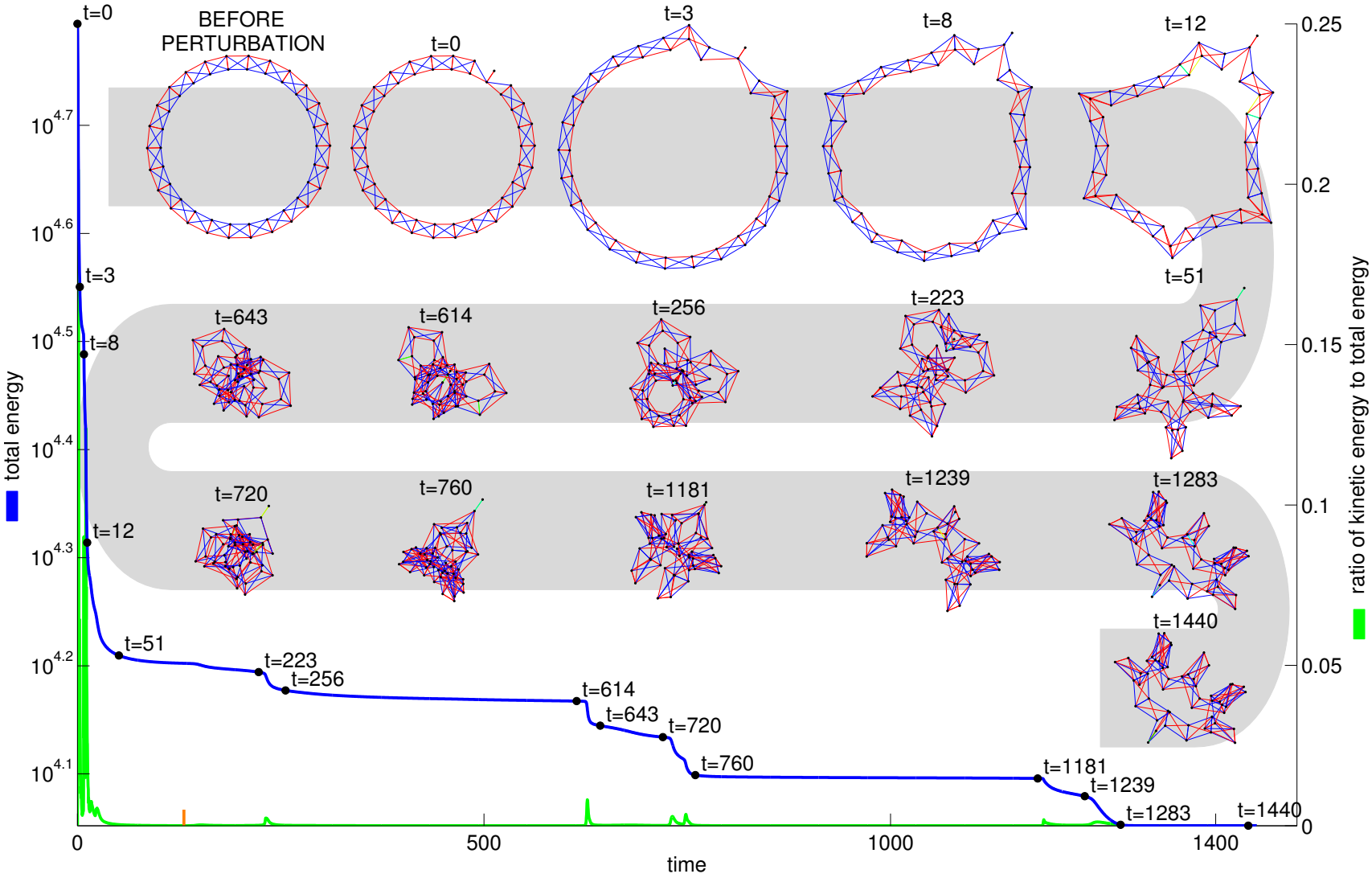


Figure 2.5: ***An example of developmental process (1).*** **See Figure 2.8.a for the corresponding genome and the text for details.**

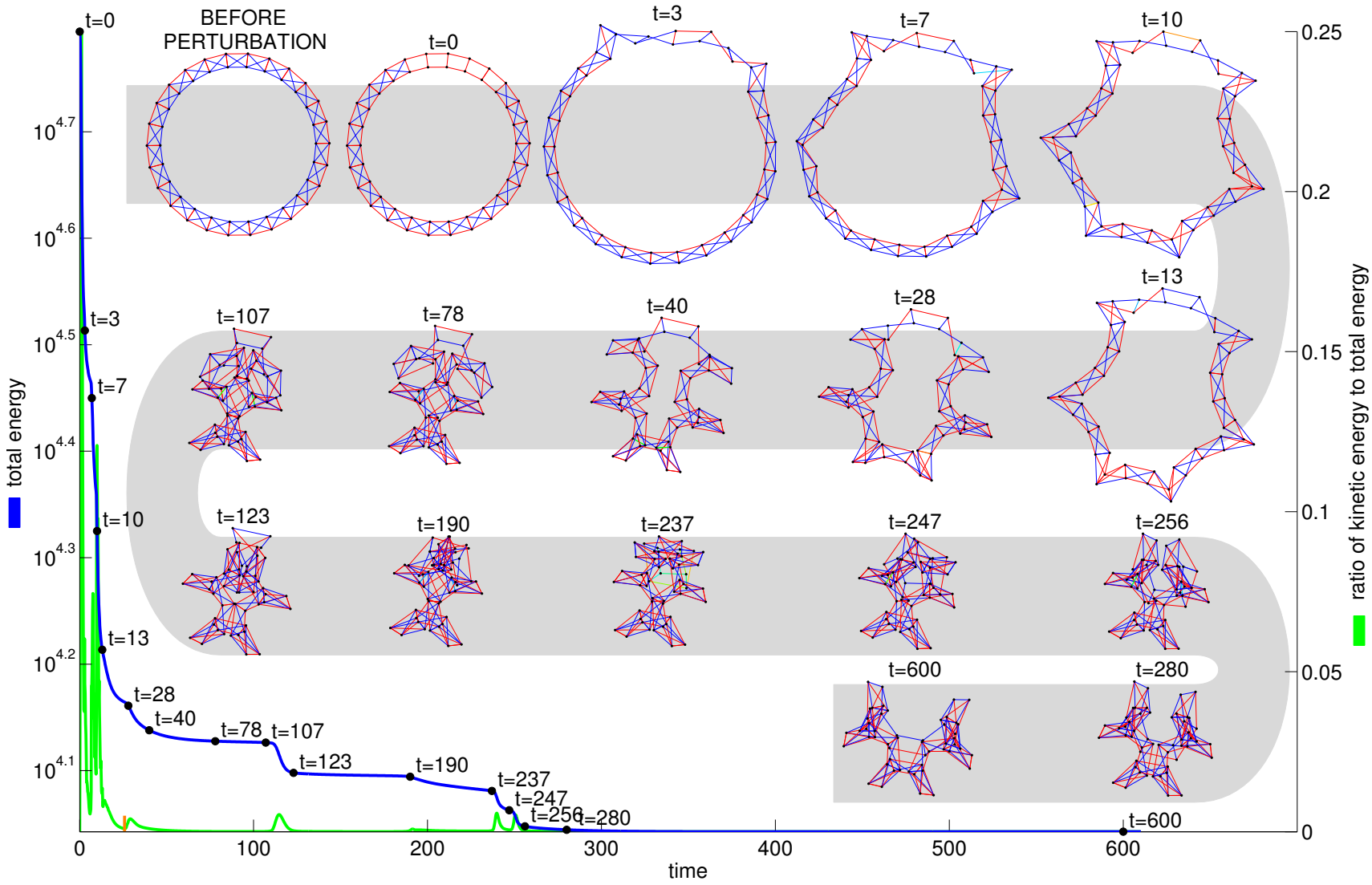


Figure 2.6: ***An example of developmental process (2).*** **See Figure 2.8.b for the corresponding genome and the text for details.**

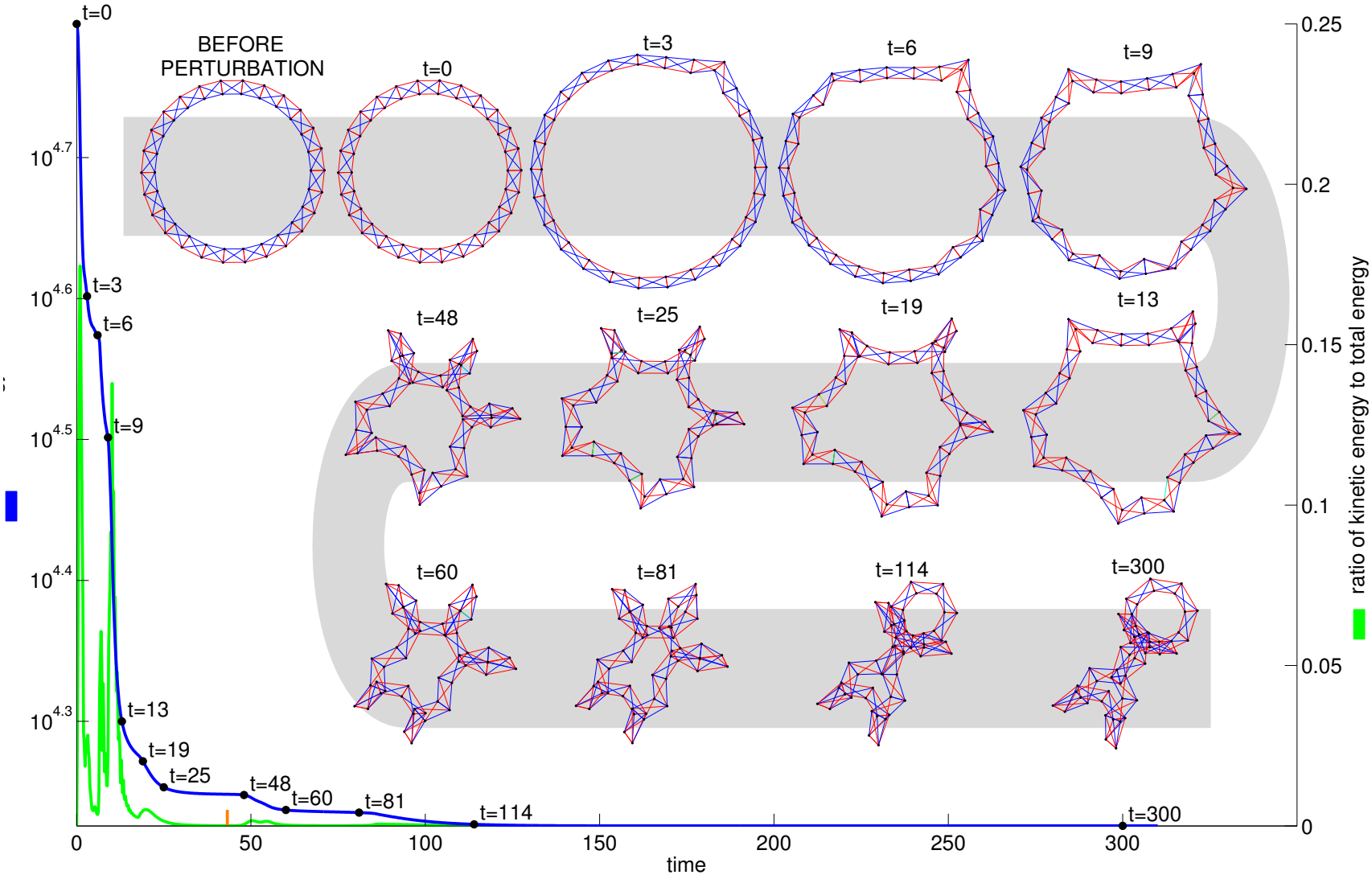


Figure 2.7: ***An example of developmental process (3).*** **See Figure 2.8.c for the corresponding genome and the text for details.**

(a)	(b)	(c)
$(1, 26, \{H_1\}, R, 2.14494510765828)$	$(1, 26, \{H_1\}, R, 2.14494510765828)$	$(1, 26, \{H_1\}, R, 2.14494510765828)$
$(1, 1, \{H_2, D_1\}, K, 0)$	$(2, 5, \{V_1, D_1, D_2\}, K, 0)$	$(1, 1, \{V_1\}, K_f, 0.81430031678588111)$
$(2, 2, \{H_2, D_2\}, K, 0)$	$(1, 1, \{V_2, D_1, D_2\}, K, 0)$	$(2, 26, \{V_2\}, K_f, 0.81430031678588111)$
		$(1, 1, \{H_2, V_1\}, K, 0.77561275506577265)$
		$(2, 2, \{H_2, V_2\}, K, 0.77561275506577265)$
See also Figure 2.5 and Figure 2.9.a	See also Figure 2.6 and Figure 2.9.b	See also Figure 2.7 and Figure 2.9.c

Figure 2.8: ***Genomes of the developmental processes.*** **Genomes are lists of perturbations (each perturbation defined as a gene) to the initial tensegrity structure. Each perturbation is a quintuple $(c_a, c_b, S, t, x,)$, such that the perturbation of type t and value x is applied to the links specified in S in the cells in the range $[c_a, c_b]$. See Section 2.2.1 for details on the meaning of the genes. Each genome corresponds to one of the three examples of the developmental processes examined in Section 2.3. For simplicity, these genomes have been simplified, combining consecutive sequences of genes applied to equivalent sets of links into equivalent single genes.**

- Developmental steps are characterized by periods of smooth changes separated by sudden transformations, during which potential energy is rapidly converted to kinetic energy, then dissipated, causing a drop in total energy. In most cases, these transitions correspond to sudden reversals in the positions of the vertices of one or several elastic links. The three figures show the typical staircase profile of total energy over time (blue line), $E(t)$, and the ratio of kinetic energy (green line), $E_k(t)/E(t)$ (drops in the former coinciding with peaks in the latter). While $\left|\dot{E}(t)\right|$ is not shown, its shape is approximately similar to $E_k(t)/E(t)$. In each figure, the small orange line marks the time of the first valley flat below 50 of $\left|\dot{E}(t)\right|$ (see Section 2.2.2 for details).

- The structure undergoes dramatic rearrangements. In each figure, several snapshots of the different stages of the corresponding developmental process are shown. The zigzagging gray path indicates their chronological order. Each snapshot is taken just before or after one of the transition steps characterized by a sudden energy drop. The time of each snapshot is indicated above it, and the corresponding locations are indicated on the blue plot of $E(t)$, indicating the position of the corresponding snapshot in time, and the energy of the structure at the snapshot..

In all three cases, the overall shape of $E(t)$ superficially resembles an exponential decay function. Accordingly, two regimes in the dynamics of $E(t)$ can be clearly distinguished:

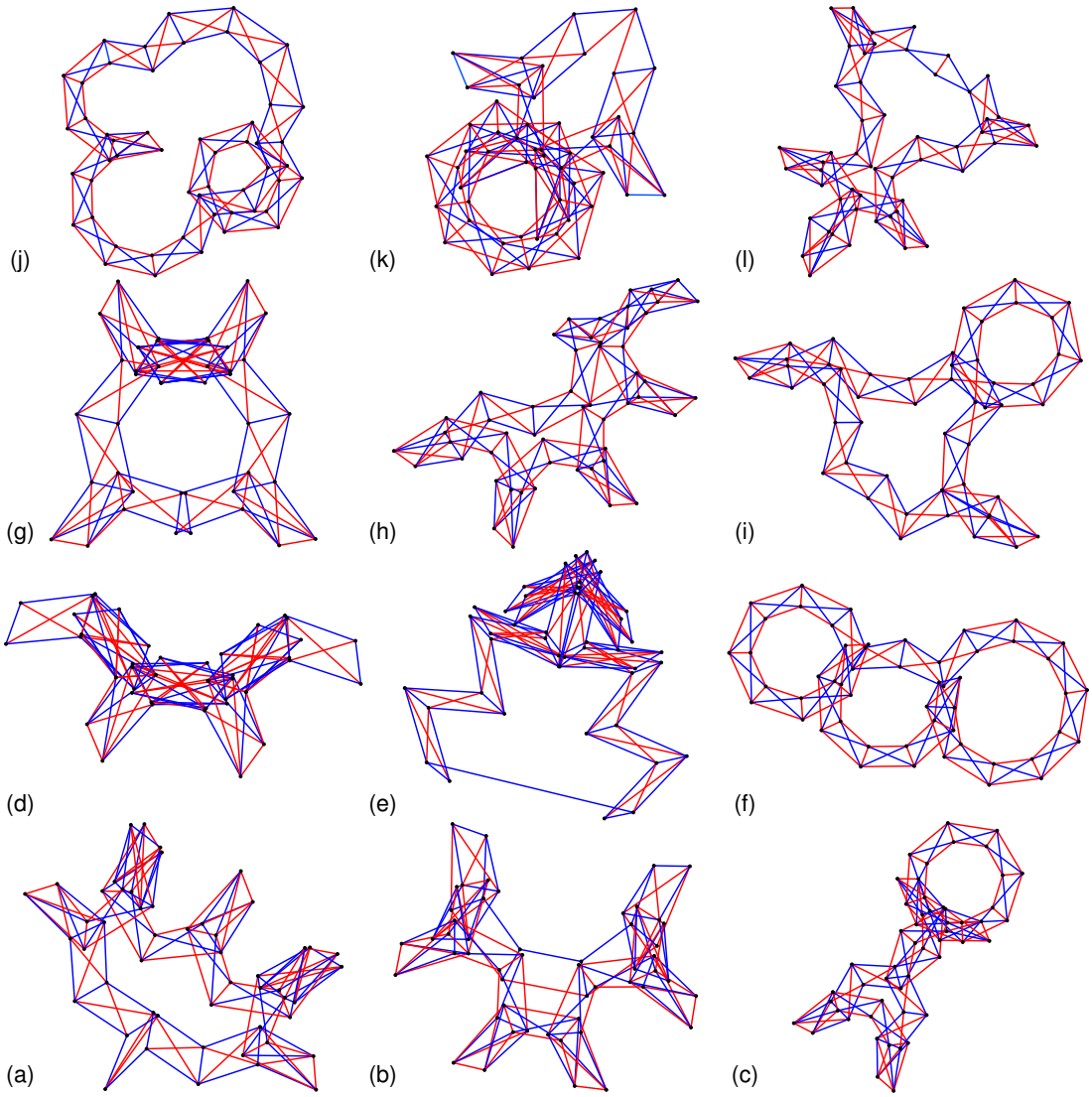


Figure 2.9: ***Examples of final morphologies.*** **Several examples of final morphologies resulting from different developmental processes, exhibiting more or less symmetry. All these diverse morphologies, are generated from the same initial structure (shown in Figure 2.3.a) by modulating the developmental process, which is entirely driven by the perturbations introduced at the beginning of the simulation (see text for details). The examples (a), (b) and (c) correspond to the developmental processes shown in Figures 2.5, 2.6 and 2.7, respectively. Relaxed links (bearing no compressive or stretching force) are not shown.**

- An initial phase where most of the energy is dissipated in a fraction of time. This first and intense *snap* corresponds to the setup of the developmental process, modulated by the local and global genomic perturbations that were stored in the structure.

- A much longer regime during which a fraction of the remaining energy is dissipated in much smaller quantities at well-delimited steps. In all three figures, the vertical axis for $E(t)$ (blue line) is a logarithmic scale to show these steps better. This slow regime corresponds to the developmental process proper, as previously explained in Section 2.2.2 (see also Figure 2.2 for an example of a simpler process with no steps).

The genomes of the three developmental processes (see Figure 2.8) are composed of a set of global as well as local perturbations (see Section 2.2.1). Global perturbations have the effect of introducing a significant amount of potential energy and destabilizing the initial structure (common to all developmental processes, see Figure 2.3). The local perturbations have the effect of canalizing the destabilized structure into an specific developmental path: in each case, the specific local perturbations modulate the location and dynamics of the initial catastrophic events (the reversals of elastic links). In turn, these initial events configure the subsequent developmental path of the structure. The final morphologies after the developmental process has finished (that is to say, when the structure finds a stable configuration) are diverse and strikingly different from each other; several examples are shown in Figure 2.9. Each example can be considered as a different sets of convoluted foldings of the initial structure, where different segments of the initial sheet of cells (Figure 2.3) are folded into various configurations with different curvatures.

2.4 Discussion

In this chapter, a model of development is proposed, able to give rise to a diversity of morphologies under minimal genetic control. While it is common to consider that, in most cases, a complex genetic control is necessary to guide an equally complex developmental process, essentially by stabilizing it and intervening at bifurcation points to switch on and off distinct, heterogeneous phases (see, e.g., [42]), the model presented here represents an extreme example of self-controlled physical structure. No genetic regulatory network is needed to guide the unfolding of its dynamics (the developmental path), since the properties of metastable tensegrity modulate themselves the sequence of transformation steps. Perturbed in some specific ways, the structure follows a complex sequence of morphological rearrangements and state transitions before settling down

into a final configuration.

This whole process is enabled by the self-stress of the circular sheet used as the initial state of the model. This tensegrity structure is stable in geometric configurations that correspond to local minima of the energy function (see Section 2.1.4), and the role of the perturbations specified by an individual's genome is precisely to radically alter the shape of this energy function. This drives the structure into a developmental path (a trajectory in phase space, in the terminology of dynamical systems), where each step is a consequence of the geometric changes created in previous steps, until a new balance of forces (i.e., a local minimum of the new potential energy function) is attained. Although the primary goal was to study development as such, this method can also be seen as a new way to generate novel tensegrity structures (see Section 2.1.4 for other methods).

2.4.1 Relevance to Biological Modeling

With respect to evo-devo, the significance of the model presented here is to cast a new light on the question still little addressed in today's predominant gene-centric view of Biology, but mostly investigated through the structuralist tradition: can organisms also be the product of complex physicochemical developmental processes not (necessarily or always) controlled by complex underlying genetics?

Each element of the described tensegrity structures influences the stress state of the other elements in highly convoluted, nonlinear ways, creating the conditions for enhanced developmental effects, while only a small amount of genetic information is sufficient to encode structural perturbations. This can also be seen in unicellular organisms whose complex cytostructure develops from an undifferentiated stage. A classical example is Goodwin's model of *Acetabularia* [70], a genus of green algae in which the genome does not explicitly code for the branching *bracts* (rings of little leaf-like elements) of a growing cell. These bracts are only the indirect result of a set of physical initial conditions, which can be modulated by the genome to produce different morphologies in different species. This represents a real-life example of developmental process primarily based on the physical properties of the developing structure, as in our model.

As seen in Figures 2.5, 2.6 and 2.7, the developmental processes studied here are characterized by their typical sequence of steps in the amount of energy over time. These steps are catastrophic geometric events, in the sense of René Thom's

theory [193], frequently involving reversals of the elastic links. These reversals cause radical changes in the balance of forces of the structure, enabling further modifications in its geometry. In Kauffman's interpretation of Waddington's epigenetic landscapes, cell types are metastable states of the cell's genetic regulatory network [98]. The morphodynamics described in this chapter could thus be considered a purely physical analog of Kauffman's metaphor, at an abstract level of self-organization. A metastable morphology represents a transient attractor in the physical dynamics of the structure, as it visits these attractors before arriving in its final state.

Although the model presented here is not related to any real-world biological system, it represents a proof of concept that complex developmental processes do no always need to be controlled by genetics in a direct and thorough way, but can largely self-regulate by relying on the physical properties of growing tissues [143]. In this respect, it lends support to the hypothesis that, in the beginnings of multicellular life, organismal development was under little to no genetic regulation [144], instead being guided by structural constraints and morphological and material properties.

2.4.2 Relevance to Bio-inspired Engineering

From the point of view of evolutionary computation and systems design, the aims of artificial development are to discover and improve algorithms by leveraging self-organizational processes to code complex solutions with small genomes [47]. Yet, even in this field especially suited to the design of embodied agents, a common assumption is that generating architectures complex enough to be capable of spontaneous innovation requires direction from a complex GRN or GRN-like mechanisms distinct from the physical model proper. As mentioned in Section 2.1.3, many AD models choose to support complexity in the development by a relative complexity in its recipe [42, 47, 85, 102, 120].

The developmental model proposed here is purely *ballistic* in the sense that genetic control is limited to providing perturbations to an initial metastable structure. In this respect, it represents a proof of concept that evolutionary algorithms based on artificial development [183] can encode solutions to form-finding problems in extremely concise ways. Earlier explorations of this concept [47] have used more complex genetic machinery, which were avoided here by using tensegrity structures.

Naturally, starting from this minimal base, it would be also perfectly possible to

extend the model and reintroduce more genetic information. For example, one could allow perturbations to be triggered not only at the beginning of development but at later times, too, through various physical parameters, like the level of pressure in a vertex, the amount of compression or stress in an elastic link, or some geometric condition, like the folding of a specific segment of cells. This would certainly allow for an even more complex and more reactive growth dynamics, where the development process is still mainly determined by the physics of the model, but the genome steers and modulates it along the way, better resembling real-world biological development.

Chapter 3

Diversity by coevolution of the body and the control system

In this chapter, agents are modeled in a detailed way and evolved using a conventional evolutionary algorithm, without direct interaction between the members of the population or a complex developmental process as in Chapter 4. Diversity emerges instead from the complex modeling of agent physics, enabling the coevolution of morphologies and the behaviors (gait patterns) associated to them. This is similar to the approach used in Chapter 2, though there is not developmental process (at least not in the same sense), but the complexity is in the interaction of the agent with the environment.

Specifically, a framework is formulated to evolve computational analogues of biological molecular motors, which are nanoscale devices capable of transforming chemical energy into mechanical work. Results from elastic network analysis and behavior-finding methods are applied to explore a subset of the configuration space of template molecular structures that are able to transform chemical energy into directed movement. The results show that molecular motion can be attained from a variety of structural configurations, each one inducing a distinct gait pattern.

This chapter is organized as follows. In Section 3.1, a comprehensive review is provided from the points of view of artificial life and molecular computational biology, introducing previous work in similar agent-based models, and the theoretical

The results presented in this chapter have been published in [54], except for Section 3.3.2.

underpinnings in molecular biology. Then, in Section 3.2, the framework is presented, thoroughly specifying all the relevant details. Section 3.3 introduces the results of a full set of computational experiments, elaborating on some practical applications of the framework. Finally, in Section 3.4 the conclusions derived from the results are discussed.

3.1 Introduction and related work

Briefly stated, the intent of this chapter is to explore agent-based models of protein-like structures to study the emergence of diverse morphologies and behaviors in the line of artificial life, while at the same time providing a formal model to study molecular proteins. Because of this double intent, the related work will be reviewed in two parts: first from the scope of artificial life, and then in the context of computational molecular modeling.

3.1.1 Evolution of morphological and behavioral diversity

In the last 15 years, one of the main research lines in evolutionary computation and artificial life has been the study of the emergence of complexity and diversity using agent-based models. To implement these models, researchers usually follow one of two strategies. The first one, less computationally intensive, uses a relatively high-level modeling of agents, endowing them with hard-wired biological characteristics in a rather ad-hoc manner (for example, a *position* which is shifted according to the agent's control system, an *energy counter* which is increased when the agent *eats* some *virtual food*). While these models are useful to study the emergent dynamics of agent behavior and interaction between agents at a high level [151, 164], hard-wiring the agent's characteristics into the model limits their scope.

The second strategy models the agents in a more detailed way, often with a more or less rigorous simulation of physical dynamics. This enables the agents to display emergent properties in a more natural way. Early studies explored the parameter space of the models by manual tweaking of the parameters [20, 162], but as computer power has increased over the years, evolutionary methods have been increasingly employed. The work of Sims provides an early and classical example: a rich and complex simulation of agents, embedded in a physically realistic environment, enables

a stunning diversity of morphologies and behaviors/strategies to evolve by using fitness functions to reward the best agents in locomotion [178] and competition tasks [177]. This diversity emerges not because of complex ecological interactions between the agents, but because the model is rich enough to enable many different solutions to the same problem. In this sense, while a physically realistic and detailed simulation is not absolutely required, it constitutes a common way to enrich the model.

In a similar line to Sims', many models have been devised in the last 15 years (including recreations of his work [27]), with many differences in their intent, scope and simulation techniques, but a similar overall goal to study complex models of embodied agents, able to generate diverse morphologies and behaviors (for example, walking, swimming, block pushing, etc.). Some models strive to create more or less complete ecologies of embodied agents, like PolyWorld [220], Framsticks [103] and Virtual Blocks [182] (also inspired in Sims' work). Others start from an even more bottom-up approach akin to Artificial Chemistry while also employing physical simulation, as JohnnyVon [48]. Even amateur online communities focused on evolution but also manual design of walking agents, like Sodarace [132] and Darwin@Home [39], have generated a diversity of walking strategies.

This research line also connects with computer graphics animation, where agents are meshes whose motion has to be automated, and the goal is to design sets of control systems to drive the mesh movement and/or behavior. Specifically crafted optimization methods are often employed to design the control systems, as in the design of artificial fish [189, 192]. However, evolutionary search techniques have been also employed to generate gait patterns for arbitrary rigid-body agents [204, 207]. The traditional formulation of evolutionary robotics is very similar: to evolve control systems for simulated (but realistic) robots [58] with a wide range of preset morphologies [88, 101].

A relatively recent trend in evolutionary robotics is the coevolution of body and control system in computational models of robotic agents, adapting to each other during evolution. In general, as in this coevolution there are much more variables to optimize, the search space becomes very high-dimensional, so indirect encoding [183] (development of both body and control system from a compact genetic specification) is used. This is the case of the work on the subject of Bongard [13] and Hornby, Lipson and Pollack [85, 158], to cite the more noted examples. However, this coevolution is also present in other studies outside evolutionary robotics proper, from already cited

examples like PolyWorld [220], Framsticks [103] and Virtual Blocks [182], to computer animation [175] and other comprehensive studies on the power of indirect encoding and body-control system coevolution, as Lobo's path followers [120]. In all cases, diverse morphologies and/or behaviors emerge through coevolution.

In almost all these models, the control system is fairly complex (often, some kind of recurrent neural network). However, it has been shown that in many cases this is unnecessarily overcomplicated. In a seminal work, Paul demonstrated that the whole body-control system is able to perform more complex computations than the control system alone [154]. This observation spawned the concept of *morphological computation*: a design methodology for robotic agents, exploiting the dynamics of interaction between the body and the control system of the agent, in order to keep the control system as simple as possible. The applications range from semi-passive bipedal robots with minimal control systems [128] to tensegrity robots whose complex, coupled non-linear dynamics are harnessed to generate a gait pattern with minimal control [154], and robots with open-loop control systems and minimal numbers of degrees of freedom, which self-stabilize fast gait patterns and generate diverse sets of behaviors through the interaction between the body and the control system [157].

Lobo's path followers [120] represent an extreme example of morphological computation, since the control is purely mechanical, with no attached abstract model of control system. However, it still relies on a complex developmental process (indirect encoding) to generate a diversity of morphologies which in turn induce diverse behaviors. It is at this point where the work presented in this chapter can be categorized: it represents, like Lobo's path followers, an extreme example of morphological computation, but the agents will not undergo a developmental process, demonstrating that a rich and complex modeling of the environment and the agents is sufficient to induce the coevolution of a diversity of morphologies and gait patterns.

3.1.2 Biological molecular motors

In the previous section, a comprehensive review has been performed to situate this work in its corresponding place within evolutionary computation and morphological computation. However, this work equally aims to be rigorously founded in biological molecular modeling, so a review in this area is also mandatory.

Cells can be deemed as complex, highly organized molecular assemblies. To achieve

such a degree of complexity and organization, cells use a wide array of means. Molecular motors, being able to transform chemical energy into (harnessable) mechanical work, are one of such means. These devices underpin most vital functions related to motion and cellular rearrangement at all scales, in all domains of life [173]. Molecular biology is the natural domain of scientific inquiry for these devices, but interest in them spans most of life sciences, from medicine [69] to cladistics [202], to cite just two disparate examples, and also nanotechnology [205].

There are many kinds of biological molecular motors. Here, the focus is put on motors able to push or walk along cytoskeletal filaments. There are three classes of such motors: myosins, kinesins and dyneins. These molecular motors perform many tasks: moving cargo around the cell and changing its shape, inducing cilia and flagella beatings, and muscle cells contractions [173, 202, 203]. As they are so crucial for most cell functions, they elicit a great deal of scientific interest. They have been extensively studied in order to understand their biochemical and structural features [203], and their *processivity*, i.e., the way some of them are able to walk along filaments taking many steps in a row [68, 209, 225]. There are also other instances of biological molecular motors which travel along other cellular filaments, as DNA helicases and RNA polymerases, though mechanical work is not their primary function.

As computers have grown more and more powerful, they have become able to tackle ever more computationally heavy and complex tasks, and computational methods have come to a paramount position in molecular biology research. Specifically, many computational studies based on simulations of kinesin and myosin molecules have been performed. The most detailed models require minute molecular dynamics simulations with explicit solvent [99, 104, 119], but the large amount of computations needed to simulate reasonable time spans, even in the nanosecond scale, renders this method impractical for performing many consecutive simulations. More lightweight approaches have been devised, such as rule-based models [77, 215], classical coarse-grained models [19, 35, 152, 191], and even very simple models based on reaction kinetics [172] and ratchet potentials [30, 97]. These lightweight approaches vary over their degree of detail, explanatory power, and rigorous foundation.

3.1.3 Molecular motor templates

The concept of *molecular motor template*, introduced in this work, is defined as an elastic network model of an abstract molecular structure, with one or more catalytic cores and docking sites, which is able to undergo a cycle of conformational changes (a *working cycle*). This cycle permits the structure, when it is suited to the task, to processively move through a filament. Although the evolution of the working cycle is a very interesting research question, it is not the subject of this work. The chemical realization or realizability of a given molecular motor template is not addressed here, either. Instead, this work focuses on gaining insights into evolutionary and structural features (that is to say, features related to shape) of actual molecular motors by studying molecular motor templates as simplified abstract models.

In this chapter, the design of processive molecular motor templates has been considered as a behavior-finding problem [120]. In this framework, structures are selected when their behavior (resulting from the interaction of its morphology with the environment) verifies a set of restrictions. Usually, the evaluation of the behavior involves the physical simulation of some task [166]. Here, the task consists in traveling along a filament, and evolutionary computation is used to explore a search space of candidate structures. Although the task is very simple and the model is very canalized (in the sense of using common heuristics to determine the configuration of all the molecular motor templates), diverse morphologies and behaviors (gait patterns) have evolved, as explained in Section 3.3.

Some other questions about proteins have been also addressed by evolutionary search techniques, as protein folding [135, 150, 190, 218, 219] and protein evolutionary dynamics [141, 219]. Another study [197], which has been a source of inspiration for this work, has applied simulated annealing to study structural properties of peptides. A related work whose approach is similar to the one presented here has been published recently [57]. However, this is the first study targeting questions about molecular motors through evolutionary search techniques.

3.1.3.1 Elastic network models

X-ray crystallography, NMR and related techniques have improved as computer machinery has become more and more powerful. The elucidation of the 3D structure

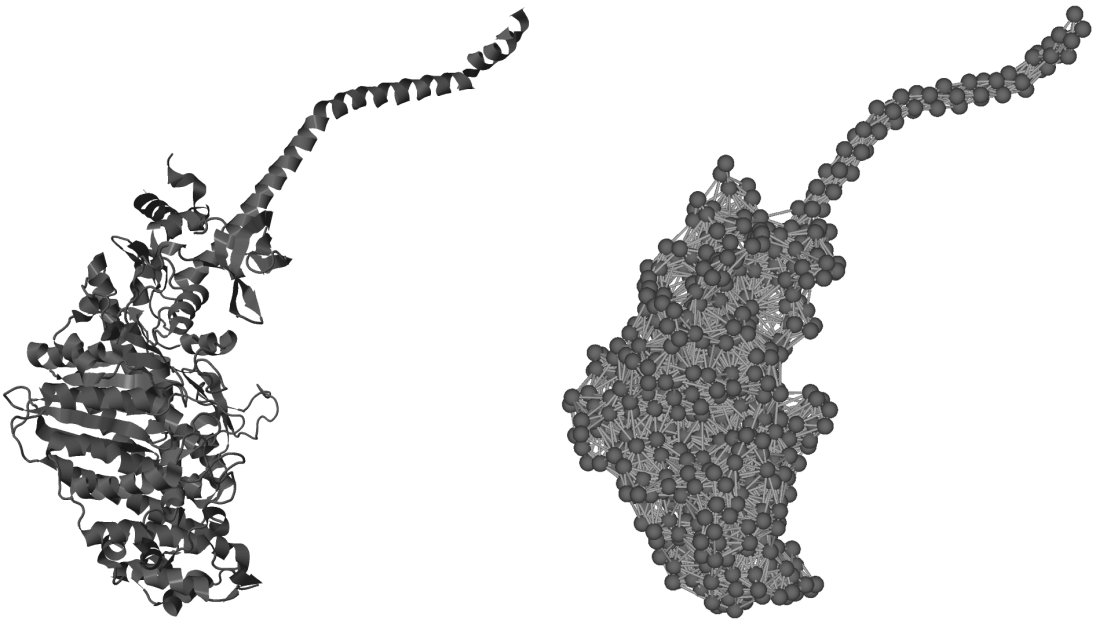


Figure 3.1: ***Elastic network model of myosin.*** **In this picture, an elastic network model of a single molecule of myosin is represented (right), where each aminoacid is replaced by a vertex (with a diameter of 3.8 Å), and vertices are connected if they are within a given cutoff distance (10 Å in this example). For comparison, the ribbon diagram of the protein is also shown (left).**

of proteins and other macromolecules has changed from an extremely slow, time-consuming process to a largely automated one, enabling a rapid growth of repositories of 3D structure data of proteins. However, the raw data provided by these techniques is still difficult to interpret, and complementary mathematical tools are needed to extract useful knowledge from the data.

In this regard, normal modal analysis (NMA) has long been used to understand the dynamics of protein structures. In NMA, a structure is modeled as a collection of objects (atoms or groups of atoms) subject to a number of energy potentials modeling their interactions. Then, the normal modes of motion of the model are analyzed to understand its dynamics. At first, very detailed NMA models were developed, explicitly modeling individual atoms and detailed quantum potentials [25]. However, researchers gradually realized that normal mode analysis was able to yield useful results and insights even using simpler and more coarse-grained NMA models [196]. Eventually, a consensus developed: for many proteins, their dynamics are largely dictated by their overall structure rather than by the specific shape and intensity of the energy potentials [41, 82, 122, 188, 227]. Thus, while complex and detailed models retained

their utility and are still being developed [81, 137], simpler and more elegant NMA models were soon formulated, as the Gaussian Network Model (GNM) [9, 79] and the Anisotropic Network Model (ANM) [7, 50].

Figure 3.1 shows how an elastic network model is obtained from the 3D structure of a protein. In the GNM, proteins are modeled as elastic networks, each residue (i.e., each aminoacid in the peptide chain) represented by a vertex placed at the position of its C^{α} atom, where vertices are connected if they are within a given cutoff distance. Two conditions are also required: (a) the vertices are assumed to fluctuate around their default positions with random isotropic Gaussian distributions, and (b) the network potential depends on the magnitude of the displacement of the vertices around their default positions (e.g., isotropically). If these conditions are met, the fluctuations of the vertices depend just on the topology of the network, up to a factor [161].

Despite its simplicity, this model consistently yields good estimates of the empirically measured fluctuations for a wide range of proteins. The normal vibrational modes predicted by GNM are computed from the Kirchhoff matrix of the underlying graph. The Kirchhoff matrix $\Gamma_{n\times n}$ (also known as the Laplacian matrix) for a graph with vertices $v_1 \dots v_n$ is defined as:

$$\Gamma_{ij} = \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } v_i \text{ and } v_j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

Where $\deg(v_i)$ represents the number of connections of the i^{th} vertex. The normal modes of vibration are given by the spectral decomposition of Γ: each eigenvector X_i represents a vibrational mode, its components representing the amplitude of the movement for each vertex, and the associated eigenvalue λ_i is proportional to the square of the frequency of the mode. In this way, the slowest vibrational modes of many proteins are fairly well predicted by GNM, a fact which has been exploited to explain with reasonable accuracy many structural features of proteins, as their unfolding pathways [187], their domain decomposition [110], their conformational changes and the position of their catalytic cores [222].

However, GNM has a drawback that becomes problematic for some applications: it tells nothing about the directionality of the fluctuations, because of the isotropic assumption (displacements from the default position affect the potential function in

the same way, regardless of direction).

The Anisotropic Network Model (ANM) is one of the simplest models dropping this assumption, and, after linearization of the equations of motion, the potential of the elastic network becomes dependent on the directions of the connections between the vertices, just like in a mass-spring system. For this reason, the ANM is able to predict the directionality of the fluctuations of the vertices, as well as the geometry of the conformational changes. However, this presents a downside: the level of agreement between experimental data and ANM predictions is lower than for the GNM, as the predicted fluctuations are higher than in the GNM, because the motion constraints are weaker [161]. This can be overcome by lengthening the cutoff distance (up to 10 Å, and sometimes even higher), at the price of a lower level of realism.

Despite these drawbacks, ANM and related models can be successfully applied to a number of problems: interpolation methods to decompose proteins in structural domains [229], describing conformational transitions [100, 156, 226], describing structural differences between related proteins [46, 114], analyzing the deformation of proteins under mechanical stress [49], and protein vibrational dynamics [133, 134, 223].

However, it is worth mentioning that normal mode analysis based on elastic network models fail for some proteins, remarkably for the motor protein kinesin [198, 228].

3.2 A formal framework for molecular motor templates

The evolutionary search method presented here evaluates, selects and mutates elastic networks. However, these elastic networks are not themselves the molecular motor templates: rather, each elastic network deterministically encodes for a template. The details will be presented in the following subsections.

3.2.1 Definition and generation of elastic networks

Elastic networks representing raw structures are created as randomly folded chains of N vertices, roughly modeling a folded peptide. Each vertex has a diameter D. The vertices are laid down in sequence: after setting the first one in place, the successive vertices are placed at random, one after the other, such that each vertex is at a distance L ($L_{min} \geq L \geq L_{max}$) from the preceding vertex. They also must be placed at a distance M ($M_{min} \leq M \leq M_{max}$) from at least K vertices, out of the C preceding

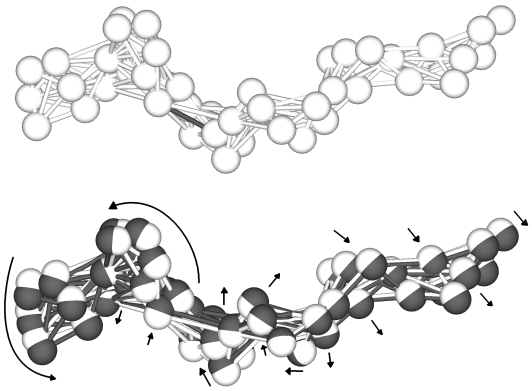


Figure 3.2: ***A mutation example.*** **A structure is to be mutated (above) by perturbing the natural length of an elastic link (dark gray). In this case, the link is lengthened. The resulting structure after relaxation is shown (below) along with the original structure, in dark gray. The vertices have been displaced, and arrows point towards the main direction of displacement in each part of the structure. The configuration of elastic links is recalculated (some links may disappear, and new ones may appear, if the distances have changed beyond the cutoff value), although in this case no great changes have been done; only natural lengths are changed. After the mutation, the catalytic cores and docking sites of the corresponding molecular motor template are also recalculated (see Figure 3.3).**

vertices in the chain ($K \leq C$). After all the vertices have been placed, elastic links are instantiated between every pair of vertices at a distance lower than M_{max}. The natural length of each elastic link is set to the distance between its corresponding vertices.

In this work these parameters have been set to $N = 50$, $K = 3$, $C = 10$, $D = 3.8$, $L_{min} = 0.9 \cdot D = 3.42$, $L_{max} = 1.1 \cdot D = 4.18$, $M_{min} = L_{min}$ and $M_{max} = 10$. This parametrization is very similar to the one used in [197], but, by setting $K = 3$ and $C = 10$, it is ensured that (almost always) no point has any rotational degree of freedom. Once these parameters are fixed, a generated elastic network can be specified by a sequence $[\mathbf{P}_1, \ldots, \mathbf{P}_N]$, where each $\mathbf{P}_i$ is the position of the vertex i of the structure.

The value chosen for D is reminiscent of other studies, where each vertex represents an aminoacid, and the distance between the locations of two consecutive aminoacids in a peptide chain is supposed to be 3.8 Å. However, this work is not concerned with the exact scale of the elastic network models. In fact, the direct interpretation (i.e., a correspondence between vertices and aminoacids) is not possible due to the small number of vertices. Instead, the structures generated with this method should be regarded as rough templates of plausible proteins.

3.2.2 Mutation operator

As evolutionary search methods are to be applied to optimize the structural templates, at least a mutation operator must be defined. A mutation operator must take one elastic network and transform it in some way to produce a new one.

The most simple way to mutate a network is to add a small and random displacement to one or more vertices [197], but this straightforward choice makes the fitness landscape too difficult to explore, since to get from some given individual to a better one, many correlated mutations might be required. For example, if an individual might become more efficient by translating or rotating some part of its structure relative to other part, this change would require many precisely coordinated mutations (perhaps tens of them) to happen in sequence, preferably in a short amount of evolutionary time.

Therefore, a new kind of mutation is needed, one able to make many small, coordinated changes to several vertices in the elastic network (from a few ones to all of them). To fulfill these requirements, a physics-based mutation has been developed (Figure 3.2). As each network is a spatial configuration of vertices connected by relaxed elastic links, a mutation consists of changing the natural length of one or several elastic links, each one by an independent, random amount. The perturbation of the natural lengths introduce potential energy in the elastic network. If it is allowed to relax, the relative positions of many vertices will change coordinately (just as originally intended) to relieve the stress. After the relaxation process, the network is validated in order to ensure that all restrictions described in Section 3.2.1 still apply to the chain of vertices. If it is so, new elastic links are calculated for the new spatial configuration (that is to say, the distance between each pair of vertices is calculated for the new spatial configuration, and elastic links are instantiated if the vertices are near enough.

3.2.3 Evaluation of molecular motor templates

In this section, the deterministic mapping from elastic networks to molecular motor templates, and the evaluation of these templates, are described.

3.2.3.1 From elastic networks to templates

As the elastic networks are intended to represent molecular motors, it is necessary to define their catalytic cores and docking sites.

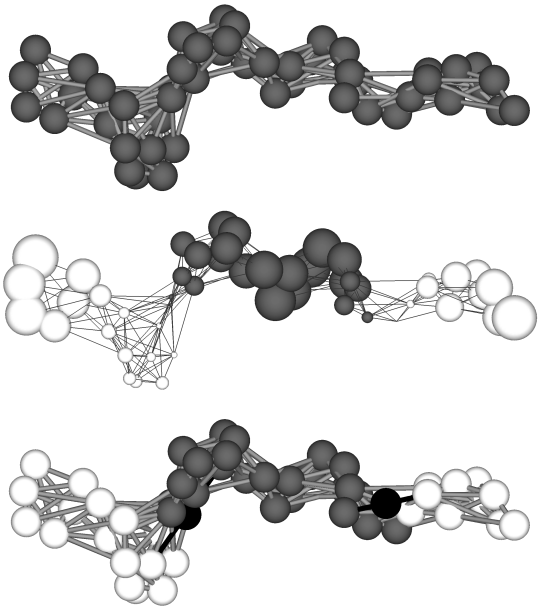


Figure 3.3: ***From structure to molecular motor template.*** **A structure (above) is processed to determine its proposed catalytic cores and docking sites. The normal mode associated to the third eigenvector X_3 of its Kirchhoff matrix is shown (middle). Each vertex v_i is associated to a component x_i of X_3, whose magnitude (size) and sign (white positive, gray negative) conveys information about the vibration of the vertex v_i in that normal mode. This normal mode splits the structure into three clusters, characterized as connected substructures whose vertices have components of the same sign. The resulting molecular motor template (below) has two motor heads, each one composed of a catalytic core (ATP and binding links shown in black) placed between a distal cluster and the central one, and a docking site (white) composed of the vertices in the distal cluster. The exact location of the catalytic core is heuristically determined to maximize the effectiveness of the conformational changes brought about by ATP ligands.**

The catalytic core (or cores) comprises the parts of an enzyme carrying out the chemical reaction mediated by it. More specifically, it is the place where the substrate (ligand) of the enzyme adheres by intermolecular forces and steric effects, and subsequently undergoes a chemical reaction. Molecular motors traveling along filaments also have well defined docking sites, the places where they attach to the filament. Each catalytic core, together with its associated docking site (consisting of a set of labeled vertices) is termed a *motor head*.

