Practical Investigations of Complex Systems

Nicolas Brodu

A Thesis
in
The Department
of
Computer Science and Software Engineering

presented in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

Concordia University
Montréal, Québec, Canada

June 2007

Your file Votre référence
ISBN: 978-0-494-31137-0
Our file Notre référence
ISBN: 978-0-494-31137-0

ABSTRACT

Practical Investigations of Complex Systems

Nicolas Brodu, Ph.D.
Concordia University, 2007

What's currently called Complexity Science suffers from an unfortunate lack of consensus as to what is meant by these terms. A review of the common notions shows a field mined by controversies, with as many frameworks for the study of Complex Systems as there are authors who propose a generic one. This document is thus not an attempt to create yet another framework, but rather an application of the traditional scientific methodology to some Complex Systems in the domain of Computer Science. It is a demonstration that even for this field, the concrete application of predictive experiments set up to challenge the extent of the main notions proves fruitful. Moreover the tools and methods that are created along the way because they were necessary to carry on experiments represent by themselves an opportunity for making progress in the domain. This is precisely the case in the present Computer Science context in the form of new algorithms, that were successfully applied to the main experiments. Hence this work is both a call for a more classical and practical approach to Complexity, and a concrete application example for that call.

Acknowledgements

My first acknowledgements are to the continuous love and support of Valérie, who not only provided motivation and attention but also the everyday life care without which much of this work would not have been possible.

I also warmly thank my supervisor Peter Grogono, who have been and still is a model for my future professional and personal development. Thanks also for the enjoyable and insightful discussions and comments, as well as your constant encouragement and support.

Thanks Nawwaf Kharma and Sudhir Mudur in particular for our conversations and for your helpful comments. Thanks also to all my committee for providing a welcome feedback.

Thanks as well to all the members of Concordia University whom I have interacted with, especially Halina Monkiewicz and Pauline Dubois for their support in all internal affairs.

Thanks to the European Aeronautics Defence and Space Research Centre and to the French Ministry of Foreign Affairs for providing a most welcome two-years financial and administrative support.

Thanks finally to my family and friends, who had to patiently endure my occasional explanations, but nonetheless provided endless encouragements.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

'Let a game begin,' said the Lady. 'And your move?' he said.

She smiled. 'I've already made it.' He looked down. 'But I don't see your pieces on the board.'

'They're not on the board yet,' she said. She opened her hand. There was something black and yellow on her palm. She blew on it, and it unfolded its wings.

It was a butterfly. *[PRAT94]*

Terry Pratchett

What is a complex system, its main features and properties? What does it mean that something is emergent?

Since the advent of modern calculus in the 17th century with Newton and Leibnitz, the dominant philosophy has been that of integration: from the reasoning on an elementary scale we could sum up and obtain global results about a system (for example, movements of planets). Of course some equations describing a system behaviour cannot be integrated, so one cannot find a shortcut that allows a direct computation for a prediction at the higher scale. Even when such a shortcut exists it is not always applicable: Exponential relations for example were sensitive to initial conditions even before the discovery (reviewed by James Gleick [GLEI87]) of Chaos theory, which brought the notion of being locally exponentially divergent and globally bounded at the same time. Yet, and especially with computers, approximate

methods and numerical integrations were developed that can produce reasonable results, and they still form the majority of industrial simulations to date.

Another approach is the study of the high-level properties of the system, considering entities defined at a global scale scale, as discussed by Russ Abbott in [ABBO06]. Then one could try modelling these entities and their interactions directly rather than by applying the more traditional integration approach. Some other systems are self-similar at different scales and may be better analysed by yet another method as explained by Benoit Mandelbrot [MAND82]. There are also universal phenomena and global properties that may be observed whatever the underlying equations. So if we now look at the problem top-down any phenomenon that we observe at the system macro-scale but that we cannot somehow relate to micro-states poses a similar problem as before but the other way around.

The reasons why a phenomenon defined at a high level cannot be related to low-level properties may be multiple, from simple ignorance of hidden relations to theoretical uncomputability. But whatever these reasons the same practical issue remains between the high-level scale and the underlying micro-scale elements: the phenomenon is then often labelled "emergent". The notion of emergence has progressed over time, and its history is reviewed by Peter A. Corning in [CORN02]. Refinements about possible reasons for the failure to relate micro and macro properties were proposed, but overall the same idea remains in one form or another.