Catalytic cores are characterized by a precise spatial arrangement of residues, cooperating to stabilize the ligand and to catalyze the chemical reaction. The chemical reaction is often preceded, accompanied and/or followed by conformational changes, which in many cases can be described in terms of deformations, rotations and/or translations of some subunits of the enzyme, relative to other subunits.

The catalytic core is often located in or near hinge sites, i.e., the sites experiencing minimal translation, rotation or distortion during a conformational change. For many enzymes, GNM can explain most of their conformational changes as combinations of a few slow normal modes. Also, the hinge sites (and so the probable location of catalytic cores) can often be explained by GNM as co-located with the local minima (in vibrational amplitude) of a slow normal mode [222].

For motor proteins, the catalytic core is the pocket (or pockets) where an ATP molecule binds to the protein. The binding and/or the hydrolysis of the ATP molecule, possibly in conjunction with other circumstances, triggers a conformational change in the protein. This conformational change, in turn, performs mechanical work.

The scope of this work is restricted to the structural level: it is not concerned with the larger (and far more difficult) question of designing or modeling catalytic cores *ab initio*, but just with the geometry of the molecule. Thus, in this work, a catalytic core is just a site where ATP is placed, plus the connections (elastic links) between the bound ATP and some of the nearby residues, just like in [197].

This method is inspired by GNM-based protein domain decomposition [110]: given that elastic networks are used, it is quite natural to use GNM theory to predict the location of the core or cores. Specifically, the third eigenvector X_3 of the Kirchhoff matrix is used. Each component x_i of X_3 is associated with a vertex v_i in the structure. For most elastic networks with an elongated shape, the normal mode associated with X_3 splits them in three clusters. Each cluster is characterized as a connected subnetwork whose vertices v_i have associated components x_i of the same sign, while the magnitude of the component x_i correlates with the vibrational amplitude of the vertex v_i in that normal mode (see Figure 3.3, middle). There are two interfaces (hinges) between the clusters, such that two of the clusters are distal (they share just one interface) while the other one is central (it shares two interfaces, each one with one distal cluster).

As predicted by GNM models of actual proteins, the sites where ATP is placed in the artificial elastic networks are near hinge sites, as in [222], which can be identified as the interface between two clusters. In this way, two catalytic cores are defined for each elastic network.

Each catalytic core is defined as a pair of vertices (v_i, v_j) at the interface, each one from a different cluster. When an ATP molecule binds to the core, it is placed exactly in the middle of the two vertices, connected by an elastic link to each vertex in the pair.

These links are stretched to model the change in potential energy brought by the ATP molecule. If $\mathbf{P}_i$ and $\mathbf{P}_j$ are the positions of the vertices v_i and v_j, $\mathbf{P}_j - \mathbf{P}_i$ is the vector from $\mathbf{P}_i$ to $\mathbf{P}_j$, and the ATP molecule is placed at a distance $q_{ij} = 0.5 \cdot \|\mathbf{P}_j - \mathbf{P}_i\|$ of each one of the two vertices of the catalytic core (see Section 3.2.3.2 for further use of the value q_{ij}).

However, in each interface there are many possible pairs, in general. In each interface, the pair of vertices (v_i, v_j) selected as catalytic core is determined by the following heuristics:

- The vertices should not be too close (to avoid pairs of vertices already connected by an elastic link) nor too distant (to avoid unrealistically large elastic links with the catalytic core). The allowed range is $M_{max} = 10 \leq \|\mathbf{P}_j - \mathbf{P}_i\| \leq 11.4 = 3 \cdot D$. If no pair fulfills these requisites, no motor template can be defined from the elastic network.

- The power stroke (as described in Section 3.2.3.2) of the motor head should maximize the movement along the direction defined by the filament. Heuristically, this criterion can be expressed as finding a pair whose vector $\mathbf{P}_j - \mathbf{P}_i$ is both, perpendicular to, and coplanar with the direction of the filament. Additionally, for the motor head to be efficient, the pair should not be too close to the filament. These three heuristics are elaborated in the following points.

- The centers of mass ($\mathbf{C}_1$ and $\mathbf{C}_2$) of each distal cluster is calculated. The line defined by the vector $\mathbf{C}_2 - \mathbf{C}_1$ is taken as the approximate direction of the filament (see Section 3.2.3.3). To measure the orthogonality of $\mathbf{C}_2 - \mathbf{C}_1$ and $\mathbf{P}_j - \mathbf{P}_i$, the absolute cosine of the angle between them is calculated:

$$u_{ij} = \frac{|(\mathbf{P}_j - \mathbf{P}_i) \cdot (\mathbf{C}_2 - \mathbf{C}_1)|}{\|\mathbf{P}_j - \mathbf{P}_i\| \cdot \|\mathbf{C}_2 - \mathbf{C}_1\|}$$

- Let v_i be the closest vertex in the pair (v_i, v_j) to the line defined by the vector $\mathbf{C}_2 - \mathbf{C}_1$, and let $\mathbf{C}_3$ be a point in the line defined by $\mathbf{C}_2 - \mathbf{C}_1$, such that the vector $\mathbf{P}_i - \mathbf{C}_3$ is perpendicular to $\mathbf{C}_2 - \mathbf{C}_1$. Let $\mathbf{Z}$ be a vector defined as $\mathbf{Z} = (\mathbf{P}_i - \mathbf{C}_3) \times (\mathbf{P}_j - \mathbf{P}_i)$. Then, the more coplanar $\mathbf{C}_2 - \mathbf{C}_1$ and $\mathbf{P}_j - \mathbf{P}_i$ are, the more perpendicular $\mathbf{C}_2 - \mathbf{C}_1$ and $\mathbf{Z}$ are. To measure the orthogonality of these

vectors, the absolute cosine of the angle between them is calculated:

$$v_{ij} = \frac{|\mathbf{Z} \cdot (\mathbf{C}_2 - \mathbf{C}_1)|}{\|\mathbf{Z}\| \cdot \|\mathbf{C}_2 - \mathbf{C}_1\|}$$

- Also, if the length of the vector $\mathbf{P}_i - \mathbf{C}_3$ is too short, the motor head will probably be inefficient, as the catalytic core will be too near the filament to produce an efficient power stroke. Heuristically, the value $D \cdot 0.05 = 3.8 \cdot 0.05 = 0.19$ is defined as a too short distance. A sigmoid function (whose value will be near 1 if the distance is not short enough) is used as the heuristic for this issue:

$$w_{ij} = 1 - \left(1 + e^{-500 \cdot \left(\frac{\|\mathbf{P}_i - \mathbf{C}_3\|}{3.8} - 0.05\right)}\right)^{-1}$$

- The selected pair (v_i, v_j) will be the one with minimal $u_{ij} + v_{ij} + w_{ij}$.

For each interface, a pair of vertices is selected as catalytic core, and the associated docking site is defined as the vertices of the nearest distal cluster (see Figure 3.3). Thus, two motor heads are defined for each elastic network.

3.2.3.2 Working cycle and interaction with the filament

The walking behavior of all known instances of molecular motors using filaments can be described as a periodic sequence of steps, comprising some combination of: attachment and detachment from the filament, switching between two or more well-defined shapes, one or more phases of advancement over the filament, binding of an ATP molecule, hydrolyzation of ATP to ADP and a P_i group, and release of the ADP and the P_i group. This is usually called the *working cycle* of the molecular motor. From a computational point of view, a working cycle can be modeled as a state machine, that is to say, a description of the different states of the motor. In fact, kinesin motion has been simulated in this way [215, 216]. As the details of the working cycles vary a great deal between different molecular motors (for example, between myosin and kinesin), different state machines are needed for each one.

In this model, since the evolution of the working cycle is not discussed, a fixed, already functional state machine will be used for all the molecular motor templates. The details of this state machine are loosely based on the working cycles of myosin

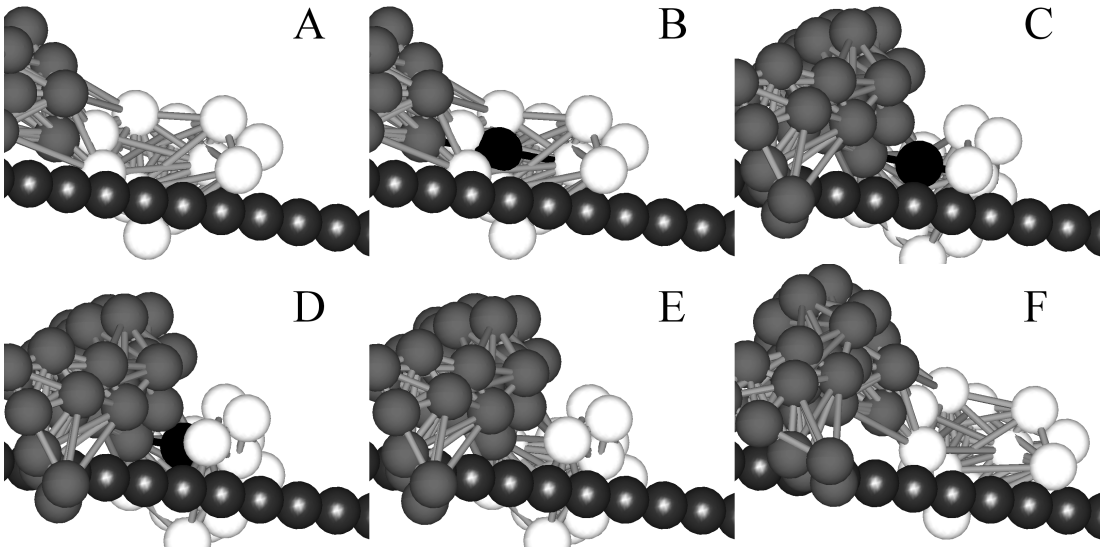


Figure 3.4: ***Working cycle of a molecular motor template.*** **To show it clearly, the motor has just one motor head: a catalytic core (ATP and binding links in black) and the corresponding docking site (white). The filament is in shiny dark gray. In subfigure (*A*), the motor head is in the *sticky* state: there is no ligand and the vertices of the docking site are almost touching the filament (shiny gray spheres). In (*B*), as the docking site is contacting the filament, the motor head transitions to the *bound* state: an ATP molecule attaches to the catalytic core, the docking site gets fixed to the filament, and the conformational change begins. In (*C*), after a fixed amount of time T_b, the conformational change is assumed to be done, ATP is hydrolyzed to ADP, and the motor head transitions to the *nonsticky* state. In this state, the vertices of the docking site are no longer fixed to the filament. In (*D*), the docking site has drifted so no vertex touches the filament; this event induces the catalytic core to have low affinity for ADP, severing its elastic links, and triggering a transition to the *relaxing* state (*E*). In (*F*), after a fixed amount of time T_r (see Section 3.2.3.3 for the values of T_b and T_r), the conformational change is assumed to be finished, and the motor head transitions to the *sticky* state, returning to (*A*).**

and kinesin, as this work is not concerned with the accurate simulation of any concrete instance of molecular motor. The proposed state machine describes how the catalytic core is coordinated with the docking site. For any motor template, each motor head (a catalytic core with its associated docking site) has a separated state machine. For each state in the machine, the characteristics of the catalytic core, the docking site and the transition to the next state are described (see also Figure 3.5 for a summary):

1. *Sticky* state: The docking site is detached from the filament, and the catalytic core is without ligand (Figure 3.4.a). This state ends when any vertex of the docking site touches the filament: then, the vertex is fixed to the filament. After this event, the catalytic core presents high affinity for ATP. In all the simulations, ATP is assumed to be present in saturated concentration, so a molecule of ATP,

represented as a vertex, gets immediately placed in the catalytic core, in the middle of the two vertices of the core, bound to them with elastic links (see Section 3.2.3.1 and Figure 3.4.b). To model the change in potential energy brought by the binding, the links are stretched by giving them a natural length of $0.5 \cdot q_{ij}$, where q_{ij} is the ideal distance from the ATP molecule to each vertex of the catalytic core, as calculated in Section 3.2.3.1). Then, the motor head transitions to the next state.

2. *Bound* state: The stretched links introduced in the transition to this state induce a conformational change (Figure 3.4.c), while the docking site remains firmly attached to the filament. The result is the power stroke of the motor head. Meanwhile, if any new vertex of the docking site touches the filament, it gets also fixed to it. After a fixed amount of time T_b passes (see Section 3.2.3.3), the ATP is hydrolyzed to ADP, and the motor head transitions to the next state.

3. *Nonsticky* state: The change from ATP to ADP induces a change in the docking site. Some of the vertices of the docking site were fixed to the filament in the previous state, but all of them lose the ability to be fixed in this state (in biochemical terms, the docking site now presents low affinity for the filament). However, the unfixed vertices remain in touch with the filament, as long as no force pushes them away from it. The needed force might come, for example, from the activity of the other motor head, or from residual elastic forces in the motor head. When no vertex from the docking site remains in touch with the filament (Figure 3.4.d), this information changes the state of the catalytic core: the ADP is unbound from the core (in biochemical terms, the core losses affinity for ADP). The elastic links binding the ADP to the catalytic core are severed (Figure 3.4.e). Then, the motor head transitions to the next state.

4. *Relaxing* state: After ADP unbinds, the absence of the severed elastic links triggers another conformational change. In this state, the docking site still presents low affinity for the filament, that is to say, its vertices do not get fixed to the filament even if they become in touch. After a fixed amount of time T_r passes (see Section 3.2.3.3), the vertices of the docking site regain the ability to get fixed to the filament, and the motor head transitions to the initial state (Figure 3.4.f), completing the cycle.

State	*Description*	*Vertices in the docking site get fixed to the filament if they touch it*	*Transition trigger*	*Transition actions*	*Next state*
sticky	Ready to attach to the filament.	yes	Any vertex from the docking site touches the filament.	ATP with stretched elastic links is bound to the catalytic core, prompting a conformational change.	bound
bound	Power stroke (conformational change).	yes	A fixed amount of time T_b passes.	Vertices in the docking site are unfixed from the filament, though they still may touch it. ATP is hydrolyzed to ADP.	nonsticky
nonsticky	Lose contact with the filament.	no	No vertex from the docking site touches the filament.	ADP is unbound from the catalytic core, prompting a conformational change.	relaxing
relaxing	Recovery (conformational change).	no	A fixed amount of time T_r passes.		sticky

Figure 3.5: ***States of the working cycle of a motor head*** **(a catalytic core and the associated docking site). In each row, a state is described. The first two columns display its name and description. the third says, for each state, if vertices from the docking site get fixed to the filament upon touching it. The remaining columns describe the transition to the next state: the condition that triggers it, the actions executed at the transition, and the next state.**

It is worth mentioning that, while actual ligands like ATP and ADP have been named in the description, this working cycle represents a simplification not relying on the actual chemical interaction between phosphorylated nucleotides and actual molecular motors, while still being a reasonable abstraction of a process similar to actual working cycles (for example, nucleotide binding and phosphorylation are known to change affinity for the filament in kinesins and myosins, respectively [169]). It can be replaced by any other cycle adequate to enable processivity in the motor templates.

Conformational changes are induced by the attachment and detachment of vertices with stretched links, as in [197]. Although this mechanism to induce conformational change can be argued to lack justification in biochemical considerations, it fits neatly in the used model, i.e. representing proteins as elastic networks. Besides, this proposal does not pursue an accurate simulation of actual macromolecules, but an abstract and computationally tractable model of molecular motor. Furthermore, evolutionary search techniques are expected to perform well at finding good molecular motor templates even if different mechanisms for conformational changes are implemented.

3.2.3.3 Simulation

Sections 3.2.3.1 and 3.2.3.2 present the way to define a molecular motor template out of an elastic network. To evaluate this motor template, it is simulated in the following setting:

- A straight filament is defined, composed of consecutive, touching spherical beads, each one with a diameter $D = 3.8$ (the same as the vertices of the motor template).
- The motor template is placed over the filament, such that the line defined by the centers of mass of its two docking sites is parallel to the filament, and at least one vertex from each docking site touches the filament.
- One motor head is set to state *sticky*, while the other is set to state *relaxing* (see Section 3.2.3.2). The election of which head is set to which state is arbitrary but deterministic.

Then, the function of the motor template is physically simulated. For each vertex v_i in the elastic network, the equation of motion is

$$\ddot{\mathbf{P}}_i = \mathbf{F}_i^e + \mathbf{F}_i^v + \mathbf{F}_i^f + \mathbf{F}_i^x$$

where:

- $\mathbf{F}_i^e$ is the sum of elastic forces on the vertex v_i:

$$\mathbf{F}_i^e = -\sum_{j=1}^{N} A_{ij} \cdot \frac{\mathbf{P}_j - \mathbf{P}_i}{\|\mathbf{P}_j - \mathbf{P}_i\|} \cdot (n_{ij} - \|\mathbf{P}_j - \mathbf{P}_i\|)$$

  where $A_{ij} = 1$ if there is an elastic link between vertices v_i and v_j, and $A_{ij} = 0$ otherwise; and n_{ij} is the natural length of the elastic link.

- $\mathbf{F}_i^v$ is a linear damp force on the vertex v_i:

$$\mathbf{F}_i^v = -\mu \cdot \dot{\mathbf{P}}_i$$

- $\mathbf{F}_i^f$ is a sum of elastic forces to avoid the crossing of the vertex v_i through the filament:

$$\mathbf{F}_i^f = -\sum_j \frac{\mathbf{G}_j - \mathbf{P}_i}{\|\mathbf{G}_j - \mathbf{P}_i\|} \cdot (D - \|\mathbf{G}_j - \mathbf{P}_i\|)$$

  where the G_j are the positions of the beads in the filament that are too close to the vertex:

$$\|\mathbf{G}_j - \mathbf{P}_i\| \leqslant D = 3.8$$

  (of course, this force is 0 if the vertex is not too close to any bead of the filament).

- If the vertex v_i is fixed to the filament (see Section 3.2.3.2), both $\ddot{\mathbf{P}}_i$ and $\dot{\mathbf{P}}_i$ are set to 0, overruling the integration of the equations of motion.

As implied by the equations of motion, the mass of each vertex and the stiffness constant of each elastic link are set to 1, while the linear damping constant is set to $\mu = 0.1$. The state machine of each motor head is also simulated. The constants T_b and T_r (defined in Section 3.2.3.2) are set to $T_b = 500$ and $T_r = 200$ time units, and the whole system is simulated using a classical fourth-order Runge-Kutta integrator with a time step of 0.1. The simulation runs for 10000 time units, which is enough for the motor template to make 10-15 steps over the filament in optimal conditions. While it might seem that the simulation of just one step is needed to measure the effectiveness of the motor template, this is not enough, since it may malfunction after this initial step (for example, by permanently losing contact with the filament).

The value of T_b is much larger than T_r, following the hypothesis that processivity in some molecular motors like myosin V is favored by spending most of its working cycle bound to the filament [174]. The actual magnitude of the time unit is not relevant here, since this work is not concerned with the exact scale of the motor templates, as stated in Section 3.2.1.

3.2.4 Evolutionary algorithm

Randomly generated elastic networks induce motor templates that usually do no walk at all. Because of this, molecular templates able to walk are generated using the following evolutionary algorithm:

1. An initial set of K randomly generated elastic networks is created, and evaluated, and the fitness of each one is calculated. This is the initial population, and each elastic network is an individual of the population.

2. For T times:

   2.1 Individuals from the previous population are selected, according to their fitness, to generate a new population of K individuals. Generally, in the i^{th} time that this loop is executed, the previous population is named the $(i-1)^{th}$ generation, and the new population the i^{th} generation.

   2.2 Mutate some of them.

2.3 Each mutated individual is evaluated and its fitness calculated.

In all the executed instances of the evolutionary algorithm (each one named an *evolutionary run*), $K = 100$, while T ranges from 100 to 200. The evaluation method for elastic networks consists of creating its associated motor template and simulating it. However, for some elastic networks, the definition of the motor template from the elastic network fails at some point (see Section 3.2.3.1):

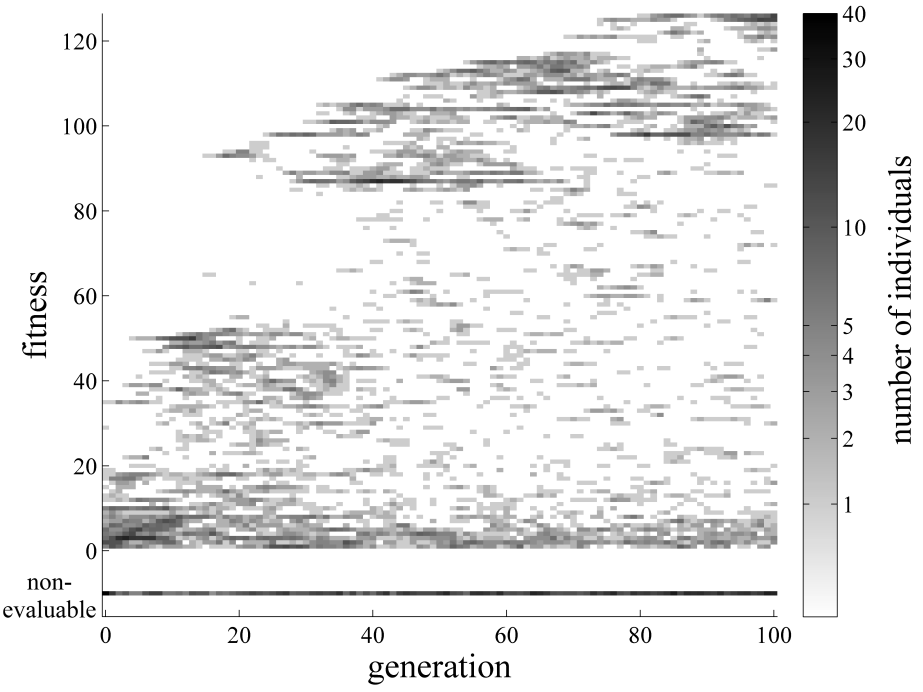
- For some networks, the third eigenvector X_3 does not split the vertices in three sequential and clearly delimited clusters.
- For other networks, no suitable pair of vertices can be found to represent a catalytic core.

In any of these cases, the elastic networks cannot be evaluated: they are discarded from the evolutionary process. They are classified as *nonevaluable*.

The most straightforward way to measure the fitness of an evaluated motor template is to measure the displacement d of its center of mass along the direction of the filament during the simulation. However, motor templates from randomly generated elastic networks present very low to low displacements almost always (see Figure 3.6), so this measure does not discriminate for templates that potentially might become good walkers after some mutations. This can be solved by taking into account the number of working cycles completed by each motor head of the template, measuring it as the number of times n_i that the motor head i transitions to the *bound* state. In this way, templates whose geometrical configuration is good for processivity have an advantage. Finally, the fitness is measured as: $f = d + n_1 + n_2$.

The procedure to select individuals from an existing population A to generate a new population B is the following:

- Let A' be the population A without the nonevaluable elastic networks.
- Let m and M be the minimal and maximal fitness in A', and f_i the fitness for the individual A'_i. Each A'_i is given a rank $r_i = 1 + p\left(\frac{f_i - m}{M - m} - 1\right)$, where p is the selective pressure. Effectively, fitness values f_i in the range $[m \dots M]$ are mapped to ranks r_i in the range $[1 - p \dots 1]$.



.

Figure 3.6: ***Evolution of fitness in one evolutionary run.*** **This is a bidimensional histogram showing the fitness for each individual evaluated in an evolutionary run (one instance of the evolutionary algorithm described in Section 3.2.4). The X axis represents evolutionary time (*generations*), while the Y axis represents the fitness. The color at each point with coordinates (i, f) represents the number of individuals at the i^{th} generation which have a fitness f (see Section 3.2.4 for details). As f cannot be calculated for nonevaluable individuals (see Sections 3.2.3.1 and 3.2.3.3 for an explanation), they are binned separately. A quick fitness improvement can be observed in the first generations, followed by a gradual refinement as the evolution progresses.**

- For each individual B_j to be in population B, a random number $d_{ij} \in [0 \dots 1]$ is generated for each individual A'_i in A'. Then, B_j is defined as the A'_i with maximal $d_{ij} r_i$: $B_j := A'_k \mid k = \underset{x}{\operatorname{argmax}} (d_{xj} r_x)$.

Therefore, the competitive advantage of an individual with good fitness over other with a bad one is modulated by the selective pressure. A relatively high selective pressure $p = 0.01$ is used.

After being selected, individuals from a population B may be mutated. For each individual B_i, a sample s_i is drawn from a Poisson distribution with $\lambda = 0.5$. If $s_i = 0$, the individual is not mutated. Otherwise, the individual is mutated by the procedure described in Section 3.2.2, and s_i is the number of elastic links e_j whose natural lengths n_j will be perturbed. The elastic links to be perturbed are selected at random, and

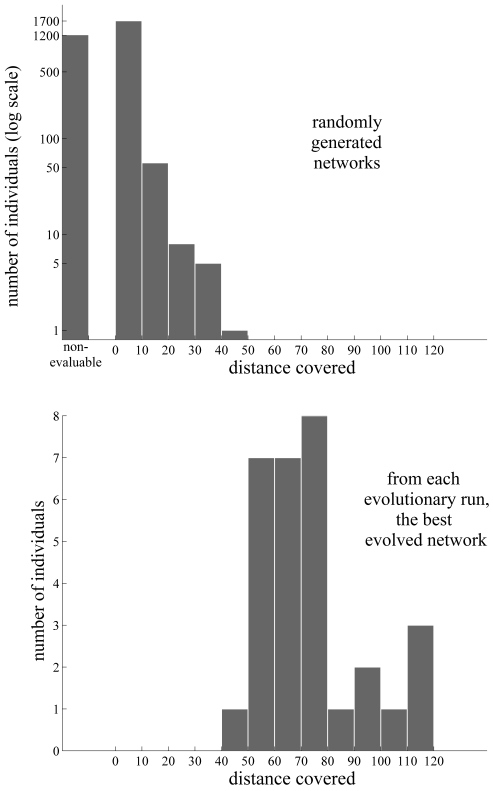


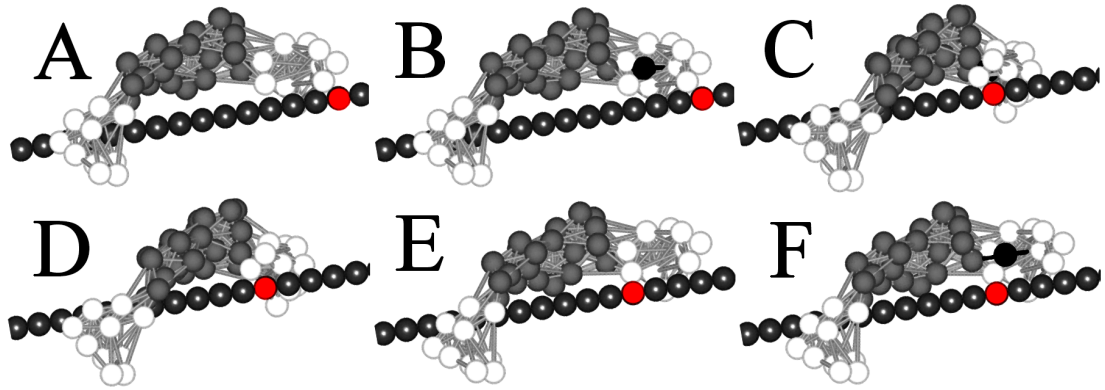
Figure 3.7: ***Comparison of results from randomly generated templates and evolved ones.*** **30 evolutionary runs have been executed. The initial population for each one consists of 100 randomly generated elastic networks (the generation procedure is given in Section 3.2.1), so 3000 random networks have been generated. The results of the simulation of the motor templates associated to these networks is shown in the first histogram (above, Y axis in logarithmic scale): a significant fraction of them ($\sim$ 1200) were nonevaluable, that is to say, no motor template could be generated for them (see Sections 3.2.3.1 and 3.2.3.3 for details). Most of the rest ($\sim$ 1700) were not able to cover a distance d (that is to say, to move along the filament) longer than $d = 10$. In the second histogram (below, Y axis in linear scale), the covered distance of the best individual in each evolutionary run is shown. The improvement over randomly generated networks is significant. The scale in the Y axis is not the same for both histograms.**

their perturbed natural lengths are $n_j^{\cdot} = n_j c_j$, where each factor c_j is a random number drawn from an uniform distribution with interval $[0.5 \ldots 1.5]$.

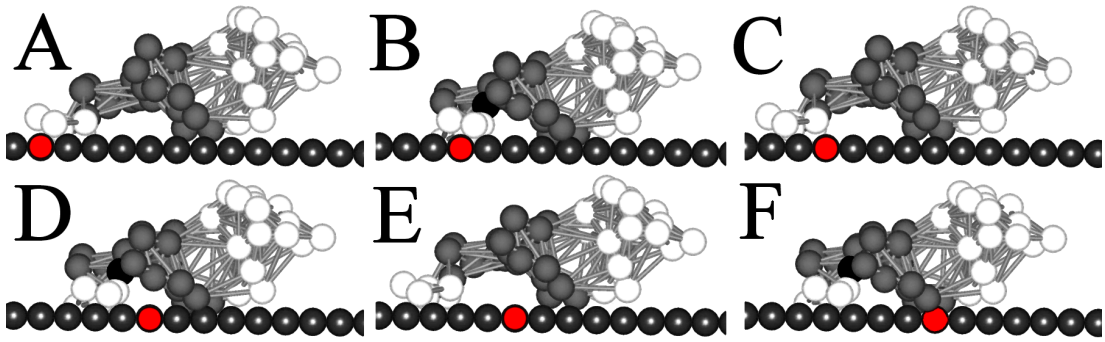
3.3 Generation of molecular motor templates by evolutionary methods

To measure the effectiveness of the evolutionary algorithm, it was performed 30 times; each one of these times is named an evolutionary run. For each run, 100 random elastic networks have been generated for the corresponding initial population, 3000 in total. Almost all of them are able to walk no farther than a distance $d = 20$ or are nonevaluable (see Section 3.2.4 for a definition). However, taking as the result of each evolutionary run the distance walked by the best individual from it, significantly improved individuals (when compared to the randomly generated ones, see Figure 3.7) can evolve. The profile of one of the evolutionary runs is shown in Figure 3.6.

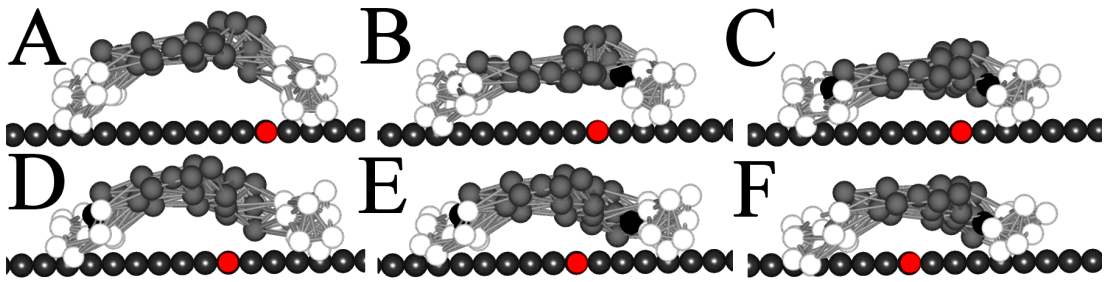
In many cases, relatively minor modifications to the elastic networks trigger



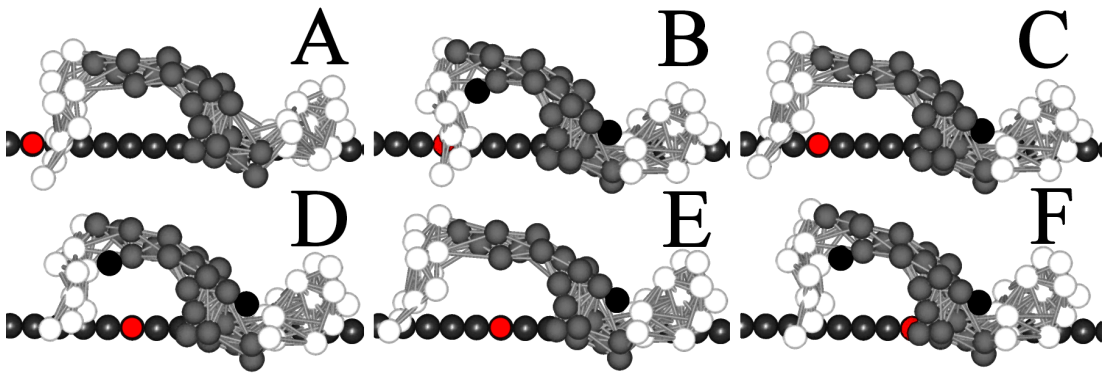
(a)



(b)

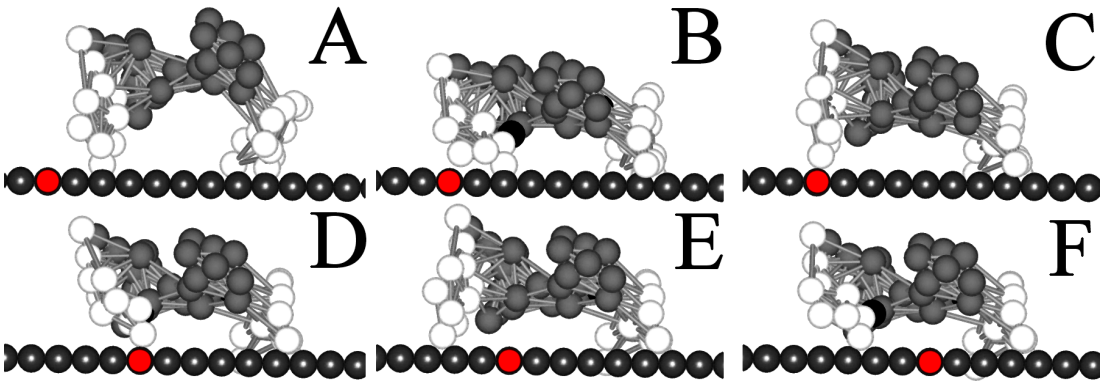


(c)

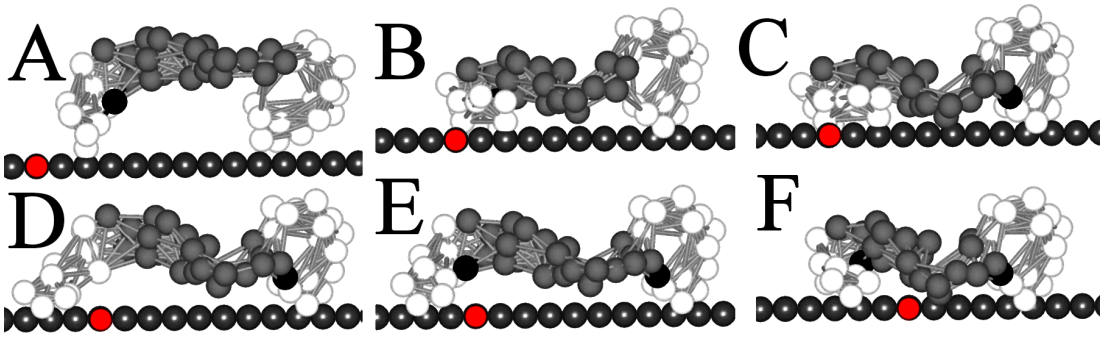


(d)

Figure 3.8: ***Six molecular motor templates.*** **This figure spans several pages.** **See continuation in page 61.**



(e)



(f)

Figure 3.8: ***(Cont. from page 60) Six molecular motor templates.*** **Six examples of the evolved templates. Each subfigure shows the gait pattern of a template with six sequential snapshots, a vertex in the filament being marked in red to provide a point of reference. The first and second ones (a, b) work using just one motor head (initially both heads are touching the filament, but after one step the template settles in a configuration with just one working motor head, the other not being able to touch the filament). The third and fourth ones (c, d) use both motor heads, but only the leading head moves the template forward, while the rear head is auxiliary: it secures the template to the filament and induces a coordinated conformational change in the template to adequately reorient it for the next power stroke of the leading motor head. The fifth and sixth ones (e, f) also use both motor heads, but the rear head does not induce any relevant conformational change; it is used just for attachment to the filament.**

significant increases in the distance covered by the corresponding motor templates, hinting that good templates need to be precisely tuned to the working cycle and to the details of the simulation.

Several walking strategies have evolved, associated to a diverse collection of shapes. Several examples are provided (see Figure 3.8). In some instances, the evolutionary algorithm has yielded templates using just one catalytic core and one docking site to move, as some forms of myosin [118], dynein [83] and kinesin [149]. It is worth pointing out that templates that use both motor heads do not use them symmetrically, i.e., in

most instances, one motor head is responsible for the motion, and the other is used to secure the template to the filament and/or to maintain a correct orientation. Some of the evolved molecular motor templates present varying degrees of rotation around the filament as they traveled along it, mainly because they are asymmetric. However, it should be noted that some studies have also reported rotation in actual molecular motors [2]. Several examples are presented:

- two examples of templates using just the leading motor head (Figure 3.8.a and b): after a first power stroke, the rear motor head loses contact with the filament, effectively becoming useless. The first one (a) presents a marked rotation around the filament as it walks over it, while the second one (b) uses a comparatively small functional limb, while the rest of its body is a wide blob which can be deemed as cargo.

- two examples of templates using both motor heads for conformational changes (Figure 3.8.c and d). In both cases, the leading motor head does the power stroke which moves forward the template, while the rear motor head attaches itself to the filament inducing a conformational change in the template.

- two examples of templates using both motor heads, but only one of them induces a conformational change (Figure 3.8.e and f). These templates are almost functionally equivalent to the ones which use just one motor head, but the leading motor head pushes the template against the filament, inducing the rear motor head to remain attached to the filament, even if no effective conformational change is generated by it because of the geometry of the template.

Several instances of motor templates walking a relatively large distance have emerged, whose motor heads complete their working cycles in a coordinated fashion. However, it must be noted that in all these cases, they move in an inchworm mode (i.e., one head was always ahead of the other one). As most instances of actual motor proteins with two motor heads have been shown (or are strongly suspected) to move in a hand-over-hand fashion [68, 209, 225], this discrepancy must be taken into account when analyzing the results.

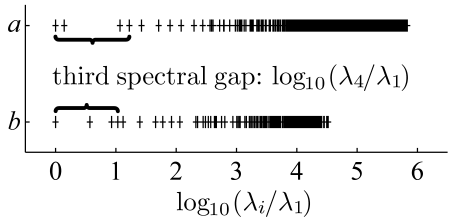


Figure 3.9: *Spectral gaps.* Spectra for two elastic networks, myosin as shown in Figure 3.1 (row *a*), and the elastic network shown in Figure 3.3 (row *b*). The n^{th}-spectral gap is the normalized distance in logarithmic scale between the first (λ_1) and the $(n+1)^{th}$ (λ_{n+1}) normal modes of the structure in ANM. The third spectral gap for myosin (1.224) and the other structure (1.033) are highlighted. Actual molecular motors and most molecular motor templates generated for this work usually have wide spectral gaps.

3.3.1 Computational testing of the *spectral gap* hypothesis

The methods presented here enable a new computational way to test hypotheses about molecular motors. To provide an example, the *spectral gap* question raised by Togashi and Mikhailov [197] can be considered. In that work, two motor proteins were studied: myosin and F_1-ATPase. Taking the ANM models of these proteins, the nonzero eigenvalues (squared frequencies) $\lambda_1, \ldots, \lambda_n$ of their normal modes were studied in normalized logarithmic scale (the numerical cutoff to consider an eigenvalue as 0 was set to 10^{-12}). They were found to feature what was called a spectral gap: the slowest normal modes (from λ_1 to λ_3) were much slower than all the other modes, thus producing a noticeable gap in the spectral signatures of the elastic networks (see Figure 3.9). This gap was argued by Togashi *et al* to significantly influence their structural properties, rendering them robust against external perturbations. As random elastic networks were generated and their spectral signature and structural properties examined, most of them were found to lack both a wide spectral gap and the same robustness as myosin and F_1-ATPase. Therefore, a wide spectral gap was hypothesized to be a salient feature of motor proteins.

Here, this hypothesis has been checked by examining the presence or absence of a wide spectral gap in molecular motor templates. To be rigorous, the n^{th} spectral gap is defined as the distance between the first and the $(n+1)^{th}$ normal modes in normalized logarithmic scale. Both myosin and F_1-ATPase present a wide third spectral gap (~ 0.5 and ~ 1.2, respectively), so it will be used here.

In Figure 3.10, data from the 30 evolutionary runs are lumped together in two curves, measuring the distribution of the third spectral gaps of elastic networks: one

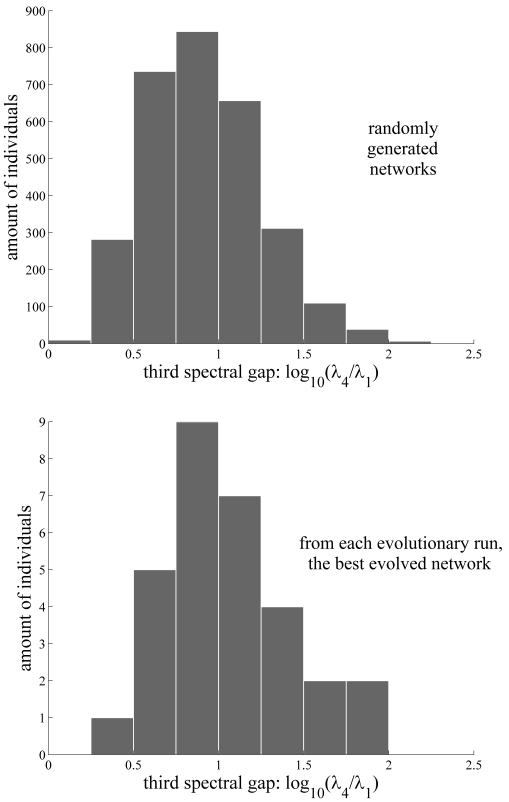


Figure 3.10: ***Spectral gap distributions for random and evolved individuals.*** **Distributions of the third spectral gap (see Figure 3.9) for randomly created and for evolved individuals in the 30 evolutionary runs from Figure 3.7. The third spectral gap is the normalized distance in logarithmic scale between the nonzero eigenvalues (squared frequencies) λ_1 and λ_4 of the first and the fourth normal modes of the structure in ANM. It is a good implementation of the concept of spectral gap to be used with molecular motor templates, as the first three normal modes have usually frequencies in the same order of magnitude. It is also a good measure of the spectral gap for the motor proteins analyzed in [197].**
Above. **Distribution of spectral gaps for 3000 randomly generated elastic networks. Both small and large spectral gaps are very rare.**
Below. **Distribution of spectral gaps for the best individual from each evolutionary run (30 in total). While the interpretation must be cautious because of the small sample size of the distribution, it is similar to the distribution for randomly generated networks, though skewed towards larger gaps.**
The scale in the Y axis is not the same for both histograms.

distribution for randomly created ones, the other for the best individuals from each evolutionary run. From both histograms it is apparent that neither very small nor very large spectral gaps are frequent, neither in randomly generated examples nor in the result of evolutionary processes. This represents an inherent bias for moderately large gaps in randomly generated elastic networks. On the other hand, the distribution for evolved elastic networks seems to be biased towards larger gaps than the distribution for random elastic networks, though the data must be interpreted with care, as the sample size of the former is very small in comparison with the latter.

Additional evolutionary runs have been performed, with the elastic networks constrained to have either short or large spectral gaps. Runs with very large spectral gaps have evolved molecular motor templates as easily as unconstrained runs. The required large spectral gap usually means that structures have a hinge in a nearly degenerate configuration (i.e., a joint almost having a degree of freedom), thus being

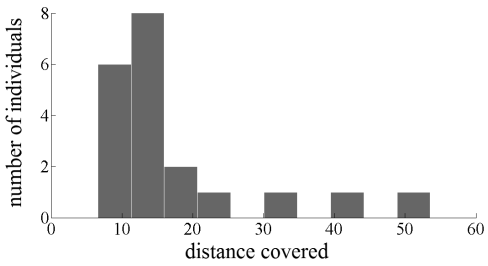


Figure 3.11: ***Performance of evolved individuals with very low spectral gaps.*** **20 evolutionary runs were performed, with the templates restricted to have an eighth spectral gap lower than 0.5. This histogram displays the distance covered by the best individual from each evolutionary run. Just one individual was able to cover a distance d longer than $d = 50$. Compare with the second histogram in Figure 3.7: most individuals evolved without this restriction are able to cover a significantly longer distance.**

able to bend easily in some direction, but that does not affect their performance as molecular motor templates.

However, evolutionary runs requiring elastic networks to have extremely low spectral gaps (an eighth spectral gap lower than 0.5, hence an even lower third spectral gap) have performed noticeably worse (see Figure 3.11), because networks with this constraint are extremely hard to evolve: most of the elastic networks are nonevaluable (see Sections 3.2.4 and 3.2.3.1), because networks with low spectral gap tend to have globular shapes which are not well partitioned by the algorithm presented in Section 3.2.3.1. Although one of these constrained motor templates is able to move a distance comparable to some results from unconstrained evolutionary runs (see the best evolved example in Figure 3.12), it moves in a strange way.

In conclusion, these results shed new light over previous studies [197], confirming that a wide spectral gap is a very important (and very probably necessary) feature of molecular motors. However, elastic networks representing randomly folded peptides present an inherent bias for relatively wide spectral gaps, and this fact might constitute a substantial contribution to explaining wide spectral gaps in actual molecular motors.

3.3.2 Bipedal templates

Many molecular motors function as dimers, that is to say, they are composed of two identical proteins, joined at a specific point, each one having one motor head in the opposite end. This suggests that bipedal templates can also be evolved. In this configuration, the generation of a molecular motor template from an elastic network is slightly different: two instances of the elastic network (one of them mirrored, to facilitate the pairing) are joined by the first vertex in the chain of vertices, and a motor

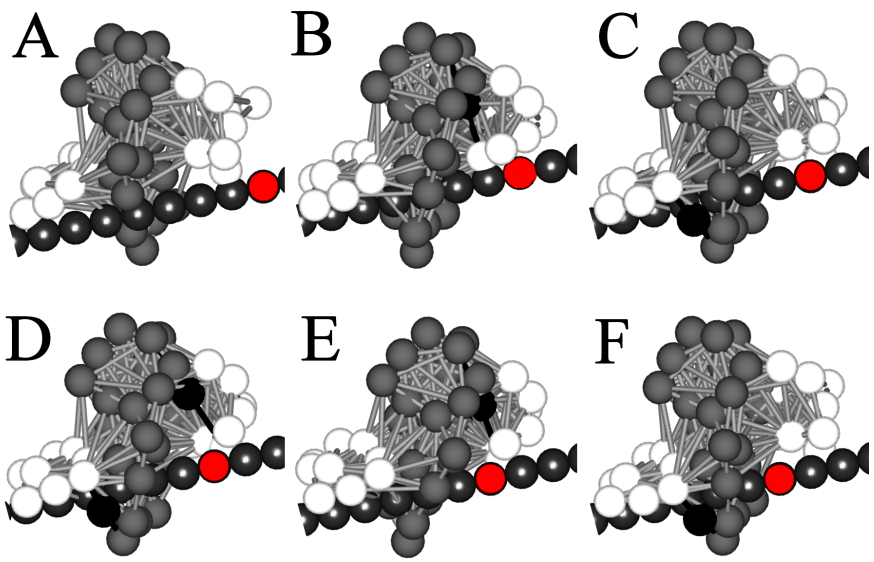
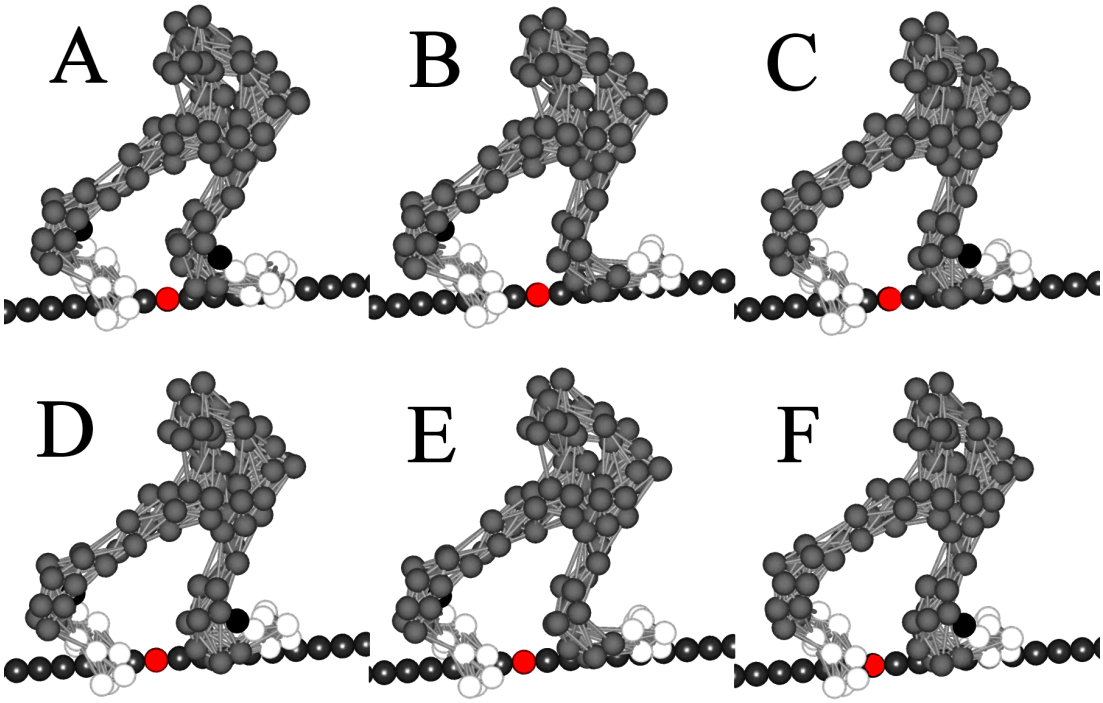


Figure 3.12: ***A molecular motor template with a very small spectral gap.*** **The figure shows the gait pattern of this template with six sequential snapshots. A vertex in the filament is marked in red to provide a point of reference. This structure is the best molecular motor evolved under the conditions described in Figure 3.11, able to cover a distance $d = 53$. It moves in a strange, jerky way, reminiscent of a ratchet.**

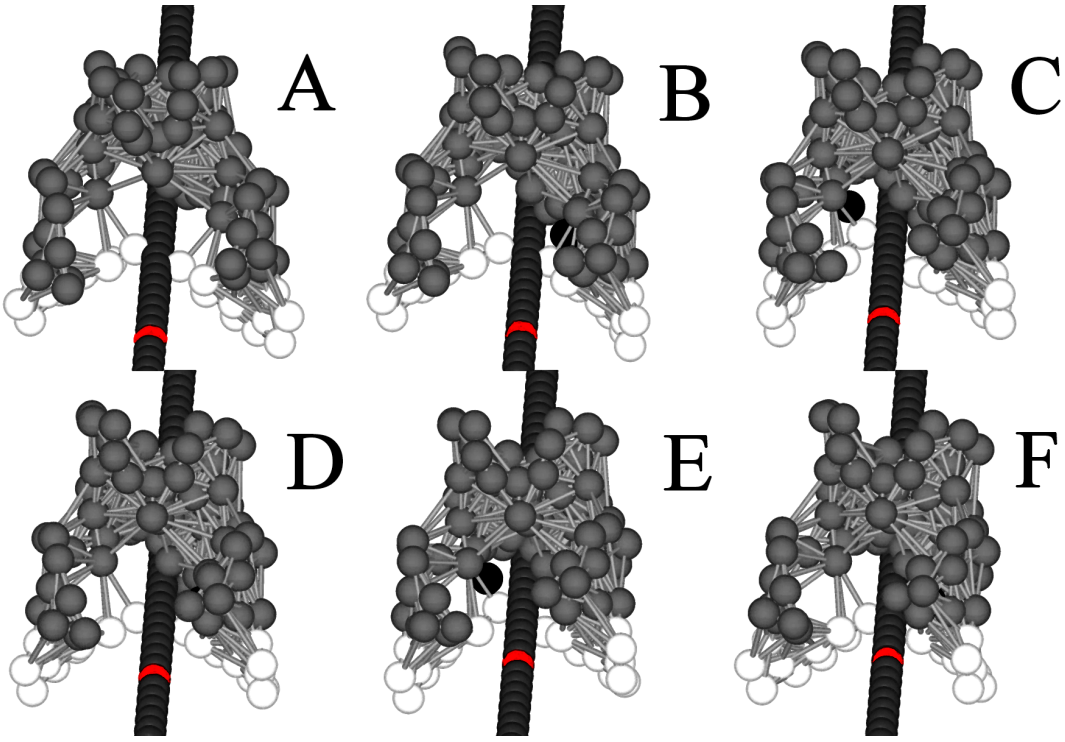
head is defined in each one of the instances (since there might be two motor heads in each instance, the one which is farthest from the joining vertex is selected). In this way, bipedal templates can be evolved. It is important to highlight that these bipedal templates are not proper models of dimer protein motors, since the dimers are composed of identical units, while the two components in the bipedal templates are specular images, like the limbs of a bilateral organism. Still, they represent an interesting experiment in the evolution of different bipedal gaits.

32 evolutionary runs were performed in bipedal configuration. In general, the evolved bipedal templates (see several examples in Figure 3.13) are not as fast as the non-bipedal evolved ones (see Figure 3.14), but they still manage to cover significant distances. It must be noted that, though many different walking strategies were readily evolved, all strategies used an inchworm movement, where one motor head was always ahead of the other one. Because of this, hand-over-hand strategies (which are thought to be the predominant mode of locomotion for dimer protein motors [68, 209, 225] can be deemed as extremely difficult to evolve in this experimental setup.

Notably, the evolved bipedal templates feature a range of shapes and gaits:

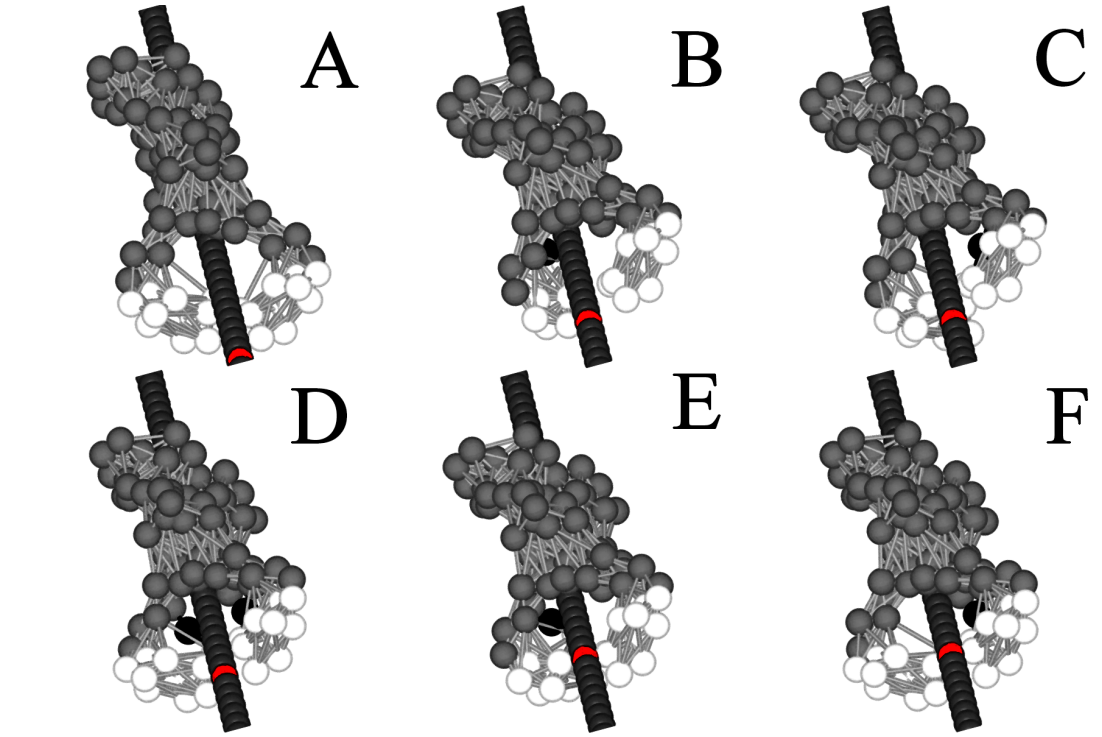


(a)

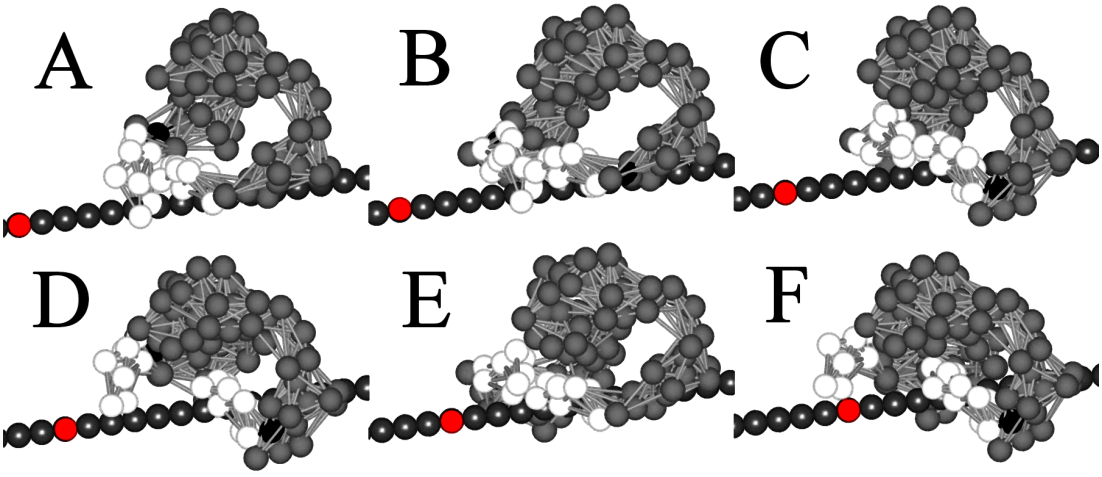


(b)

Figure 3.13: ***Four bipedal templates.*** **This figure spans several pages. See continuation in page 68.**



(c)



(d)

Figure 3.13: *(Cont. from* **page 67***) Four bipedal templates.* **Four examples of the evolved bipedal templates. Each subfigure shows the gait pattern of a template with six sequential snapshots, a vertex in the filament being marked in red to provide a point of reference. The first one (a) resembles two humanlike legs, while the second and third ones (b, c) are more reminiscent of molecular motors with attached cargo. The last one (d) is the fastest.**

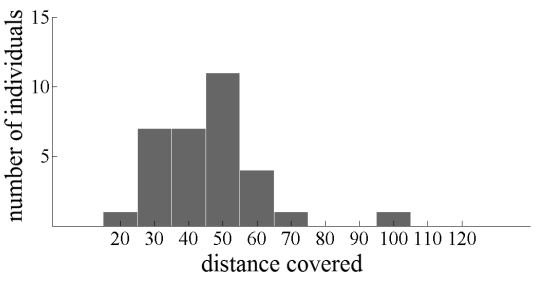


Figure 3.14: ***Performance of evolved bipedal templates.*** **32 evolutionary runs were performed in bipedal configuration. This histogram displays the distance covered by the best individual from each evolutionary run. Compare with the second histogram in Figure 3.7: the results are slightly worse, as the templates must coordinate the movement of two loosely connected parts.**

- walking pseudo-legs (Figure 3.13.a), taking short but secure alternative steps. The example shown here presents the peculiarity that the legs get attached to the filament in different angles, yet they still produce a steady gait.

- slow but well-secured pullers (Figure 3.13.b), with a firm grip over the filament: observe that the limbs grip the filament from below, while they join above it. This example rotates around the filament as it moves through it.

- hopping templates (as in Figure 3.13.c), whose gait consist in thrusting itself with both motor heads in an alternate way, only occasionally having both legs attached to the filament. In the example provided here, most of the length of both limbs are entangled in a single mass, effectively acting as cargo, moved by comparatively small actuating limbs.

- short but fast pulling pseudo-limbs (Figure 3.13.d), which are the fastest bipedal templates evolved in these experiments. This example has the peculiarity that the phase difference between both legs shifts as time passes.

3.4 Conclusions and discussion

As stated in the introduction to this chapter, a diversity of shapes and gait patterns has been generated by a classical (in the sense of evolutionary computation) evolutionary optimization: the individuals do not directly compete against each other but have to perform individually a simple task as best as possible: to move as far as possible. The resulting structures are interpreted as molecular motor templates; however, they can also be interpreted as models of robotic agents suspended in a viscous fluid.

Many aspects of the model are specifically designed to be as simple as possible: the genomic model is extremely simple (just a fixed-width sequence of 3D points),

and the evolutionary algorithm is also very simple, with no crossover and just one mutation operator, which is also fairly simple (in the sense of not using any heuristics or analytical tools to optimize the elastic networks in any way). The model finds viable gait patterns because individuals are canalized in two ways:

- the working cycle (a very simple reactive model) and the specification of the motor heads are hard-wired.
- the mutation operator is based on physical relaxation after the application of perturbations to the structure, so it induces a fitness landscape more correlated to the physical properties of the structure, which play a key role in the configuration of gait patterns.

However, these represent relatively low level features of the model, as they do not constrain in any precise way the gait patterns of the templates. Thus, the diversity of shapes and gait patterns is enabled by these characteristics, and the fact that the individuals compete in a 3D virtual world, making decisive the fine details of their shapes, as subtle differences can enable significant differences in fitness.

The model represents an extreme example on morphological computation [157]: the gait pattern emerges from the interaction between the properties of the motor heads and the physics and geometry of the both the templates and the filament, with no typical control subsystem. While this concept is similar to Lobo's path followers [120], there is an important difference: no indirect encoding (developmental process) has been used, because the problem of the high dimensionality of the parameter space has been solved in other way: the configuration of the motor heads (which can be loosely considered as the control system) is induced by the geometry of the elastic network (the body of the agent), so a coevolution of body and control system emerges from the evolutionary algorithm. Thus, diversity arises from a detailed modeling of the interaction between the individuals and the world were they act, enabling many different solutions to the problem of motion through a filament to appear through the (implicit) coevolution of the body (morphology of the structure) and the control system (the configuration of the motor heads), instead of relying on indirect encoding.

The mutation operator can also be considered as a mode of morphological computation: instead of using heuristics based of the analysis of the characteristics of the structures, they are mutated by perturbing some elastic links and performing a

physical simulation of the resulting relaxation process, which naturally induces many coordinated changes into the structure.

3.4.1 Discussion on molecular modeling

From the point of view of computational molecular modeling, a set of methods to simulate and evolve structural templates of motor proteins have been developed. By using them, research questions about motor proteins can be tackled from a new angle, providing new insights. An example in this regard has been provided: while it is apparent that the shape of an actual molecular motor is exquisitely adapted to the way it moves through the filament, the evolutionary processes driving this adaptation cannot be directly examined, and researchers must instead rely on comparative studies of related motors [109]. However, analyzing the results of the evolutionary runs, some insights can be made about this evolutionary process. Specifically, the results suggest that, while the shape of a molecular motor must be exquisitely tuned to its working cycle, the characteristics of the filament and the interaction with it, the most difficult aspect of the evolution of motor proteins is not dominated by this tuning process, which seems to be relatively straightforward in the evolutionary runs, but by the evolution of a functional working cycle, which has been taken for granted here.

It should be further stressed that in the model presented here structures evolve, but the working cycle (described in Section 3.2.3.2) is fixed, thus restricting the results to searching structures able to walk using this working cycle. However, it is clear that structure and working cycle have coevolved in actual motor proteins. While this observation obviously limits the scope of the results, they are still valuable, because in the coevolution of structure and working cycle, the results suggest that the former one evolves easily, while the latter one is the most important issue. It can be expected that if the working cycle is not hard-wired into the model but emerges from a lower-level simple model of chemistry, the coevolution of the working cycle together with the shape of the template will enable even more diversity of shapes and gait patterns.

The methods presented here are based on several results from elastic network analysis, a mature and well-proved collection of tools to understand biological macromolecules, as it has been stated in the introduction. The definition of the structures studied here is based on ANM, so the presented method shares some characteristics with it, particularly, the long cutoff distance relative to the separation

between vertices in the chain. For practical considerations, the size of the structures has been kept small, with few vertices, so they can be regarded as very rough and coarse-grained models of hypothetical motor proteins. The GNM model provides theoretical justification to the way catalytic cores and docking sites are placed within the structures. The basic outline of the model of interaction between the protein structures and ATP molecules has been inspired by previous studies on conformational changes of elastic models of peptides by Togashi and Mikhailov [197]. A very similar approach by Flechsig and Mikhailov [57] has been developed independently, but it concentrates on modeling just one protein: HCV helicase, an enzyme which moves trough DNA strands unzipping them. In a way, the methods presented here can be regarded as reverse-engineering applications of structural research based on GNM. In this regard, only structural considerations have guided the method, avoiding to tackle extremely complex issues, like the shape, arrangement, coordination and working cycles of catalytic cores and docking sites based on actual biochemical considerations.

From the point of view of molecular modeling, it can be argued that elastic network models, and more precisely NMA techniques, are not valid universal models of molecular structures. They are not valid for some proteins [156], and, specifically, for the motor protein kinesin [198, 228]. The conformational changes of the working cycle of kinesin cannot be accounted for by models of interaction (between ATP and a catalytic core) as simple as the one proposed here, but need to be modeled using other approaches [208]. It is apparent that somewhat important portions of the configuration space of possible molecular motors are unreachable by the method presented here. However, this issue is not as troublesome as it might appear, as the dynamics of many other proteins (among them other molecular machines as myosin [228, 229] and DNA helicase [56]) can be described to varying degrees of accuracy by elastic network models and NMA techniques. Besides, any model taking into account the complex dynamics of kinesin would probably not be as computationally lightweight as the one presented here. And, as evolutionary algorithms require many structures to be evaluated, these performance considerations become very important.

There is another point of possible concern: the structures are not immersed in a thermal bath (simulation of solvent with Brownian fluctuations), which is not negligible at the scale of molecular motors. The justification can be provided by examining the work presented in [36]. In that work, the simulation of the working cycle of a

molecular machine [197] was coupled to a multiparticle collision method [123, 124], a computationally costly hydrodynamic model of thermal bath. The resulting simulations were fairly similar to simulations without the solvent. These results can be interpreted as supporting the view that the dynamics of the working cycle can be described (up to a given level of realism) without taking into account the solvent. Furthermore, when the hydrodynamic solvent model was substituted by a less costly (but also less realistic) simulation of thermal bath, the working cycle became very erratic, as several conservation laws were violated.

Therefore, the lack of a (necessarily very costly) model of thermal bath stems from performance considerations. Of course, a thermal bath simulation might be important for some walking strategies; specifically, it has been argued that the hand-over-hand movement of myosin and kinesin includes biased diffusive steps greatly aided by the random component added by the thermal bath [86, 97], i.e., a Brownian ratchet mechanism. On the other hand, in [57], the molecular motor HCV helicase is simulated with methods very similar to the ones presented here, and also without taking into account Brownian fluctuations.

UNIVERSIDAD
DE MÁLAGA

Chapter 4

Diversity by emergent evolutionary dynamics

In this chapter, it is presented a novel method to generate morphological diversity through an agent-based, simple ecological framework with evolutionary dynamics. The complexity of the modeling of agents is significantly lower than in Chapters 2 and 3; diversity emerges instead by evolutionary interaction between the agents, which compete in an open-ended world instead of just optimizing some pre-arranged fitness function. The framework is analyzed not only from the point of view of morphological diversity; other interesting emerging evolutionary properties are also studied.

In the framework proposed in this chapter, the agents are virtual plants which are characterized by very simple abstractions of genetic, developmental and physiological processes. On the population scale, the heterogeneous spatial structure of the plant community emerges from the evolution of its component plants. As the virtual plants compete for shared resources, the ensuing evolutionary dynamics generate a variety of biological emergent phenomena, from the diversification and complexification of the forms of the plants to the auto-adjustment of the mutational robustness to environmental factors. The results demonstrate that diversity can spontaneously emerge in a community of mutually interacting individuals under the influence of specific environmental conditions.

This chapter is organized as follows. In Section 4.1 the related work for the proposed

The results presented in this chapter have been published in [55], except for the data and the discussion about Figures 4.5, 4.6, 4.7 and 4.8.

model is reviewed. In Section 4.2 the evolutionary model of virtual population is introduced, specifying mechanisms at the genetic level, the laws of interaction with the environment, and the mutation/selection rules. Section 4.3 presents the detailed results of a full set of numerical experiments. Finally, Section 4.4 presents the conclusion and a summary of the properties that emerge from the evolutionary dynamics of the model.

4.1 Introduction and related work

This chapter focuses on analyzing diversification under evolutionary conditions. A virtual community of plants is simulated to study the emergence and dynamics of genomic and phenotypic variation during evolution. *Rewriting systems*, a family of formal methods widely investigated in theoretical computer science and capable of encoding and generating complex structures, are used to model plants. In this formal framework, parts of an initial object, generally strings of symbols, are iteratively replaced by other parts, generally longer string segments, following a given set of rewriting rules. One of the most popular and best studied rewriting systems are *L-systems*. First proposed by Lindenmayer [117] (hence its name), L-systems are a class of formal grammars that proceed by parallel application of the rewrite rules to the current string. Parallel transformation (context-free or context-sensitive) makes L-systems especially suitable for the computational modeling of the development of unicellular colonies and multicellular organisms [92]. In particular, they have been extensively used in plant simulation [160] due to the ease of converting generated strings into 2D or 3D tree-like structures by means of *turtle geometry*, e.g., such as the Logo computer language [131, 159]. With the years, L-systems have found many applications in plant anatomy, physiology and morphogenesis [3, 31, 66, 168]. Remarkably few of these cases, however, have examined plant generation on the *evolutionary* scale, nor do they use evolutionary algorithms. Yet, L-systems provide a rather straightforward model suitable for evolutionary exploration: just as in biological development, where complex organisms can be considered as emerging from the interpretation of one-dimensional strings (genomes), L-systems can encode complex structures through fairly simple sets of rules. Encoding large phenotypes with small genotypes *via* an intermediate developmental stage has been termed *indirect* or *implicit encoding* in the evolutionary computation community [183].

A first attempt at evolving L-systems was proposed by Koza [108]: it used genetic programming and represented the rewrite rules as labeled tree graphs. The goal of the search was to find specific rules that could generate a structure identical to a predefined one. Since then, other works based on evolutionary complex structures encoded by L-systems have been proposed. Jacob [92, 93] presented a variant of genetic programming to evolve context-free and context-sensitive L-systems that could generate plants similar to real ones. The fitness of individuals was based on the volume and number of blooms and leaves of the virtual plant during each iteration of the system. He also experimented with an ecosystem of different coevolving plant species [94]. Ochoa [147] designed a model for generating 2D plant morphologies from D0L-systems (the simplest class of L-systems, where 'D' stands for deterministic and '0' for context-free). It resembles the one which is proposed here (see below), although her work was more focused on exploring the model itself rather than using it to investigate evolutionary dynamics. Another similar, albeit only briefly sketched framework, was proposed by Mock [136]. Ebner *et al.* [43, 45] presented a light seeking 3D plant model based on L-systems. They also observed [44] the *Red Queen Effect* under competitive simulations, i.e., the fitness either remained constant or decreased while evolutionary progress was still going on. However, their simulation evolved to a dominant stable strategy with no further progress. Toussaint [200] defined a very similar model to study the role of neutral mutations in the evolvability of genetic representations, although his model had additional layers of complexity in the genetic representation. Finally, Bornhofen and Lattaud [15, 16] evolved generic virtual plants including their full life cycle. Their simulations ended in ecosystems of tall plants with no phenotypic diversification. In [17], they concluded that the genetic search space of the D0L-system was too limited, constraining evolutionary dynamics too severely and preventing it from evolving a wide variety of plant morphologies.

Here, in contrast to Bornhofen and Lattaud's conclusions, a morphological model based on a D0L-system is analyzed, showing a wide range of concurrent diversification in an open-ended evolutionary process (i.e., evolution without a definite maximum fitness, hence no optimal goal or solution to reach). Regarding the issue of genetic search space, an important aspect when designing evolutionary models of complex structures (and especially so when using an indirect encoding) concerns the proper choice of *genetic operators*. For its part, this work largely relies on *gene duplication*. In biology, gene

duplication (leading to the creation of paralogous genes) is a form of silent mutation in the sense that it is generally neutral with respect to selective pressure. Although in most cases copied genes could be considered junk DNA (non-coding DNA with no direct effect over the phenotype), since they simply seem to increment redundancy, they provide in fact a fertile coding substrate for the advent of new proteins and new functions. In some models, gene duplication has been combined with traditional evolutionary operators, as in [210], which concluded that creating copies of key portions of the genome was useful because it could both retain their function and leave the copies open to incremental improvement. In the same spirit, the model presented here shows the performance of various gene duplication operators combined with usual point mutation operators (alteration, deletion, insertion), and how this can create initially non-disruptive, but later highly productive genetic modifications.

4.2 An evolutionary model of virtual plant population

Following standard practice in evolutionary computation studies, the model is presented in four sections: the genome and development of plant phenotypes (Section 4.2.1), the genetic operators (Section 4.2.2), the environment and associated fitness function (Section 4.2.3), and the evolutionary scheme (Section 4.2.4). The term *individual* will be used sometimes to refer to a virtual plant.

4.2.1 Genome and development

Like other works in virtual botany, a grammar-based indirect encoding has been chosen, defining plant growth as a rewriting process. More precisely, a plant develops according to a deterministic, context-free L-system also called *bracketed D0L-system*. In this type of abstract framework, gene regulation is greatly simplified since all symbols are rewritten in parallel and the only remaining control parameter is the length of the derivation process, i.e., the number of global rewriting stages, denoted n. Since this work is mainly concerned with the structural evolution of genomes, this parameter will be constant for all practical purposes and set to $n = 3$ in all individuals.

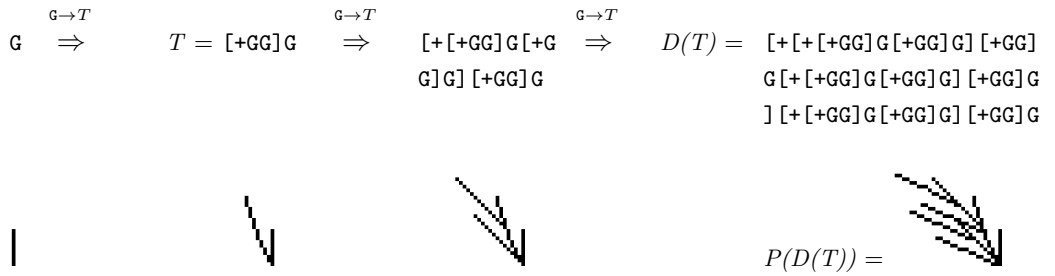


Figure 4.1: ***Genotype-phenotype mapping.*** **Going from a string genotype T to a graphical phenotype $P(D(T))$ via the developed string $D(T)$. (Top) Genotype T, here `[+GG]G`, is interpreted as the right-hand side of the unique rule of a D0L-system. Starting from axiom `G`, the rule $\texttt{G} \to T$ is applied three times to produce a string version of the phenotype, $D(T)$. (Bottom) To obtain the final graphical phenotype $P(D(T))$, this string is then interpreted by a Logo-style turtle routine, where each symbol corresponds to a command: `+` and `-` change the turtle's direction (on the trigonometric circle, counterclockwise for positive angles), `G` moves forward by 10 units (drawing a branch over the pixels it covers) and brackets `[` and `]` respectively push and pop the turtle's stacked state (position *and* direction). Note the repetitive nature of $D(T)$. In this figure, a graphical equivalent is shown for each string in the rewriting process from `G` to $D(T)$, but only the last one corresponds to the phenotype of the plant, $P(D(T))$.**

4.2.1.1 Genotype representation

The D0L-systems used in this work are defined as a triplet ($\{\texttt{G,+,-,[,]}\}$, `G`, $\{\texttt{G} \to T\}$), where there is only one non-terminal symbol, denoted `G`. Accordingly, there is a single production rule $\texttt{G} \to T$, whose right-hand side is a string T representing the *genome* of the plant, since it is the only part of the D0L-system that changes from plant to plant. Genome T is composed of symbols from the alphabet $\{\texttt{G,+,-,[,]}\}$ and includes at least one instance of the symbol `G`, with the additional syntactic restriction that brackets must be balanced. Genetic expression and organism development in the proposed model consist of rewriting the axiom `G` by applying the rule $\texttt{G} \to T$ exactly $n = 3$ times, using the production rule encoded by the genome. In this way, the encoding method is rather straightforward, as first remarked in [147]: since only one rewrite rule applies, its right-hand side codes fully, albeit indirectly, for the plant's structure.

The final string generated after 3 applications of the rule $\texttt{G} \to T$ is referred to as the *developed string*, and it is denoted $D(T)$ to highlight the deterministic nature of the system. The top row of Figure 4.1 illustrates this derivation process. Despite its simplicity compared to most works in L-system evolution, this encoding allows for the definition of robust genetic mutation operators (see Section 4.2.2 below).

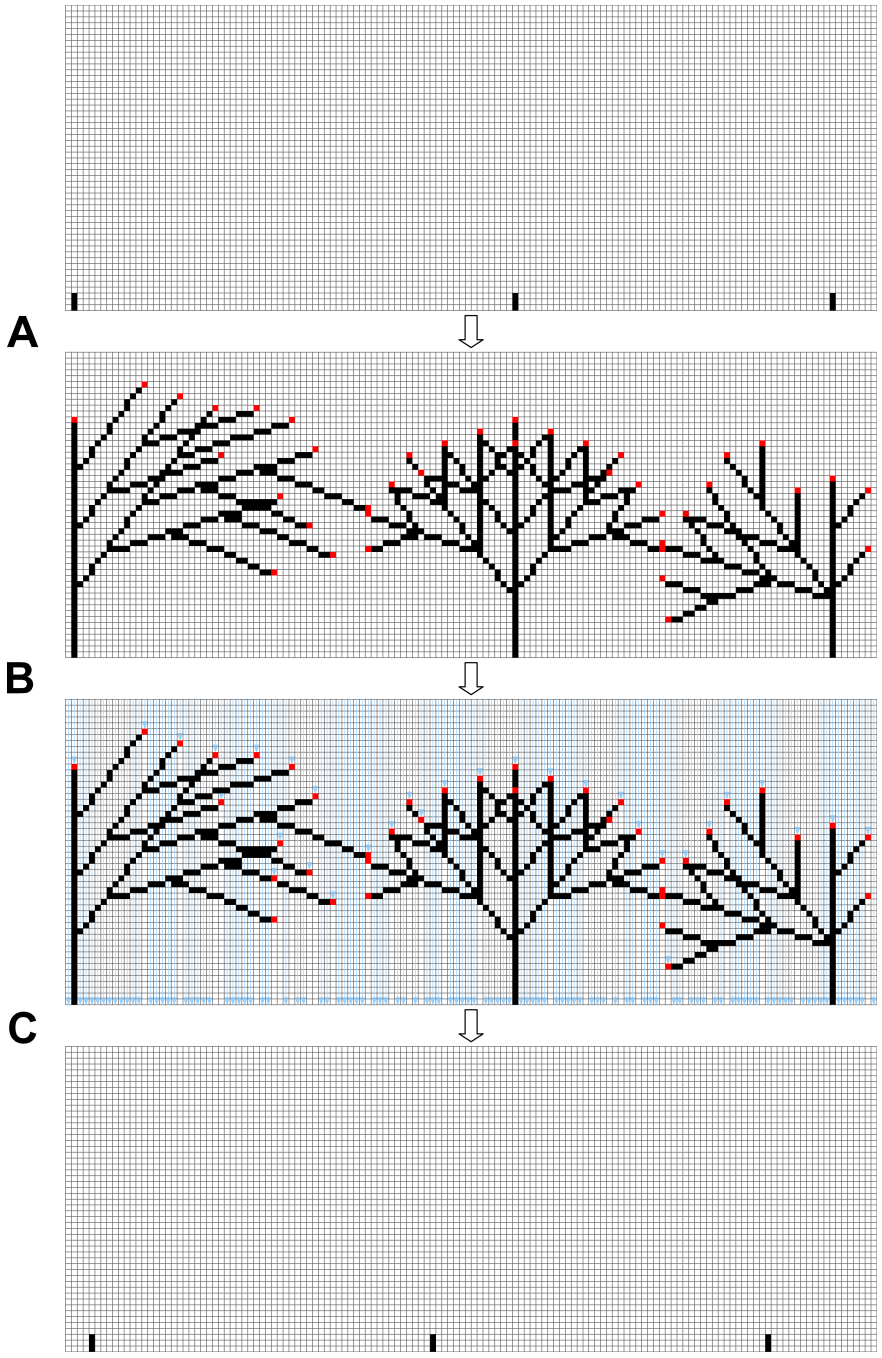


Figure 4.2: ***Simulation cycle of a plant population.*** **(A) Plants' phenotypes grow in the environment from stolons (stem bases) spawned by the previous generation. Branches pixels are colored in black, leaves pixels in red. (B) Once plants are fully grown, they compete for ambient light in the form of vertical beams (here in blue) falling from above on the leaves. (C) The number of descendants of each plant in the next generation is a function of its fitness, calculated according to the amount of light it could capture and its number of branches. Taller plants overshadowing smaller plants are able to absorb light first, hence get a competitive advantage. The genotypes of the simple plants displayed here are, from left to right:** `G[--GG][G]`**,** `G[G][--G][++G]`**, and** `G[G][--G]++`**.**

4.2.1.2 Phenotype development

The phenotype of an individual, denoted $P(D(T))$, is the complete matrix of pixels (bottom of Figure 4.1; see also a close-up view in Figure 4.2) produced from the developed string $D(T)$, also in deterministic fashion. To create the phenotype, the developed string is interpreted graphically through a Logo-style geometrical routine [1]. Initially, a graphic *turtle pen* points upwards and its memory stack is empty, then each symbol of $D(T)$ is executed as follows:

- `+` / `-`: the turtle changes its current direction by adding/subtracting an angle of 22°.
- `[`: the turtle pushes (saves) onto the stack its current position and direction.
- `]`: the turtle pops (restores) from the stack the last saved position and direction.
- `G`: the turtle advances 10 units in its current direction, marking the pixels on the way in black (part of the plant) and the last pixel in red (representing a leaf, see Figure 4.2 and an explanation of its role in Section 4.2.3).

The 22° value of the angle parameter was chosen for its ability to generate a great variety of natural-looking plant forms (see, e.g., Figure 4.10). It belongs to an interval of angles that present a good compromise between too linear, broom-like shapes (at smaller angles) and too irregular, contorted shapes (at larger angles; Figure 4.3). Three examples of relatively simple phenotypes are displayed in Figure 4.2. Their genotypes are (from left to right): `G[--GG][G]`, `G[G][--G][++G]`, and `G[G][--G]++`. The corresponding developed strings are not shown, as they are rather long and repetitive (which is often the case, even when the initial genotype is not).

4.2.1.3 Genotype-phenotype mapping

At this point, it is important to remark that closely similar genotypes can map to very different phenotypes (divergence) and, conversely, the exact same phenotype can be produced by very different genotypes (convergence). Divergence is illustrated in Figure 4.4, where genotypes `G[G+[G][-G]]` and `G[G+[G][--G]]` differ only by one symbol, yet they produce widely dissimilar graphical forms. Convergence is mostly due (but not exclusively) to *redundancy* in genotypes and consequently in developed

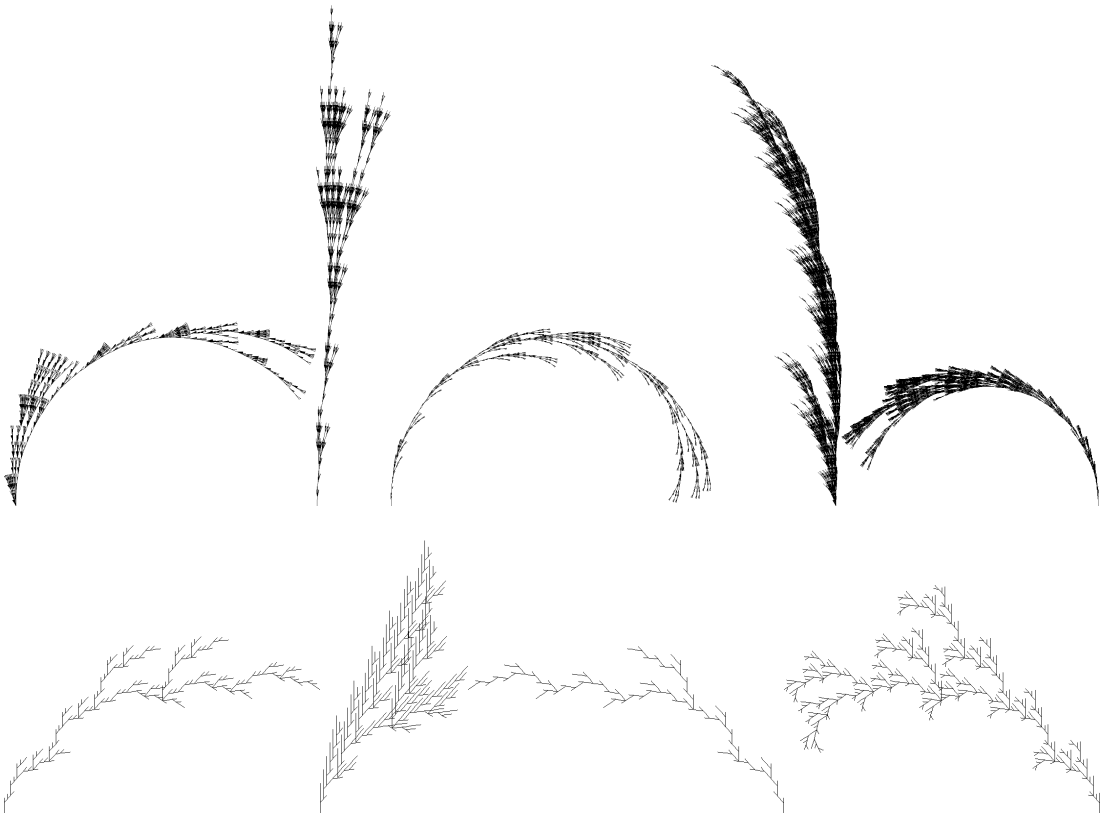


Figure 4.3: ***Choice of growth angle.*** **When constructing a graphical phenotype $P(D(T))$ from a developed string structure $D(T)$, the `+` and `-` symbols are interpreted as adding and subtracting a given angle to the turtle's direction. While this parameter was set to $22°$, other values were tried in the evolutionary experiments but not retained. (Top) Specimens obtained with an angle of $5°$: here, plants present a mostly linear, broom-like structure, which can be more or less bent into an arch. (Bottom, scaled up) Specimens obtained with an angle of $41°$: these plants tend to contain less material, and generally present a highly irregular, contorted structure. The value was set to $22°$ as an interesting compromise able to generate a greater variety of natural-looking plant forms (see Figure 4.10).**

strings: for example, the sequence `[G][G]G` has the effect of drawing the same branch three times, as every closed bracket followed by an open bracket takes the graphical turtle back to its previous bifurcation point. This also means that the total number of branches of the graphical phenotype $P(D(T))$ is not equal to the number of instances of `G` in $D(T)$ (which is the cube of the number of instances of `G` in T). For example, for $T =$ `G[--GG][G]` (left individual in Figure 4.2), the corresponding developed string $D(T)$ contains $4^3 = 64$ instances of `G`, whereas the corresponding phenotype $P(D(T))$ only has 33 distinct branches. This issue of relative uncorrelation between genomic similarity and phenotypic similarity when defining distance measures is discussed again

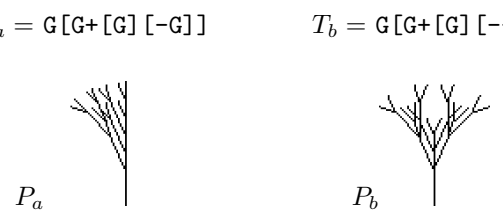


Figure 4.4: ***Phenotypic divergence.*** **Very similar genotypes can map to very dissimilar phenotypes. In this example, the genotypes T_a and T_b of two individuals (top row) differ by a single - symbol in the eighth position, yet their respective phenotypes P_a and P_b are strikingly different (bottom row).**

in Section 4.3.2.3. The specific case of genotypic redundancy is addressed in the next section.

4.2.1.4 Reduction of genotypes

The mapping from genotypes T to phenotypes $P(D(T))$ is many-to-one: there exist different genotypes that can be graphically translated into the exact same phenotype. Without loss of generality, since the mapping from T to $D(T)$ is one-to-one, let us consider only the transformation of $D(T)$ into $P(D(T))$.

For example, the string $D(T) =$ `G` creates the simplest phenotype of all, consisting of only one vertical branch. This would also be the case of strings $D(T) =$ `[[G]]` (the additional brackets have no effect), $D(T) =$ `[G][G]G` (the same branch is drawn three times by the turtle), and $D(T) =$ `[][+-G++]` (no effect is produced by empty brackets, `+-` pairs, or any combination of `+/-` symbols at the end of a bracket). This raises interesting questions, in particular: given a phenotype $P(D(T))$, what is the minimal developed string $D(T)$, hence minimal genotype T that can generate it?

While this question cannot be easily solved for every case, simple heuristics can be applied to a genotype T to remove the most blatant redundancies in order to substantially reduce the computational cost of the simulations, producing then what can be called the *reduced* genotype $R(T)$. It is important to note, however, that the reduced genotype is not part of the biological model, but only a tool to analyze this model (see Section 4.3.2). The phenotype is invariant by reduction of the genotype: $P(D(T)) = P(D(R(T)))$, while T's length increases at a significantly higher rate than $R(T)$'s length, as it will be later explained in Section 4.3.2.

The algorithmic reduction used here simply applies the following four transforma-

tion rules iteratively until no further change is made to the genotype:

1. Review all bracketed segments and remove any trailing `+/-` symbols inside them. Also remove all `+-` and `-+` pairs wherever they appear. For example: `G+-+[G+-+]` becomes `G+[G]`.

2. Remove empty brackets and extra layers of nested brackets, except the last pair. For example: `[[[G[]][]]]` becomes `[G]`.

3. Reduce expressions of the form `[X]X` to `X` and `[X][X]` to `[X]`, where `X` does not contain any brackets (to accomplish this, regular pattern matching are used for performance reasons).

4. Un-nest expressions of the form `[[`X_1`][`X_2`]...[`X_n`]]` to produce `[`X_1`][`X_2`]...[`X_n`]`, and expressions of the form `[[`X_1`][`X_2`]...[`X_n`]`X_{n+1}`]` to produce `[`X_1`][`X_2`]...[`X_n`][`X_{n+1}`]`, *unless* the outer pair of brackets encloses the whole genotype T. In the last case, removing it would dramatically change the phenotype, since the rule `G` $\rightarrow T$ is repeated three times to generate $D(T)$ (whereas it would not change anything if removed from around $D(T)$).

Other, more sophisticated reduction schemes could be used, e.g., ones that would take into account semantic constraints in addition to the syntactic structure, but they were ruled out in favor of simplicity and speed, because they would come at a higher computational cost.

4.2.2 Mutation operators

While clonal reproduction ideally creates perfect copies of successful genomes, other genetic mechanisms alter the genomic information passed on to descendants, ultimately causing evolution. A mutation operator transforming one genotype into another is denoted as M: $T' = M(T)$. In the present model, mutations are constrained to preserve well-balanced brackets in the symbolic expressions. Two types of mutations are modeled here: *point mutations* and *duplications*. Point mutation operators affect the genome at the level of a single symbol or a bracketed expression in three different ways:

- M_A: *alteration*, replacing a symbol (other than brackets) by another symbol (other than brackets);

- M_D: *deletion*, removing a symbol (other than brackets) or a whole bracketed expression of the genotype (under the constraint that the resulting genotype still includes at least one `G` symbol);
- M_I: *insertion*, adding a symbol or an empty bracketed expression `[]` (itself susceptible to be filled later on) at a random position in the genome.

Duplication mutation operators, on the other hand, only apply to bracketed expressions and provide the genome with an additional copy of that expression in three possible ways:

- M_R: *random duplication*, inserting the copy of the bracketed expression at any position in the genome (including possibly inside itself);
- M_L: *level duplication*, inserting the copy of the bracketed expression at any position in the genome that is located at the same bracket-nesting level than the original (where the nesting level of a position can be computed as the number of open brackets minus the number of closed brackets leading to that position);
- M_T: *tandem duplication*, inserting the copy of the bracketed expression immediately after the original (note that this kind of mutation is always silent according to the genotype-to-phenotype mapping, e.g., $P(D(\texttt{[G]})) = P(D(\texttt{[G][G]}))$, until further mutations change either the original or the copy).

A full mutation function is calculated by composing the above six elementary operators, where each type of operator is given in turn an independent probability of occurrence of 0.05. If more than one mutation is activated, then these are applied in a sequence (i.e., each operator transforms the string just produced by the previous operator). Point mutation operators are applied if they meet the following restriction: after the mutations, the first symbol outside brackets (that is, at nesting level 0) must be `G`, not `+` / `-`, in order to minimize the occurrence of an otherwise common evolutionary scenario (favored by the unbiased application of the mutation operators), where some individuals evolve into giant, slanted (in some cases even diagonal) forms with few branches, swiftly replicating into huge populations and exterminating smaller plants under their shadows (see Figure 4.7).

4.2.3 Environment and fitness function

The plants' environment is simply the discretized 2D plane containing the pixel-matrix occupied by the plant phenotypes. Accordingly, a pixel of the environment can be either empty or occupied by one or several overlapping plant bits (branches or leaves). This 2D world is bounded only downwards by the horizontal ground. It extends without bounds left, right and up. Plants are modeled solely by their aerial part, ignoring the roots, and can grow in width or height without restriction. This means that a plant can cover an unlimited number of pixels, and the length of occupied ground can increase as far as plants can reach.

Plants do not interfere with one another during growth; they develop independently as if in isolation. When immersed in the environment, the base of each plant's stem (i.e., where the turtle starts drawing) is assigned a horizontal coordinate on the ground, then the plant's branches and leaves freely expand in the open environment. As explained above, the exact phenotype $P(D(T))$ is a deterministic product of the turtle-graphic interpretation of the thrice rewritten string $D(T)$. If any branch of the plant hits the ground, i.e., if any pixel is going to be drawn below ground level, then the plant is marked as unfit and removed from the environment and the evolving population.