This main micro-macro relation difficulty is in my opinion the major characteristic of what we call today a "Complex System". Of course this definition is elusive: If "Complex System Science" were to solve that micro-macro relation difficulty then by definition the system would be put outside the "complex" class. The definition is also circular, hence not really a definition: emergence is when

micro-macro relations are too complex to be understood, and complex systems science is the study of emergence. Yet, many of the notions presented in the next chapter (self-organisation, synergy, etc.) rely on such circular concepts. This is unfortunate because many arguments in Complex Systems Science take the form of "X and Y are properties some entities may have in a complex system. Entities having X are observed to also have property Z, while entities having Y do not. Therefore X is the reason why Z". Of course, the last step is erroneous in its generalisation (it is a correlation, not a causal link). For example, self-organisation is often given a causal power, while not even defined unambiguously. This is the kind of reasoning that probably lead to the citation of Steven Weinberg given in introduction to the next chapter (and which he actually made in reaction to Stephen Wolfram [WOLF02]), and one of the reasons why I expand on the notion of causality in Section 2.2.

Attempts at formalisation (for example by Cosma R. Shalizi [SHAL01] and by Aleš Kubík [KUBI03]) must rely on some mathematical framework and do not encompass (to date) all aspects that were proposed by other definitions. What is called emergent by some is outside the definition of others. Too broad definitions are rejected because they are either inapplicable or they would include a range of phenomena that we intuitively do not label as "emergent"; while too restricted definitions miss one or another of such phenomena. My guess is that no definition of emergence may satisfyingly correspond to our intuition (I explain this point in Section 2.4), and conversely, any successful hypothetical theory on emergence would include counter-intuitive effects.

Nevertheless, I think this short introduction outlines the main idea. As Robert B. Laughlin notes in [LAUG06], "science has now moved from the Age of Reductionism

to the Age of Emergence, a time when the ultimate cause for things shifts from the behaviour of the parts to the behaviour of the collective". The next chapter explains why this might be the case, by reviewing the current notions related to complex systems and emergence.

However there is no reason to think that Complexity Science cannot be handled by the traditional approach exemplified by Thomas S. Kuhn [KUHN62]: by using incremental steps, with predictive testing, refinement of the main concepts, that allow to validate or not the main theories, etc. This is what Kuhn calls "normal science", posed as a necessary condition for further ground-breaking discoveries. Trying to skip the intermediate steps in the hope to jump to the last is in my opinion neither wise nor particularly productive: Indeed, without the intermediate steps to back up the final claims, these will remain mere speculations.

Hence the present dissertation goal is to demonstrate that the traditional approach is also effective for Complex Systems, based on the selected examples which I had the opportunity to investigate during my PhD studies. But this document is also a dissertation in Computer Science, not Philosophy. The tools and techniques that are created along the way are often of interest in themselves ([KUHN62] again) and in the particular field of Computer Science these tools take the form of new algorithms necessary to carry on the experiments. Thus I also concentrated on generalising any valuable algorithm I had to create in order to run the experiments, so as to make these algorithms independently applicable pieces of work that form contributions in their own.

This dissertation is organised the following way:

- Chapter 2 reviews the general notions related to Complex Systems and emergence and discusses what are possibly the main sources of controversies. A

personal view and modest contribution is proposed at the end of this chapter, regarding implications for computer simulations.

- Chapter 3 extends on the above remark of applying the general scientific methodology to complex systems. It introduces the research problems and explains the methodology that I applied for this work.

- Chapter 4 presents the experiments and frameworks I devised to assert the predictive value of two notions related to complexity: downward causation and the Edge of Chaos hypothesis (see the next chapter). These experiments were also additionally published in different articles in the Artificial Life (Section 4.1) and Artificial Intelligence (Section 4.2) research domains.

- Chapter 5 describes three algorithmic contributions that I created to fill the needs presented by Chapter 4; the previously mentioned tools necessary to realise the experiments. State of art techniques in each domain are also presented together with the new algorithms so these can be appreciated in their own context.

- Finally Chapter 6 summarises the main advances of knowledge that are provided by the present work and concludes on the issues encountered in this study. Possible future projects that would complement and extend this work are also suggested.

All in all, these PhD studies were for me the occasion to learn and practice in a variety of domains, which I have enjoyed as much as I enjoyed presenting this work when I had the occasion. But as far as research is concerned this not an end, rather a beginning.

Chapter 2: Background review

> *Particle physicists like to say that the theory of complexity is the most exciting new thing in science in a generation, except that it has the one disadvantage of not existing.* [WEIN02]
>
> *Steven Weinberg*

This chapter proposes a review of current notions about complex systems. The goal is to present a fair description of the field, so the research problem introduced in Chapter 3 can be fully appreciated.

Section 2.1 reviews the different concepts related to emergence and complex systems and how they relate with each other. Section 2.2 then presents possible reasons for the controversies presented in Section 2.1. Personal comments on numerical simulations are presented in Sections 2.3 and 2.4. These comments also introduce the motivation for the methodology described in Chapter 3.