The environment offers a single source of energy: light, distributed homogeneously and uniformly over space and constantly over time. Light projects from the top, vertically, one beam per pixel column, and is captured exclusively by the plants' leaves (the red terminal pixels), not by their branches. In other terms, only leaves are opaque: a unit of light hitting a leaf is fully absorbed and contributes to the energy captured by the plant. It does not propagate further down, nor is it reflected in any direction. The only two other events involving light can be: falling through a branch or vanishing into the ground. If a unit of light hits a pixel that is occupied by several overlapping leaves from different plants, then the energy contribution will be captured by only one of these, chosen at random.

Based on this virtual energy source, each individual a in the environment has a replicating potential or *fitness*, determined in part by the total amount of light, denoted l_a, that it is able to capture. The other part contributing to the fitness, but in an inverse way, is the size of the foliage f_a, measured in number of branches. This number corresponds to the subset of those instances of `G` in $D(T)$ that caused the turtle to switch a white pixel to black (i.e., it excludes other instances of `G` that only caused

the turtle to redraw an existing branch; see previous discussion in Section 4.2.1.3). In summary, the fitness F_a of each individual is based on the ratio of gained energy l_a over spent energy f_a, thus represents the *ease* of building its structure (the higher F_a, the better):

$$F_a = \frac{l_a}{f_a^{\alpha}}$$

in which $\alpha \in [0, 1]$ is a constant exponent. Parameter α tunes the environmental cost of growing and branching, and for this reason α can be described as the *harshness* of the environment: the higher α, the more difficult it is for the plant to increase its surface exposure to light without compromising its overall fitness. Typical values of α will be chosen in the [0.5, 1] interval (see Section 4.3), which corresponds to fitness values F_a roughly of the order of 1.

4.2.4 Evolutionary schedule

Numerical simulations of plant populations are organized in synchronous cycles or *generations*. At each time step, the previous generation of plants is entirely removed from the environment and a new generation is calculated by applying the following four transformations (Figure 4.2):

(A) *Development*: Each plant of the i^{th} generation (offspring of another plant from the $(i-1)^{th}$ generation) is fully developed from the ground up by calculating its phenotype $P(D(T))$ from its genotype T. Phenotypes with branches projecting below the ground are discarded from the population.

(B) *Competition*: Each plant captures the light arriving to its leaves and calculates its fitness value F. This is done by positioning the phenotype's stem in the environment at a location that was randomly chosen when the $(i-1)^{th}$ generation was spawned. As explained in the previous section, F can be affected by the overlap with neighboring plants.

(C) *Reproduction*: Each plant then spawns a certain number of descendants, function of its fitness, and each descendant is assigned a random position in the environment near its parent (see details below).

(D) *Mutation*: Finally, each descendant is subject to the mutation operators explained in Section 4.2.2, potentially modifying its genotype. After mutation, the set of all resulting genotypes make the $(i+1)^{th}$ generation, which is ready to be processed in the same way as the previous generation, starting again from step (A).

In stage (C) the number of descendants of an individual is calculated by rounding to the nearest integer (up or down) the sum of its fitness plus a random value uniformly drawn from the interval [0, 0.6]. The rationale behind 0.6 is twofold: giving individuals with zero fitness a chance of staying in the game by maintaining one descendant, and giving individuals with fitness approximately equal to 1 a chance to have more than one descendant. Subsequently, offspring is positioned in the vicinity of the parent in a way that resembles the strategy of stolons in vegetative reproduction by clonal growth [12]: positions along the horizontal axis of the ground are uniformly drawn in a 100-pixel range centered around the stem of the parent. The width of the range influences evolutionary pressure, as it modulates the ease of colonizing empty space.

The initial population is composed of a single plant, whose genome codes for the simplest L-system able to reproduce: the single-character string `G`. This single-individual/single-symbol initialization scenario is at the core of the main motivation in this work, which is to study the evolution of diversity in purely emergent communities. It is thus best to start from the simplest forms of organization, both at the organism and the population levels. If the simulation were started from a population of already developed plant shapes, this would bias the results of the study.

The (A)...(D) loop is run for a preset amount of generations, 500 in this work. It can also be stopped earlier if it becomes too computationally costly due to occasional explosions in size of T, hence $D(T)$ and $P(D(T))$.

4.3 Experimental evolution of virtual plant communities

Here, the results of a number of experiments are presented in detail. The main observation is that evolution drives a single individual containing the simplest possible genome $T =$ `G` toward an increasingly large and complex population that expands along the horizontal ground in both directions and grows vertically. The simulations of virtual plant communities illustrate how organisms proliferate, diversify and eventually colonize their world. With each individual able to grow, absorb energy, replicate and

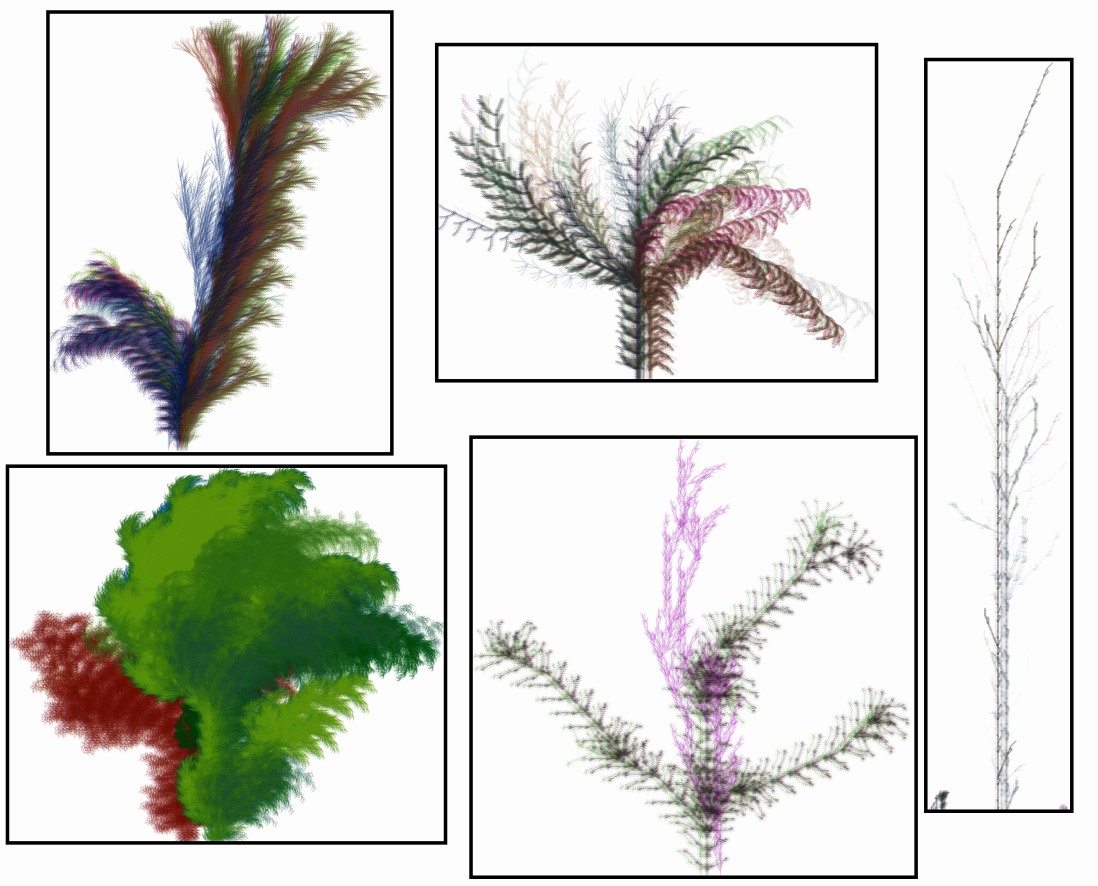


Figure 4.5: ***Several examples of simulations in mild environments.*** **Five examples of final populations of simulations in mild environment (with $\alpha = 0.5$). The plants are not shown at the same scale, as the proportions of the images vary greatly. The common motif is the evolution of very large and branchy plants that drive smaller ones to extinction. In all cases, the simulation had to be stopped well before the 500^{th} generation.**

mutate, experiments show that the size of the population first increases, then starts to diversify. It can then take different evolutionary paths. Final distributions of phenotypes range from a spread-out collection of simple individuals to winner-take-all situations where a few individuals have developed dramatically large and complex morphologies, extinguishing all simpler types – after which they start a Red-Queen race among each other toward even greater heights and widths.

The evolutionary dynamics of the model are modulated by the exponent α (see definition above in Section 4.2.3). Three classes of *environment* can be recognized, corresponding to different values of α:

- *mild environment*, corresponding to low α values (see Figures 4.5 and 4.9): the

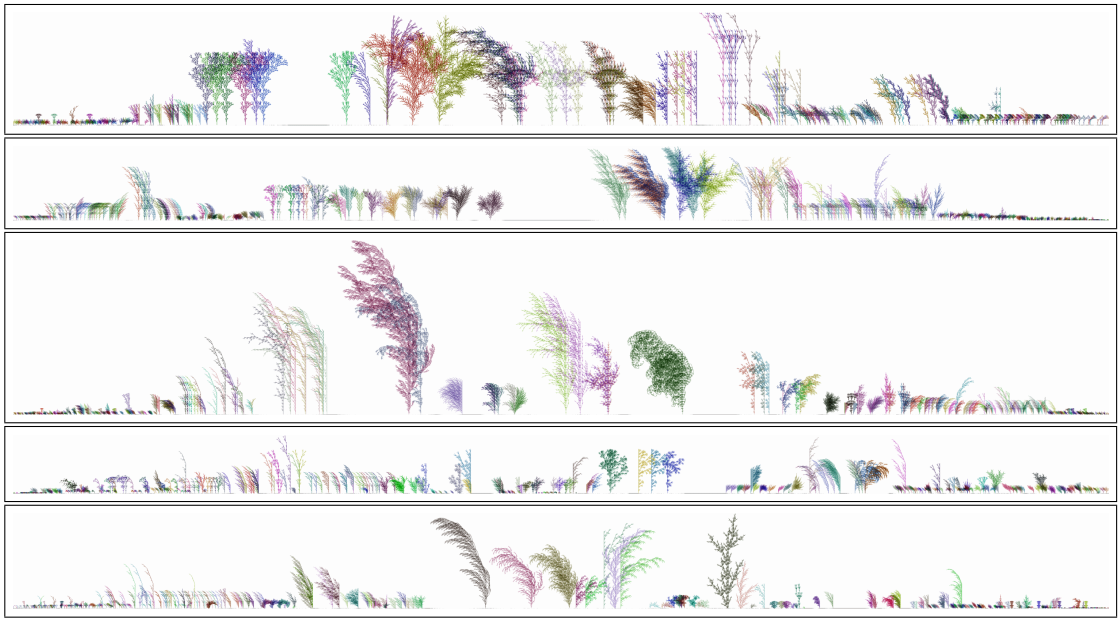


Figure 4.6: ***Several examples of simulations in harsh environments.*** **The 300^{th} generation of five simulations in harsh environment. The first three have $\alpha = 0.7$, while the last two have $\alpha = 0.75$. The simulations were ran up to the 500^{th} generation, but the resulting images are too long to be adequately shown in this page. The simulations are not to the same scale, as the proportions of the images vary greatly. The common motif is the evolution of complex, large shapes (though far smaller than in Figure 4.5) while smaller ones continue to thrive at the edges of the populations.**

cost of growing is negligible, yet the population collapses rapidly in most cases, because a handful of individuals manage to evolve toward very tall and/or branchy forms, blocking most of the light and swiftly eliminating (either all or almost all) smaller competitors. The time to collapse is highly variable. As the required memory and computation time increase exponentially, mild environments cannot be simulated for many generations.

- *harsh environment*, corresponding to medium α values (see Figures 4.6 and 4.10): the cost of growing is higher than in the previous case, thus simple individuals can proliferate while more complex individuals develop, compete and eventually become extinct, resulting in a fluctuating and/or slowly declining population size. In some cases, the population size collapses as in mild environments.

- *very harsh environment*, corresponding to high α values (see Figure 4.11): the cost of growing is so high that all individuals remain simple and proliferate near the ground.

Figure 4.7: ***Tall and slanted plant population.*** **100^{th} generation of one evolutionary simulation under $\alpha = 0.5$. The plants have evolved toward very tall, slanted shapes (the world represented in this image is 41262-pixel wide by 46519-pixel high) with very few branches, enabling them to be tightly packed without significantly competing for light, so high population sizes can be attained. This evolutionary outcome is rare (just 8 instances out of 131 in the sample shown in Figure 4.8) unless point mutation operators are unbiased (see Section 4.2.2).**

Generally, the type of environment is reflected in the evolution of two global variables: the *population size*, equal to the number of plants, and the global *biomass*, defined here as the total number of branches of all plants. Even at an early simulation time, simulations of different environments can be roughly characterized by these variables (Figure 4.8). Roughly speaking, the harsher the environment, the higher the population size but the lower the biomass. For mild and very harsh environments, these correlations tend to become increasingly marked as the simulation time (number of generations) advances, while the parameters of harsh environments tend to fluctuate. As α increases, the transition from mild to harsh environment is stochastic. For example, at $\alpha = 0.65$, some simulations exhibit typical characteristics of mild environment, while others are more typical of harsh environment. In contrast, the transition from harsh to very harsh environment is smoother, without a clear transition at some value of α: plants simply become smaller and less branchy as α increases.

In some simulations with low/medium α values, very tall, slanted plants with very few branches evolve (see an example in Figure 4.7). These plants can be packed tightly without significantly competing for light, so they can attain high population sizes and biomass at the same time, and significantly deviate from typical evolutionary dynamics (for example, in cases with low α, the population size and the biomass become very high at the same time). This is the case in eight instances of the sample shown in Figure 4.8.

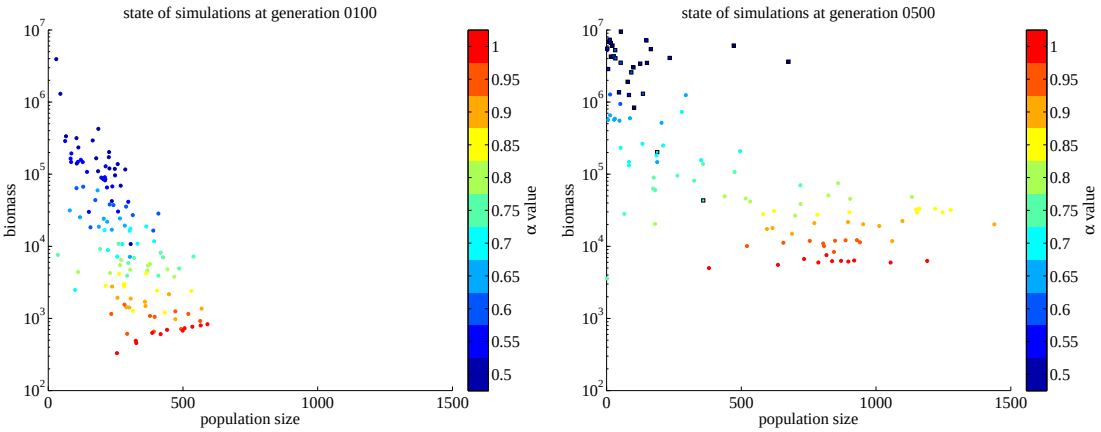


Figure 4.8: ***Evolutionary dynamics are modulated by the environment.*** **131 simulations with several different α values (from 0.5 to 1 in steps of size 0.05) are shown as points whose coordinates are their corresponding population size (number of individuals) and biomass (total amount of branches of all plants) at the 100^{th} (*left*) and the 500^{th} (*right*) generations of the corresponding simulation. In the right side, simulations which were stopped before the 500^{th} generation (because of considerations about the computational cost) are shown as squares instead of points. The color of each point represents the α value of the simulation (the higher α value, the harsher is the environment, as branches are more penalized). Mild, harsh and very harsh environments approximately correspond to bluish, greenish and reddish colors. Roughly speaking, the harsher the environment, the higher the population size but the lower the biomass.**

Three simulations (one with $\alpha = 0.5$, other with $\alpha = 0.75$, and other with $\alpha = 1$) from the sample in Figure 4.8, representing typical examples for each type of environment (respectively mild, harsh and very harsh), will be analyzed through the remainder of this section from several points of view, highlighting similarities and differences between the corresponding environments. *Distance* measures are also presented in Section 4.3.2 for a more rigorous quantitative assessment of the visible qualitative differences between the plants from these three dynamical regimes.

4.3.1 Overview of the population dynamics

Figures 4.9, 4.10 and 4.11 display a full example of one simulated population for each type of environment. Individuals are colored randomly for visualization purposes. The first population (Figure 4.9) is typical of a mild environment where plants grow to a great magnitude, both in height and breadth, and are intensely competing with neighboring individuals. The second population (Figure 4.10) shows a moderately harsh environment filled with a higher diversity of individuals presenting a more limited growth area and interfering less with other individuals. Simpler plants grow at the

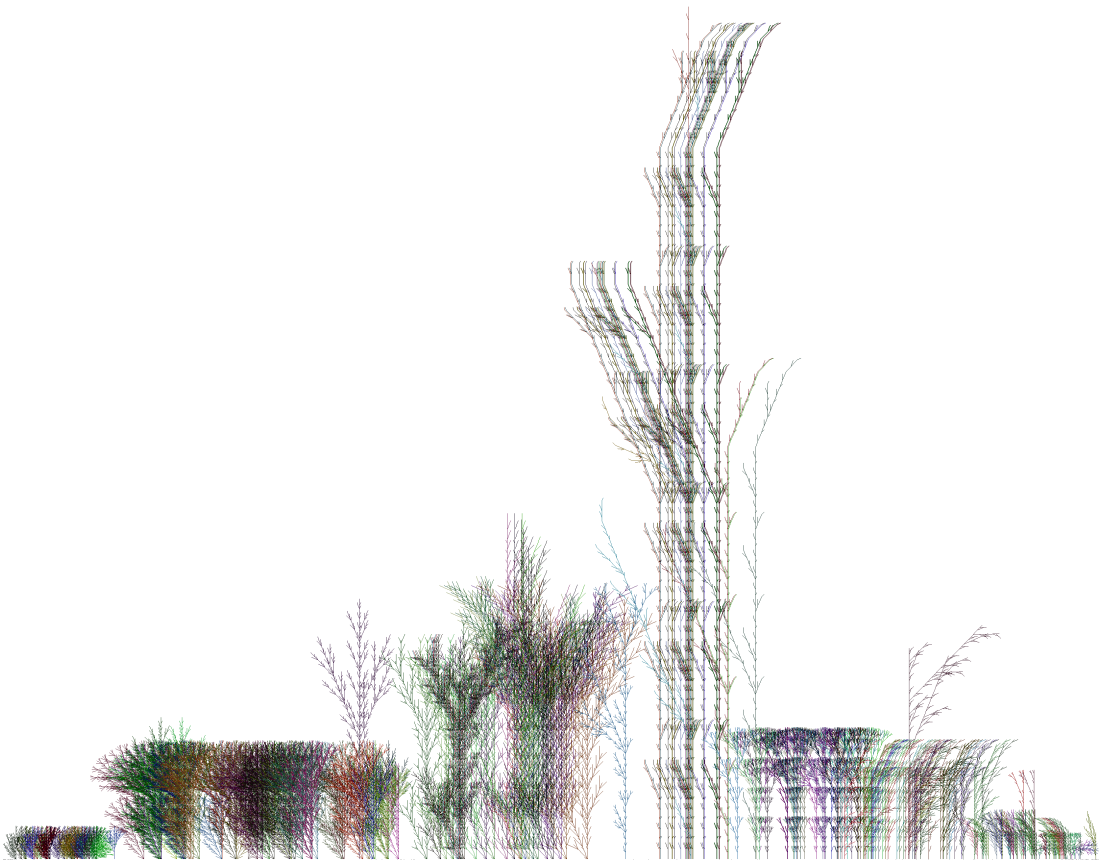


Figure 4.9: ***Typical plant world in a mild environment.*** **100^{th} generation of one evolutionary simulation under $\alpha = 0.5$, which has been analyzed in detail in this chapter (see Figures 4.12, 4.13, 4.14, 4.16 and 4.18). The world covered by this plant population is 3333-pixel wide by 2602-pixel high. Already in the early stages, extreme competitive pressure is driving a group of plants toward a runaway Red Queen effect, in which they try to overshadow each other, while smaller individuals are at a disadvantage under the taller ones and become progressively extinct. Due to an exponential increase in CPU time and memory, simulations under these conditions have to be stopped earlier than the other conditions, here at the 150^{th} generation. A sequence of snapshots of the simulation corresponding to this image is provided in Section B.1.**

two ends of the population's domain. Finally, the last population (Figure 4.11) is characteristic of a very harsh environment, where plant forms remain simple and short and colonize the ground efficiently. A series of images depicting the evolution of each one of these three simulations can be found in Appendix B. In the remainder of the chapter, these three simulations will be analyzed from several points of view.

The dynamics of the population size and the biomass characterize the environments (Figure 4.12). In a mild environment (black line), a fast spike in the number of plants is followed by a great extinction event, as tall individuals eliminate smaller ones by

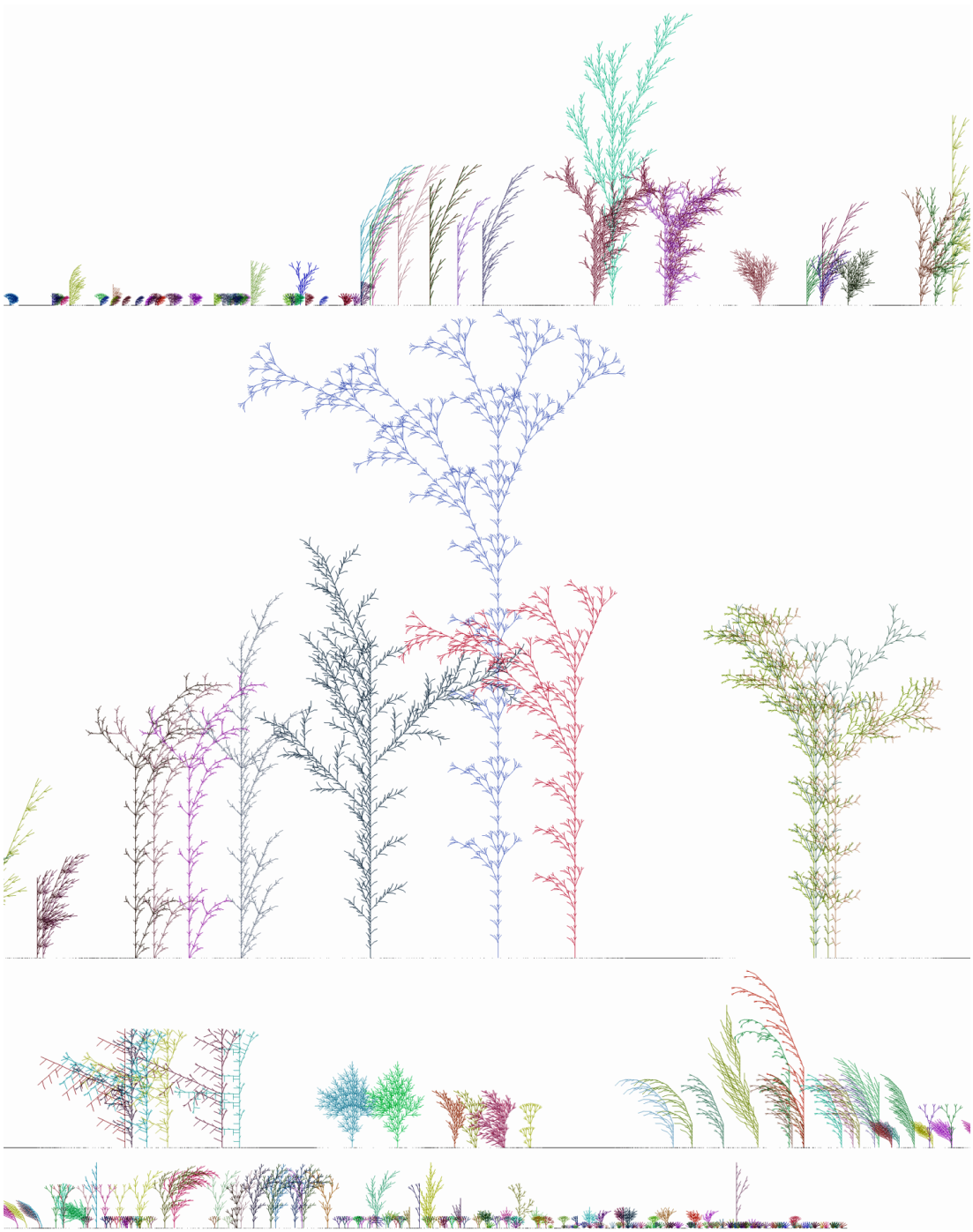


Figure 4.10: ***Typical plant world in a harsh environment.*** 340^{th} **generation of one evolutionary simulation under** $\alpha = 0.75$**, which has been analyzed in detail in this chapter (see Figures 4.12, 4.13, 4.14, 4.16 and 4.17). The world covered by this plant population is 12627-pixel wide by 2199-pixel high and, for clarity, has been divided into four consecutive segments, all of them at the same scale. Individuals in the central region have become more and more complex as they compete for light. Large and complex individuals sometimes cause the appearance of desolate areas, as they decimate smaller neighbors and occasionally disappear themselves due to an overload of branches, thereby opening the way for simpler and faster-expanding organisms to colonize new regions. A sequence of snapshots of the simulation corresponding to this image is provided in Section B.2.**

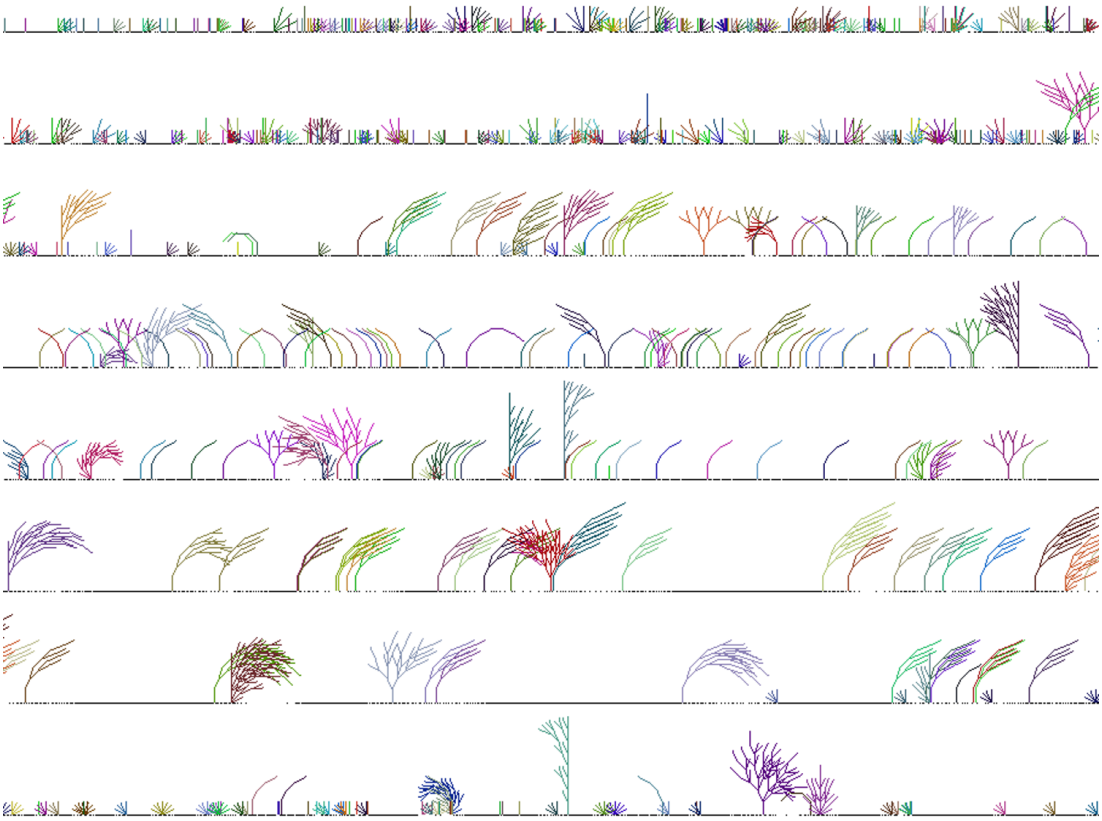


Figure 4.11: ***Typical plant world in a very harsh environment.*** **340^{th} generation of one evolutionary simulation under $\alpha = 1$, which has been analyzed in detail in this chapter (see Figures 4.12, 4.13, 4.14, 4.16 and 4.18). The world covered by this plant population is 7220-pixel wide by 92-pixel high and, for clarity, it has been divided into eight consecutive segments. Individuals remain simple and efficient, as biomass acquisition is severely penalized. A sequence of snapshots of the simulation corresponding to this image is provided in Section B.3.**

overshadowing their canopies. Even in this scenario of dwindling population size, however, the few remaining plants evolve toward such gigantic forms, and do this so fast, that total biomass increases exponentially. By contrast, in a harsh environment (dark gray), the population size fluctuates as colonization of new territories is punctuated by local extinctions due to tall plants. Biomass also fluctuates, but tends to grow over larger time periods, as competition triggers the evolution of larger plants. Finally, in a very harsh environment (light gray), the population usually increases faster than in the previous case, but fluctuations remain present, since a limited degree of complexification still takes place while plants compete for light. Biomass grows very slowly, however,

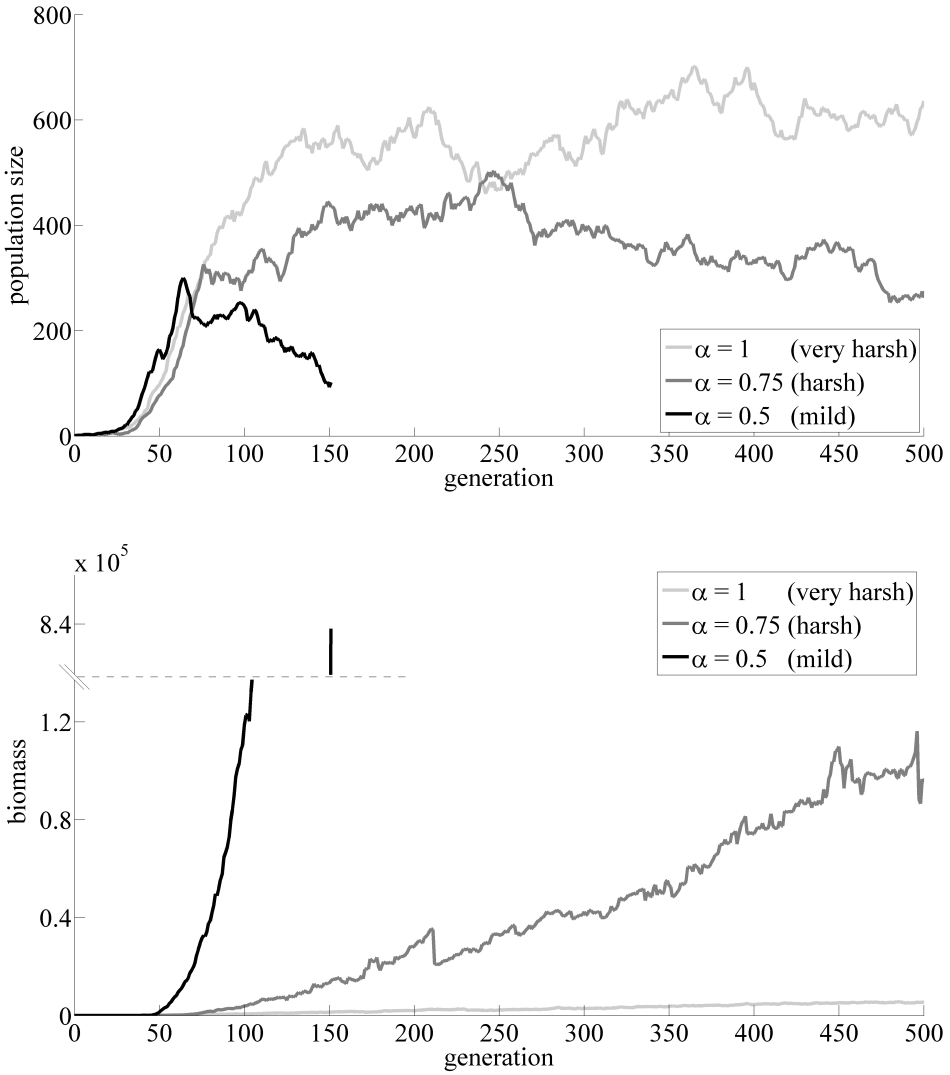


Figure 4.12: ***Evolution of population size and biomass.*** **(Top) Population size is the total number of individuals. (Bottom) Biomass is the total quantity of branches across all plants. In mild simulations ($\alpha = 0.5$, black curves, see also Figure 4.9), population size spikes briefly before tall plants start exterminating shorter ones, as they compete to become ever taller and larger at a fast pace. Meanwhile, however, biomass continues its exponential increase (reaching about 840,000 after the 150^{th} generation), mostly sustained by these few gigantic specimens. In harsh simulations ($\alpha = 0.75$, dark gray curves, see also Figure 4.10), the population size grows fast at first, but soon starts to fluctuate, as episodic and local extinction events take place. Sudden large drops in biomass are associated with the extinction of large individuals. In very harsh simulations ($\alpha = 1$, light gray curves, see also Figure 4.11), the population size increases faster, while plant morphologies remain very simple, and, consequently, biomass grows very slowly compared to the other conditions. Plants are still subject to a complexification process, albeit at a smaller scale, and population size also fluctuates a little as they compete.**

mostly as a consequence of the small size of the plants.

Mean fitness values ($\bar{F}$) in each kind of environment are also revealing (Figure 4.13). In very harsh simulations, where $\alpha = 1$, $\bar{F}$ is always less than or equal to 1 because by definition the number of light beams l_a captured by an individual a cannot be higher than its number of leaves, hence branches f_a. In milder simulations, where $\alpha < 1$, some individuals can reach fitness values much larger than 1. However, results show that the milder the environment, the lower the mean fitness level. This is due to the fact that milder environments are more crowded because plants are more prolific than in harsher environments and offspring's positions are statistically closer to their parents. Crowded conditions then create more intense competition, resulting in a significant decrease of the mean fitness. In fact, Figure 4.13 clearly shows that the mean fitness never rises above 1 in any environment, which indicates that high-fitness individuals constitute a minority whose sum never outweighs the rest of the low-fitness population (with respect to reference level 1). Note also an interesting phenomenon of episodic collapse of the fitness distribution during the early generations in all environments: the mean reaches exactly 1, while the standard deviation drops to 0. These particular generations correspond to populations composed exclusively of elementary stick plants of length 1, whose fitness is exactly 1 by definition (1 unit of light divided by 1 branch, for any exponent α). In these moments, more complex individuals have all died out, albeit temporarily.

It is also interesting to analyze the growth of genomic length and reduced genomic length as the simulation time advances. Interestingly, the growth depends on the type of environment. In Figure 4.14, the mean genomic length and mean reduced genomic length are plotted for three simulations, one of each type. The X axis represents simulation time. In all three simulations, but especially in the very harsh and harsh ones, the growth is approximately linear, both for reduced and nonreduced mean genomic length, the latter being dramatically shorter then the former: while the size of the genome increases continuously, the portion of it that is actually coding for the structure is kept more or less constant. Reduced genomic length grows at a slightly faster rate for the harsh simulation than for the very harsh one. However, the picture is very different in the mild simulation: as the population is driven to become ever more taller and more branchy by a very strong evolutionary pressure, this is reflected on the mean genomic length, growing at a much faster (although still linear) rate. In

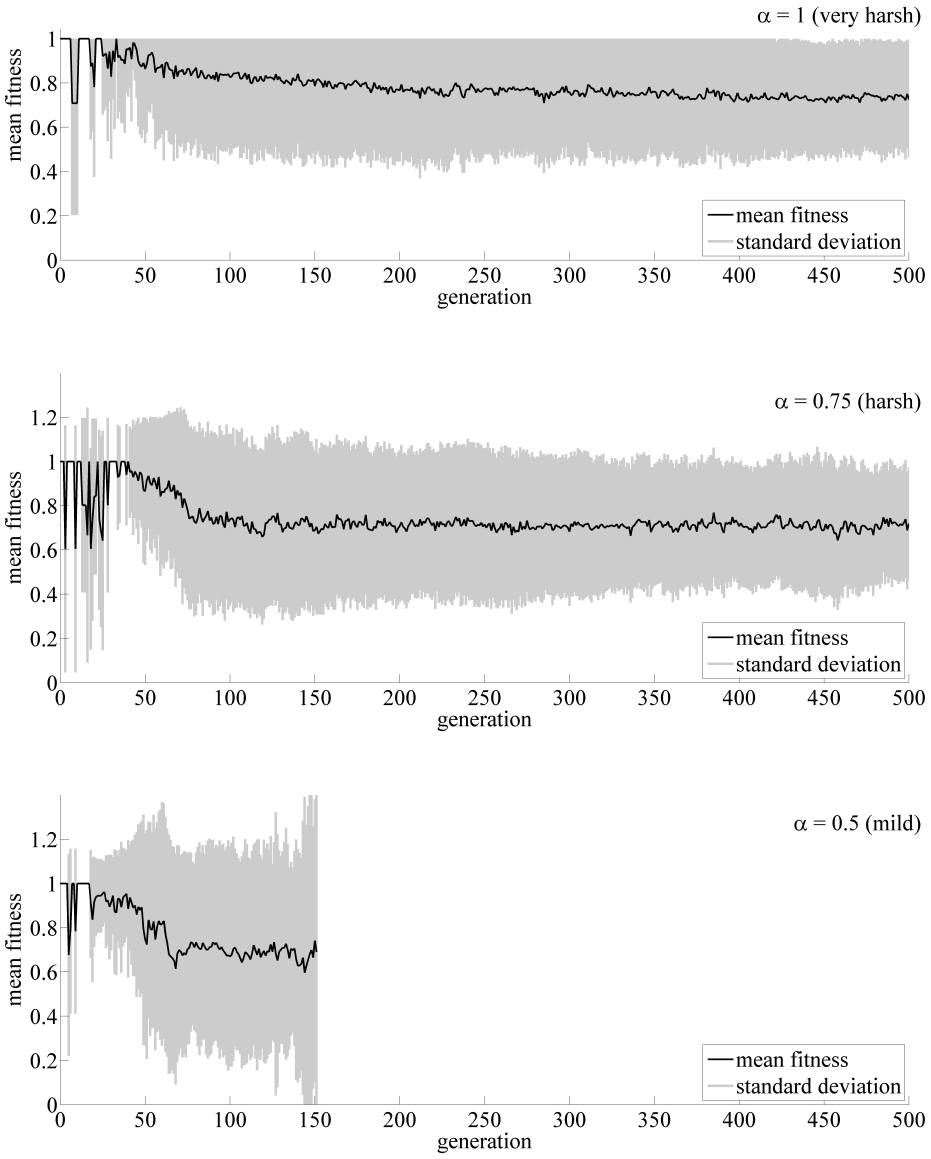


Figure 4.13: ***Evolution of mean fitness.*** **At each time step, the mean fitness (black curves) is defined as the sum of all fitness values divided by the population size. Standard deviation bars are also shown (gray areas). (Top) Very harsh condition ($\alpha = 1$, see also Figure 4.11). (Middle) Harsh condition ($\alpha = 0.75$, see also Figure 4.10). (Bottom) Mild condition ($\alpha = 0.5$, see also Figure 4.9). In all simulations, the initial value is 1, corresponding to the fitness of a single individual of genotype `G` (receiving one unit of light divided by one branch) in the absence of competition. Later, the mean fitness drops below 1 as plants become larger and more complex and compete for light, but the few individuals with fitness higher than 1 (when $\alpha < 1$) never outweigh the low-fitness majority. Note also episodic collapses of the fitness distribution to populations composed exclusively of 1-stick plants (fitness 1, deviation 0), during the early generations. The milder the environment, the faster the mean fitness drops, and the higher the standard deviation becomes, since more plants tend to die out under the shadow of larger individuals. In the harsh and very harsh environments, the mean fitness seems to converge to slightly different values, close to 0.7.**

contrast with the other two cases, the mean reduced genomic length is much higher, and represents a significant part of the raw genomic length, again evidencing the strong evolutionary drive to produce taller and more branchy plants. The sudden drop on mean genomic length just at the end of the simulation is due to an extinction event.

Note that the mild environmental condition $\alpha = 0.5$ is also characterized by an exploding growth in computational cost, both in memory and time. This is why it was decided to stop the simulation after 151 generations, while the time to generate the phenotypes and evaluate the fitness of the individuals was still within reasonable limits. In order to estimate the computational cost of pursuing the simulation toward later generations, an exponential curve was fitted to a 2D cloud of experimental (i, t_i) points, where i is the generation number and t_i the CPU time required to calculate the plants up to generation i. Extrapolating from this curve, the CPU time to reach generation 200 increased 100-fold, while for generation 300 it was 70000-fold. The computational cost of harsh and very harsh simulations does not grow so quickly, but they still can become difficult to be simulated after several hundreds of simulations. In this work, they were up to 500 generations, when computational costs begin to be too high.

4.3.2 A distance-based measure of diversity

To quantitatively assess the amount of *diversity* of the plant population in each environment, a measure of *distance* between individuals is needed. Two definitions of distance are proposed here: a genetic distance and a phenotypic distance, then they are analyzed to study their abilities to differentiate among the three types of environment.

4.3.2.1 Definition of a genetic distance

The *edit distance* $\mathcal{E}(T_a, T_b)$ between two strings of symbols T_a and T_b (the genomes of two individuals a and b) is defined as the minimal number of insertions, deletions and alterations of symbols in T_a needed to transform it into T_b, or vice versa (it is indeed symmetrical, as every insertion can be reverted by a deletion and every alteration by the opposite alteration). In real-world genomics, the genetic distance between nucleotide sequences is measured in many cases by a weighted edit distance [11], using one of several possible weighting schemes [4] specifically fitted for each specific task. In many cases, however, genomic metrics take into account some statistical model of mutation [63]. Here, genetic diversity is measured using a raw edit distance

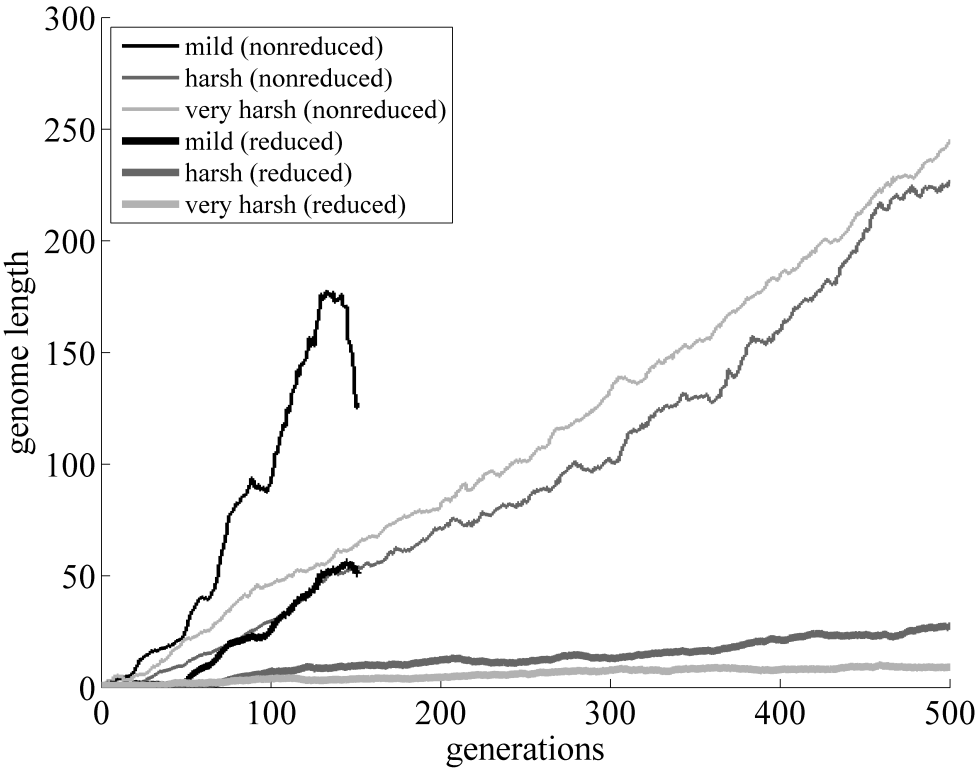


Figure 4.14: ***Evolution of mean genomic length and mean reduced genomic length*** **for the three simulations, one of each type: mild ($\alpha = 0.5$, see also Figure 4.9), harsh ($\alpha = 0.75$, see also Figure 4.10) and very harsh ($\alpha = 1$, Figure 4.11). For very harsh and harsh simulations, genetic material is gained at a more or less linear rate over evolutionary time, but most of it is useless. For mild simulations, the rate of growth is dramatically higher for both reduced and non-reduced genomic lengths, indicating the strong evolutionary pressure for the individuals to become ever more complex, in a runaway Red Queen effect. The sudden drop in mean genomic length that can be seen at the end of the mild simulation is due to an extinction event of a subpopulation with particularly large genomes.**

(without weighting). For example, if T_a = `G+[G[-G]G]+` and T_b = `G[G[--G]G]G`, then $\mathcal{E}(T_a, T_b) = 3$, as T_a can be transformed into T_b by removing the second symbol from T_a, adding a symbol `-` before the third `G`, and replacing the last symbol `+` by `G`.

4.3.2.2 Definition of a phenotypic distance

For a measure of the degree of dissimilarity between morphological phenotypes, the *Jaccard distance* (denoted $\mathcal{J}$) is used. This distance calculates the ratio of overlapping pixels with respect to the total number of pixels occupied by both plants, which are aligned and superimposed to make their stem bases match. Thus if P_a and P_b are the

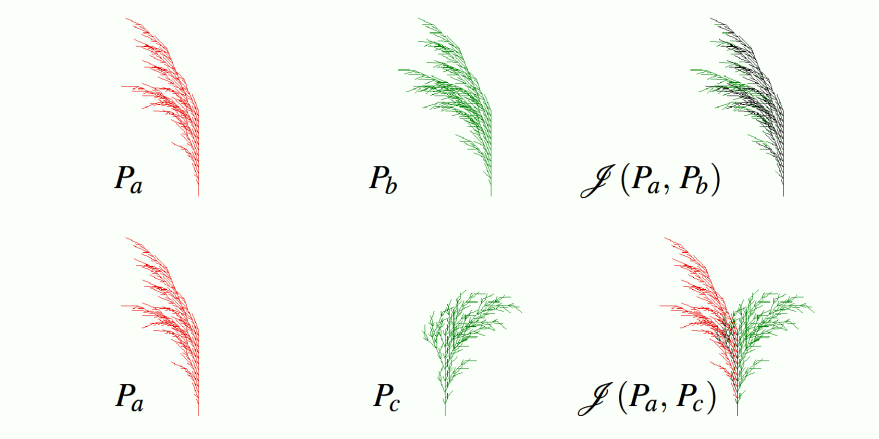


Figure 4.15: ***Illustration of the Jaccard distance.*** **Graphical basis of the measure of similarity between individual phenotypes. (Left) One same individual a, top and bottom. (Center) Two different individuals b (top) and c (bottom). (Right) Superimposition of a and b (top), and a and c (bottom) by alignment of their stem bases. Black pixels represent the portion of phenotype shared by the two individuals, while green and red pixels show their symmetric difference. The Jaccard distance between two phenotypes is then calculated by dividing the number of common pixels (in black) by the total number of pixels (union of the three colored areas) and subtracting the result from 1, which gives here: $\mathcal{J}(P_a,\ P_b) = 0.31$ (top) and $\mathcal{J}(P_a,\ P_c) = 0.92$ (bottom).**

pixel sets of two aligned individuals a and b, the Jaccard distance reads:

$$\mathcal{J}(P_a,\ P_b) = 1 - \frac{|P_a \cap P_b|}{|P_a \cup P_b|} \tag{4.1}$$

Figure 4.15 shows three distinct phenotypes P_a, P_b and P_c, illustrating the Jaccard distance between two similar ones ($\mathcal{J}(P_a,\ P_b) = 0.31$) and two dissimilar ones ($\mathcal{J}(P_a,\ P_c) = 0.92$).

4.3.2.3 Comparing genetic and phenotypic diversity

For a given distance $\mathcal{D}$, whether $\mathcal{E}$ or $\mathcal{J}$, the diversity $\delta_\mathcal{D}$ of the population is defined as the mean over all the values of the distance matrix $[\mathcal{D}_{ab}]_{N\times N}$:

$$\delta_\mathcal{D} = \frac{1}{N^2}\sum_{a=1}^{N}\sum_{b=1}^{N}\mathcal{D}_{ab} \tag{4.2}$$

where N is the size of the population, and $\mathcal{D}_{ab} = \mathcal{E}(R(T_a),\ R(T_b))$ or $\mathcal{D}_{ab} = \mathcal{J}(P_a,\ P_b)$ (similar definitions, such as *discrepancy*, have been used elsewhere [186]). Figure 4.16 shows the evolution of $\delta_\mathcal{E}$ and $\delta_\mathcal{J}$ over time, using each of the two distances above. Note that the edit distance $\mathcal{E}$ was applied to reduced genotypes $R(T)$ only; interestingly,

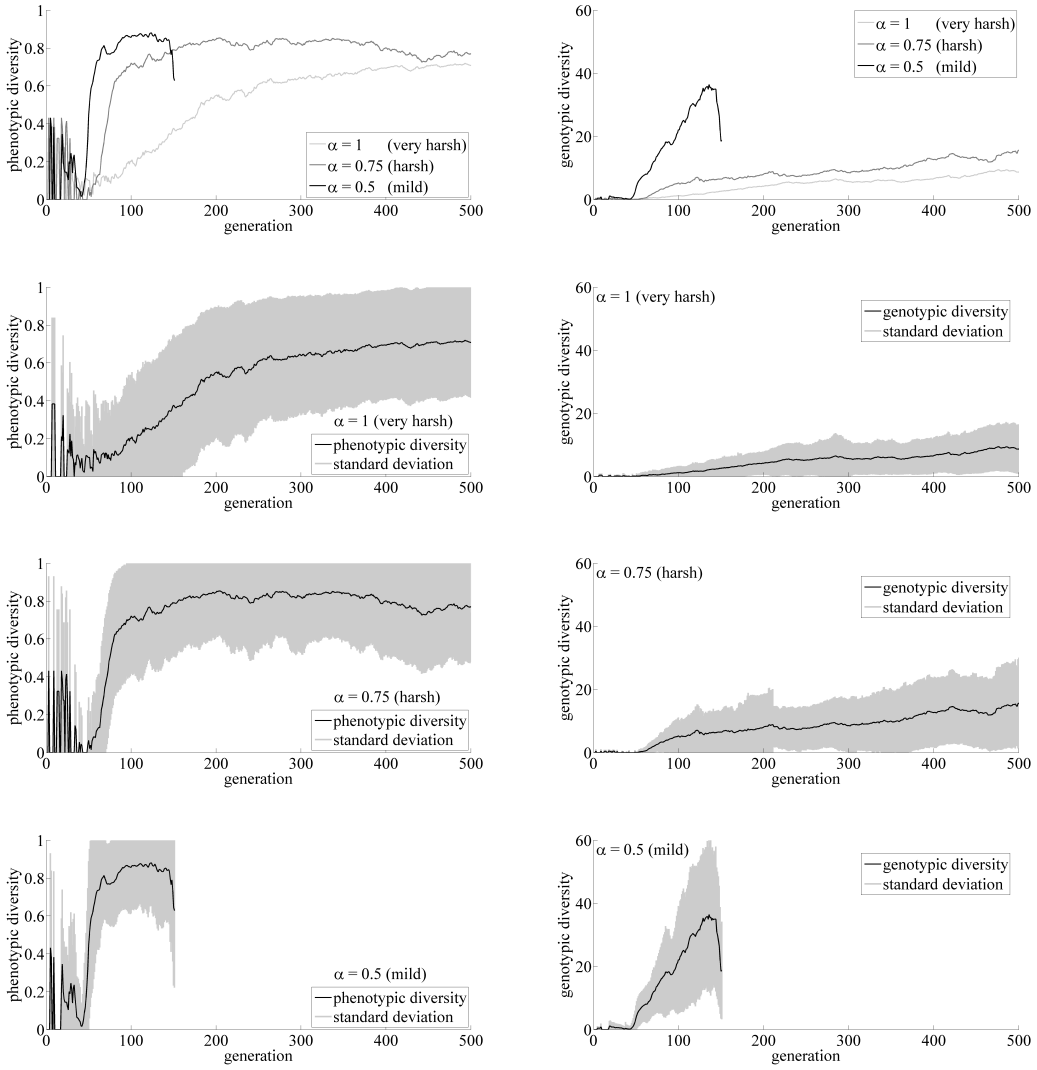


Figure 4.16: ***Evolution of diversity.*** **(Left) Phenotypic diversity $\delta_{\mathcal{J}}$ is defined as the mean of the Jaccard distance matrix $\mathcal{D}_{ab} = \mathcal{J}(P_a, P_b)$ over all pairs of individuals (a, b). (Right) Genotypic diversity $\delta_{\mathcal{E}}$ is defined as the mean of the edit distance matrix $\mathcal{D}_{ab} = \mathcal{E}(R(T_a), R(T_b))$ applied to reduced genomes. (Top) Comparing diversity values in the three different kinds of environments, mild ($\alpha = 0.5$, black curves, see also Figure 4.9), harsh ($\alpha = 0.75$, dark gray curves, see also Figure 4.10) and very harsh ($\alpha = 1$, light gray curves, see also Figure 4.11). (Bottom three rows) Same diversity curves, separately for each environment, adding standard deviation bars (gray areas). The initial high-amplitude fluctuations in the phenotypic diversity are an artifact of the small population size, in which the emergence and extinction of a few mutated individuals appear large. The drop in diversity at the end of the mild simulation is due to the extinction of a large number of relatively small individuals. The sudden reduction of variance in the genotypic distance of the harsh environment, around generation 210, is due to the extinction of one plant with a large genome.**

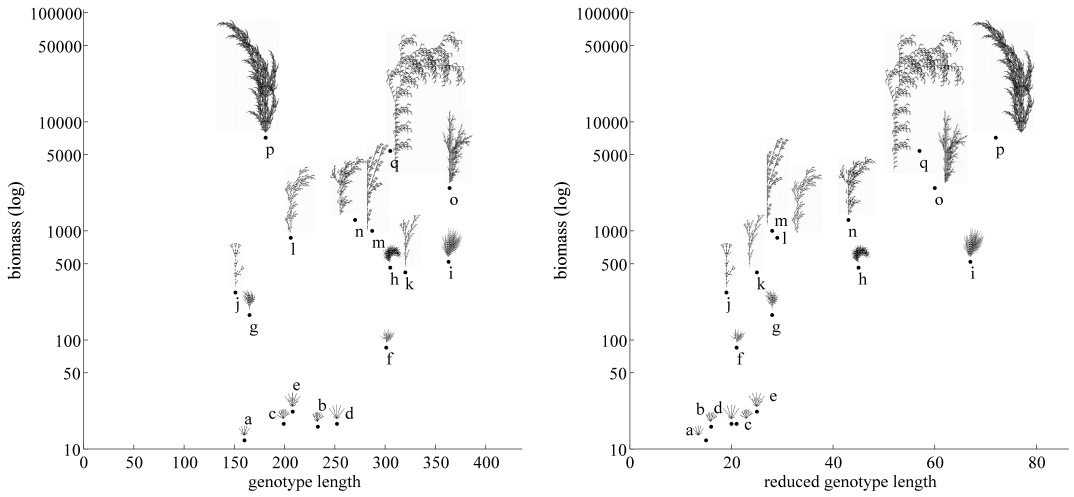


Figure 4.17: ***Biomass vs. genome length.*** **Snapshot of a few specimens from the 500^{th} generation of a harsh-environment simulation ($\alpha = 0.75$, see also Figure 4.10). The biomass of a plant is defined as its total number of branches. (Left) Biomass does not seem to be correlated with raw genotype length T in any meaningful way. (Right) Showing the same individuals, dependency of biomass on the *reduced* genotype length $R(T)$, however, appears more correlated. Here, three trends of increasing biomass vs. increasing $R(T)$ can be identified: one for individuals a-e (scaled 4x), other for individuals f-i (scaled 2x), and other for individuals j-p (scaled 1x). Smaller individuals are represented at a magnified scale, otherwise they would appear too small.**

results obtained with original genotypes T turned out to be very similar, essentially differing in scale, so they were omitted for clarity.

In milder environments, diversity $\delta_{\mathcal{J}}$ based on the Jaccard distance increases faster toward its highest value 1 (in particular, the growth is considerably slower for the very harsh simulation). The evolution of diversity $\delta_{\mathcal{E}}$) based on the edit distance between reduced genomes, for its part, is rather similar in the harsh and very harsh environments, while the mild conditions clearly stick out through high values. This process of accelerated diversification is driven by a high growth rate in the length of the reduced genotypes.

The observed differences between the genetically based and phenotypically based notions of diversity and their evolution over generations can be ascribed to the decorrelation introduced by the indirect developmental mapping. As discussed in Section 4.2.1.3, a consequence of this mapping is that changes in the genotype may have wildly different consequences on the phenotype: some mutations may cause only slight modifications in the phenotype, or none at all, while others can cause major

transformations. This is famously the case of homeobox genes, which play a critical role in the establishment of the overall body plan of metazoan organisms [146], and whose mutation can give rise to ectopic structures, such as the development of legs in the place of antennae in *Drosophila*. Conversely, the phenomenon of convergent evolution also hints at the possibility of very different genomes associated with very similar phenotypes. Thus genetic diversity and phenotypic diversity are not necessarily correlated. Figure 4.4 shows a case where similar genotypes give rise to clearly different phenotypes, while, on the contrary, related phenotypes encoded by very different genotypes can be seen in Figure 4.17.

4.3.3 Robustness to mutations

Evolutionary search relies on the application of genetic operators to the individuals of a population (i.e., in this work, to their genomic strings T, *not* their developed strings $D(T)$) in order to create variability. Since reproduction is asexual here, these operators intervene only in the form of mutations. By mutating, individuals explore the fitness landscape in search of solutions adapted to the problem imposed by the environment via the fitness function. Mutation operators introduce alterations in the genome, which provoke changes in the phenotype through a developmental process. One could say that biology makes use of indirect strategies to encode organisms, in the sense that the genotype-to-phenotype mapping is generally highly complex. In real-world multicellular organisms, it is the result of a gigantic self-generating and self-assembling process involving between thousands and trillions of cells. In this abstract virtual model, it is produced by the linear expansion of a string that can typically contain between a dozen and a few thousand characters, giving rise to a geometric morphology covering hundreds to millions of pixels.

In this section, the Jaccard distance $\mathcal{J}$ (see definition above, Section 4.3.2.2) is used to study the extent of the disruption provoked by the different types of mutation operators in morphological phenotypes (only single mutations this time, not in sequence). The first five types of mutation operators described in Section 4.2.2 are considered: alteration (M_A), deletion (M_D), insertion (M_I), random duplication (M_R), and level duplication (M_L), leaving out tandem duplication. The notation $M \in \{M_A, M_D, M_I, M_R, M_L\}$ is used to refer to any one of these mutation operators.

The main motivation is to test the hypothesis that evolution encourages the

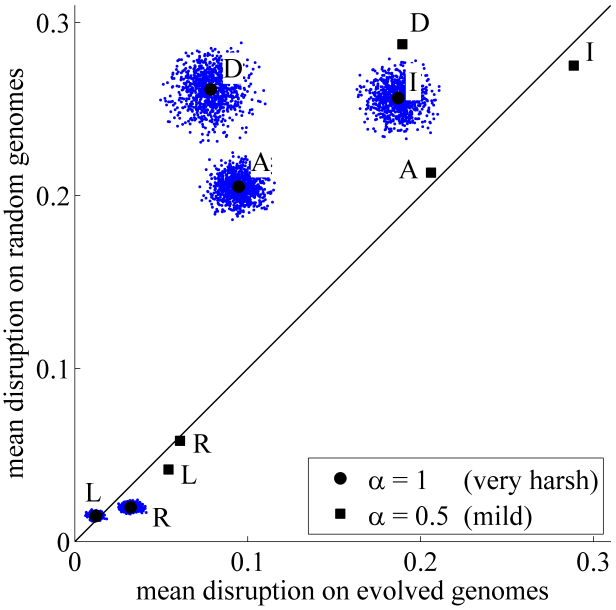


Figure 4.18: ***Robustness to mutation.*** **Mean disruption caused by five different types of mutation operators in pools E_m, E_{vh} of evolved genomes (horizontal axis) vs. corresponding pools R_m, R_{vh} of random genomes with the same statistical properties in each case (vertical axis). The disruption value for each genome in each pool is simply defined as the mean Jaccard distance between its phenotype and the phenotypes of 100 mutated versions of that genome. The coordinates of each point represent a pair of mean disruption values of a given mutation over an evolved pool and the corresponding random pool (see Section 4.3.3) in two different environments (squares: mild, $\alpha = 0.5$, see also Figure 4.9; circles: very harsh, $\alpha = 1$, see also Figure 4.11). It is important to note that the sizes of the pools are not identical: $|E_m| = |R_m| = 64$, while $|E_{vh}| = |R_{vh}| = 593$. To show that the results are statistically robust, 1000 random subsamples of size 64 have been drawn from the pools R_{vh} and E_{vh}, plotting the mean disruption values for each subsample as blue dots. It can be seen that the points cluster neatly around global mean values without far outliers. Mutations are labeled A for alteration, D for deletion, I for insertion, R for random duplication and L for level duplication. Point mutations (A, D, I) are generally more disruptive than duplication mutations (R, L) while, in the very harsh condition, they are significantly less disruptive in evolved genomes than in random ones. Plants evolved in a very harsh environment are also more resilient to mutation-induced phenotypic changes than plants evolved in a mild environment.**

selection of genomes that are more tolerant to mutation-induced disruption [214]. The concept of mutational robustness has been correlated with the evolutionary advantage of lower-than-optimal fitness plateaus under high mutation rates [212], the existence of large networks of neutral mutations [213] and the variability of the environment [170]. In the model, an evolved genome might show a more robust underlying structure than a random genome, in the sense that it would be able to minimize the effects of various mutations, especially if the environmental conditions remain stable during evolution.

Thus, for each mutation operator M, we are interested in measuring its effect on the phenotype of an individual a. To this aim, if b is the mutated individual such that $T_b = M(T_a)$, the *quantity of disruption* is defined as the Jaccard distance between a and b, $\mathcal{J}(P_a,\ P_b)$.

Then, the mean quantities of disruption caused by various mutation operators are calculated on six different pools of genomes:

- Three pools E_m, E_h, E_{vh} of evolved genotypes, one for each kind of environment, respectively mild, harsh, and very harsh, composed of all the distinct genotypes produced by the last generation of the corresponding simulation (for E_m the 150^{th} generation, due to the high computational cost discussed at the end of Section 4.3.1; for E_h and E_{vh}, the 500^{th} generation). These pools contain 64, 245 and 593 genotypes, respectively.

- Three other pools R_m, R_h, R_{vh} of randomly generated genotypes. Each pool R_x matches the corresponding evolved pool E_x as follows: it contains the same number of genotypes and its random strings were generated in such a way that they have the same statistical properties as the strings of E_x (average length and probability of appearance of each symbol, under the constraint of well-balanced brackets).

For each string T_a in each pool, 100 different mutated strings $T_b = M(T_a)$ were generated per mutation operator M (thus totaling 500 mutated strings per T_a) and the corresponding disruption values $\mathcal{J}(P_a,\ P_b)$ were evaluated. Finally, for each mutation operator M and each pool $E \in \{E_m, E_h, E_{vh}\}$, $\delta_{\mathcal{J}}^{M,E}$ was defined as the mean of all the values $\mathcal{J}(P_a,\ P_b)$ between each string P_a in the pool E and the corresponding strings P_b generated through mutation operator M:

$$\delta_{\mathcal{J}}^{M,E} = \frac{1}{||E||} \sum_{a=1}^{||E||} \left(\frac{1}{100} \sum_{b=1}^{100} \mathcal{J}(P_a, P_b) \right)$$

Mutatis mutandis, the corresponding $\delta_{\mathcal{J}}^{M,R}$ are also defined for pools $R \in \{R_m, R_h, R_{vh}\}$. Thus, $\delta_{\mathcal{J}}^{M,E}$ and $\delta_{\mathcal{J}}^{M,R}$ are referred as the *mean disruption* from the original to the mutated phenotypes for each pool and operator. Therefore, since 5 mutation operators and 6 pools were considered, a total of $5 \times 6 = 30$ mean disruption values were generated. These values can also be considered as a special type of diversity measure, applied

here between two different worlds (based on before-mutation/after-mutation pairs of individuals) instead of the same world (based on all possible pairs of individuals).

The final results are displayed in Figure 4.18. As the results for harsh and very harsh environments were nearly identical, only the results for the mild and very harsh environments are presented here (20 values). The 2D coordinates of each one of the 10 points represent the pairs of mean disruption quantities of a given mutation type over the evolved and random pools, i.e. ($\delta_{\mathcal{J}}^{M,E}$, $\delta_{\mathcal{J}}^{M,R}$). The main observations suggested by this chart can be summarized as follows:

- In all cases, point mutations (M_A, M_D, M_I) are far more disruptive than duplication mutations (M_R,M_L).
- In the very harsh condition, *point mutations are significantly less disruptive in evolved genomes than in random ones*, while duplication mutations are about the same (at a low level).
- In the mild condition, disruption caused by point mutations, while still lower for evolved genomes, is shifted toward significantly higher values compared to the harsh condition.

In light of these results, the starting hypothesis that evolution selects for genomes that are robust against mutations is confirmed in the very harsh and harsh environments. This is to be expected, since in these environments the plants are constrained to grow to small or moderate sizes, and the overall environmental conditions do not change significantly over the course of the simulation (at least on a global scale). At the same time, the resulting genotypes have been subject to a relatively high mutation rate for several hundreds of generations, and the risk of obliteration (by becoming too heavy in biomass or developing underground branches after a mutation) is always present, thus selective pressure encouraging robustness against mutations seems to be a logical effect. This would support the idea that selection in this model sculpts the structure of genomes in a direction that makes them more robust to the disruptive effects of mutations. For mild environments, however, the competitive pressure among individuals (in contrast to the pressure from the environment) is so high that the environmental conditions created by the neighbors change rapidly (as new individuals reach taller and taller sizes), and evolution selects for genomes that are *more* sensitive

to all types of mutations, especially point mutations, in order to keep an edge over the ever larger competitors. In summary, these results reinforce the view that the environmental conditions must remain relatively constant for mutational robustness to emerge [170].

4.4 Conclusions and discussion

This chapter combines advanced (yet minimal) models based on L-systems, genetic expression, biologically inspired mutations, and open-ended evolution in a population of interacting individuals to create a framework of evolving virtual plants. In order to characterize and understand the dynamics of diversification, data produced by the simulations was processed by simple statistical analysis based on measures of distance. While the overall model has been kept as simple as possible (through a reduced number of rules and parameters), results show the evolution of plant-like organisms capable of increasing their degree of complexity and diversification at the same time that they are competing for common resources.

The biological value and relevance of the model is twofold. On the one hand, experiments can be fully recorded at the genomic level, since organisms are represented by one-dimensional chains of symbols, which deterministically develop into plant-like structures. Because mutations act upon genotypes, the genomic dynamics can be characterized by a statistical analysis of the pool of character strings. On the other hand, this simple genotypic model maps to a highly complex phenotypic space [24] allowing the study of rich evolutionary processes oriented toward greater organism complexity and population diversity.

It is important to stress the simplicity of the model. Other models of evolving plant communities based on L-systems are generally more complicated, making use of a great number of rules and symbols to encode the various parts of the plants. In some of these works, the conclusion was that the genetic search space of D0L-systems was too limited [17]. However, this might be precisely a consequence of an overly complicated framework, since the high-dimensional parameter space that these models generate can be particularly difficult to explore and analyze. In contrast, the minimal system presented here facilitates the identification of relevant biological phenomena during the simulations, in which the great volume of generated data helps, rather than hinders,

the understanding of finer causality links among interacting individuals.

It is also important to note that the present model is not based on data or parameters derived from real-world ecosystems, such as densities of species or rates of interactions between individuals. It is an abstract, artificial-life model that is not committed to any particular plant community found in nature. Yet, general parallels with biological ecology can still be drawn. More specifically, the model shows the emergence of at least two different evolutionary strategies: (1) developing a simple morphology while yielding large offspring to escape competition and colonize new areas; (2) developing a large and branchy morphology to better compete in overcrowded environments, but at the cost of reduced offspring. These two dynamics closely resemble the ecological transition from newly colonized areas to old-growth forest. In fact, the results presented here show that very harsh environments are characterized by the first strategy and mild environments by the second, reflecting their differences in developmental costs.

In summary, the biological implications of the experimental results are the following:

1. Population diversity and individuals' complexity strongly depend on environmental conditions, confirming experimental [167] and theoretical results [95].

2. Population dynamics can be extremely varied, ranging from low-diversity populations that tend to increase in size, to very small and competitive communities of highly complex individuals.

3. Environmental conditions must remain relatively constant for mutational robustness to emerge.

4. The evolution of phenotypic complexity is based upon the dynamics of genetic mutations. As these are quick to produce larger genomes, the evolutionary exploration is biased toward more complicated structures.

5. Eventually, one individual undergoes a fast excursion toward significantly more complex morphologies, extinguishing the intermediate individuals from which it evolved. These complex structures then dominate during a certain period of time, eventually themselves disappearing due to the combined effect of high internal competition and excessive enlargement.

It is remarkable that such a simple model as the presented here induces such a variety of biological phenomena and diversity of plant shapes. Plants compete against their neighbors directly (instead of evaluating their fitness separately, like in more conventional evolutionary algorithms), and in an natural and emergent way, without the need to explicitly and arbitrarily pitch the individuals in battle, as in Sims' work [177]. Multiple levels of selection emerge: competition between organisms, adaptation to the environment, and adaptation against deleterious mutations (mutational robustness). Particularly, repeated simulations of mild environments produce a wide diversity of very elaborated forms (see Figure 4.5). In general, this model has demonstrated that diversity can emerge through evolutionary dynamics without the need of a complex and detailed modeling of the agents.

Chapter 5

Conclusions

This final chapter presents a global summary of each one of the parts of the dissertation in Section 5.1. Then, the main contributions are summarized in Section 5.2, and finally the conclusions are presented in 5.3.

5.1 Summary

In Chapter 1, the main themes of this dissertation are presented, relating them to the models presented in subsequent chapters and the global objective of studying the evolution of diversity. Biological and engineering perspectives are present in these themes. From the biological perspective, three main themes are introduced: the rising interest in agent-based computational models, the use of models of molecular motors to understand the evolution of diversity under very specific and constrained conditions, and evolutionary developmental biology (evo-devo) as a framework to understand the role of development in the evolution of diversity. From the engineering perspective, a brief introduction to evolutionary computation, highlighting the recent subfield of artificial embryogeny as an application of evo-devo to evolutionary algorithms; also, the concept of morphological computation, born in Robotics, is presented and put in context, also presenting the concept of tensegrity as enabler of morphological computation.

Chapter 2 presents a discussion on the evolutionary origin of novelty in the context of evo-devo, as the modulation at an evolutionary scale of the interplay in developmental processes between genomes and developing phenotypes. Tensegrity structures are also

introduced, explaining their properties and their roles in Biology (in this context, Appendix A provides a novel mathematical methodology to analyze some properties of tensegrity structures). A novel model of developmental process is presented and described, in which the genome plays a very minimal role, just specifying some initial conditions. The subsequent developmental process is largely self-regulated by the physical properties of the constituent parts of a tensegrity structure. The use of tensegrity structures as the substrate for the phenotype represents the key innovation, enabling the model to minimize the role of the genome during the developmental processes, as the non-linear dynamics of tensegrity structures substitute the genome as the control system of the development, in an example of morphological computation. An evolutionary algorithm is used to evolve long developmental processes with many and complex well-timed steps; as a side effect, a diversity of final morphologies (i.e., final forms produced as the final stage of developmental processes) is produced.

In Chapter 3, the molecular biology of protein motors is described. Mathematical tools from elastic network theory (applied to protein structures) are used as a formal background to build a framework to find artificial models (*templates*) of molecular protein motors through evolutionary algorithms. These templates are useful as computational models to assess hypotheses about biological molecular motors, as exemplified in Section 3.3.1. The framework follows a long-standing tradition in computational biology and artificial life to enrich agent-based models with physical dynamics in order to produce an emergent diversity of morphologies and behaviors. From this point of view, the framework can be characterized as specifying a very precise optimization task (advance as fast as possible along a straight filament), and a very precise set of rules to interact with the environment (the heuristic rules to specify the motor heads and the events in the working cycle are rigidly prespecified). Yet, in spite of this rigid setting, the coevolution of morphology and function in this framework yields a diversity of shapes and gait patterns, because the underlying model is relatively detailed and fine-grained, enabling the evolutionary algorithm to find innovative solutions to the problem of walking over a filament.

Chapter 4 presents a study of the evolutionary dynamics induced by the ecological interactions between very simple models of plants characterized by very simple abstractions of genetic, developmental and physiological processes. Plant genomes are character strings interpreted as very simplified Lindenmayer systems, and phenotypes

are generated by the classical interpretation in terms of turtle geometry. Ecological interactions between individuals within a population are driven by the locality of reproduction (immediate descendants grow close to the site of their ancestors), and are limited to the competition for light in discrete rounds of competition. The amount of collected light determines the reproductive success of the individual for the next round (thus using a biological interpretation of fitness, instead of a proper evolutionary algorithm). The reproduction is asexual, with occasional mutations. In spite of these simple premises, evolutionary dynamics emerge at the genomic, phenotypic and populational levels; in particular, a wide diversity of plant shapes evolve as they compete for light under different environments.

5.2 Contributions

This dissertation represents an interdisciplinary work, presenting a wide range of contributions, tied together under the common theme of the evolution of diversity, though there are also contributions beyond this specific subject. This section provides a summarized account of all these contributions.

In the field of evolutionary computation, a new kind of mutational operator for discrete elastic objects has been introduced in Chapter 3. This mutational operator is based of elastic perturbations to the objects, and enables a more efficient structuring of the fitness landscape for these objects, as it is able to generate coordinated and mechanically relevant changes to the whole of the structure. More broadly, from the point of view of engineering design optimization, that chapter presents a novel methodology for the design of elastic structures able to advance through a line in viscous fluids.

Additionally, also in evolutionary computation, but more specifically in the subfield of artificial embryogeny, a new indirect encoding for the evolutionary search of tensegrity structures has been introduced in Chapter 2. This indirect encoding is notable for using an extremely concise form of genotype; the morphology of the phenotype is almost entirely determined by an artificial developmental process.

The previous contribution can also be interpreted in the context of evolutionary developmental biology, related to the nature of genetic control in developmental processes. Specifically, it represents a novel model for developmental processes with

minimal genetic control, providing a conceptual example of complex and convoluted developmental processes which are mostly self-regulated by the properties of the structures under development, instead of being tightly regulated by genetic control. Since these morphologies emerge by evolution while selecting for complex developmental processes, this example demonstrates that the evolution of a diversity of morphologies can be a side-effect of the complexification of developmental processes, regardless of the complexification of the genetic regulation of the process.

To enable the self-regulation of developmental processes, the previously mentioned developing structures must have a substrate with rich, non-linear dynamical properties. The concept of tensegrity provides this substrate. However, precisely because these properties, tensegrity structures are notoriously difficult to design and analyze. The studied developmental process suggest a new way to generate new engineering designs of tensegrity structures. Also, while not directly related to the theme of evolution of diversity, mathematical tools for the analysis of tensegrity structures represent also a contribution of this thesis; specifically, to analyze the self-stress of tensegrity structures (Appendix A), which in turn critically influences the dynamical properties of these structures.

In the domain of molecular biology, the main contribution is the framework for the evolution of molecular motor templates; which are simplified agent-based models of protein motors walking over filaments (Chapter 3). Apart from the shape, which is more or less freely adaptable, this framework specifies a very restrictive way to determine the characteristics of the molecular motor templates; and a very restricted environment to test them. The results show that diverse morphologies and gait patterns can be evolved in spite of these severe constraints, because of the detailed simulation of the molecular motors, enabling the coevolution of the structure and its control system.

Finally, regarding the analysis of the evolutionary dynamics of diversity, a novel agent-based model of evolution is presented in Chapter 4, whose main defining characteristic is the emergence of open-ended evolutionary dynamics (in the sense that the agents have no explicit fitness function beyond surviving) in an extremely simplified model, without a careful fine-tuning of the ecological parameters. This example demonstrates that, even under extremely simplified models of ecological interaction, evolutionary dynamics can emerge, driving the evolution of complex and diverse communities of agents with rich and interesting dynamics at the population

level. In the framework of this simple model, a series of techniques are provided to measure diversity and other interesting parameters of the simulated populations.

5.3 Conclusions and discussion

The main purpose of this dissertation has been to study the evolution of diversity of form (morphology and structure) and function (behavior and development) in three agent-based models of biological systems from a biological perspective, but also with a strong focus on evolutionary computation. These models have been discussed from several different of points of view: evolutionary developmental biology, molecular biology and evolutionary dynamics in the context of Biology, and indirect encoding in evolutionary computation and engineering design; all of them complementing the unifying theme of the evolution of diversity. Consequently, this dissertation can be unequivocally considered as interdisciplinary. This final section will review the conclusions derived from the findings presented in this dissertation.

In a typical dissertation, a research question is posed, and then just one model (or at most a family of more or less closely related models) is introduced, described, simulated, analyzed and the results put in context, throughout several chapters. In contrast, the structure of this dissertation represents a significant departure from that usual structure: each one of the previous three chapters have presented a model (significantly different from the others) in a mostly self-contained format.

The reason for this unusual structure is the following: the main purpose or research question of this dissertation is to study from a computational perspective the origin of diversity of morphology and function in biological systems. However, this purpose is exceedingly vast in scope; a compromise must be reached; in fact, this dissertation is not, and cannot provide, an exhaustive account of all the possible ramifications related to this broad research question. The usual way to solve this issue is to formulate just one model to address a very precise and restricted version of the originally broad research question (giving rise to the usual dissertation structure discussed in the previous paragraph). However, as a side effect, this means a considerable restriction in the scope of the results. An alternative may be to define a model with multiple layers, to study the origin of diversity at multiple levels within a single model. However, this hypothetical model would be difficult to manage and analyze. Instead, the strategy

chosen in this dissertation is to use several simple models, each one suited to study a different aspect of the origin of diversity; in this way, a wider range of relevant topics can be analyzed and discussed.

Specifically, each model is related to an aspect of the original research question about the evolution of diversity:

The role of developmental processes.
In the context of evo-devo, developmental processes are postulated to play a key role in the evolution of diversity. This role is often interpreted in terms of the genetic control of this process: a complex genetic regulatory network orchestrates a complex process of development from zygote to adult phenotype, and thus the evolution of diverse morphologies is the result of the evolution of complexity in genetic regulatory networks [14]. In this setting, we pose the following question: is the evolution of complex genetic regulatory networks a requirement to induce diverse morphologies through developmental processes? However, in Chapter 2, it is presented an abstract model of self-regulated development with no genetic regulation, but just genetic modulation of the initial conditions of the process. Yet, diverse morphologies evolve by running several times a simple evolutionary algorithm, set to heuristically find instances of long and complex developmental processes. In this setting, the regulatory character of the genome is taken over by the dynamical properties of tensegrity structures; therefore, the evolution of diverse morphologies is directly the result of the complexification of the developmental processes, disregarding the need for a complex genetic regulatory network.

The interplay between the body and its control system.
In the context of computational biology, evolutionary robotics and artificial life, the coevolution of the body and the control system of the agents can produce diverse and coadapted morphologies and behaviors [58]. Previous work in this area has traditionally used complex control systems and, in some cases, developmental processes as complex genotype-to-phenotype mappings [103, 158, 178]. However, the concept of morphological computation [154] suggests that this coevolution can be effective even if the control system is extremely simple and mostly implicit in the morphology and structure of the agent, as in the case of Lobo's path followers [120]. Chapter 4 presents an example in this vein, but simpler: no developmental process but a direct

(albeit highly heuristic) genotype-to-phenotype mapping, a simple control system (the reactive working cycle) and a very simple task (advance as fast as possible) in a simple environment (a straight filament in an empty 3D space). In spite of this simplicity, diverse morphologies and behaviors (gait patterns) evolve. Globally, this model constitutes a fairly minimal example of body-control system coevolution through morphological computation.

The role of evolutionary dynamics.

The scientific literature on the evolution of diversity is rich and spans many scientific disciplines. In the context of agent-based simulation, many different models have been proposed over the years. Many of them feature complex agents in a rich and complex environment, with a detailed modeling of their characteristics, as Sims' [178]. Others try to model many relevant details to build complex ecological communities [182, 220]. In these cases, the resulting diversity in the simulations can be attributed to many different features of the models. Other agent-based models feature very simple agents, in order to enable an exhaustive mathematical analysis of the model [26, 221], at the cost of a very high-level modeling of the characteristics of the agents. Our contribution in this area is the model presented in Chapter 4, whose defining characteristics are a bottom-up approach and the simplicity of the modeling at all levels: in the genotype, the phenotype and its mapping with the genotype, in the determination of biological fitness and the asexual reproductive model, and in the ecological interactions; however, on the whole, the approach is not too abstract. Because of this simplicity at all levels, the resulting evolution of diverse morphologies can be attributed to the evolutionary dynamics induced by the ecological interactions between the agents, rather than to any other aspect of the model.

UNIVERSIDAD
DE MÁLAGA

Appendix A

Atom-based analysis of tensegrity structures

In Chapter 2, tensegrity structures have been used to produce complex developmental processes, giving rise to a diversity of morphologies. This appendix provides a supplementary contribution on the mathematical analysis of tensegrity structures.

Up to date, formal analysis and design of tensegrity structures has been conducted using mathematical tools like potential energy functions [32], and a variety of form-finding methods [138, 195]. However, in [78] a new approach was developed: tensegrity structures were shown to be decomposable into a sum of elemental tensegrity units, called atoms (see Figure A.3). That result was applied to develop a new form-finding method, finding tensegrity configurations for a given abstract graph, provided that some conditions hold (see [78] for the complete definition). This works the other way, too: it is possible to assemble new tensegrity structures by summing tensegrity atoms.

In general, tensegrity structures can be decomposed in many possible ways, and there is currently no way to define a *canonical* decomposition. However, for any tensegrity structure, there is at least a class of distinguished decompositions, whose characteristic is to use a minimal amount of atoms. This appendix presents a mathematical analysis on the length of decompositions of tensegrity structures. This study represents the initial phase of a research line on a new way to analyze and generate tensegrity structures based on atomic decomposition.

The mathematical results presented in this appendix have been published in [52].

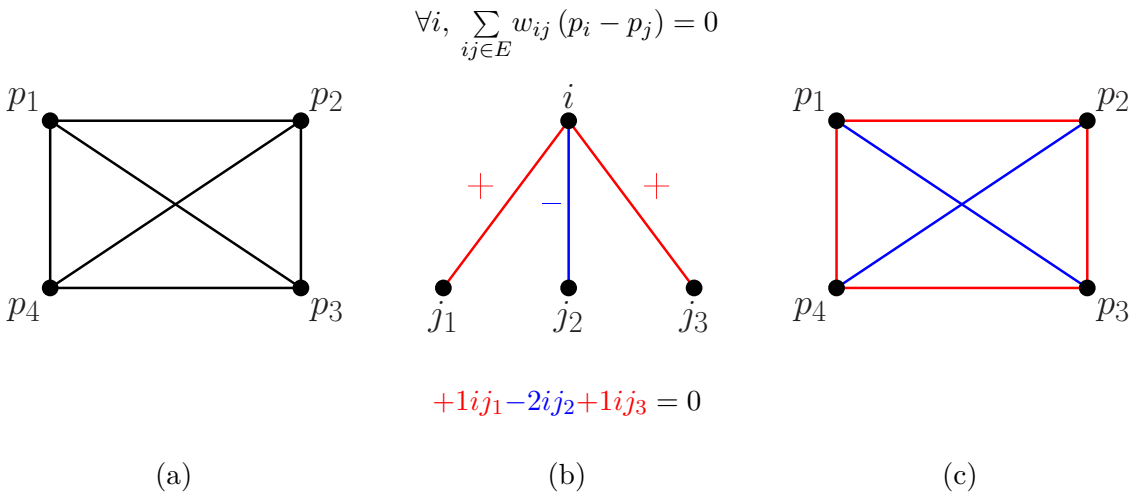


Figure A.1: ***Basic concepts.*** **(a) A framework is an embedding of an abstract graph (in this case K_4) with straight edges in some space $\mathbb{R}^d$ (in this case, $\mathbb{R}^2$). (b) A self-stress is an assignment of real values (stresses) to the edges of a framework such that the sum of incident vectors scaled by their values is zero in every edge. Here, the convention is to use blue to denote a negative stress and red to denote a positive stress. (c) A tensegrity structure is a framework with a self-stress.**

A.1 Theoretical definitions

Before starting the discussion, some concepts and results (formulated in [78]) will be briefly presented, since they are needed to understand the rest of the section:

- A finite *point configuration* $P := \{p_1, ..., p_n\}$ in $\mathbb{R}^d$ is in *general position* if no $d + 1$ points lie on the same hyperplane. More restrictively, if the points are algebraically independent, the position is *generic*.

- A *framework* $G(P)$ in $\mathbb{R}^d$ is an embedding of the abstract graph $G = (V, E)$ on a finite point configuration P in $\mathbb{R}^d$ in general position, with straight edges. See Figure A.1.a.

- A *self-stress* w on a framework is an assignment of scalars w_{ij} (called tensions) to its edges, such that for each vertex i, the scaled sum of incident vectors $p_i - p_j$ is zero (see Figure A.1.b):

$$\forall i, \quad \sum_{ij\in E} w_{ij}\,(p_i - p_j) = 0$$

  Observe that self-stresses form a vector space. In relation to Chapter 2, the stress is equivalent to the force exerted by the elastic link, divided by the length of the

link (force density in the literature [138, 195]).

- If a framework $G(P)$ has n vertices and e edges, its associated *rigidity matrix* $R(P)$ has e rows and nd columns, such that:

  – There is a row per edge ij of the framework, with $i < j$ and in lexicographic order.

  – Each block of d columns is associated to a vertex p_i, and it contains zeros except for each row corresponding to an incident edge ij, where it contains the d coordinates $p_i - p_j$.

  Observe that if w is a self-stress on $G(P)$, then $w \cdot R(P) = 0$.

- A framework with a self-stress non-null on every edge is called a *tensegrity structure*, denoted as $G(P, w)$. See Figure A.1.c.

- An *atom* A in dimension d is a complete graph K_{d+2} embedded in $\mathbb{R}^d$. If the embedding is in general position, its space of self-stresses has dimension 1. Otherwise, it only admits a null stress, that is to say, $w_{ij} = 0$, $\forall i, j$. See Figure A.2.

- An *atomic decomposition* of a tensegrity structure $G(P, w)$ is a finite set of atoms (each atom corresponding to a set of $d+2$ points in P) such that the sum of their self-stresses is w. Note that the atoms in the decomposition may have edges ij not present in G, which cancel out to $w_{ij} = 0$ when the atom stresses are added up. In general, decompositions are not unique. The *length* of the decomposition is the cardinality of the set of atoms. See Figure A.3.

One of the main results in [78] is the development of an algorithm to generate atomic decompositions for tensegrity structures. The algorithm can be applied to a tensegrity structure $G(P, w)$ (Theorem 3.2 in [78]), or to an abstract graph $G = (V, E)$ (Algorithm 3.4 in [78]). These variants will be referred as the *geometric* and *combinatorial algorithms*, respectively. Below, the combinatorial version of the algorithm is reproduced (see Figure A.4 for a graphical example):

Algorithm A.1. (Adapted from Algorithm 3.4 in [78]) *Atomic combinatorial decomposition*

INPUT: abstract graph $G = (V, E)$ and dimension d.

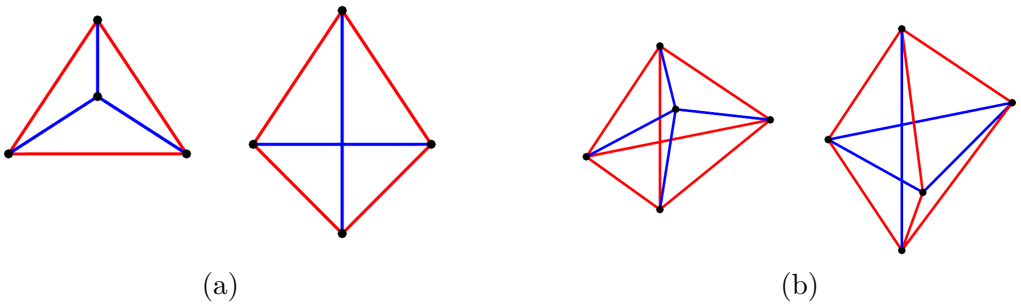
OUTPUT: (L, M, F), where L is a list of "atoms" (subsets of $(d + 2)$ elements of V), M is a list containing the number of edges added for each atom in L, and F is a list of intermediate graphs.

1. Initialize $L = \varnothing$, $M = \varnothing$, $F = [G]$.
2. While E is not empty, choose a vertex $a \in V$ and:
   - 2.1 If a has degree $d+1$, let $a_0, \ldots, a_d$ be its neighbors. Remove the edges aa_i from E. Let E' be the set of all the edges a_ia_j between the neighbors that were not in E.
   - 2.2 If a has degree at least $d+2$, choose $d+1$ neighbors $a_0, \ldots, a_d$ of a. Remove the edge aa_0 from E. Let E' be the set of all the edges a_ia_j between the neighbors that were not in E.
   - 2.3 If a has degree $\leq d$, remove its incident edges from E.

   In cases 2.1 and 2.2, also add the edges from E' to E, insert the atom $\{a, a_0, \ldots, a_d\}$ to the list L, and $|E'|$ to the list M. In any case, also update the graph with the new set of edges and removing unconnected vertices, and add it to the list F.
3. Return (L, M, F).

It is important to note that both the geometric and the combinatorial algorithms are non-deterministic: different decompositions can be obtained by making different sets of choices at several points in the algorithms. In the combinatorial algorithm, the resulting decomposition of a graph G represents a set of constraints between the positions of vertices and/or self-stresses of edges of tensegrity structures with underlying graph G (more details in [78]).

The acronym SAL will be used to refer to the smallest atomic length, i.e., the smallest size of the list L over all the possible outputs of Algorithm A.1. Studying the SAL is interesting because it can be seen as a tool to analyze a tensegrity structure: it represents the minimal way to interlock a set of one-dimensional atoms to generate the structure.



(a) (b)

Figure A.2: *Atoms in* $\mathbb{R}^2$ *and* $\mathbb{R}^3$. (a) All atoms in $\mathbb{R}^2$ share the geometric configuration (either a triangle with an interior point or a quadrilateral) and self-stress scheme (pattern of stress signs) of one of these two atoms (or can be obtained by flipping the stress signs in any of these two examples, that is to say, switching colors from blue to red and vice versa). (b) The same applies in $\mathbb{R}^3$ for these other two atoms.

Remark A.2. While the combinatorial algorithm formally operates in the domain of abstract graphs, it implicitly assumes that the graph is embedded in some unspecified tensegrity structure. In this context, it is significant to note that some valid tensegrity structures $G(P, w)$ might have a SAL shorter than the combinatorial SAL for G. Consider the decomposition shown in Figure A.5: if a general point configuration P is set and a self-stress w which is the sum of atoms $1, \ldots, 4$ is defined, a tensegrity structure $G(P, w)$ can be obtained, whose underlying graph G is the same as the one depicted in the figure. Therefore, while the geometric SAL of this specifically constructed $G(P, w)$ is 4, the combinatorial SAL for G is 6, because the combinatorial algorithm tacitly assumes that G is implicitly embedded in a tensegrity structure $G(P, w)$ as generic as possible, both in terms of the position P and of the self-stress w. In spite of these considerations, the combinatorial problem can be reduced to the geometric one, as it will be seen in Section A.2.1.

Remark A.3. In the geometric case, it is important to note that tensegrity structures in non-general position can still be decomposed into sums of atoms, under the condition that each one of the atoms in the decomposition is in general position. This might lead in some cases to minimal geometric decompositions longer than the minimal combinatorial decompositions for the underlying abstract graph, since the combinatorial decomposition may use atoms which in the geometric case are not in general position, thus not usable. On the other hand, in the geometric case, the step 2.3 may sometimes remove more than one edge, rendering the combinatorial analysis of these cases more difficult.

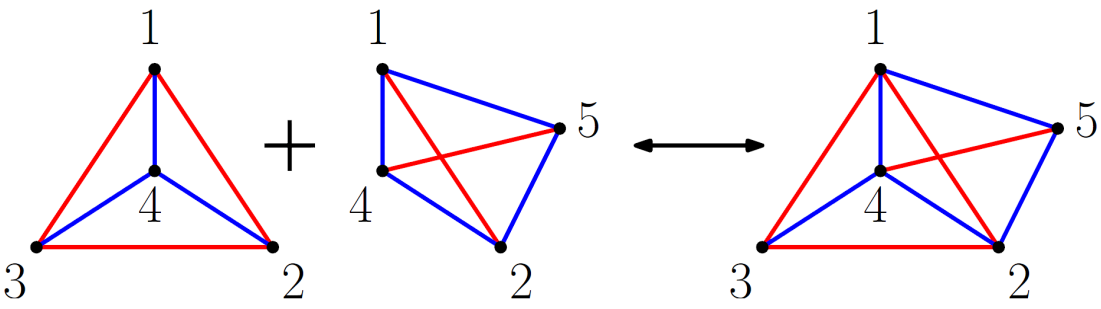


Figure A.3: ***Atomic decomposition/assembly.*** **All tensegrity structures can be decomposed into a sum of tensegrity atoms [78], such that the self-stress of the overall structure is the sum of the self-stresses of the atoms. The atoms may have edges not present in the tensegrity structure, such that the sum of stresses over all atoms for those edges is zero (this does not happen in the simple example depicted in this figure). Equivalently, a sum of atoms with shared edges results in the assembly of a tensegrity structure.**

From rigidity theory [74], it is known that the set W of all possible self-stresses of a framework $G(P)$ is, in fact, the left kernel of the matrix $R(P)$. Similarly, the space of infinitesimal motions is the right kernel of $R(P)$, whose dimension (or number of degrees of freedom) is 0 if $G(P)$ is rigid. The following proposition relates the dimensions of both spaces:

Proposition A.4. (Theorem 2.4.1 in [74]) *Let $G(P)$ be a framework in general position P in dimension d with $G = (V, E)$, $\left|W_{G(P)}\right|$ the dimension of the self-stress space and* $\mathrm{df}(G(P))$ *the number of degrees of freedom of $G(P)$. Then:*

$$\mathrm{df}(G(P)) = \begin{cases} \left|W_{G(P)}\right| - \binom{d+1}{2} + d \cdot |V| - |E|\,, & \text{if } |V| \geq d \\ \binom{|V|}{2} - |E|\,, & \text{if } |V| \leq d+1 \end{cases} \tag{A.1}$$

A.2 The structure of the space of self-stresses

For all practical purposes, only generically rigid graphs will be considered, that is to say, graphs which are rigid in any generic position. Non-generically rigid graphs can be embedded in tensegrity structures only in very degenerated positions. The notation $|W_G|$ will be used to denote the dimension of $W_{G(P)}$ if P is a generic position, since it is constant for every generic position.