2.1 Review of emergence and related concepts

The following subsections contain common notions associated with complex systems. The basic constituents are presented in Section 2.1.1, they form the building blocks for the definitions of Section 2.1.2. These definitions are descriptive only: they may be used to clarify the domain by classifying and qualifying the properties of complexity and emergence, but they have no predictive value (so far). Quantifiable aspects of complexity are described in Section 2.1.3. These quantities are necessarily dependent on some formalization, hence become "reductionist"

compared to the holistic concepts presented in the Sections 2.1.1 and 2.1.2. However, they do have a predictive value, and may thus form the most promising approach for a formalized theory of emergence.

2.1.1 Ingredients for a complex recipe

The notions presented in this subsection form a common basis for complex systems frameworks. These ideas are generic and applicable to many systems.

The idea there are **Levels of investigation** correspond to the intuitive notion that was presented in Chapter 1: That a system can be studied at different scales, or at least at a micro-level and at a global level. When the process can be repeated for yet another level this defines a hierarchy. This notion is not new: Philip E. Agre [AGRE03] reviews and explains the static vs. dynamic hierarchies issue that was presented by Herb A. Simon in [SIMO69], related to how the levels are defined. Russ Abbott [ABBO06] also considers hierarchies and static and dynamic emergence, and these notions are presented in Section 2.1.2. Peter A. Corning [CORN02] proposes an historical perspective where older articles also convey the ideas of hierarchies, especially in the life sciences.

In either case there are observed entities made of smaller constituents, and some features of the entities are not easily linked (or reduced) to the constituents. See also the "whole and the sum of parts" entry in this section. The hierarchical organisation of levels occurs when such entities form themselves the basic constituents of yet another larger entity, and so on. For example, a cell, an organ, an organism, a social organisation, etc. In the particular context of life sciences John Maynard Smith and Eörs Szathmáry [SS95] explore explicitly the transitions from one level to the other.

Yet the boundary between levels is not always very clear, and some constituents may interact at different scales. Alternatively, scope, together with resolution and the states of a system were proposed by Alex J. Ryna in [RYNA06] as better notions: study should then be done on entities defined at their proper resolution in space and time, and whether there are other such entities at the same scale to form a "level" is irrelevant. In this view, the levels themselves could possibly be emergent properties.

While attractive, the scope/resolution approach does not solve the main issue of the relations between the components, irrespectively of how they are defined. Yet, Russ Abbott [ABBO06] in particular shows that reasoning on entities directly rather than on global levels solves a number of definition issues and thus clarifies the situation.

The notion that there are "levels" of investigation is a handy conceptual tool, but it is unfortunately defined precisely because it allows to pose the question of emergence (higher-level entities are said to emerge from a lower-level), thus forming a circular definition. Jaegwon Kim [KIM99] states that "a layered model [..] provides an essential framework needed to formulate the emergentist/reductionist debate". The problem is also that the "layer" defined implicitly by one "emergent" entity may not correspond to the one of another. Hence there is no global layer but rather a continuum of scales with their own properties and entities, defined with respect to other entities at a lower or equal scale.

The whole and the sum of parts refer to the statement by Aristotle [CORN02] that both are not identical. By extension, this is the same idea as the one of synergy between components: a higher-level entity comprising lower-level elements is a "whole" that is not just the mere juxtaposition of these elements. A reductionist approach is that the "sum" in "sum of parts" is more complicated than a simple

linear combination and thus explains our apparent inability to relate the whole with the parts. Then, since linear relations form only a small fraction of all possible relations[1], this explains the apparent universality of the "whole vs part" issue, though there is nothing special going on and the notion is better investigated on a case by case basis. On the holistic side [KAUF93], the parts are said to self-organise due to their relations, but there is also irreducibility of some higher-level function of the whole (consequently this functionalist view is not just a matter of non-linear relations). In any case, "this parts and whole" approach to emergence is perhaps historically [CORN02] the first approach, and it is still a topic of controversy. There are also complications with the notion of causality, which I detail in Section 2.2.

Interactions between the elements must be taken into account, and they must be sufficiently complex so there can be a "whole" which is not just elements side by side. Interaction graphs and networks then define as much of the global "whole" as the elements own nature. Such networks then offer a connection with dynamical systems and graph theory. They can be simulated and their properties can be studied on a large scale (see Andrew Wuensche [WUEN02] and Réka Albert and Albert-László Barabási [AB02]). John Holland in [HOLL98] models the relations between the elements as **constrained generating procedures**. Similarly, when the parts can learn and adapt to their environment the system is called a **Complex Adaptive System** [HOLL98].