For generically rigid graphs, the space of self-stresses of the intermediate graphs (list

F) in the decomposition algorithm changes according to the number of intermediate edges inserted along the decomposition (list M), and these numbers are directly related to the dimension of the space of self-stresses. To show this, we start with the following proposition:

Proposition A.5. *If a graph $G = (V, E)$ is generically rigid in $\mathbb{R}^d$, then every intermediate graph in any decomposition in $\mathbb{R}^d$ will be generically rigid.*

Proof. Suppose that, for a given decomposition, the list of intermediate graphs is $F = [\ldots, G_i, G_{i+1} \ldots,]$. The proposition can be proven by showing that if an intermediate graph G_i is generically rigid, then G_{i+1} also is. Doing it by cases:

- If the transition is done by step 2.1, let E' be the set of the added edges from G_i to G_{i+1}. If G_i is generically rigid, consider the rigidity of the graph G'_{i+1}, induced from G_{i+1} by removing the edges in E'. If G'_{i+1} is not generically rigid, then the edges incident to a in G_i must remove all degrees of freedom from G'_{i+1}, which allows the movement of some of its neighbors relative to others. In G_{i+1}, all distances between these neighbors are fixed by the edges added in E', so G_{i+1} is also generically rigid.

- If the transition is done by step 2.2, let aa_1 be the only edge removed from G_i to G_{i+1}. In G_{i+1}, the subgraph induced by the vertices $a, a_1, \ldots, a_d$ is a clique K_{d+2} minus an edge. A clique K_{d+2} is a generic rigidity circuit in dimension d (Theorem 3.11.9.a in [74]). If an edge is removed from a rigidity circuit, the resulting subgraph is still generically rigid (in chapter 3 in [74]). Therefore, the relative positions of a and a_1 are fixed in G_{i+1}, and hence this is also generically rigid.

- If the transition is done by step 2.3, at most d edges have been removed from G_i to G_{i+1}. Reasoning by the number of vertices in G_i:

  - if $|V_i| > d$, the vertex a must have exactly d incident edges for G_i to be rigid, and any self-stress must be always zero in these edges. Therefore, the equilibrium at other vertices in G_i is independent from these edges, and $|W_G|$ remains the same in G_i and G_{i+1}. By applying the first case of Equation A.1, $\mathrm{df}(G_{i+1}) = 0$, so G_{i+1} is also generically rigid.

– if $|V_i| \leq d$, G_i must be a complete graph in order to be generically rigid by the second case of Equation A.1, so G_{i+1} will also be a complete graph, hence also generically rigid.

□

Therefore, generic rigidity is a property conserved through all intermediate graphs in the decomposition algorithm. To take advantage of this, the *Laman bound* is defined as the dimension of the self-stress space in generic position:

Definition A.6. The *Laman bound* of a generically rigid graph $G = (V, E)$ in dimension d is defined as:

$$B = \begin{pmatrix} d+1 \\ 2 \end{pmatrix} - d \cdot |V| + |E|$$

The Laman bound is modified by the decomposition algorithm in a very specific way:

Proposition A.7. *Let G_i, G_{i+1} be two successive intermediate graphs in a combinatorial atomic decomposition in dimension d, with every vertex in G_i having degree at least d. Let B_i and B_{i+1} be their Laman bounds, respectively, and let e_i be the number of edges added from G_i to G_{i+1}. Then, $B_{i+1} = B_i + e_i - 1$.*

Proof. By the definition of Laman bound, it is easy to see that the equality holds in any case (steps 2.1, 2.2 and 2.3). □

Remark A.8. Note, as explained in Remark A.3, that in the geometric case, the step 2.2 in Algorithm A.1 sometimes removes more than one edge. Proposition Proposition A.7 (and consequently Proposition A.9) does not hold in these cases. However, the propositions still hold in the geometric cases induced as general instances of combinatorial cases (see Remark A.11).

If a graph is generically rigid and has at least d vertices, then $B = |W_G|$ by Proposition A.4. Hence, by Proposition A.5, each atom considered in the decomposition algorithm changes the dimension of the space of self-stresses of the intermediate graph, according to the number of edges added to the graph. Atoms adding no edges represent an independent dimension in W_G, atoms adding one edge must be tuned to cancel out

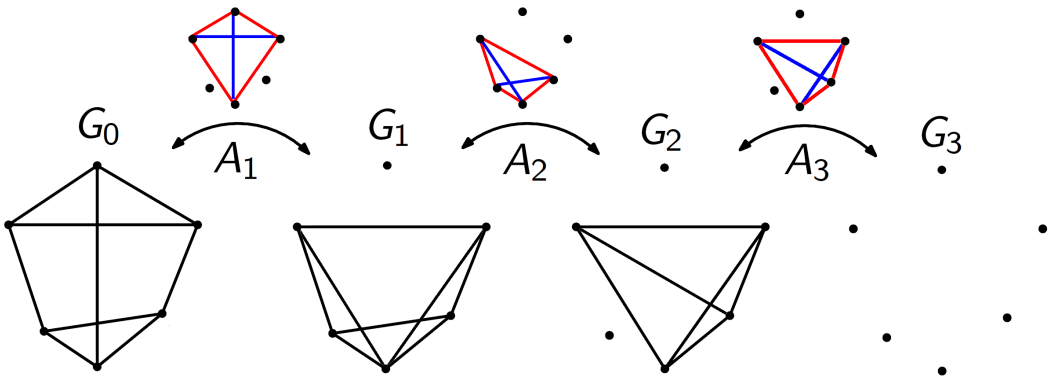


Figure A.4: ***Atomic decomposition algorithm.*** **A graphical example of the Algorithm A.1, used to decompose an abstract graph into a sum of symbolic atoms, which can be used to find a suitable geometry to define a tensegrity structure using the abstract graph, under some circumstances (see [78]). Roughly speaking, the algorithm works by symbolically subtracting atoms from the abstract graph. In each substraction, some edges are added and some others are removed. After several steps, all edges are substracted, and the decomposition ends.**

the stress in that edge, so they do not affect the dimension of W_G, and atoms adding two or more edges represent an interlock between several other atoms. These considerations are summarized in the following result:

Proposition A.9. *Let G be a generically rigid graph and B its Laman bound, and consider the lists L (of atoms) and $M = \left[e_1, \ldots, e_{|L|}\right]$ (of amounts of added edges) produced by an atomic decomposition of G. Then, the number of atoms in the decomposition is the Laman bound plus the total amount of (possibly repeated) edges added during the decomposition:*

$$|L| = B + \sum_{i=1}^{|L|} e_i \tag{A.2}$$

Proof. As G is generically rigid, Proposition A.7 can be applied to every intermediate graph generated by the algorithm, so an atom introducing e edges changes B by $e - 1$. Equation A.2 is implied by combining Proposition A.7 with the fact that the Laman bound must change from $B_0 = B$ to $B_{|L|} = 0$. □

The previous proposition provides a characterization of the SAL: it corresponds to the decompositions introducing the fewest edges in the intermediate steps. This holds even if the graph is not generically rigid.

Definition A.10. A decomposition is defined as *atomistic* it no edges are added in any intermediate step. By extension, a graph is *atomistic* if it admits an atomistic decomposition.

It is clear that, in an atomistic decomposition, the self-stresses of the atoms form a basis for W_G. It is also evident that if a graph $G = (V, E)$ is decomposed and E' is the set of all edges added in the intermediate steps, then the graph $G' = (V, E \cup E')$ is atomistic. Also, chordal graphs that are generically rigid are also atomistic, and so are cliques K_n, whose minimal decomposition length is $\binom{n-d}{2}$.

A.2.1 An algebraic characterization

Given a tensegrity structure $G(P, w)$, the geometric decomposition algorithm (check the combinatorial version in Algorithm A.1 or the geometric version in [78] admits an algebraic reformulation.

Remark A.11. Note that in this section the domain is being shifted from combinatorics to geometry. As described in Remark A.2, some valid tensegrity structures might have a SAL shorter than the combinatorial SAL for G. If a generically rigid graph $G = (V, E)$ can be embedded in a tensegrity structure $G(P, w)$, the way to find P and w to construct a tensegrity structure $G(P, w)$ depends on the value of $|W_G|$:

- If $|W_G| > 0$, and no edge in E must have a null self-stress for a generic position P, any generic P will be suited to construct a $G(P, w)$. Given a generic P (uniformly random configurations are almost certainly generic), a suitable self-stress can be generated almost certainly by taking a basis $\{w_1, \ldots, w_k\}$ for $W_{G(P)}$, and finding a linear combination $w = \sum a_i w_i$ where coefficients a_i are drawn from a uniform distribution $U(0, 1)$.

- If $|W_G| = 0$ and a tensegrity structure $G(P, w)$, can be defined, then the configuration P must be non-generic. In some cases, it can be found relatively easily (for example, for graphs with edge-inserting decompositions, as described in [78]).

Definition A.12. Let P be a point configuration in dimension d. Q_m is defined as the collection of all sets with exactly m vertices from P. Also, S is defined as the *atomic self-stresses matrix* as follows:

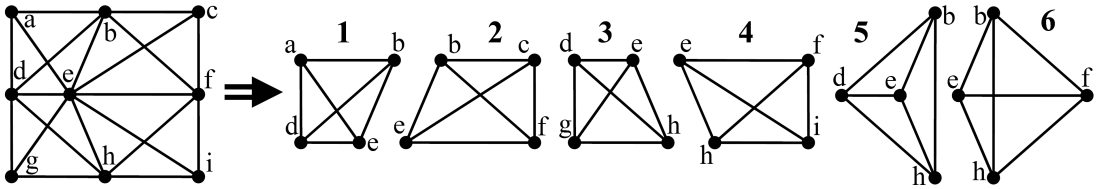


Figure A.5: An abstract graph with one example of a combinatorial atomic decomposition with six atoms. Tensegrity structures with the same underlying abstract graph but only four atoms (from 1 to 4) can be formulated, showing that the minimal combinatorial decomposition is an upper bound for geometric instances of the problem.

- There is a row for each possible set $\{p_i, p_j\} \in Q_2$. The rows are ordered by the indices i, j, with $i < j$ and in lexicographic order.

- There is a column for each possible set $\{p_1, \ldots, p_{d+2}\} \in Q_{d+2}$. As these $d+2$ points represent an atom A, let w_A be an unitary, non-null self-stress of A. Then, the column associated to A contains, for each row corresponding to a pair of points p_i, p_j of A, the self-stress assigned by w_A to the edge $p_i p_j$. In every other row, the value is 0. As with the rows, the columns are ordered in lexicographic order by the indices of the points.

Definition A.13. For a given tensegrity structure $G(P, w)$, the self-stress w can be represented as a column vector $\overline{w}$ of length $\binom{n}{d+2}$, where there is a position for each possible pair p_i, p_j of vertices of the framework, with $i < j$ and in lexicographic order, and the value in $\overline{w}$ for the position corresponding to the pair p_i, p_j is w_{ij} if there is an edge between them, and 0 otherwise. See Figure A.6.

It is easy to see that if w is a valid self-stress of $G(P)$, then every solution x to the underdetermined system of linear equations $S \cdot x = \overline{w}$ represents a linear combination of atoms which can be used to construct a tensegrity structure $G(P, w)$. Furthermore, let $\|x\|_0$ be number of non-zero elements in x. The problem of finding the SAL can be recast as finding a sparsest solution (with minimal $\|x\|_0$) to $S \cdot x = \overline{w}$. Since by Remark A.11 the combinatorial algorithm can be restated in these terms for most graphs, this leads to some interesting properties for S:

Corollary A.14. *If a configuration P in n vertices is in general position, the rank of the corresponding atomic self-stresses matrix S is $\binom{n-d}{2}$.*

Proof. K_n is an atomistic graph, so any minimal decomposition corresponds to a basis

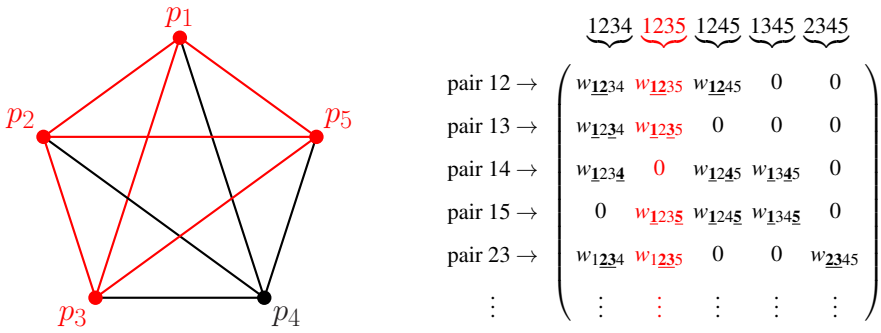


Figure A.6: ***Atomic self-stresses matrix*** **(see Definition A.12). For a given framework $G(P)$ in dimension d (*left*), the atomic self-stresses matrix S (*right*) has one row for each possible pair of points in the framework (not only edges, but any all pairs), and one column for each possible set of $d+2$ points (representing all possible atoms in the framework). For example, in $d=2$, each column represents the self-stress of the corresponding atom: for column $ijkl$, there are non-zero values on rows corresponding to the pairs of points for that column: ij, ik, il, jk, jl, kl (even if these pairs are not edges in $G(P)$), and zeroes in all other rows. These non-zero values are the normalized values of self-stress of a framework with underlying graph K_{d+2} and the points $ijkl$ corresponding to that column. In this way, the self-stresses matrix S compiles all possible atomic self-stresses for $G(P)$.**

for W_{K_n}. As the columns of the matrix S are the self-stresses for all possible atoms in K_n, its rank must be exactly the size of this basis, i.e., $\binom{n-d}{2}$. □

It is interesting to consider how Proposition A.9 translates into this algebraic setting. First, some definitions are needed:

Definition A.15. Let $G(P)$ be a framework, and Q_2, Q_{d+2} as in Definition A.12. Given a set of pairs $D \subseteq Q_2$, its collection of *associated atoms* $A_D \subseteq Q_{d+2}$ is defined as the collection of all sets of $d+2$ points including some member of D, i.e., $A_D = \{\{\ldots, p_i, \ldots, p_j, \ldots\} \in Q_{d+2} \mid \{p_i, p_j\} \in D\}$.

Definition A.16. Let $G(P, w)$ be a tensegrity structure, with $G = (V, E)$ its underlying graph, S its atomic self-stresses matrix and $\overline{w}$ the column vector associated to w, as in Definition A.12. Let $E' \subseteq Q_2$ be a set of edges containing E. Let $G' = (V, E')$ be a graph formed by adding the edges in $E' - E$ to G. Let $A_{E'} \subseteq Q_{d+2}$ be the set of atoms associated to E'. The rows (resp. columns) of S correspond one-to-one to sets in Q_2 (resp. Q_{d+2}). The *core* of S (resp. $\overline{w}$) with respect to E', denoted $S_{E'}$ (resp. $\overline{w}_{E'}$), is defined as a the submatrix of S (resp. $\overline{w}$) induced by E', whose rows correspond to E' and whose columns correspond to $A_{E'}$. See Figure A.7.

Now, it is worth to note that any solution $x_{E'}$ to $S_{E'} \cdot x_{E'} = \overline{w}_{E'}$ induces a solution x (generated by padding $x_{E'}$ with zeros for the rows in S but not in $S_{E'}$) to $S \cdot x = \overline{w}$

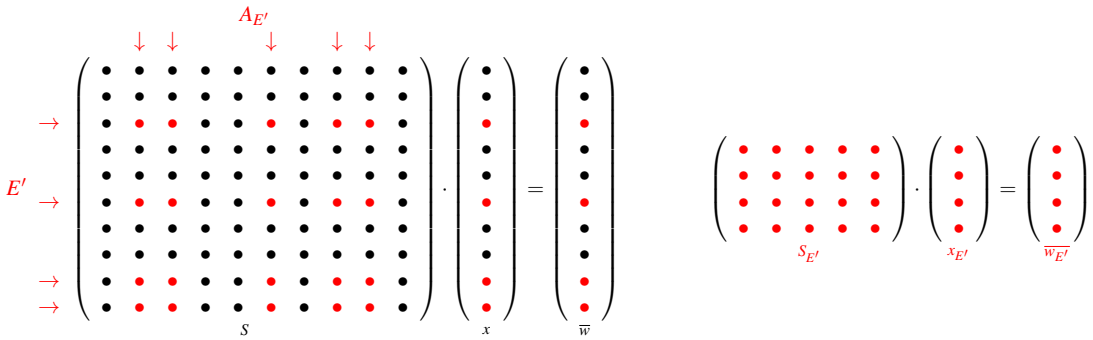


Figure A.7: ***Core of a system of linear equations*** **(see Definition A.16). For a given tensegrity $G(P,w)$, any atomic decomposition can be characterized as a solution x to the linear system of equations $S \cdot x = \overline{w}$, where S is the atomic self-stresses matrix (see Figure A.6). Any set of pairs of points $E' \supseteq E$ (a superset of the set of edges of the framework) induces a set of atoms $A_{E'}$ such that all atoms in $A_{E'}$ have at least an edge in E'. Then, the restricted matrix $S_{E'}$ can be defined as the submatrix of S with the rows corresponding to E' and the columns corresponding to $A_{E'}$. The restricted vectors $x_{E'}$ and $\overline{w_{E'}}$ are subvectors of x and $\overline{w}$ defined in a similar way. Then, the subsystem $S_{E'} \cdot x_{E'} = \overline{w_{E'}}$, called the *core* of $S \cdot x = \overline{w}$, is solvable if and only if $A_{E'}$ is a superset of the atoms of a decomposition (see Proposition A.17).**

(in fact, $\|x_{E'}\|_0 = \|x\|_0$). Without loss of generality, all pairs in E' can be supposed to be included in some element of $R' \subseteq Q_{d+2}$, as pairs not fulfilling this requirement are associated to rows whose elements are all zero in $[S_{E'} \,|\, \overline{w}_{E'}]$. The following insight relates the structure of a combinatorial decomposition to the solution to the linear system of equations $S \cdot x = \overline{w}$:

Proposition A.17. *Let $G(P,w)$ be a tensegrity structure with graph $G = (V,E)$, such that Proposition A.9 holds (as in Remark A.11). Then, the SAL corresponds to a minimal set of edges E_a such that, for $E' = E \cup E_a$, it holds that* $\operatorname{rank}(S_{E'}) = \operatorname{rank}([S_{E'} \,|\, \overline{w}_{E'}])$*, where $S_{E'}$ and $\overline{w}_{E'}$ are the cores of S and $\overline{w}$. See Figure A.7.*

Proof. Let L be any geometric decomposition of $G(P,w)$, let E_a be the set of edges added during decomposition L, and let $E' = E \cup E_a$. L can be recast as a solution $x_{E'}$ to $S_{E'} \cdot x_{E'} = \overline{w}_{E'}$. Then, Proposition A.9 means that $\|x_{E'}\|_0$ is the sum of the dimension of the space of self-stresses of $G(P)$ and the number of rows in $S_{E'}$ corresponding to edges in E_a. Therefore, a SAL will correspond to a minimal set of edges E_a such that the system $S_{E'} \cdot x_{E'} = \overline{w}_{E'}$ is solvable. The solvability condition can be restated in terms of the ranks of the matrix and the augmented matrix, $\operatorname{rank}(S_{E'}) = \operatorname{rank}([S_{E'} \,|\, \overline{w}_{E'}])$. □

As a result, atomistic graphs can be characterized in algebraic terms: if a suitable tensegrity structure $G(P, w)$ is defined on a graph $G = (V, E)$ (as in Remark A.11), then it is atomistic if and only if $\operatorname{rank}(S_E) = \operatorname{rank}([S_E \,|\, \overline{w}_E])$.

A.2.2 Computational complexity

The problem of computing the SAL seems to be NP-complete, but the question remains open. As the combinatorial atomic decomposition is defined through an algorithm with some non-deterministic steps, a choice (selected atom) taken in a step affects in highly convoluted ways to the choices available in all subsequent steps. Because of this, it is very difficult to reason directly about the computational complexity of finding the SAL. While several NP-complete problems seem to be very related to it, no obvious ways to reduce them to it have been devised:

The NP-complete **fixed clique covering problem** (FCC) [33, 105] is (for any given n) the problem of finding the minimal amount of copies of K_n needed to cover all the edges of a graph G, where these copies are not required to be induced subgraphs of G. FCC can be deemed as a non-trivial lower bound on the SAL in dimension d, but, unfortunately, the FCC tends to grossly underestimate the SAL.

The NP-complete **minimum fill-in** (MFI) [224] of a graph G is the minimal amount of edges whose addition makes the graph chordal. If a graph can be instantiated as a tensegrity structure in dimension d, its MFI plus its Laman bound can be regarded as an upper bound on its SAL, as the resulting chordal supergraph will necessarily have at least as many edges as a minimal atomistic supergraph of G (in fact the chordal supergraph will induce an atomic decomposition of G, although not necessarily minimal). As chordal graphs are also atomistic, the MFI induces a SAL in many cases. These facts hint that, even if the problem is not NP-complete, it must be relatively hard to solve.

By Definition A.12, finding the SAL is equivalent to finding a sparsest solution to $S \cdot x = \overline{w}$. While this algebraic formulation of the problem may seem more tractable than the combinatorial one, the highly structured nature of the matrix S implies severe restrictions on the ways to reduce any known NP-complete problem to this one. Interestingly, the superproblem of finding a **sparsest solution to a general linear system of equations** $A \cdot x = b$ (over the real numbers) has been thoroughly studied [6, 22, 23, 40, 61, 64, 75, 111, 116, 125, 201], as it is relevant in many

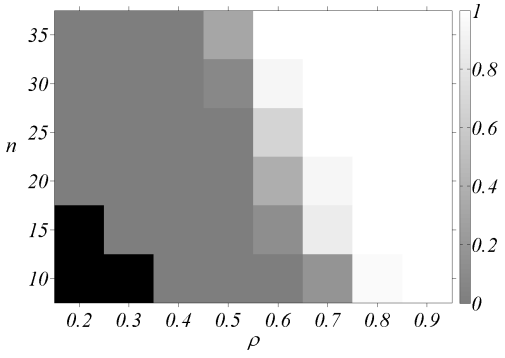


Figure A.8: Ratio of atomistic graphs in each sample (n, ρ) of random and generically rigid graphs with a given number of vertices n and a given density ρ, for dimension $d = 2$. Pitch black rectangles represent empty samples (too sparse to be rigid). A phase transition can be appreciated.

engineering applications. While this problem is widely reported to be very hard, no rigorous proof of NP-completeness has been found after a fairly exhaustive literature review (a superproblem, the sparsest approximation to the solution, is known to be NP-complete [142]). However, many theoretical results have been developed, defining conditions on the matrix A for the sparsest solution to be calculable in polynomial time [29], provided that A is full-rank or at least has a large spark. Unfortunately, the matrix S has always low rank (Corollary A.14) and the spark is always extremely low (at most seven for dimension $d = 2$, since six vertices induce seven linearly dependent atoms). As a result, these methods yield very sub-optimal solutions to the SAL in most cases.

A.2.3 Heuristics

Because no practical method to find the SAL has been found, heuristics are the natural choice. While many heuristics were tried (notably, heuristics adapted from methods to find the sparsest solution to a linear system of equations, as described in Section A.2.2), only the two best ones will be defined here, alongside the original decomposition algorithm:

RAW is the original decomposition algorithm, where at each step, the vertex with minimal degree is selected to ensure that the algorithm always halts.

MINDEG is an heuristic which greedily prunes the decision tree of the decomposition algorithm: in general, when considering an intermediate graph G_i in the combinatorial algorithm, there are many possible next graphs G_{i+1}. For each possible G_{i+1}, consider the sorted list vertex degrees, D. Select as the next graph the one with minimal D in the lexicographic sense. Therefore, these vertices will be more likely to

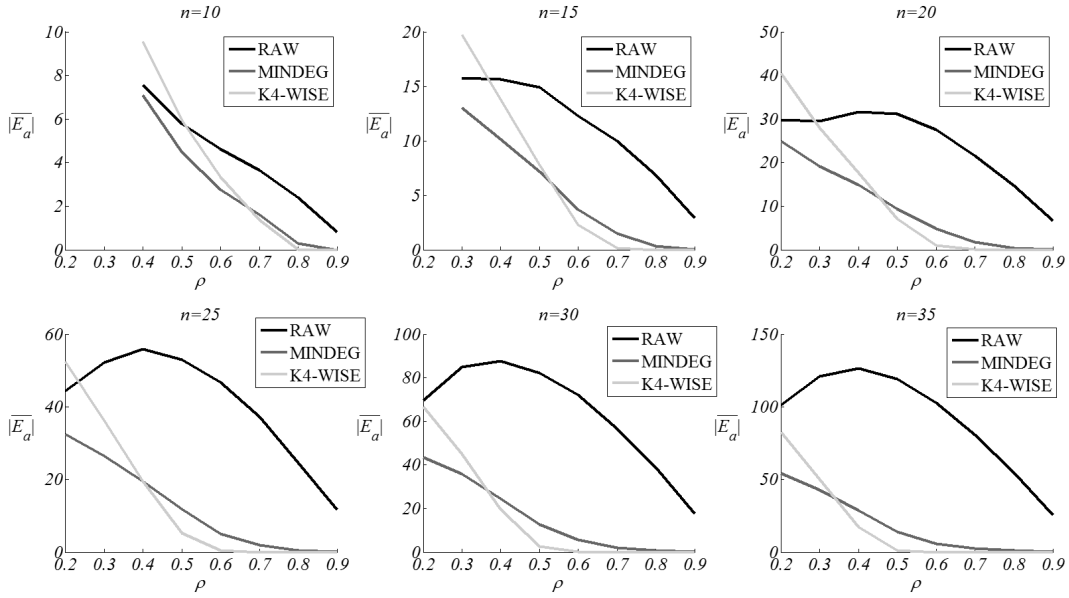


Figure A.9: Performance in dimension $d = 2$ of the heuristics MINDEG and K4-WISE compared to the RAW method. For each sample (n, ρ), all decompositions for all graphs are considered, and the mean of the number of added edges $|\overline{E_a}|$ is calculated for each different method. There is a subgraph for each different number of vertices n. For all subgraphs, the X axis represents graph density ρ, and the Y axis represents the mean of the number of added edges for the corresponding sample (n, ρ). See Section A.2.3 for details.

be selected in subsequent steps: the net effect is a strategy tending to remove as many edges as possible in clusters of vertices with small degree. This way, these clusters tend to be removed as soon as possible.

K4-WISE is an heuristic based on the algebraic characterization of the problem, which greedily adds edges to a set E_a until $\operatorname{rank}(S_{E'}) = \operatorname{rank}([S_{E'} \mid v_{E'}])$, with $E' = E \cup E_a$ (Proposition A.17). In dimension $d = 2$, to decide which edges to add to E_a, the heuristic considers all the the induced subgraphs of four vertices having between 3 and 5 edges (modulo all the induced subgraphs formed by a triangle and an unconnected vertex). Then, in each step, it selects them randomly one by one, adding to the graph the set of edges in the complement of the corresponding subgraph, and testing if it has become atomistic after each addition. When the condition is fulfilled, the added sets of edges are considered in some other random order, removing the ones that leave the graph as still atomistic. Note that, while finding the rank of a single matrix is computationally expensive, better methods may be possible if the rank is calculated over a series of gradually changing matrices [62].

Samples of relatively large graphs have been generated for several combinations of numbers of vertices $n \in [10, 15, 20, 25, 30, 35]$ and graph densities $\rho \in [0.2, 0.3, \ldots, 0.9]$, to study the heuristics as the number of vertices and the density grow. For each combination (n, ρ), a sample of 50 random and generically rigid graphs has been generated (except for combinations $(10, 0.2)$, $(10, 0.3)$, and $(15, 0.2)$, which cannot yield rigid graphs). 50 trials of each method (RAW, MINDEG and K4-WISE) were performed for each graph. From this sample of random graphs, a phase transition has been detected: at a certain density depending on n, the ratio of random atomistic graphs switches from nearly 0 to nearly 1 (Figure A.8).

The results (Figure A.9) show that MINDEG is always significantly better than RAW, while K4-WISE is significantly worse (even than RAW in some cases) for small and sparse graphs, but it gets better and better for larger and larger graphs, and surpasses the performance of MINDEG for dense graphs of any size. This is not surprising, since K4-WISE is able to detect atomistic graphs (which predominate at high densities), while MINDEG still tends to add unnecessary edges to them. The point at which it is better to switch from MINDEG to K4-WISE shifts to lower densities as the number of vertices increases, and is always at slightly lower densities than the phase transition.

As the number of vertices increases, minimal decompositions are expected to become an exponentially smaller and smaller fraction of all possible combinatorial decompositions for sparse graphs. For these graphs, no heuristic seems to perform very much better than the raw method. This makes sense, as sparse graphs will have an associated generic self-stress space of very low dimensionality, requiring many precisely added edges to become atomistic.

A.3 A proposal to analyze self-stresses

The atomic decomposition of a tensegrity structure shows how to build its self-stress from elemental units (atoms). Thus, if the self-stress w of a tensegrity structure $G(P, w)$ is analyzed from a purely static point of view, it makes sense to use the length of its minimal decomposition (the SAL) to measure its complexity $C(w)$, since it represents the smallest number of elemental units (atoms) needed to build the structure: $C(w) =$ SAL. Note that this applies to the self-stress of the structure, not to the structure in

itself.

However, under this definition, the self-stresses of cliques would be maximally complex. In general, very dense graphs tend to have a relatively long SAL, mainly because of having a self-stress space of relatively high dimension. However, we can use the relationship between atomic decompositions and the structure of the self-stress space $W_{G(P)}$ of tensegrity structure $G(P, w)$, in order to get a more descriptive definition. Since in most tensegrity structures each edge added during the decomposition represents an interlock between the self-stresses of several atoms (Section A.2), the more edges, the more complex must be the interlock between the self-stresses of the atoms in the decomposition, in order to form the desired self-stress of the tensegrity structure being analyzed. Thus, a more meaningful way to define the complexity of the self-stress of a tensegrity structure $G(P, w)$ is to substract the dimension of the self-stress space from the SAL: $C(w) = \text{SAL} - \left|W_{G(P)}\right|$. This definition will be used in future research on the generation of new tensegrity structures by addition of atoms.

Appendix B

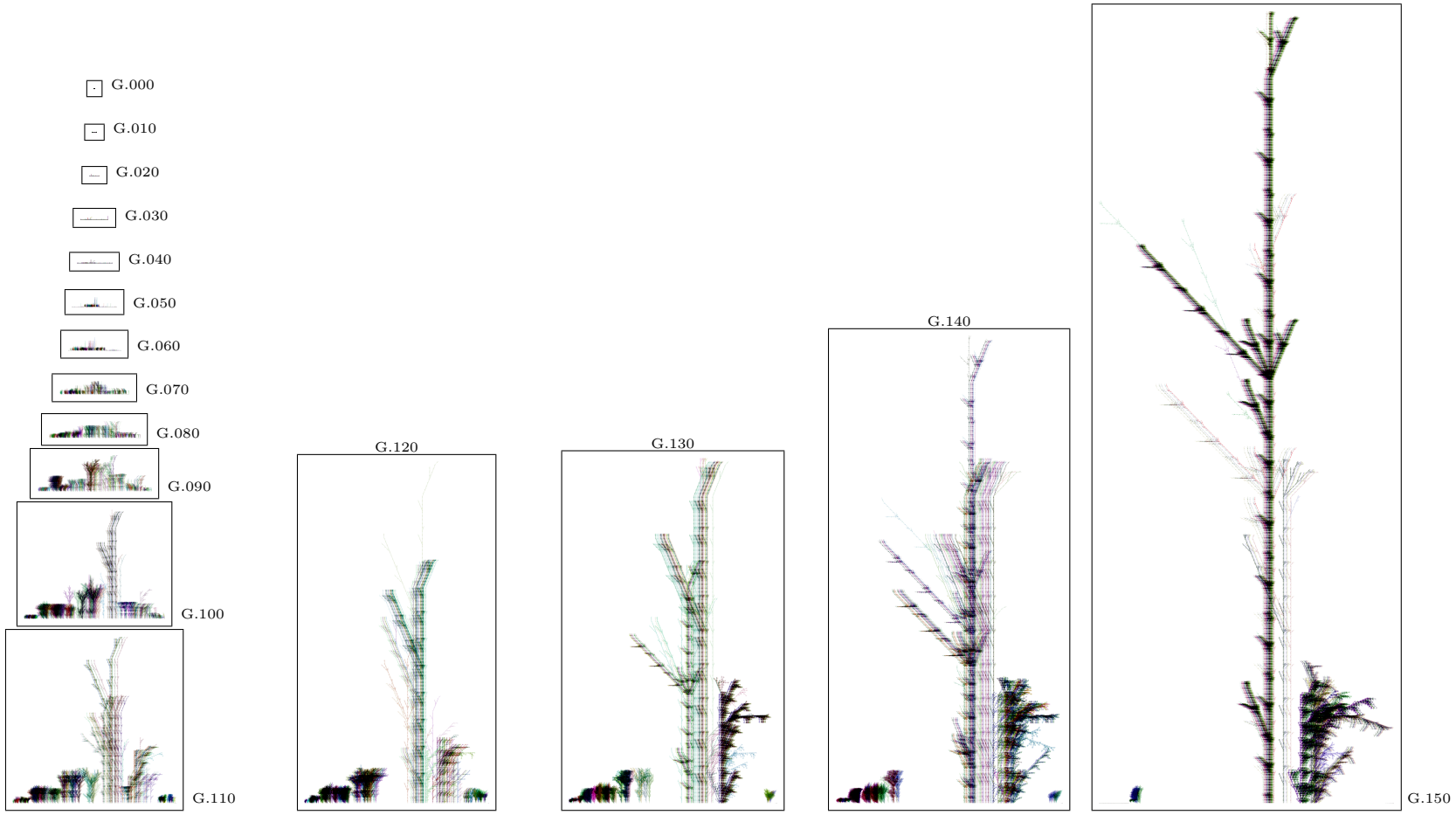
Virtual plant simulations

In Section 4.3, three simulations of virtual plants were thoroughly analyzed. Here, we include complementary data: a series of images for each simulation, showing the populations at intervals of 10 generations. Section B.1 corresponds to the simulation in Figure 4.9, Section B.2 to Figure 4.10 and Section B.3 to Figure 4.11, respectively. For each simulation, the set of the images corresponding to that simulation are shown to scale, to appreciate how the population and the individuals grow in size. However, the sets are not all to the same scale, as their proportions vary too greatly.

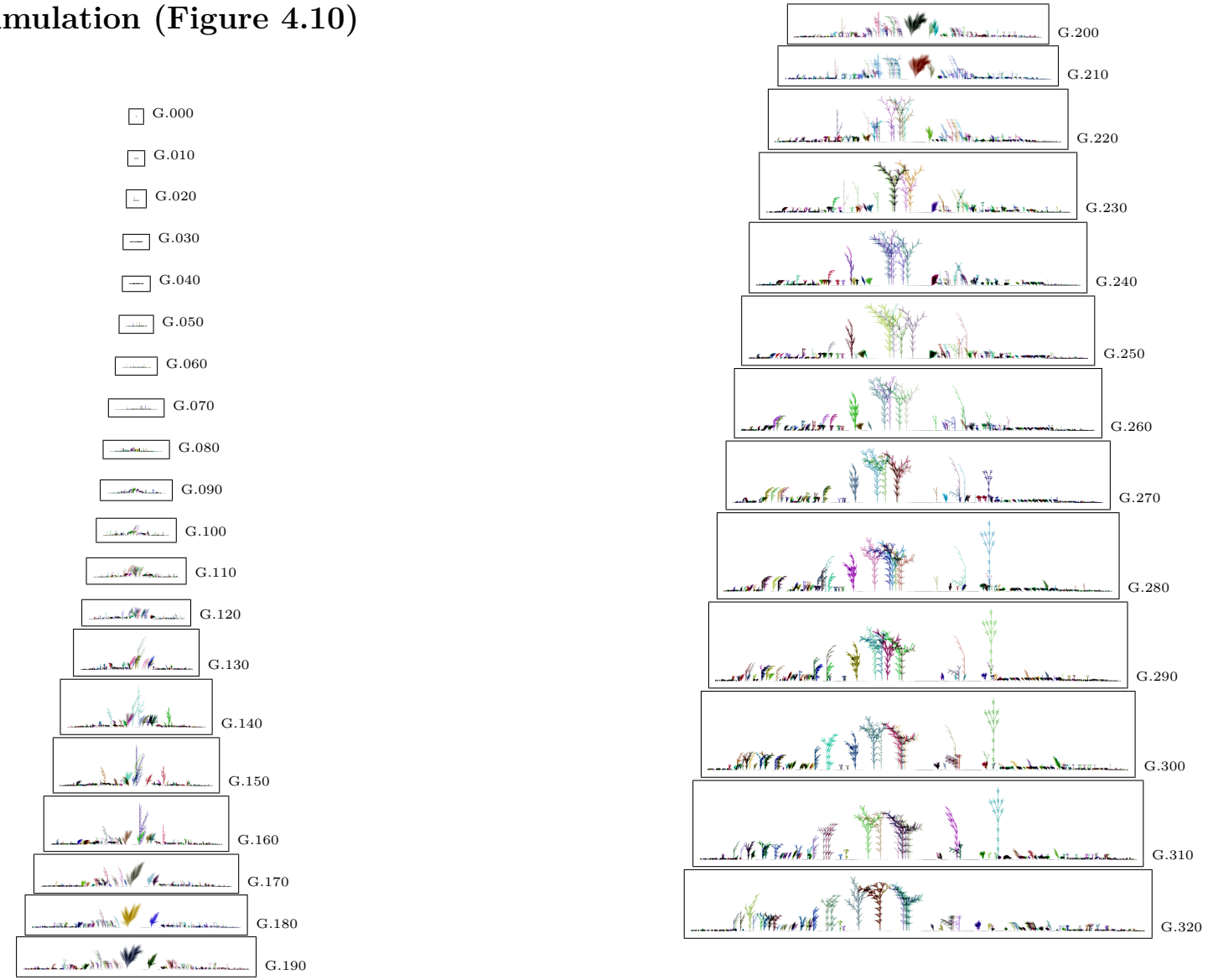
The images for mild and harsh simulations have been scaled down before inclusion in this document, as they are too big, so the images for the initial generations are too small, almost invisible, and show no detail. However, the images of the very harsh simulation retain all the original details, which can be seen in the electronic version of this document with full resolution, if it is appropriately zoomed in.

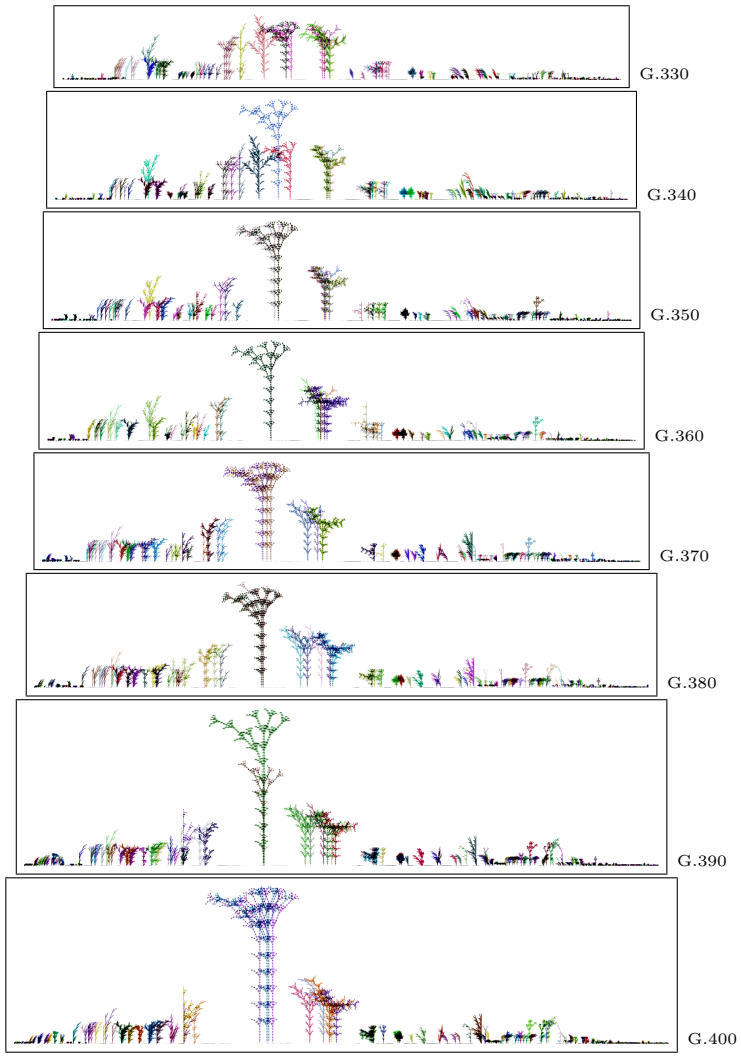
The colors of the plants are completely random, and they are used just to visualize the plants which are superposed; they do not convey any information.

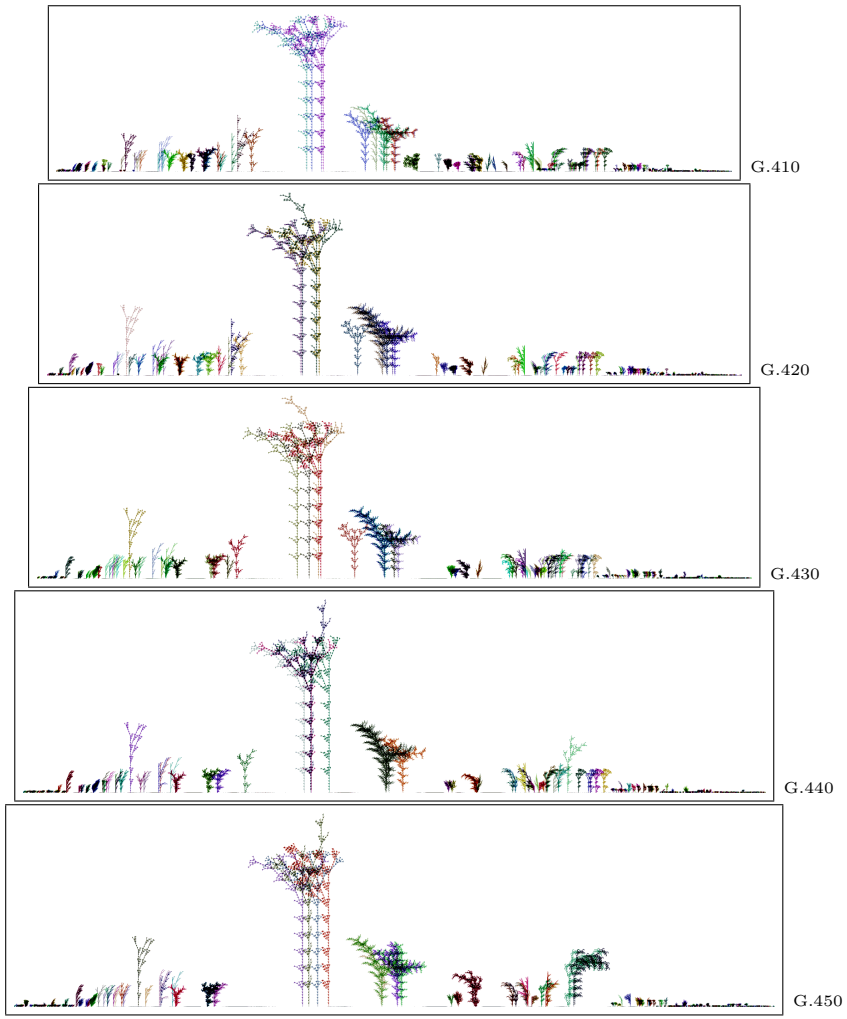
B.1 Mild simulation (Figure 4.9)



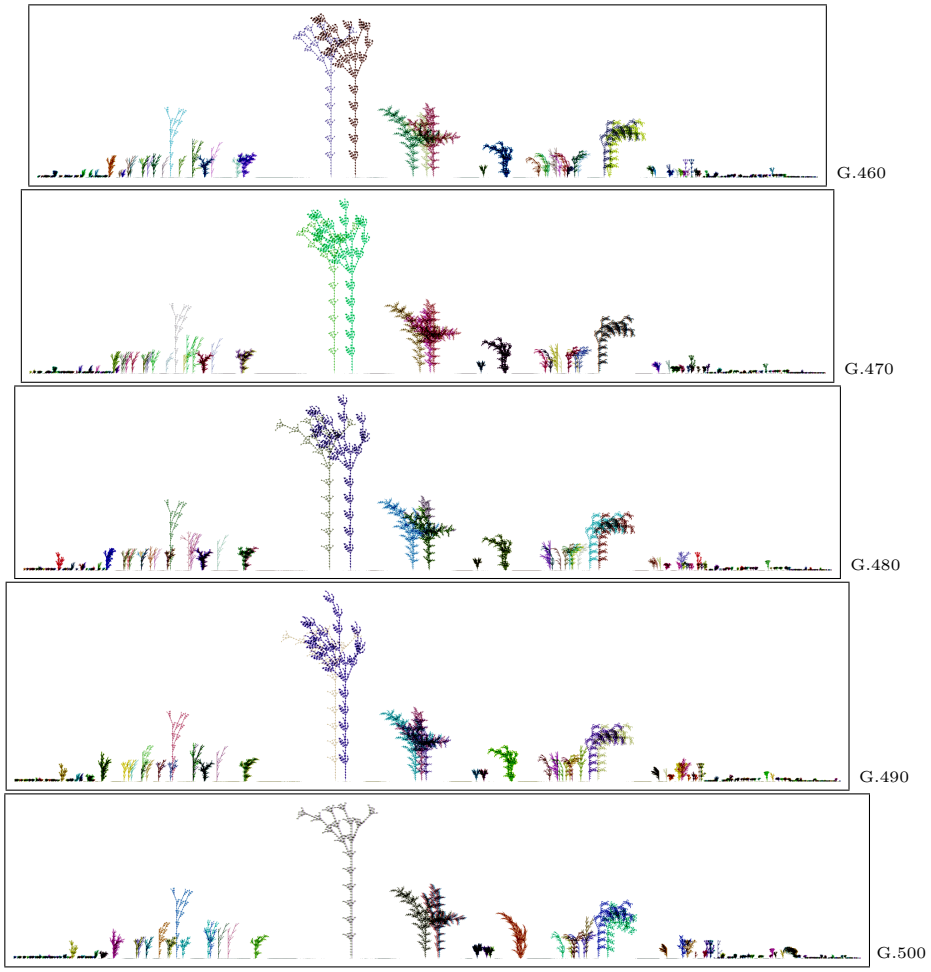
B.2 Harsh simulation (Figure 4.10)

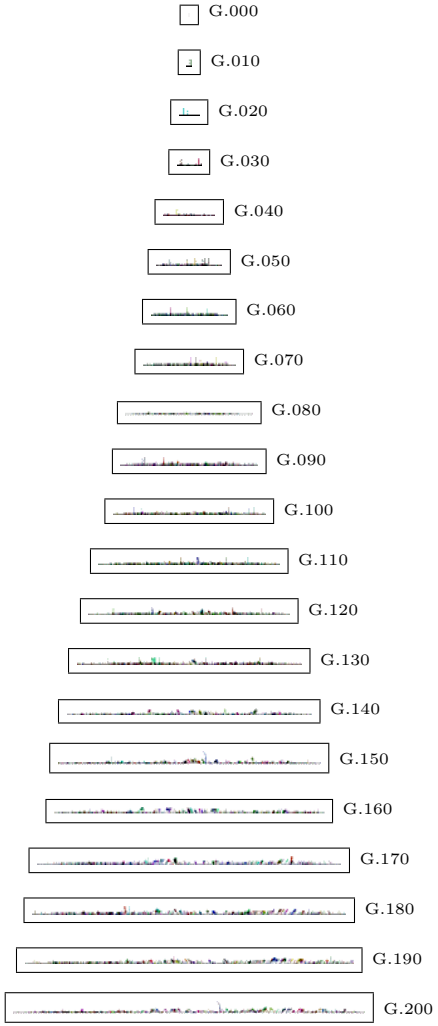


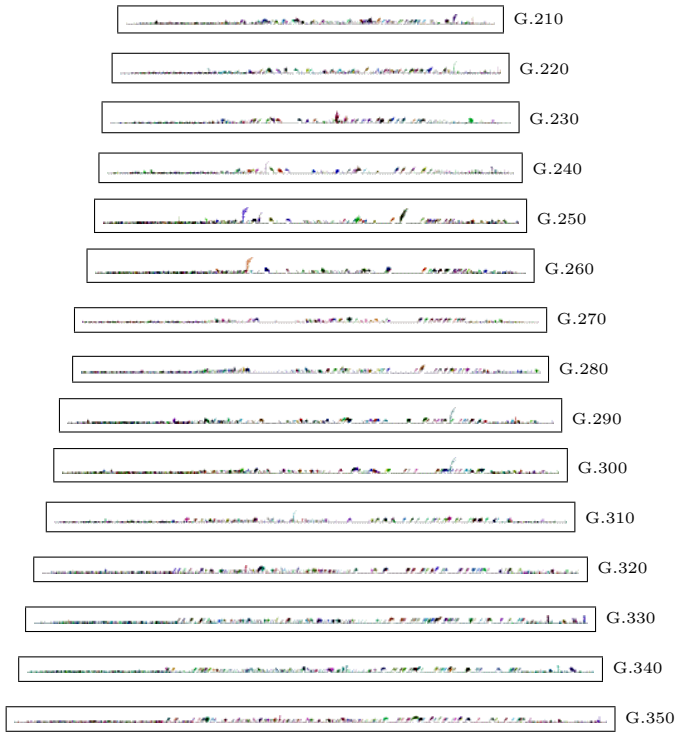

G.330
G.340
G.350
G.360
G.370
G.380
G.390
G.400

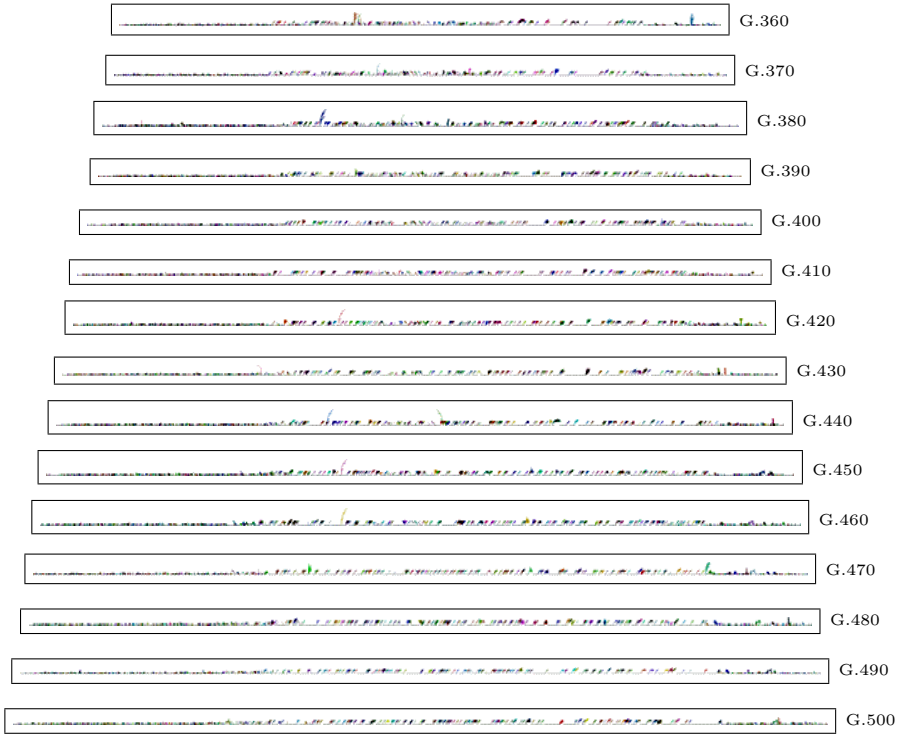

G.410
G.420
G.430
G.440
G.450

B.3 Very harsh simulation (Figure 4.11)






G.210
G.220
G.230
G.240
G.250
G.260
G.270
G.280
G.290
G.300
G.310
G.320
G.330
G.340
G.350


G.360
G.370
G.380
G.390
G.400
G.410
G.420
G.430
G.440
G.450
G.460
G.470
G.480
G.490
G.500

Appendix C

Implementation details

This appendix presents a summarized account of the implementation details of the hardware (Section C.1) and software (Section C.2) employed in this dissertation.

C.1 Hardware configuration and system software

Since the simulations and evolutionary algorithms presented in this dissertation have a relatively high computational cost, a Beowulf computer cluster, named *terclus*, was used. This cluster was built and administered by the members of the research group of Francisco Vico, and it was entirely at our disposal. *terclus* was built with the following hardware configuration:

- A master node with 2 AMD Opteron Quad-Core processors at a clock speed of 2.2 Gigahertz (a total of 8 CPUs), 8 Gigabytes of main memory, and a hard drive with a capacity of one Terabyte as the main store unit in the cluster.
- 12 slave nodes with 2 AMD Opteron Dual-Core processors at a clock speed of 1.9 Gigahertz (4 CPUs per slave, a total of 48 CPUs), 4 Gigabytes of main memory, and a very small hard-drive (just 80 Gigabytes) as local storage for the operating system and temporary files.
- A data network to communicate the nodes for computational purposes and to make accessible the main store unit in the master node to all the slave nodes (therefore, the store unit could be used to share data between the nodes). This

data network was implemented as two Gigabit Ethernet networks configured as a channel bonding on the nodes.

- An administrative network for configuring purposes and command-line access to any slave node from the master node. This administrative network was implemented using a Fast Ethernet Network.

The master node also had another network link connected to the intranet of the university, in order to be used remotely. In total, *terclus* had 56 CPUs and 56 Gigabytes of main memory, so one Gigabyte of main memory was available per CPU, roughly speaking. Memory swapping was disabled in all nodes by default, since in most cases available memory was enough, and whenever swapping started to happen, severe thrashing almost always ensued, leading to an almost unrecoverable state, with all the activity of the node virtually halted.

All the nodes in *terclus* ran a `Debian GNU/Linux` operating system[1]. Job scheduling was implemented through the `TORQUE` Resource Manager[2] and the `MAUI` Cluster Scheduler[3] (Adaptive Computing, Inc.), in order to balance the computing load of the cluster. `TORQUE` was configured to manage a pool of 52 computational units (48 CPUs from the slave nodes, and 4 more CPUs from the master node). The 4 remaining CPUs in the master node were left to power controlling processes in the master node and various administrative tasks.

In summary, the *terclus* cluster was configured to be used in the following manner: login remotely to the master node through `SSH`, and run controlling processes in it, which in turn would spawn concurrent computing processes in the slave nodes, using `TORQUE/MAUI`. Additionally, computing processes could be directly run in the slave nodes, accessing to them from the master node by `SSH`. The `screen`[4] GNU application was available in all nodes as a means of keeping sessions running while the user was not logged in, and `Ganglia`[5] was used to monitor the computational load of the nodes of the cluster (in terms of memory usage, used network bandwidth and CPU usage). While command-line access was the most common way to operate software on the cluster, `X11` GUI applications were also remotely executable in the master node.

[1] `http://www.debian.org/`
[2] `http://www.adaptivecomputing.com/products/torque.php`
[3] `http://www.clusterresources.com/pages/products/maui-cluster-scheduler.php`
[4] `http://www.gnu.org/software/screen/`
[5] `http://ganglia.info/`

Additionally, another computer cluster (*Picasso-cluster*) operated by the Supercomputing Center[6] of the Universidad de Málaga was also used. This cluster had 10 nodes, each one with 8 CPUs (Intel Xeon) and 16 Gigabytes of main memory (in total, 80 CPUs Intel Xeon and 160 Gigabytes of main memory). *Picasso-cluster* also could be accessed by `SSH` (though no `X11` GUI applications could be launched) and also used `TORQUE` to schedule batches of computing processes. However, the conditions of use were very restrictive, as this cluster was shared by the entire scientific community of the university. In order to cope with these limitations, the algorithms to be executed in this cluster had to be partially reimplemented (see next Section).

C.2 Algorithms and simulations

The models exposed in Chapters 2, 3 and 4, from the respective evolutionary algorithms to the agent simulations were prototyped in the `MATLAB`® scripting language[7] (The MathWorks, Inc.). Then, in the computer cluster, they were compiled into standalone executable files using the `MATLAB`® `Compiler`™ tool[8], and executed. The same procedure was used for the statistical testing of the heuristic algorithms presented in Appendix A.

In the evolutionary algorithms in Chapters 2 and 3, the cost of selecting and specifying the mutations to the individuals in the evolving population were negligible, while, at the same time, the task of evaluating the individuals of the population was embarrassingly parallel. Therefore, these evolutionary algorithms were designed to be executed as controlling processes in the master node of *terclus*. In each generation of the algorithm, the individuals of the population had to be evaluated, so `TORQUE` was used to dispatch a batch of processes to the slave nodes to evaluate each one of the individuals. In each case, the evaluation consisted of the physical simulation of the model (with a relatively high computational cost), and the calculation of the fitness based on the simulation. The physical simulation itself was fairly simple, at the core being a function calculating the velocities and accelerations of the nodes in the tensegrity structures and elastic networks in each simulation step. In the case of the evolutionary algorithm in Chapter 3, the mutation was specified by the main evolutionary algorithm, but the

[6] `http://www.scbi.uma.es/`
[7] `http://www.mathworks.com/products/matlab/`
[8] `http://www.mathworks.com/products/compiler/`

mutation itself was the result of a physical simulation, which was performed just before evaluating the mutated individual, also in the processes executed in the slave nodes. The necessary network bandwidth in the computer cluster was fairly low, since the controlling processes just had to send the genome of the individuals to the slave nodes to evaluate them, and only the numerical fitness and a few statistical parameters from each individual where sent back from the slave nodes to the controlling processes.

For the simulations in Chapter 3, *Picasso-cluster* was also used. This cluster had restrictive conditions of use:

- No controlling process could be executed in the cluster for more than five minutes of CPU time.
- Processes scheduled through `TORQUE` were required to spend a significative amount of CPU time (above an hour); in contrast, each evaluation of a molecular motor took a few minutes.

In order to cope with these limitations, the evolutionary algorithm was modified to be executed from a personal computer. As in the *terclus* implementation, it consisted of a cycle of selection - mutation - evaluation of individuals, where all parts of the process had a small computational cost, except for the physical simulations of the mutations and the evaluations of the individuals. In each cycle (or generation), these physical simulations were evaluated remotely in the cluster: the evolutionary algorithm automatically logged by `SSH` to *Picasso-cluster*, sending the necessary data to dispatch a small batch of processes, each one evaluating several individuals. After dispatching the processes, the evolutionary algorithm periodically logged by `SSH` in the cluster to ask if all processes were finished, in order to retrieve the results all at once. Once all the individuals were evaluated and the results retrieved, the evolutionary algorithm moved to the next cycle of selection - mutation - evaluation.

For Chapter 2, about 20 instances of the evolutionary algorithm were run, using in total about 8 years of CPU time (entirely executed in *terclus*). For Chapter 3, about 250 instances of the evolutionary algorithm were run (mostly preliminary tests to refine the model), using in total about 12 years of CPU time[9]. From the figures in the previous

[9]The computational cost per simulation was lower in the latter case because the duration of the simulation was strictly defined and bounded, while in the former case the simulation was run until a stable configuration was reached, a very long time in many cases.

sentence, 86 instances of the evolutionary algorithm were run in *Picasso-cluster* (see previous Section), amounting to about 2 years and 4 months of CPU time.

In the case of the evolutionary simulation in Chapter 4, while it was indeed parallelizable, the calculation procedure of the fitness was not as embarrassingly parallel as in the other cases, since the fitness was the result of the interaction between the individuals in a common environment. As a consequence, the simulation was not parallelized, in order to cut programming time. Instead, multiple instances of the evolutionary simulation were run concurrently in different nodes of the computer cluster. Every ten generations, the environment with all the trees in the population was saved as a PNG snapshot image file. About 1000 instances of the evolutionary simulation were run (mostly preliminary tests to refine the model), using in total about one year of CPU time (entirely executed in *terclus*). Most of the computational cost went into simulations with mild environments, the gigantic trees emerging in these simulations being extremely costly to evaluate (some snapshots of these populations were so big that their PNG files were over one hundred Megabytes).

The statistical tests of the heuristic algorithms in Appendix A were also run in the computer cluster, though the computational cost was negligible in comparison to the costs previously described for Chapters 2, 3 and 4.

In all cases, the results collected from the numerical simulations and experiments were analyzed using the `MATLAB`® language. Also, all the figures in all chapters and appendices presenting tables, plots and histograms have been generated using `MATLAB`®, as well as the depictions of 2D tensegrity structures (Chapter 2), 3D elastic networks (Chapter 3), and 2D virtual plants (Chapter 4 and Appendix B). Especially in the model in Chapter 4, some analytical procedures (as the calculation of the mean Jaccard distance in evolving populations, Figure 4.16) were very computationally demanding. As a consequence, they were also computed in *terclus* in a distributed way.

UNIVERSIDAD
DE MÁLAGA

Appendix D

Spanish summary and conclusions

In this Appendix, a summary and the conclusions of the dissertation are translated to Spanish, as per the rules of the Universidad de Málaga[1].

Declaración de financiación: durante la realización de esta tesis doctoral yo, Jose David Fernández Rodríguez, he recibido una beca FPU (AP2007-03704) del Gobierno de España. Además, he recibido financiación de dos proyectos de investigación: BioEmergences (código 28892) del Sexto Programa Marco de la Unión Europea, y el projecto GENEX (P09-TIC-5123) de la Junta de Andalucía.

D.1 Resumen

En las últimas décadas, la línea divisoria entre la Biología y la Informática se ha hecho cada vez más difusa. Por un lado, una gran cantidad de métodos computacionales de todo tipo han tomado como inspiración otros tantos sistemas y procesos biológicos (por lo que apropiadamente reciben el nombre colectivo de *Computación Bioinspirada*). Por otro lado, la mayoría de los campos de investigación en Biología se han vuelto (o están en proceso de volverse) totalmente dependientes de muchos tipos de métodos computacionales, conforme el creciente flujo de datos ha hecho que los investigadores los incluyan en sus métodos habituales de trabajo. De la misma forma, se encuentra fuera de toda duda la prevalencia en muchos campos del uso de modelos computacionales

[1] `http://www.uma.es/`

para comprender sistemas y procesos biológicos. Dentro de este contexto, esta tesis doctoral constituye un trabajo interdisciplinar, situado precisamente en la frontera entre Informática y Biología.

Puede decirse que la teoría de la evolución es la disciplina más permeable en este intercambio de ideas entre Biología e Informática. Considerando el flujo de ideas de la primera a la segunda, en las últimas décadas se ha visto cómo los procesos y dinámicas evolutivas pueden abstraerse fácilmente en esquemas algorítmicos, inspirando a muchos ingenieros informáticos a crear técnicas de optimización metaheurísticas basadas en la evolución, llamadas *Computación Evolutiva* en su conjunto. En lo que se refiere al flujo de ideas en sentido contrario, no solamente se han incorporado métodos computacionales a la investigación de la teoría de la evolución, sino que también se usan cada vez más modelos computacionales puramente sintéticos para investigar cuestiones de dinámica evolutiva.

Un fascinante aspecto de la teoría de la evolución es la evolución de la diversidad: en los últimos cientos de millones de años, nuestro planeta ha sido el escenario de la evolución de una increíble variedad en tamaño, morfología, función, organización y comportamiento de seres vivos. Se puede decir que, actualmente, la cuestión de por qué estos organismos presentan tanta diversidad en tantos aspectos no ha podido ser respondida satisfactoriamente, y sigue siendo un fascinante campo de investigación. Como tantas otras relacionadas, esta cuestión tiene también su reflejo en Computación Evolutiva, en la cual se investiga activamente la generación y mantenimiento de la diversidad de soluciones en algoritmos evolutivos.

Frecuentemente, la ciencia se enfrenta a cuestiones y problemas muy difíciles, que podrían parecer intratables si se abordan directamente. En muchos casos, las respuestas y soluciones se encuentran analizando modelos matemáticos y simplificados de la realidad. Sin embargo, en relación a la cuestión previamente formulada de los orígenes evolutivos de la diversidad, es razonable esperar que ningún marco conceptual pueda contener por sí solo la respuesta a tan formidable cuestión, sino que una gama más o menos amplia de éstos contribuyan a explicar la evolución de la diversidad a distintos niveles. En este marco, esta tesis doctoral presenta tres modelos computacionales simples, cada uno de ellos profundamente diferente de los otros dos. De esta manera, cada modelo sirve para estudiar diferentes aspectos y causas de la evolución de diversidad. Específicamente, los modelos exploran los roles en la evolución de la

diversidad de los procesos de desarrollo (Capítulo 2), la coevolución de la morfología y el sistema de control de estructuras biológicas (Capítulo 3) y las dinámicas y procesos evolutivos (Capítulo 4).

El resto de esta sección presenta una pequeña introducción a las principales líneas tratadas en esta tesis doctoral (Secciones D.1.1 a D.1.5) y un resumen separado para cada uno de los capítulos mencionados en el párrafo anterior (Secciones D.1.6 a D.1.8).

D.1.1 Modelos basados en agentes en Biología Computacional

La investigación en la teoría de la evolución se ocupa de esclarecer los detalles de los procesos evolutivos que han configurado todos los organismos vivos que han poblado la Tierra a los largo de su historia. De especial interés es el estudio del cambio evolutivo al nivel de especies o incluso superior, conocido popularmente como *macroevolución*. Este cambio macroevolutivo se ha estudiado tradicionalmente solamente de forma indirecta, al principio con herramientas de Taxonomía, Sistemática y Paleontología, después con Genética de Poblaciones, y con nuevas herramientas cada vez más sofisticadas como la Filogenia Molecular. Sin embargo, es difícil estudiar los procesos evolutivos en detalle, debido a que la evidencia es escasa e incompleta, y debe de componerse a partir de fuentes muy diversas como la estructura de los organismos actualmente existentes (a todas las escalas, de la anatómica a la genética) y de los registros fósiles.

Por lo tanto, se usan muchas vías alternativas para estudiar los procesos evolutivos, en particular, en lo que concierne a este trabajo, *modelos computacionales basados en agentes*. Estos modelos describen comunidades de individuos en un esquema de abajo hacia arriba, a través de las propiedades de sus miembros individuales, las reglas que siguen a distintos niveles, y las interacciones entre ellos y con el entorno [76]. Se ha propuesto un gran número de modelos computacionales basados en agentes, desde un nivel abstracto de grandes poblaciones de individuos que interaccionan entre sí de forma muy esquematizada [26, 221] a un nivel mucho más detallado, como por ejemplo el crecimiento pluricelular [51]. Los agentes se pueden modelar de muchas maneras distintas más allá de como abstracciones simplificadas de organismos biológicos, como por ejemplos programas informáticos [163], estructuradas derivadas de sistemas de Lindenmayer [17] o incluso autómatas en el contexto de la teoría de lenguajes formales [127].

Potencialmente, los modelos basados en agentes pueden usarse para producir

experimentos computacionales a gran escala que condensen largos periodos evolutivos en tiempos de cómputo relativamente cortos, a la vez que se registran vastas colecciones de datos para su análisis exhaustivo (de una manera imposible de igualar en trabajos puramente biológicos) y se extraen sus propiedades más relevantes [115, 121]. De esta manera, conforme se incrementa la potencia de cómputo y la capacidad de almacenamiento disponible para los investigadores, se hace factible la simulación de modelos de procesos evolutivos con un poder explicativo cada vez mayor.

Los tres modelos presentados en esta tesis doctoral están basados en agentes, aunque cada uno es profundamente diferente de los otros dos. El Capítulo 4 presenta un modelo sintético de poblaciones de plantas basado en agentes para el estudio de dinámicas evolutivas de diversificación, mientras que en los Capítulos 2 y 3 se usan algoritmos evolutivos para optimizar poblaciones de agentes con un criterio distinto en cada capítulo; la diversidad de forma y función emerge de los distintos aspectos de ambos modelos antes que de dinámicas evolutivas propiamente dichas.

D.1.2 La Biología Evolutiva del Desarrollo

La principal corriente de pensamiento evolutivo actualmente en vigor es aún la *Síntesis Evolutiva Moderna* [130], usada por muchos biólogos evolutivos como el marco conceptual de referencia para la teoría de la evolución, con la genética de poblaciones como la principal herramienta para entender el hecho evolutivo. Bajo este punto de vista, la diversidad de forma y función es una consecuencia secundaria de las dinámicas evolutivas a nivel de población, incrementándose gradual y pasivamente conforme diversos conjuntos de alelos se originan y compiten en diferentes subpoblaciones de individuos, y estos alelos se expresan en consecuentemente diversos conjuntos de fenotipos, adaptándose a nuevos entornos. Esto lleva a la tradicional visión de los fenotipos como versiones decodificadas de los correspondientes genotipos [129]. Sin embargo, este punto de vista hace bastante difícil la investigación de las causas de la evolución de la diversidad a un nivel más directo, menos abstracto [112].

Un punto de vista alternativo se origina en la siguiente observación: el fenotipo de un individuo se genera mediante un *proceso de desarrollo*, que transforma un cigoto en un organismo plenamente funcional. Este proceso se puede describir, en esencia, como una complejísima coreografía de eventos exquisitamente cronometrados, conforme las células del organismo en desarrollo se dividen, cambias sus propiedades fisicoquímicas,

y se disponen en capas y otras configuraciones celulares, plegándose en formas cada vez más complejas [60].

Por lo tanto, la evolución de los fenotipos se puede entender en gran medida como la evolución de sus correspondientes procesos de desarrollo. En este contexto, la disciplina de la *Biología Evolutiva del Desarrollo* ha emergido como el estudio de los procesos de desarrollo desde una perspectiva evolutiva. De hecho, los procesos de desarrollo parecen ser un factor clave para entender la evolución de la diversidad [14].

Desde un punto de vista funcional, los procesos de desarrollo se pueden interpretar como una interacción extremadamente intrincada entre el genoma y las propiedades geométricas y fisicoquímicas del organismo en desarrollo [60], donde el genoma se expresa funcionalmente como una red reguladora genética que guía y modula el proceso de desarrollo. Bajo esta interpretación, las reglas geométricas y fisicoquímicas del proceso de desarrollo adquieren un papel al menos tan importante como el del propio genoma, ya que estos aspectos de los procesos de desarrollo pueden imponer sesgos inherentes a las dinámicas evolutivas [5].

Incluso, desde un punto de vista más radical, es opinable que el genoma tenga que ser la principal fuente de control regulador en los procesos de desarrollo: características no genéticas de naturaleza geométrica y/o fisicoquímica podrían ser incluso más determinantes en algunos contextos, especialmente en las fases iniciales de la evolución de los organismos pluricelulares [144] y en algunos organismos unicelulares [71]. En el Capítulo 2 se presenta un modelo donde la diversidad de forma emerge como una consecuencia cuando los procesos de desarrollo se hacen gradualmente más complejos, incluso bajo un control genético mínimo, mientras que en los otros capítulos no hay un componente tan fuerte de procesos de desarrollo.

D.1.3 Motores moleculares

Los motores moleculares son máquinas biológicas a escala nanométrica capaces de transformar energía química en trabajo mecánico. Estos motores están asociados a muchas funciones vitales a todas las escalas, desde la subcelular a la anatómica [173]. Aunque los motores moleculares se estudian primordialmente desde la perspectiva de la Biología Molecular, suscitan gran interés en muchas otras disciplinas biológicas, desde la Medicina [69] a la Sistemática [202], e incluso en otras áreas como la Nanotecnología [205].

Desde un punto de vista de simulación y modelado computacional, las características y dinámica de estos motores se estudian a múltiples escalas temporales, con muchos tipos de modelos, que van desde simulaciones de dinámica molecular muy detalladas y precisas con rangos temporales extremadamente cortos [99, 104, 119], a modelos extremadamente simples y aproximados de sus *ciclos de trabajo* [77, 215]. Los motores moleculares se han estudiado extensivamente para entender sus características bioquímicas y estructurales [203], y su *procesividad*, es decir, la manera en que muchos de ellos son capaces de *caminar* a lo largo de filamentos dando una cierta cantidad de pasos seguidos sin despegarse completamente [68, 209, 225].

Sin embargo, estos estudios se han hecho hasta ahora en las relativamente pocas instancias conocidas en Biología Molecular. En el Capítulo 3, se usan resultados de análisis de redes elásticas [161] y métodos de configuración de función (*behavior-finding methods*) [120] para explorar un subconjunto de espacio de configuraciones de las *plantillas* de estructuras moleculares capaces de transformar energía química en movimiento direccional. De esta manera este dominio particular y muy específico (el estudio de los motores moleculares) se toma como plataforma de experimentación para el estudio de la evolución de la diversidad bajo condiciones muy específicas y limitadas, con el interesante resultado de que, de hecho, el movimiento de los motores moleculares se puede conseguir desde una gran diversidad de configuraciones estructurales, con una consecuente diversidad de patrones de funcionamiento, ambos en coevolución.

D.1.4 Computación Evolutiva

La Naturaleza siempre ha sido una gran fuente de inspiración para muchos científicos e ingenieros. En este aspecto, los ingenieros informáticos no son una excepción: muchas técnicas algorítmicas encuentran el origen de su inspiración en la naturaleza; nombrando sólo de algunas de las más relevantes y/o actuales, podemos hablar de las redes neuronales artificiales, los sistemas inmunes artificiales, la optimización por colonia de hormigas, y los algoritmos evolutivos [67].

Los algoritmos evolutivos son técnicas metaheurísticas que se usan para encontrar soluciones aceptables a problemas de optimización. La mayoría de los algoritmos evolutivos siguen un patrón común: un conjunto cambiante de soluciones candidatas (la *población*) sufre un ciclo (repetido múltiples veces) de evaluación, selección y reproducción con variación. El primer paso en el algoritmo consiste en generar las

soluciones candidatas del conjunto inicial, comúnmente de forma más o menos aleatoria. Cada candidata se evalúa mediante una *función de fitness*, es decir, una regla heurística para medir su calidad como solución al problema[2]. Después de la fase de evaluación, se efectúa la selección: un nuevo conjunto de soluciones candidatas se genera a partir del conjunto anterior; cada candidata se copia un número de veces que es proporcional (aunque aleatorio en cierta medida) a su nivel de fitness. Este paso disminuye la diversidad de la población, que es hasta cierto punto restaurada aplicando a una fracción de las soluciones candidatas algún operador diseñado para incrementar la variación (por ejemplo, operadores de cruce o mutación). Estos pasos (los previamente mencionados de evaluación, selección y reproducción con variación) se aplican múltiples veces. El efecto acumulativo es un cambio más o menos lento en la media (y la mejor instancia) del valor de fitness en las soluciones candidatas, conforme las mejores soluciones son promovidas a través de la selección y nuevas variantes van siendo creadas.

Aunque este patrón algorítmico es común a todos los algoritmos evolutivos, cada uno de ellos usa diferentes conjuntos de reglas de selección, operadores de variación y codificaciones de las soluciones. Las variedades más relevantes son los *algoritmos genéticos* [84], las *estrategias evolutivas* [8], la *programación evolutiva* [59] y la *programación genética* [107]. Entre estas variantes, los algoritmos genéticos, definidos por primera vez por Holland, son los más populares.

En un algoritmo genético, cada solución candidata tiene un genotipo y un fenotipo, que son respectivamente la forma codificada que es manipulada por el algoritmo, y la solución en sí misma, que es evaluada por la función de fitness. La formulación original de Holland está fuertemente asociada con una codificación muy sencilla y directa de los genotipos como cadenas de bits, aunque éste es no es necesariamente el caso en todas las instancias de algoritmos genéticos. De hecho, los modelos presentados en los Capítulos 2 y 3 usan lo que se puede describir perfectamente como algoritmos genéticos (aunque con codificaciones considerablemente más elaboradas) para diseñar evolutivamente diversas estructuras. No obstante, para evitar cual malentendido potencial, éstos serán denominados, más genéricamente, como algoritmos evolutivos.

[2] éste es el significado de fitness en Computación Evolutiva, y es el significado usado más comúnmente en esta tesis doctoral. Está inspirado en (pero no se debería confundir con) el concepto de *fitness* en Biología Evolutiva: una medida de la habilidad de un individuo para propagar sus genes (frecuente pero incorrectamente asumido como restringido al éxito reproductivo inmediato). En general, en esta tesis doctoral, la palabra *fitness* se usa con el significado de Computación Evolutiva en los Capítulos 2 y 3, mientras que el significado puramente biológico se usa en el Capítulo 4.

Una subdisciplina de la Computación Evolutiva, de reciente formulación, es la *Embriogénesis Artificial*, que consiste en el uso de procesos de desarrollo artificiales como codificaciones indirectas para las soluciones candidatas [183]. Es decir, en vez de codificar las soluciones de un modo más o menos directo, se codifican como algún tipo de especificación para dirigir o modular algún tipo de proceso de desarrollo más o menos abstracto. El objetivo es acometer problemas más grandes y complejos mediante algoritmos evolutivos, ya que si la codificación indirecta se diseña correctamente, permite reducir (a veces dramáticamente) el tamaño, dimensionalidad y complejidad de los espacios genéticos en los que se lleva a cabo la búsqueda evolutiva [47]. El Capítulo 2 proporciona un ejemplo extremo de codificación indirecta a través de procesos de desarrollo, mientras que en el Capítulo 4 los individuos también se codifican indirectamente, aunque de un modo menos intrincado, usando un proceso de desarrollo muy simple basado en reglas de reescritura en el contexto de la teoría de lenguajes formales.

D.1.5 Computación morfológica y tensegridad

El concepto de *computación morfológica* ha surgido en los últimos años en el campo de la Robótica, para nombrar la idea de que el comportamiento de un robot no sólo depende de la naturaleza de sus sistema de control y la manera en que éste se ajusta al cuerpo del robot, sino también de la morfología del cuerpo mismo [154, 157]. Esto significa que, si la morfología del robot se toma en cuenta en el diseño, el sistema de control puede ser más simple de lo que podría esperarse, aprovechando la dinámica de interacción entre el cuerpo y el sistema de control. Paul [154] proporcionó una prueba de concepto para esta idea al diseñar un robot con un comportamiento de tipo XOR, aunque su sistema de control esta basado en simples perceptrones monocapa (incapaces de generar la función XOR). El comportamiento de tipo XOR era el resultado de la interacción entre el sistema de control y la estructura del robot.

Este concepto se ha aplicado al diseño de robots bípedos con sistemas de control mínimo [128] y robots con sistemas de control sin retroalimentación y mínima cantidad de grados de libertad, que son capaces de generar patrones de movimiento rápidos y estables y generar amplios conjuntos de comportamientos, todo ello a través de la interacción entre el cuerpo y el sistema de control de los robots [157]. La clave es el modo de construir los robots, no solo su morfología, sino las características de los

materiales constituyentes, que pueden modular de forma no lineal la actividad de los sistemas de control.

Aunque este concepto se ha usado primordialmente en el campo de la Robótica, representa un paradigma útil en otros campos. Un ejemplo de computación morfológica en modelos basados en agentes es el modelo de *path-followers* de Lobo [120]: agentes relativamente simples evolucionados para seguir caminos curvados, cuyos sistemas de control están implícitos en sus propias estructuras. Como ejemplo en el contexto de la Biología, los motores moleculares [173] pueden considerarse como dispositivos a escala nanométrica con un grado significativo de computación morfológica, debido a su inherente naturaleza como enzimas capaces de caminar sobre filamentos citoesqueléticos dentro de las células. Su locomoción se puede considerar como el resultado auto-organizado de la interacción entre su morfología y los detalles bioquímicos (que se pueden considerar colectivamente como su sistema de control) del ciclo de interacción con el filamento. En el Capítulo 3, se usa un algoritmo evolutivo para generar una diversidad de estructuras inspiradas en los motores moleculares. Sus (igualmente diversos) patrones de funcionamiento emergen de la compleja interacción entre sus estructuras y sus ciclos de operación, por computación morfológica.

Por otro lado, las estructuras de tensegridad también tienen potencial para permitir otros tipos de computación morfológica. Se trata de estructuras estables compuestas de elementos rígidos que soportan fuerzas de compresión, sostenidos en equilibrio por una red de cables que soportan fuerzas de tensión, de tal manera que se produce un equilibrio de fuerzas global [138]. Debido a que unos elementos soportan compresión y otros tensión (el así llamado *auto-estrés* de la estructura, que se analiza desde una perspectiva matemática en el Apéndice A), las estructuras de tensegridad poseen una cierta cantidad de energía potencial elástica; como efecto colateral, su dinámica es altamente no lineal. Estos hechos han sido aprovechados para diseñar robots ambulantes basados en tensegridad [153]; cuyos patrones de movimientos son modulados por las interacciones no lineales entre el sistema de control y las características de la estructura. En los procesos de desarrollo, el genoma puede ser considerado como el sistema de control. En el Capítulo 2 se presenta un algoritmo evolutivo usando un modelo basado en estructuras de tensegridad para generar diversos procesos de desarrollo (que a su vez inducen diversas morfologías finales). El rol primario como sistema de control no es proporcionado por el genoma, como es usual, en la forma de una red reguladora

genética, sino por la dinámica no lineal de las propias estructuras de tensegridad.

D.1.6 Capítulo 2: Diversidad mediante procesos de desarrollo complejos

En el Capítulo 2 se presenta una discusión sobre los orígenes evolutivos de la diversidad en el contexto de la Biología Evolutiva del Desarrollo (*evo-devo*), mediante la modulación a escala evolutiva de la interacción entre los genomas y los fenotipos durante los procesos de desarrollo. Se introduce el concepto de tensegridad, explicando sus propiedades y los roles de las estructuras con tensegridad en Biología (en este contexto, el Apéndice A proporciona una nueva metodología matemática para analizar algunas propiedades de las estructuras con tensegridad).

A continuación, se presenta un nuevo modelo abstracto de proceso de desarrollo, en el que el genoma juega un papel mínimo, simplemente especificando un conjunto de condiciones iniciales. El subsiguiente proceso de desarrollo se encuentra autoregulado principalmente por las propiedades dinámicas de las partes constituyentes de una estructura de tensegridad. El uso de estas estructuras como sustrato físico del proceso de desarrollo representa la innovación crucial aportada, que permite al modelo minimizar el papel regulador del genoma durante el proceso de desarrollo, ya que las propiedades dinámicas de las estructuras de tensegridad sustituyen al genoma como sistema de control del proceso de desarrollo, en lo que podría considerarse un caso de computación morfológica.

Buscando diseñar de instancias abstractas de procesos de desarrollo que sean largos y complejos, con muchos pasos intermedios, se usa un algoritmo evolutivo pensado para encontrar heurísticamente estos rasgos en los procesos de desarrollo. Como efecto colateral de la búsqueda de ejemplos de procesos de desarrollo complejos, evoluciona una diversidad de morfologías finales (resultados finales de los procesos de desarrollo).

D.1.7 Capítulo 3: Diversidad mediante la coevolución de la morfología y su sistema de control

En el Capítulo 3 se describen los motores moleculares biológicos desde las perspectiva de la Biología Molecular, a la vez que se da una vista complementaria desde la perspectiva de la Computación Evolutiva y la Vida Artificial. Se presentan varias herramientas matemáticas de la teoría de redes elásticas aplicadas al estudio de la estructura de las

proteínas (especialmente el *Gaussian Network Model* y el *Anisotropic Network Model*), que a continuación se usan como respaldo formal para un marco de trabajo para diseñar ejemplos artificiales (*plantillas*) de posibles configuraciones estructurales para motores moleculares. Desde el punto de vista de la Biología Molecular, estas plantillas son útiles como medio de evaluación computacional de hipótesis sobre la estructura de los motores moleculares, como en el ejemplo estudiado en la Sección 3.3.1.

El marco de trabajo propuesto en este capítulo sigue una larga tradición en Biología Computacional, que consiste en añadir aspectos físicos a las simulaciones basadas en agentes, de modo que emerja una diversidad de morfologías y formas de funcionamiento. Desde este punto de vista, el marco de trabajo puede caracterizarse de la siguiente manera: una tarea de optimización muy precisa (avanzar tan lejos como sea posible sobre un filamento rectilíneo), junto con un conjunto de reglas igualmente muy preciso para especificar la forma en que las plantillas de motores moleculares interaccionan con el filamento (el *ciclo de trabajo*).

Sin embargo, a pesar de esta rígida especificación, la coevolución de la morfología de las estructuras y el los detalles que la acoplan a su patrón de funcionamiento (su *sistema de control*) hace que emerja una amplia diversidad de plantillas con distintas morfologías y formas de funcionamiento. Esto se debe a que el modelo subyacente es relativamente detallado y suficientemente realista, permitiendo que el algoritmo evolutivo encuentre soluciones innovadoras al problema de *caminar* sobre un filamento.

D.1.8 Capítulo 4: Diversidad mediante procesos evolutivos emergentes

En el Capítulo 4 se presenta un estudio de las dinámicas evolutivas inducidas por las interacciones ecológicas entre modelos de plantas muy simples basados en agentes. Los agentes (plantas) se caracterizan por abstracciones extremadamente simples de varios procesos en los niveles genético, de desarrollo y fisiológico. Los genomas son cadenas de caracteres interpretadas como sistemas de Lindenmayer muy simples (D0L-systems con una única regla), y los fenotipos se generan mediante la clásica interpretación de los L-systems en términos de *geometría de tortuga*.

Las interacciones ecológicas entre individuos (plantas) dentro de una población son motivadas por la localidad de la reproducción (es decir, los descendientes inmediatos crecen en las inmediaciones de sus ancestros en generaciones anteriores), y se limitan a

rondas discretas de competición (una por generación) por la luz que cae verticalmente en el entorno. En cada ronda o generación, la cantidad de luz recogida determina el éxito reproductivo del individuo, es decir, la cantidad de descendientes en la siguiente generación (usando, de esta manera, el concepto biológico de *fitness*, en oposición al concepto relacionado pero distinto usado en trabajos más comunes de Computación Evolutiva).

La reproducción es asexual, con mutaciones ocasionales. A pesar de este marco de funcionamiento tan simple, surgen dinámicas evolutivas en los niveles genético, fenotípico y de población. En particular, una gran diversidad de morfologías evoluciona conforme las plantas compiten por la luz bajo diversas condiciones ambientales.

D.2 Conclusiones

El propósito principal de esta tesis doctoral has sido estudiar la diversidad de forma (morfología, estructura) y función (comportamiento, desarrollo) en tres modelos basados en agentes de sistemas biológicos, desde una perspectiva biológica, pero también con un fuerte enfoque en Computación Evolutiva. Estos tres modelos han sido considerados desde un amplio rango de puntos de vista, desde la Biología Evolutiva del Desarrollo, la Biología Molecular y las dinámicas evolutivas hasta la codificación indirecta en Computación Evolutiva y el diseño de estructuras, todo ello complementando el tema unificador de la evolución de la diversidad. Consecuentemente, esta tesis doctoral puede considerarse inequívocamente como interdisciplinar. Esta última sección revisará las principales conclusiones derivadas de los estudios presentados en esta tesis doctoral.

Específicamente, cada uno de los tres modelos está relacionado con un aspecto de la evolución de la diversidad:

El papel de los procesos de desarrollo.
En el contexto de la Biología Evolutiva del Desarrollo, los procesos de desarrollo juegan un papel clave en la evolución de la diversidad. Este papel se interpreta frecuentemente en términos de la regulación genética del desarrollo: una compleja red reguladora genética orquesta un complejo proceso de desarrollo que lleva del cigoto al fenotipo adulto, por lo que la evolución de diversas morfologías es el resultado de la evolución de la complejidad en las redes reguladoras genéticas [14]. En este

contexto, formulamos la siguiente cuestión: ¿la evolución de redes reguladoras genéticas complejas es un requisito necesario para inducir diversas morfologías a través de procesos de desarrollo? Sin embargo, en el Capítulo 2, se presenta un modelo abstracto de procesos de desarrollo auto-regulados, sin regulación genética, solamente modulación genética de las condiciones iniciales del proceso. Aun así, evoluciona una diversidad de morfologías finales usando un simple algoritmo evolutivo, con una heurística que se limita a encontrar instancias de procesos de desarrollo largos y complejos. En este modelo, las propiedades dinámicas de las tensegridades se apropian del carácter regulador del genoma. Por lo tanto, la evolución de morfologías diversas se puede considerar el resultado directo del incremento evolutivo de complejidad en los procesos de desarrollo, sin necesidad de invocar un papel absolutamente necesario para las redes reguladoras genéticas complejas.

La interacción entre la estructura y su sistema de control.
En el contexto de la Biología Computacional, la Robótica Evolutiva y la Vida Artificial, la coevolución del cuerpo y su sistema de control puede producir diversas morfologías y comportamientos coadaptados [58]. En este área, el trabajo previo ha usado tradicionalmente complejos sistemas de control y, en algunos casos, igualmente complejas codificaciones indirectas mediante procesos de desarrollo u otros esquemas de traducción de genotipo a fenotipo [103, 158, 178]. Sin embargo, el concepto de computación morfológica [154] sugiere que esta coevolución puede hacerse efectiva incluso si el sistema de control es extremadamente simple y está en su mayor parte implícito en la morfología y estructura de los agentes, como en los *path-followers* de Lobo [120]. En el Capítulo 4 se presenta un ejemplo en esta línea, pero más simple: sin proceso de desarrollo sino una traducción directa (aunque altamente heurística), y una tarea por resolver también muy simple (avanzar tan rápido como sea posible) en un entorno igualmente sencillo (un filamento rectilíneo). A pesar de esta simplicidad, coevolucionan diversas morfologías y patrones de funcionamiento. Globalmente, este modelo constituye un ejemplo bastante mínimo de coevolución de la estructura y de su sistema de control a través de computación morfológica.

El papel de las dinámicas y procesos evolutivos.
La literatura científica sobre la evolución de la diversidad es muy rica y se distribuye a través de varias disciplinas científicas. En el contexto, de las simulaciones basadas

en agentes, se han propuesto muchos modelos diferentes a lo largo de los últimos años. Muchos de ellos proponen agentes complejos en entornos ricos e igualmente complejos, con un detallado modelado de sus características, como en los famosos organismos artificiales de Sims [178]. Otros intentan modelar muchos de los detalles relevantes a la hora de construir comunidades ecológicas complejas [182, 220]. En estos casos, la diversidad resultante en los agentes que evolucionan en las simulaciones se puede atribuir en mayor o menor grado a las diferentes características y complejidades de los propios modelos. Aún en otros casos, se formulan modelos con agentes extremadamente simples, para permitir un análisis matemático exhaustivo de las propiedades de los modelos [26, 221], con el coste de modelar las características de los agentes a un nivel muy abstracto. Nuestra contribución en este área es el modelo presentado en el Capítulo 4, cuya características definitorias son un enfoque de abajo a arriba (los detalles de la simulación ecológica emergen de los detalles de los agentes) y la simplicidad del modelado a todos los niveles: en el genético, el fenotípico y el proceso de traducción entre ambos, el modelo reproductivo asexual y las interacciones ecológicas; pero todo ello sin llegar a un nivel excesivamente abstracto. A causa de esta simplicidad a todos los niveles, la evolución resultante de diversas morfologías puede atribuirse a las dinámicas evolutivas inducidas por las interacciones ecológicas entre los agentes, antes que a cualquier otro aspecto del modelo, como podría ocurrir en otros casos según ya se ha mencionado.

Bibliography

A DOI (Digital Object Identifier) is a permanent code which uniquely identifies a digital resource, used by some editorial organizations to secure a permanent digital identifier for journal articles, congress communications and book chapters. Through a web interface[3], DOIs can be used as permanent URLs for the corresponding digital resources.
In this bibliography, where possible, the DOI is provided as the last item of the reference. In the electronic version of this document, a DOI-based permanent web link can be accessed by clicking with the mouse over the title. The same is applicable for references without a DOI but with a reasonably stable web link.

[1] Abelson, H. and diSessa, A. *Turtle geometry: the computer as a medium for exploring mathematics*. The MIT Press, 1986.

[2] Ali, M. Y., Uemura, S., Adachi, K., Itoh, H., Kinosita, K., and Ishiwata, S. **Myosin V is a left-handed spiral motor on the right-handed actin helix.** *Nature Structural Biology*, **9**(6):464–467, 2002. DOI:10.1038/nsb803.

[3] Allen, M., Prusinkiewicz, P., and DeJong, T. **Using L-systems for modeling source-sink interactions, architecture and physiology of growing trees: the L-PEACH model**. *New Phytologist*, **166**(3):869–880, 2005. DOI:10.1111/j.1469-8137.2005.01348.x.

[4] Apostolico, A. and Giancarlo, R. **Sequence alignment in molecular biology**. *Journal of Computational Biology*, **5**:173–196, 1998. DOI:10.1089/cmb.1998.5.173.

[5] Arthur, W. *Biased embryos and evolution*. Cambridge University Press, 2004.

[3] http://dx.doi.org/

[6] Aspremont, A. and El Ghaoui, L. **Testing the nullspace property using semidefinite programming**. *Mathematical Programming*, **127**:123–144, 2010. DOI:10.1007/s10107-010-0416-0.

[7] Atilgan, A. R., Durell, S. R., Jernigan, R. L., Demirel, M. C., Keskin, O., and Bahar, I. **Anisotropy of fluctuation dynamics of proteins with an elastic network model.** *Biophysical Journal*, **80**(1):505–515, 2001. DOI:10.1016/S0006-3495(01)76033-X.

[8] Back, T., Hoffmeister, F., and Schwefel, H.-P. **A survey of evolution strategies**. In *Proceedings of the 4th International Conference on Genetic Algorithms*, pp. 2–9, 1991.

[9] Bahar, I. **Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential**. *Folding and Design*, **2**(3):173–181, 1997. DOI:10.1016/S1359-0278(97)00024-2.

[10] Bart-Smith, H., Moored, K. W., Taylor, S. A., and Bliss, T. K. **Optimization of a tensegrity wing for biomimetic applications**. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 2288–2293, 2006. DOI:10.1109/CDC.2006.377421.

[11] Batzoglou, S. **The many faces of sequence alignment**. *Briefings in Bioinformatics*, **6**(1):6–22, 2005. DOI:10.1093/bib/6.1.6.

[12] Bell, A. *Plant form: an illustrated guide to flowering plant morphology*, chapter **Vegetative multiplication**, p. 206. Oxford University Press, 1991.

[13] Bongard, J. *Incremental approaches to the combined evolution of a robot's body and brain*. Doctoral dissertation, Universitat Zurich, 2003.

[14] Borenstein, E. and Krakauer, D. C. **An end to endless forms: epistasis, phenotype distribution bias, and nonuniform evolution**. *PLoS Computational Biology*, **4**(10):e1000202, 2008. DOI:10.1371/journal.pcbi.1000202.

[15] Bornhofen, S. and Lattaud, C. **Life history evolution of virtual plants: trading off between growth and reproduction**. In *Proceedings of the*

9th International Conference on Parallel Problem Solving from Nature (PPSN), pp. 808–817. Springer, 2006. DOI:10.1007/11844297_82.

[16] Bornhofen, S. and Lattaud, C. **On hopeful monsters, neutral networks and junk code in evolving L-systems**. In *Proceedings of the 10th Genetic and Evolutionary Computation Conference (GECCO)*, pp. 193–200. ACM, 2008. DOI:10.1145/1389095.1389127.

[17] Bornhofen, S. and Lattaud, C. **Competition and evolution in virtual plant communities**. *Natural Computing*, **8**(2):349–385, 2009. DOI:10.1007/s11047-008-9089-5.

[18] Bowler, P. *Evolution: the history of an idea*. University of California Press, 2003.

[19] Bowling, A. and Palmer, A. F. **The small mass assumption applied to the multibody dynamics of motor proteins**. *Journal of Biomechanics*, **42**(9):1218–1223, 2009. DOI:10.1016/j.jbiomech.2009.03.017.

[20] Braitenberg, V. *Vehicles: experiments in synthetic psychology*. The MIT Press, 1984.

[21] Brangwynne, C., Mackintosh, F., Kumar, S., Geisse, N., Talbot, J., Mahadevan, L., Parker, K., Ingber, D., and Weitz, D. **Microtubules can bear enhanced compressive loads in living cells because of lateral reinforcement**. *Cell Biology*, **173**:733–745, 2006. DOI:10.1083/jcb.200601060.

[22] Bruckstein, A. M., Donoho, D. L., and Elad, M. **From sparse solutions of systems of equations to sparse modelling of signals and images**. *SIAM Review*, **51**(1):34–81, 2009. DOI:10.1137/060657704.

[23] Candès, E., Wakin, M., and Boyd, S. **Enhancing sparsity by reweighted L_1 minimization**. *Journal of Fourier Analysis and Applications*, **14**:877–905, 2008. DOI:10.1007/s00041-008-9045-x.

[24] Casas, M. and Vico, F. **On the performance of some bioinspired genetic operators in complex structures evolution**. In *Proceedings of the 11th*

Genetic and Evolutionary Computation Conference (GECCO), pp. 1841–1842, 2009. DOI:10.1145/1569901.1570193.

[25] Case, D. **Normal mode analysis of protein dynamics**. *Current Opinion in Structural Biology*, **4**(2):285–290, 1994. DOI:10.1016/S0959-440X(94)90321-2.

[26] Champagnata, N., Ferrierea, R., and Meleardb, S. **Unifying evolutionary dynamics: from individual stochastic processes to macroscopic models**. *Theoretical Population Biology*, **69**:297–321, 2006. DOI:10.1016/j.tpb.2005.10.004.

[27] Chaumont, N., Egli, R., and Adami, C. **Evolving virtual creatures and catapults**. *Artificial Life*, **13**(2):139–157, 2007. DOI:10.1162/artl.2007.13.2.139.

[28] Chen, C., Mrksich, M., Huang, S., Whitesides, G., and Ingber, D. **Geometric control of cell life and death**. *Science*, **276**(5317):1425–1428, 1997. DOI:10.1126/science.276.5317.1425.

[29] Chen, S. S., Donoho, D. L., and Saunders, M. A. **Atomic decomposition by basis pursuit**. *SIAM Review*, **43**(1):129–159, 2001. DOI:10.1137/S003614450037906X.

[30] Ciudad, A., Sancho, J., and Lacasta, A. **Dynamics of an inchworm nano-walker**. *Physica A: Statistical Mechanics and its Applications*, **371**(1):25–28, 2006. DOI:10.1016/j.physa.2006.04.099.

[31] Collado-Vides, L., Gómez-Alcaraz, G., Rivas-Lechuga, R., and Gómez-Gutierrez, V. **Simulation of the clonal growth of Bostrychia radicans (Ceramiales-Rhodophyta) using Lindenmayer systems**. *Biosystems*, **42**(1):19–27, 1997. DOI:10.1016/S0303-2647(96)01681-4.

[32] Connelly, R. and Whiteley, W. **Second-order rigidity and prestress stability for tensegrity frameworks**. *SIAM Journal on Discrete Mathematics*, **9**(3):453–491, 1996. DOI:10.1137/S0895480192229236.

[33] Corneil, D. G. and Fonlupt, J. **The complexity of generalized clique covering**. *Discrete Applied Mathematics*, **22**:109–118, 1989. DOI:10.1016/0166-218X(88)90086-8.

[34] Coughlin, M. F. and Stamenović, D. **A tensegrity structure with buckling compression elements: application to cell mechanics**. *Journal of Applied Mechanics*, **64**(3):480–486, 1997. DOI:10.1115/1.2788918.

[35] Craig, E. M. and Linke, H. **Mechanochemical model for myosin V**. *Proceedings of the National Academy of Sciences of the United States of America*, **106**(43):18261–18266, 2009. DOI:10.1073/pnas.0908192106.

[36] Cressman, A., Togashi, Y., Mikhailov, A. S., and Kapral, R. **Mesoscale modeling of molecular machines: cyclic dynamics and hydrodynamical fluctuations**. *Physical Review E*, **77**(5):050901+, 2008. DOI:10.1103/PhysRevE.77.050901.

[37] Davidson, E. H. **Emerging properties of animal gene regulatory networks**. *Nature*, **468**(7326):911–920, 2010. DOI:10.1038/nature09645.

[38] Dawkins, R. *The Selfish Gene*. Popular Science. Oxford University Press, 1989.

[39] de Jong, G. **Darwin@Home**. http://www.darwinathome.org/.

[40] Donoho, D. **For most large underdetermined systems of linear equations the minimal L_1-norm solution is also the sparsest solution**. *Communications on Pure and Applied Mathematics*, **59**(6):797–829, 2006. DOI:10.1002/cpa.20131.

[41] Doruker, P. and Jernigan, R. L. **Functional motions can be extracted from on-lattice construction of protein structures**. *Proteins: Structure, Function, and Genetics*, **53**(2):174–181, 2003. DOI:10.1002/prot.10486.

[42] Doursat, R. *Organic Computing*, chapter **Organically grown architectures: creating decentralized, autonomous systems by embryomorphic engineering**, pp. 167–199. Springer, 2008. DOI:10.1007/978-3-540-77657-4_8.

[43] Ebner, M. **Evolution and growth of virtual plants**. In *Lecture Notes in Computer Science*, pp. 228–237. Springer, 2003. DOI:10.1007/978-3-540-39432-7_25.

[44] Ebner, M. **Coevolution and the Red Queen effect shape virtual plants**. *Genetic Programming and Evolvable Machines*, **7**(1):103–123, 2006. DOI:10.1007/s10710-006-7013-2.

[45] Ebner, M., Grigore, A., Heffner, A., and Albert, J. **Coevolution produces an arms race among virtual plants**. In *Proceedings of the 4th European Conference on Genetic Programming (GECCO)*, pp. 114–170. Springer, 2002. DOI:10.1007/3-540-45984-7_31.

[46] Echave, J. **Evolutionary divergence of protein structure: the linearly forced elastic network model**. *Chemical Physics Letters*, **457**(4-6):413–416, 2008. DOI:10.1016/j.cplett.2008.04.042.

[47] Eggenberger, P. **Genome-physics interaction as a new concept to reduce the number of genetic parameters in artificial evolution**. In *Proceedings of the 5th IEEE Congress on Evolutionary Computation (CEC)*, **1**, pp. 191–198, 2003. DOI:10.1109/CEC.2003.1299574.

[48] Ewaschuk, R. and Turney, P. **Self-replication and self-assembly for manufacturing**. *Artificial Life*, **12**(3):411–433, 2006. DOI:10.1162/artl.2006.12.3.411.

[49] Eyal, E. and Bahar, I. **Toward a molecular understanding of the anisotropic response of proteins to external forces: insights from elastic network models.** *Biophysical Journal*, **94**(9):3424–3435, 2008. DOI:10.1529/biophysj.107.120733.

[50] Eyal, E., Yang, L.-W., and Bahar, I. **Anisotropic network model: systematic evaluation and a new web interface**. *Bioinformatics*, **22**(21):2619–2627, 2006. DOI:10.1093/bioinformatics/btl448.

[51] Federici, D. and Downing, K. **Evolution and development of a multicellular organism: scalability, resilience, and neutral complexification.** *Artificial Life*, **12**(3):381–409, 2006. DOI:10.1162/artl.2006.12.3.381.

[52] Fernández, J. D. and Orden, D. **On the atomic decomposition length of graphs and tensegrities**. In *Actas del Séptimo Encuentro de Matemática Discreta*, pp. 27–34, 2011.

[53] Fernández, J. D., Vico, F. J., and Doursat, R. **Complex and diverse morphologies can develop from a minimal genomic model**. In *Proceedings of the 14th Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 2012.

[54] Fernández, J. D. and Vico, F. J. **Automating the search of molecular motor templates by evolutionary methods**. *Biosystems*, **106**:82–93, 2011. DOI:10.1016/j.biosystems.2011.07.002.

[55] Fernández, J. D., Vico, F. J., Lobo, D., Martín, G. M., and Doursat, R. **Emergent diversity in an open-ended evolving virtual community**. *Artificial Life*, **18**(2):(pages to be determined), 2012 (accepted 2011). DOI:10.1162/artl_a_00059.

[56] Flechsig, H. and Mikhailov, A. **Elastic network modeling of molecular motor HCV helicase: inchworm translocation and DNA unzipping cycles**. In *Proceedings of the 29th Dynamic Days Europe Conference (session on Networks and Time Series)*, p. 6, 2009.

[57] Flechsig, H. and Mikhailov, A. S. **Tracing entire operation cycles of molecular motor hepatitis C virus helicase in structurally resolved dynamical simulations.** *Proceedings of the National Academy of Sciences of the United States of America*, **107**(49):20875–80, 2010. DOI:10.1073/pnas.1014631107.

[58] Floreano, D. and Keller, L. **Evolution of adaptive behaviour in robots by means of Darwinian selection**. *PLoS Biology*, **8**(1):e1000292, 2010. DOI:10.1371\%2Fjournal.pbio.1000292.

[59] Fogel, L. J., Owens, A. J., and Walsh, M. J. *Artificial intelligence through simulated evolution*. John Wiley, 1966.

[60] Forgács, G. and Newman, S. *Biological physics of the developing embryo*. Cambridge University Press, 2005.

[61] Foucart, S. and Lai, M.-J. **Sparsest solutions of underdetermined linear systems via L_q-minimization for** $0 < q <= 1$. *Applied and Computational Harmonic Analysis*, **26**(3):395–407, 2009. DOI:10.1016/j.acha.2008.09.001.

[62] Frandsen, G. S. and Frandsen, P. F. **Dynamic matrix rank**. *Theoretical Computer Science*, **410**:4085–4093, 2009. DOI:10.1016/j.tcs.2009.06.012.

[63] Freeman, S. and Herron, J. *Evolutionary Analysis*, chapter **14**, p. 562 (box 14.1). Pearson Prentice Hall, 2003.

[64] Fuchs, J.-J. **Sparsity and uniqueness for some specific under-determined linear systems**. In *Proceedings of the 30th IEEE International Conference on Acoustics, Speech, and Signal Processing*, **5**, pp. 729–732, 2005. DOI:10.1109/ICASSP.2005.1416407.

[65] Fuller, B. *Synergetics: explorations in the geometry of thinking*. Macmillan, 1975.

[66] Gautier, H., Mech, R., Prusinkiewicz, P., and Varlet-Grancher, C. **3D architectural modelling of aerial photomorphogenesis in white clover (Trifolium repens L.) using L-systems**. *Annals of Botany*, **85**:359–370, 2000. DOI:10.1006/anbo.1999.1069.

[67] Gendreau, M. and Potvin, J.-Y., editors. *Handbook of metaheuristics*. Springer, 2010.

[68] Gennerich, A. and Vale, R. D. **Walking the walk: how kinesin and dynein coordinate their steps.** *Current Opinion in Cell Biology*, **21**(1):59–67, 2009. DOI:10.1016/j.ceb.2008.12.002.

[69] Goldstein, L. S. B. **Molecular motors: from one motor many tails to one motor many tales**. *Trends in Cell Biology*, **11**(12):477–482, 2001. DOI:10.1016/S0962-8924(01)02143-2.

[70] Goodwin, B. *How the leopard changed its spots: the evolution of complexity*. Princeton University Press, 2001.

[71] Goodwin, B. *How the leopard changed its spots: the evolution of complexity*, chapter **Living form in the making**, pp. 77–114. In [70], 2001.

[72] Gould, S. *The structure of evolutionary theory*. Belknap Press of Harvard University Press, 2002.

[73] Graells Rovira, A. and Mirats Tur, J. M. **Control and simulation of a tensegrity-based mobile robot**. *Robotics and Autonomous Systems*, **57**(5):526–535, 2009. DOI:10.1016/j.robot.2008.10.010.

[74] Graver, J. E., Servatius, B., and Servatius, H. *Combinatorial Rigidity*, **volume 2** of *Graduate Studies in Mathematics*. American Mathematical Society, 1993.

[75] Gribonval, R. and Nielsen, M. **Sparse representations in unions of bases**. *IEEE Transactions on Information Theory*, **49**(12):3320–3325, 2003. DOI:10.1109/TIT.2003.820031.

[76] Grimm, V. and Railsback, S. F. *Individual-based modeling and ecology*. Princeton Series in Theoretical and Computational Biology. Princeton University Press, 2005.

[77] Gruenert, G., Ibrahim, B., Lenser, T., Lohel, M., Hinze, T., and Dittrich, P. **Rule-based spatial modeling with diffusing, geometrically constrained molecules**. *BMC Bioinformatics*, **11**(1):307+, 2010. DOI:10.1186/1471-2105-11-307.

[78] Guzman, M. and Orden, D. **From graphs to tensegrity structures: geometric and symbolic approaches**. *Publicacions Matematiques*, **50**:279–299, 2006. DOI:10.5565/PUBLMAT_50206_02.

[79] Haliloglu, T., Bahar, I., and Erman, B. **Gaussian dynamics of folded proteins**. *Physical Review Letters*, **79**(16):3090–3093, 1997. DOI:10.1103/PhysRevLett.79.3090.

[80] Henderson, J. and Carter, D. **Mechanical induction in limb morphogenesis: the role of growth-generated strains and pressures**. *Bone*, **31**(6):645–653, 2002. DOI:10.1016/S8756-3282(02)00911-0.

[81] Hills, R. D., Lu, L., and Voth, G. A. **Multiscale coarse-graining of the protein energy landscape**. *PLoS Computational Biology*, **6**(6):e1000827+, 2010. DOI:10.1371/journal.pcbi.1000827.

[82] Hinsen, K. **Analysis of domain motions by approximate normal mode calculations**. *Proteins: Structure, Function, and Genetics*, **33**(3):417–429,

1998. DOI: 10.1002/(SICI)1097-0134(19981115)33:3<417::AID-PROT10>3.0.CO;2-8.

[83] Hirakawa, E., Higuchi, H., and Toyoshima, Y. Y. **Processive movement of single 22S dynein molecules occurs only at low ATP concentrations**. *Proceedings of the National Academy of Sciences of the United States of America*, **97**(6):2533–2537, 2000. DOI:10.1073/pnas.050585297.

[84] Holland, J. *Adaptation in natural and artificial systems*. University of Michigan Press, 1975.

[85] Hornby, G. S., Lipson, H., and Pollack, J. B. **Generative representations for the automated design of modular physical robots**. *IEEE Transactions on Robotics and Automation*, **19**(4):703–719, 2003. DOI:10.1109/TRA.2003.814502.

[86] Houdusse, A. **Myosin motors: missing structures and hidden springs**. *Current Opinion in Structural Biology*, **11**(2):182–194, 2001. DOI:10.1016/S0959-440X(00)00188-3.

[87] Hughes, A. and Lambert, D. **Functionalism, structuralism, and ways of seeing**. *Journal of Theoretical Biology*, **111**(4):787–800, 1984. DOI:10.1016/S0022-5193(84)80267-2.

[88] Ijspeert, A. J. and Kodjabachian, J. **Evolution and development of a central pattern generator for the swimming of a lamprey**. *Artificial Life*, **5**(3):247–269, 1999. DOI:10.1162/106454699568773.

[89] Ingber, D. **Cellular tensegrity: defining new rules of biological design that govern the cytoskeleton**. *Journal of Cell Science*, **104 (Pt 3)**:613–627, 1993.

[90] Ingber, D. **Cellular mechanotransduction: putting all the pieces together again**. *The FASEB Journal: Official Publication of the Federation of American Societies for Experimental Biology*, **20**(7):811–827, 2006. DOI:10.1096/fj.05-5424rev.

[91] Ingber, D. **Mechanical control of tissue morphogenesis during embryological development**. *Developmental Biology*, **50**:255–266, 2006. DOI:10.1387/ijdb.052044di.

[92] Jacob, C. **Genetic L-system programming**. In *Proceedings of the 3rd International Conference on Parallel Problem Solving from Nature (PPSN)*, pp. 334–343. Springer, 1994. DOI:10.1007/3-540-58484-6_277.

[93] Jacob, C. **Evolving evolution programs: genetic programming and L-systems**. In *Proceedings of the 1st European Conference on Genetic Programming*, pp. 107–115, 1996.

[94] Jacob, C. *Illustrating evolutionary computation with Mathematica*. Morgan Kaufmann, 2001.

[95] Jansen, V. and Mulder, G. **Evolving biodiversity**. *Ecology Letters*, **2**:379–386, 1999. DOI:10.1046/j.1461-0248.1999.00100.x.

[96] Juan, S. and Miratstur, J. **Tensegrity frameworks: static analysis review**. *Mechanism and Machine Theory*, **43**(7):859–881, 2008. DOI:10.1016/j.mechmachtheory.2007.06.010.

[97] Jülicher, F., Ajdari, A., and Prost, J. **Modeling molecular motors**. *Reviews of Modern Physics*, **69**(4):1269–1282, 1997. DOI:10.1103/RevModPhys.69.1269.

[98] Kauffman, S. A. *The origins of order: self-organization and selection in evolution*. Oxford University Press, USA, 1993.

[99] Kawakubo, T., Okada, O., and Minami, T. **Molecular dynamics simulations of evolved collective motions of atoms in the myosin motor domain upon perturbation of the ATPase pocket**. *Biophysical Chemistry*, **115**(1):77–85, 2005. DOI:10.1016/j.bpc.2004.12.049.

[100] Kim, M. **Elastic models of conformational transitions in macromolecules**. *Journal of Molecular Graphics and Modelling*, **21**(2):151–160, 2002. DOI:10.1016/S1093-3263(02)00143-2.

[101] Kodjabachian, J. and Meyer, J.-A. **Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance**

in artificial insects. *IEEE Transactions on Neural Networks*, **9**(5):796–812, 1998. DOI:10.1109/72.712153.

[102] Komosinski, M. and Rotaru-Varga, A. **Comparison of different genotype encodings for simulated 3D agents**. *Artificial Life*, **7**(4):395–418, 2001. DOI:10.1162/106454601317297022.

[103] Komosinski, M. and Ulatowski, S. **Framsticks: towards a simulation of a nature-like world, creatures and evolution**. In Floreano, D., Nicoud, J.-D., and Mondada, F., editors, *Proceedings of the 5th European Conference on Advances in Artificial Life (ECAL)*, pp. 261–265. Springer-Verlag, Springer-Verlag, 1999. DOI:10.1007/3-540-48304-7_33.

[104] Koppole, S., Smith, J. C., and Fischer, S. **Simulations of the myosin II motor reveal a nucleotide-state sensing element that controls the recovery stroke**. *Journal of Molecular Biology*, **361**(3):604–616, 2006. DOI:10.1016/j.jmb.2006.06.022.

[105] Kou, L. T., Stockmeyer, L. J., and Wong, C. K. **Covering edges by cliques with regard to keyword conflicts and intersection graphs**. *Communications of the ACM*, **21**:135–139, 1978. DOI:10.1145/359340.359346.

[106] Kowaliw, T. and Banzhaf., W. **Augmenting artificial development with local fitness**. In *Proceedings of the 11th IEEE Congress on Evolutionary Computation (CEC)*, pp. 316–323, 2009.

[107] Koza, J. R. *Genetic programming: On the programming of computers by means of natural selection (complex adaptive systems)*. The MIT Press, 1992.

[108] Koza, J. **Discovery of rewrite rules in Lindenmayer systems and state transition rules in cellular automata via genetic programming**. In *Symposium on Pattern Formation*, pp. 1–19, 1993.

[109] Kull, F. J., Vale, R. D., and Fletterick, R. J. **The case for a common ancestor: kinesin and myosin motor proteins and G proteins**. *Journal of Muscle Research and Cell Motility*, **19**(8):877–886, 1998. DOI:10.1023/A:1005489907021.

[110] Kundu, S., Sorensen, D. C., Phillips, G. N., and Jr. **Automatic domain decomposition of proteins by a Gaussian Network Model**. *Proteins: Structure, Function, and Bioinformatics*, **57**(4):725–733, 2004. DOI:10.1002/prot.20268.

[111] Lai, M. **On sparse solutions of underdetermined linear systems**. *Journal of Concrete and Applicable Mathematics*, **8**:296–327, 2010.

[112] Laland, K. N., Sterelny, K., Odling-Smee, J., Hoppitt, W., and Uller, T. **Cause and effect in biology revisited: is Mayrs proximate-ultimate dichotomy still useful?** *Science*, **334**:1512–1516, 2011. DOI:10.1126/science.1210879.

[113] Larson, E. *Evolution: the remarkable history of a scientific theory.* Modern Library Chronicles, 2006.

[114] Leo-Macias, A., Lopez-Romero, P., Lupyan, D., Zerbino, D., and Ortiz, A. R. **An analysis of core deformations in protein superfamilies.** *Biophysical Journal*, **88**(2):1291–1299, 2005. DOI:10.1529/biophysj.104.052449.

[115] Levin, S., Grenfell, B., Hastings, A., and Perelson, A. **Mathematical and computational challenges in population biology and ecosystems science**. *Science*, **275**(5298):334–343, 1997. DOI:10.1126/science.275.5298.334.

[116] Li, Y. and Amari, S.-I. **Two conditions for equivalence of L_0-norm solution and L_1-norm solution in sparse representation**. *IEEE Transactions on Neural Networks*, **21**:1189–1196, 2010. DOI:10.1109/TNN.2010.2049370.

[117] Lindenmayer, A. **Mathematical models for cellular interaction in development: Parts I and II**. *Journal of Theoretical Biology*, **18**(3):280–315, 1968. DOI:10.1016/0022-5193(68)90080-5.

[118] Lister, I., Schmitz, S., Walker, M., Trinick, J., Buss, F., Veigel, C., and Kendrick-Jones, J. **A monomeric myosin VI with a large working stroke**. *The EMBO Journal*, **23**(8):1729–1738, 2004. DOI:10.1038/sj.emboj.7600180.

[119] Liu, Y., Scolari, M., Im, W., and Woo, H.-J. **Protein-protein interactions in actin-myosin binding and structural effects of R405Q mutation:**

a molecular dynamics study. *Proteins: Structure, Function, and Bioinformatics*, **64**(1):156–166, 2006. DOI:10.1002/prot.20993.

[120] Lobo, D. *Evolutionary development based on genetic regulatory models for behavior-finding*. Doctoral dissertation, Universidad de Malaga, 2010.

[121] Lobo, D. and Vico, F. J. **Evolutionary development of tensegrity structures**. *Biosystems*, **101**(3):167–176, 2010. DOI:10.1016/j.biosystems.2010.06.005.

[122] Lu, M. **The role of shape in determining molecular motions**. *Biophysical Journal*, **89**(4):2395–2401, 2005. DOI:10.1529/biophysj.105.065904.

[123] Malevanets, A. and Kapral, R. **Mesoscopic model for solvent dynamics**. *Journal of Chemical Physics*, **110**:8605–8613, 1999. DOI:10.1063/1.478857.

[124] Malevanets, A. and Kapral, R. **Solute molecular dynamics in a mesoscale solvent**. *Journal of Chemical Physics*, **112**:7260–7269, 2000. DOI:10.1063/1.481289.

[125] Malioutov, D., Cetin, M., and Willsky, A. **Optimal sparse representations in general overcomplete bases**. In *Proceedings of the 29th IEEE International Conference on Acoustics, Speech, and Signal Processing*, **2**, pp. 793–796, 2004. DOI:10.1109/ICASSP.2004.1326377.

[126] Mammoto, T. and Ingber, D. E. **Mechanical control of tissue and organ development.** *Development*, **137**(9):1407–1420, 2010. DOI:10.1242/dev.024166.

[127] Martín, G. M. *Evolving complexity and similarity in an Artifcial Life framework based on formal language theory*. Doctoral dissertation, Universidad de Málaga, 2010.

[128] Matsushita, K., Lungarella, M., Paul, C., and Yokoi, H. **Locomoting with less computation but more morphology**. In *Proceedings of the 22nd IEEE International Conference on Robotics and Automation.*, pp. 2008–2013, 2005. DOI:10.1109/ROBOT.2005.1570408.

[129] Mayr, E. **Cause and effect in biology**. *Science*, **134**:1501–1506, 1961. DOI:10.1126/science.134.3489.1501.

[130] Mayr, E. *What evolution is*. Basic Books, 2001.

[131] McCormack, J. *Complex systems: from biology to computation*, chapter **Interactive evolution of L-system grammars for computer graphics modelling**, pp. 118–130. IOS Press, 1993.

[132] McOwan, P. and Burton, E. *Artificial Life models in software*, chapter **Sodarace: continuing adventures in Artificial Life**, pp. 61–78. Springer-Verlag New York, 2009.

[133] Micheletti, C., Carloni, P., and Maritan, A. **Accurate and efficient description of protein vibrational dynamics: comparing molecular dynamics and Gaussian models**. *Proteins: Structure, Function, and Bioinformatics*, **55**(3):635–645, 2004. DOI:10.1002/prot.20049.

[134] Ming, D. and Wall, M. E. **Allostery in a coarse-grained model of protein dynamics**. *Physical Review Letters*, **95**(19):198103+, 2005. DOI:10.1103/PhysRevLett.95.198103.

[135] Mirny, L. A., Abkevich, V. I., and Shakhnovich, E. I. **How evolution makes proteins fold quickly.** *Proceedings of the National Academy of Sciences of the United States of America*, **95**(9):4976–4981, 1998. DOI:10.1073/pnas.95.9.4976.

[136] Mock, K. **Wildwood: the evolution of L-system plants for virtual environments**. In *Proceedings of the 2nd IEEE World Congress on Computational Intelligence*, pp. 476–480, 1998. DOI:10.1109/ICEC.1998.699854.

[137] Moritsugu, K. and Smith, J. **Coarse-grained biomolecular simulation with REACH: realistic extension algorithm via covariance hessian**. *Biophysical Journal*, **93**(10):3460–3469, 2007. DOI:10.1529/biophysj.107.111898.

[138] Motro, R. *Tensegrity: structural systems for the future*. Butterworth-Heinemann, 2003.

[139] Muñoz, J., Conte, V., and Miodownik, M. **Stress-dependent morphogenesis: continuum mechanics and truss systems**. *Biomechanics and Modeling in Mechanobiology*, **9**(4):451–467, 2010. DOI:10.1007/s10237-009-0187-9.

[140] Müller, G. and Newman, S., editors. *Origination of organismal form: beyond the gene in developmental and evolutionary biology*. The Vienna series in theoretical biology. MIT Press, 2003.

[141] Munoz, E. and Deem, M. W. **Amino acid alphabet size in protein evolution experiments: better to search a small library thoroughly or a large library sparsely?** *Protein Engineering, Design and Selection*, **21**(5):311–317, 2008. DOI:10.1093/protein/gzn007.

[142] Natarajan, B. K. **Sparse approximate solutions to linear systems**. *SIAM Journal on Computing*, **24**:227–234, 1995. DOI:10.1137/S0097539792240406.

[143] Nelson, C., Jean, R., Tan, J., Liu, W., Sniadecki, N., Spector, A., and Chen, C. **Emergent patterns of growth controlled by multicellular form and mechanics**. *Proceedings of the National Academy of Sciences of the United States of America*, **102**:11594–11599, 2005. DOI:10.1073/pnas.0505939102.

[144] Newman, S. *Origination of organismal form: beyond the gene in developmental and evolutionary biology*, chapter **From physics to development: the evolution of morphogenetic mechanisms**, pp. 221–240. In Muller and Newman [140], 2003.

[145] Nowak, M. A. *Evolutionary dynamics: exploring the equations of life*. Belknap Press of Harvard University Press, 2006.

[146] Nusslein-Volhard, C. and Wieschaus, E. **Mutations affecting segment number and polarity in *drosophila***. *Nature*, **287**(5785):795–801, 1980. DOI:10.1038/287795a0.

[147] Ochoa, G. **On genetic algorithms and Lindenmayer systems**. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN)*, pp. 335–344, 1998. DOI:10.1007/BFb0056876.

[148] Odell, G., Oster, G., Alberch, P., and Burnside, B. **The mechanical basis of morphogenesis: epithelial folding and invagination**. *Developmental Biology*, **85**(2):446–462, 1981. DOI:10.1016/0012-1606(81)90276-1.

[149] Okada, Y. and Hirokawa, N. **A processive single-headed motor: kinesin superfamily protein KIF1A**. *Science*, **283**(5405):1152–1157, 1999. DOI:10.1126/science.283.5405.1152.

[150] Ortiz, A. and Skolnick, J. **Sequence evolution and the mechanism of protein folding**. *Biophysical Journal*, **79**(4):1787–1799, 2000. DOI:10.1016/S0006-3495(00)76430-7.

[151] Pachepsky, E., Taylor, T., and Jones, S. **Mutualism promotes diversity and stability in a simple artificial ecosystem**. *Artificial Life*, **8**:5–24, 2002. DOI:10.1162/106454602753694747.

[152] Parker, D., Bryant, Z., and Delp, S. **Coarse-grained structural modeling of molecular motors using multibody dynamics**. *Cellular and Molecular Bioengineering*, **2**(3):366–374, 2009. DOI:10.1007/s12195-009-0084-4.

[153] Paul, C., Cuevas, V. F., and Lipson, H. **Design and control of tensegrity robots for locomotion**. *IEEE Transactions on Robotics*, **22**(5):944–957, 2006. DOI:10.1109/TRO.2006.878980.

[154] Paul, C. **Morphological computation: a basis for the analysis of morphology and control requirements**. *Robotics and Autonomous Systems*, **54**(8):619–630, 2006. DOI:10.1016/j.robot.2006.03.003.

[155] Paul, C., Lipson, H., and Valero Cuevas, F. J. **Evolutionary form-finding of tensegrity structures**. In *Proceedings of the 7th Genetic and Evolutionary Computation Conference (GECCO)*, pp. 3–10. ACM, 2005. DOI:10.1145/1068009.1068011.

[156] Petrone, P. and Pande, V. **Can conformational change be described by only a few normal modes?** *Biophysical Journal*, **90**(5):1583–1593, 2006. DOI:10.1529/biophysj.105.070045.

[157] Pfeifer, R., Iida, F., and Gomez, G. **Morphological computation for adaptive behavior and cognition**. *International Congress Series*, **1291**:22–29, 2006. DOI:10.1016/j.ics.2005.12.080.

[158] Pollack, J. B., Lipson, H., Hornby, G., and Funes, P. **Three generations of automatically designed robots**. *Artificial Life*, **7**:215–223, 2001. DOI:10.1162/106454601753238627.

[159] Prusinkiewicz, P. **Graphical applications of L-systems**. In *Proceedings on Graphics Interface 86 / Vision Interface 86*, pp. 247–253. Canadian Information Processing Society, 1986.

[160] Prusinkiewicz, P. and Lindenmayer, A. *The algorithmic beauty of plants*. Springer-Verlag, 1990.

[161] Rader, A. J., Chennubhotla, C., Yang, L.-W., and Bahar, I. *The Gaussian Network Model: theory and applications*, chapter **3**, pp. 41–63. Chapman & Hall/CRC, 2006.

[162] Raibert, M. H. and Hodgins, J. K. **Animation of dynamic legged locomotion**. In *Proceedings of the 18th Annual Conference on computer graphics and interactive techniques (SIGGRAPH)*, **25**, pp. 349–358. ACM, 1991. DOI:10.1145/127719.122755.

[163] Ray, T. S. **An approach to the synthesis of life**. In *Proceedings of the 2nd International Workshop on the Synthesis and Simulation of Living Systems (Artificial Life II)*, pp. 371–408. Addison-Wesley, 1991.

[164] Reynolds, C. W. **Flocks, herds and schools: a distributed behavioral model**. In *Proceedings of the 14th Annual Conference on computer graphics and interactive techniques (SIGGRAPH)*, **21**, pp. 25–34. ACM, 1987. DOI:10.1145/37401.37406.

[165] Rieffel, J., Cuevas, F. V., and Lipson, H. **Automated discovery and optimization of large irregular tensegrity structures**. *Computers & Structures*, **87**(5-6):368–379, 2009. DOI:10.1016/j.compstruc.2008.11.010.

[166] Rieffel, J. A., Valero-Cuevas, F. J., and Lipson, H. **Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion**. *Journal of The Royal Society Interface*, **7**(45):613–621, 2010. DOI:10.1098/rsif.2009.0240.

[167] Rosenzweig, M. *Species diversity in space and time*. Cambridge University Press, 1995.

[168] Runqiang, B., Chen, P., Burrage, K., Hanan, J., Room, P., and Belward, J. *Developments in applied Artificial Intelligence*, chapter **Derivation of L-system models from measurements of biological branching structures using genetic algorithms**, pp. 321–324. Springer, 2002.

[169] Sablin, E. **Nucleotide switches in molecular motors: structural analysis of kinesins and myosins**. *Current Opinion in Structural Biology*, **11**(6):716–724, 2001. DOI:10.1016/S0959-440X(01)00265-2.

[170] Sarkozy, M. **Evolution of genetic potential**. *PLoS Computational Biology*, **1**(3):e32, 2005. DOI:10.1371/journal.pcbi.0010032.

[171] Savin, T., Kurpios, N. A., Shyer, A. E., Florescu, P., Liang, H., Mahadevan, L., and Tabin, C. J. **On the growth and form of the gut**. *Nature*, **476**(7358):57–62, 2011. DOI:10.1038/nature10277.

[172] Schilstra, M. J. and Martin, S. R. **An elastically tethered viscous load imposes a regular gait on the motion of myosin-V. simulation of the effect of transient force relaxation on a stochastic process**. *Journal of The Royal Society Interface*, **3**(6):153–165, 2006. DOI:10.1098/rsif.2005.0098.

[173] Schliwa, M. and Woehlke, G. **Molecular motors**. *Nature*, **422**(6933):759–765, 2003. DOI:10.1038/nature01601.

[174] Sellers, J. and Veigel, C. **Walking with myosin V**. *Current Opinion in Cell Biology*, **18**(1):68–73, 2006. DOI:10.1016/j.ceb.2005.12.014.

[175] Shim, Y.-S. and Kim, C.-H. **Generating flying creatures using body-brain co-evolution**. In *Proceedings of the Symposium on Computer Animation (part*

of the 30th SIGGRAPH), SCA '03, pp. 276–285, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[176] Simona-Mariana, C. and Gabriela-Catalina, B. **Tensegrity applied to modelling the motion of viruses**. *Acta Mechanica Sinica*, **27**:125–129, 2011. DOI:10.1007/s10409-011-0402-7.

[177] Sims, K. **Evolving 3D morphology and behavior by competition**. *Artificial Life*, **1**(4):353–372, 1994. DOI:10.1162/artl.1994.1.353.

[178] Sims, K. **Evolving virtual creatures**. In *Proceedings of the 21st Annual Conference on computer graphics and interactive techniques (SIGGRAPH)*, pp. 15–22. ACM, 1994. DOI:10.1145/192161.192167.

[179] Smith, J. *Evolutionary Genetics*. Oxford University Press, 1998.

[180] Snelson, K. **Continuous tension, discontinuous compression structure (US patent 3169611)**, 1965.

[181] Souza, T. R., Fonseca, S. T., Gonsalves, G. G., Ocarino, J. M., and Mancini, M. C. **Prestress revealed by passive cotension at the ankle joint**. *Journal of Biomechanics*, **42**(14):2374–2380, 2009. DOI:10.1016/j.jbiomech.2009.06.033.

[182] Spector, L., Klein, J., and Feinstein, M. **Division blocks and the open-ended evolution of development, form, and behavior**. In *Proceedings of the 9th Genetic and Evolutionary Computation Conference (GECCO)*, pp. 316–323. ACM, 2007. DOI:10.1145/1276958.1277019.

[183] Stanley, K. and Miikkulainen, R. **A taxonomy for artificial embryogeny**. *Artificial Life*, **9**(2):93–130, 2003. DOI:10.1162/106454603322221487.

[184] Sterelny, K. *Dawkins vs. Gould: survival of the fittest*. Icon Books, 2007.

[185] Struhl, G. **A homoeotic mutation transforming leg to antenna in *drosophila***. *Nature*, **292**(5824):635–638, 1981. DOI:10.1038/292635a0.

[186] Studer, M., Ritschard, G., Gabadinho, A., and Müller, N. S. *Advances in knowledge discovery and management*, chapter **Discrepancy analysis of complex objects using dissimilarities**, pp. 3–19. Springer-Verlag, 2010.

[187] Su, J. **Protein unfolding behavior studied by elastic network model**. *Biophysical Journal*, **94**(12):4586–4596, 2008. DOI:10.1529/biophysj.107.121665.

[188] Takano, M., Higo, J., Nakamura, H. K., and Sasai, M. **On the model granularity to simulate protein dynamics: a biological physics view on biomolecular computing**. *Natural Computing*, **3**(4):377–393, 2004. DOI:10.1007/s11047-004-2639-6.

[189] Tan, J., Gu, Y., Turk, G., and Liu, C. K. **Articulated swimming creatures**. In *Proceedings of the 38th Annual Conference on computer graphics and interactive techniques (SIGGRAPH)*, pp. 58:1–58:12. ACM, 2011. DOI:10.1145/1964921.1964953.

[190] Taverna, D. M. and Goldstein, R. A. **Why are proteins marginally stable?** *Proteins*, **46**(1):105–109, 2002. DOI:10.1002/prot.10016.

[191] Taylor, W. R. and Katsimitsoulia, Z. **A coarse-grained molecular model for actin-myosin simulation**. *Journal of Molecular Graphics and Modelling*, **29**:266–279, 2010. DOI:10.1016/j.jmgm.2010.06.004.

[192] Terzopoulos, D., Tu, X., and Grzeszczuk, R. **Artificial fishes: autonomous locomotion, perception, behavior, and learning in a simulated physical world**. *Artificial Life*, **1**(4):327–351, 1994. DOI:10.1162/artl.1994.1.327.

[193] Thom, R. *Structural stability and morphogenesis*. Addison Wesley Publishing Company, 1989.

[194] Thompson, D. *On Growth and Form*. University press, 1917.

[195] Tibert, A. and Pellegrino, S. **Review of form-finding methods for tensegrity structures**. *International Journal of Space Structures*, **18**:209–223, 2003. DOI:10.1260/026635103322987940.

[196] Tirion, M. M. **Large amplitude elastic motions in proteins from a single-parameter, atomic analysis**. *Physical Review Letters*, **77**(9):1905–1908, 1996. DOI:10.1103/PhysRevLett.77.1905.

[197] Togashi, Y. and Mikhailov, A. S. **Nonlinear relaxation dynamics in elastic networks and design principles of molecular machines**. *Proceedings of the National Academy of Sciences of the United States of America*, **104**(21):8697–8702, 2007. DOI:10.1073/pnas.0702950104.

[198] Togashi, Y., Yanagida, T., and Mikhailov, A. S. **Nonlinearity of mechanochemical motions in motor proteins**. *PLoS Computational Biology*, **6**(6):e1000814, 2010. DOI:10.1371/journal.pcbi.1000814.

[199] Toprak, E., Veres, A., Michel, J.-B., Chait, R., Hartl, D., and Kishony, R. **Evolutionary paths to antibiotic resistance under dynamically sustained drug selection**. *Nature Genetics*, **44**:101–105, 2012. DOI:10.1038/ng.1034.

[200] Toussaint, M. **Demonstrating the evolution of complex genetic representations: an evolution of artificial plants**. In *Proceedings of the 5th Genetic and Evolutionary Computation Conference (GECCO)*, **2723**, p. 200. Springer Berlin Heidelberg, 2003. DOI:10.1007/3-540-45105-6_8.

[201] Tsaig, Y. and Donoho, D. L. **Breakdown of equivalence between the minimal L_1-norm solution and the sparsest solution**. *Signal Processing*, **86**(3):533–548, 2006. DOI:10.1016/j.sigpro.2005.05.028.

[202] Vale, R. D. **The molecular motor toolbox for intracellular transport**. *Cell*, **112**(4):467–480, 2003. DOI:10.1016/S0092-8674(03)00111-9.

[203] Vale, R. D. and Milligan, R. A. **The way things move: looking under the hood of molecular motor proteins**. *Science*, **288**(5463):88–95, 2000. DOI:10.1126/science.288.5463.88.

[204] Panne, M. van de and Fiume, E. **Sensor-actuator networks**. In *Proceedings of the 20th Annual Conference on computer graphics and interactive techniques (SIGGRAPH)*, pp. 335–342. ACM, 1993. DOI:10.1145/166117.166159.

[205] Heuvel, M. G. L. van den and Dekker, C. **Motor proteins at work for nanotechnology**. *Science*, **317**(5836):333–336, 2007. DOI:10.1126/science.1139570.

[206] Waddington, C. H. **Genetic assimilation of the bithorax phenotype.** *Evolution*, **1**:1–13, 1956.

[207] Wampler, K. and Popović, Z. **Optimal gait and form for animal locomotion**. *ACM Transactions on Graphics*, **28**(3):1–8, 2009. DOI:10.1145/1531326.1531366.

[208] Wang, Z., Feng, M., Zheng, W., and Fan, D. **Kinesin is an evolutionarily fine-tuned molecular ratchet-and-pawl device of decisively locked direction**. *Biophysical Journal*, **93**(10):3363–3372, 2007. DOI:10.1529/biophysj.107.108233.

[209] Warshaw, D. **Differential labeling of myosin V heads with quantum dots allows direct visualization of hand-over-hand processivity**. *Biophysical Journal*, **88**(5):L30–L32, 2005. DOI:10.1529/biophysj.105.061903.

[210] Watson, J., Hanan, J., and Wiles, J. **Modeling the fitness of plant morphologies across three levels of complexity**. *Biosystems*, **94**(1-2):182–190, 2008. DOI:10.1016/j.biosystems.2008.05.023.

[211] West-Eberhard, M. J. **Phenotypic plasticity and the origins of diversity**. *Annual Review of Ecology and Systematics*, **20**(1):249–278, 1989. DOI:10.1146/annurev.es.20.110189.001341.

[212] Wilke, C., Wang, J., Ofria, C., Lenski, R., and Adami, C. **Evolution of digital organisms at high mutation rates leads to survival of the flattest**. *Nature*, **412**:331–333, 2001. DOI:10.1038/35085569.

[213] Wilke, C. **Adaptive evolution on neutral networks**. *Bulletin of Mathematical Biology*, **63**:715–730, 2001. DOI:10.1006/bulm.2001.0244.

[214] Wilke, C. O. and Adami, C. **Evolution of mutational robustness**. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, **522**(1-2):3–11, 2003. DOI:10.1016/S0027-5107(02)00307-X.

[215] Wilson, R. J. **Simulating the kinesin walk: a small step towards understanding dementia**. *European Symposium on Computer Modeling and Simulation (UKSIM)*, **0**:226–231, 2008. DOI:10.1109/EMS.2008.49.

[216] Wilson, R. J. **Kinesin's walk: springy or gated head coordination?** *Biosystems*, **96**(2):121–126, 2009. DOI:10.1016/j.biosystems.2008.12.002.

[217] Woods, R. J., Barrick, J. E., Cooper, T. F., Shrestha, U., Kauth, M. R., and Lenski, R. E. **Second-order selection for evolvability in a large *escherichia coli* population**. *Science*, **331**(6023):1433–1436, 2011. DOI:10.1126/science.1198914.

[218] Wroe, R., Bornberg-Bauer, E., and Chan, H. S. S. **Comparing folding codes in simple heteropolymer models of protein evolutionary landscape: robustness of the superfunnel paradigm.** *Biophysical Journal*, **88**(1):118–131, 2005. DOI:10.1529/biophysj.104.050369.

[219] Xia, Y. and Levitt, M. **Simulating protein evolution in sequence and structure space**. *Current Opinion in Structural Biology*, **14**(2):202–207, 2004. DOI:10.1016/j.sbi.2004.03.001.

[220] Yaeger, L. **Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or PolyWorld: life in a new context**. In *Proceedings of the 3rd Artificial Life Conference*, 1994.

[221] Yamamura, N., Higashi, M., Behera, N., and Wakano, J. **Evolution of mutualism through spatial effects**. *Journal of Theoretical Biology*, **226**(4):421–428, 2003. DOI:10.1016/j.jtbi.2003.09.016.

[222] Yang, L.-W. W. and Bahar, I. **Coupling between catalytic site and collective dynamics: a requirement for mechanochemical activity of enzymes.** *Structure*, **13**(6):893–904, 2005. DOI:10.1016/j.str.2005.03.015.

[223] Yang, L., Song, G., and Jernigan, R. L. **Comparisons of experimental and computed protein anisotropic temperature factors.** *Proteins*, **76**(1):164–175, 2009. DOI:10.1002/prot.22328.

[224] Yannakakis, M. **Computing the minimum fill-in is NP-complete**. *SIAM Journal on Algebraic and Discrete Methods*, **2**(1):77–79, 1981. DOI:10.1137/0602010.

[225] Yildiz, A., Tomishige, M., Vale, R. D., and Selvin, P. R. **Kinesin walks hand-over-hand**. *Science*, **303**(5658):676–678, 2004. DOI:10.1126/science.1093753.

[226] Zheng, W. **A unification of the Elastic Network Model and the Gaussian Network Model for optimal description of protein conformational motions and fluctuations**. *Biophysical Journal*, **94**(10):3853–3857, 2008. DOI:10.1529/biophysj.107.125831.

[227] Zheng, W., Brooks, B. R., and Thirumalai, D. **Low-frequency normal modes that describe allosteric transitions in biological nanomachines are robust to sequence variations**. *Proceedings of the National Academy of Sciences of the United States of America*, **103**(20):7664–7669, 2006. DOI:10.1073/pnas.0510426103.

[228] Zheng, W. and Doniach, S. **A comparative study of motor-protein motions by using a simple elastic-network model**. *Proceedings of the National Academy of Sciences of the United States of America*, **100**(23):13253–13258, 2003. DOI:10.1073/pnas.2235686100.

[229] Zheng, W. and Thirumalai, D. **Coupling between normal modes drives protein conformational dynamics: illustrations using allosteric transitions in myosin II**. *Biophysical Journal*, **96**(6):2128–2137, 2009. DOI:10.1016/j.bpj.2008.12.3897.