Open dissipative structures were initially defined in a thermodynamical framework by Grégoire Nicolis and Ilya Prigogine [NP77]. The idea may be extended: so long as the underlying assumptions allow for a definition of a generic notion of energy, and the system under consideration allows that energy to flow, then entities in that system may "use" this energy [ABBO06]. Extensions to this

1 Stanislaw Ulam compared non-linear mathematics to non-elephant zoology...

framework are when the entities can store energy and then use that reserve later in time [KAUF00], and when the entities simply use the energy to perpetuate themselves, which then leads to the notion of **autonomous structures** [ABBO06]. Other extensions consider how an autonomous structure may use the output flow in relation to its environment as a mean of action (with the corresponding form of causal relationship, see Section 2.2). The notion of **empowerment** by Klyubin *et al.* [KPN05] represents precisely this ability to act on the environment, but also relates it to the feedback the autonomous agent may get from its actions.

The notion of **energy** may be abstracted in a functionalist point of view. For example, in social contexts, energy may be related to available skills, money, or time; In artificial life contexts energy may be CPU execution slots; In discrete dynamical systems energy may be related to a system state change (and its dissipation would be the fusion of trajectories). Generally speaking, energy is functionally defined by the capacity of the entities in the system to use it. Of course this leads to a circular argument. Howard Pattee's semantic closure concept [PATT95] can also be used as a justification for a separation of the emergent level, when the usage of the energy has an intrinsic signification for the entities in the system (see also the semantic vs syntactic entry in Section 2.1.2). The notion of energy may then be used formally in the higher level.

Self-organisation is concerned with the internal structure of a system, and how that structure evolves without external intervention. [KAUF93] proposes that self-organisation is the result of positive feedback loops (see above). The term self-organisation is credited to William Ross Ashby [ASHB56] in a pioneer work on cybernetics, but the notion has now extended to a point where it is ubiquitous. Cosma Rohilla Shalizi presents an extensive effort [SHAL01] to clarify the notion in

the context of time series, and equates self-organisation to a rise in statistical complexity (with a working data-based algorithm for computing this value, see the corresponding entry in 2.1.3). Another definition for self-organisation is the "state-space description" proposed by Francis Heylighen [HEYL01]: "self-organisation as the appearance of coherence or correlation between the system's components is equivalent to the reduction of entropy", which is in some cases contradictory with the statistical complexity interpretation. There are other definitions, like the positive feedback loop approach previously mentioned. A generally applicable and consensual notion of self-organisation has thus yet to be defined.

Hypercycles [ES70] are another name for **positive feedback loops**, applied in a pre-biotic biological context. Proto-cells in the form of compartments allow different chemicals to concentrate and then react. The feedback loop is when the resultant of one reaction enhances the next, in this case RNA strings are translated to enzymes which catalyse the next reaction. This mechanism is a "natural principle of self-organisation" and an important hypothesis for the appearance of life [SS95]. Stuart A. Kauffman [KAUF93] makes an argument for autocatalytic cycles and extends the notion to other domains, deriving the notion of an "order for free" [KAUF95] that would counter the second thermodynamic law and entropy in dynamic systems [KAUF00].

Autopoiesis, defined by Varela *et al.* [VMU74] is the idea of a structure that is: 1. Defined in space, it has a boundary with the external environment. 2. Able to reproduce itself. This is a variation on the theme of autonomous structures and self-organisation applied in a biological setup. The notion has attracted much controversy (related by Barry McMullin in [MULL04]) as to whether it is a suitable model for living entities, and the application of the definition has itself rooted out

numerous problems (such as how to define the structure boundary, and what permeability is allowed so it can interact with its environment [BEER04]). However, when viewed in a larger framework of autocatalytic cycles and autonomous structures, the concept rejoins the view [KAUF00] that self-organisation is anterior to evolution and adaptation (See also Arantza Etxeberria [ETXE04]).

Synergetics is the name of an inter-disciplinary approach founded by Herman Haken [HW73]. The best definition is perhaps the one given by the Center of Synergetics, headed by Haken himself: "Synergetics deals with complex systems that are composed of many individual parts (components, elements) that interact with each other and are able to produce spatial, temporal or functional structures by self-organisation"[2]. The initial topics of investigation were focused on physics, but the field has enlarged and the current domains of research of the institute are brain theory and psychology. As the etymology "science of synergy" suggests, "synergetics" calls preferentially for a holistic approach of emergence. Carlos Gershenson also proposes in [GERS07] a methodology for controlling complex systems that is well suited to this approach.

2.1.2 Descriptive qualifiers of emergence

The definitions presented in this section are used to classify the different kinds of complexity, emergence, or properties the entities under investigation should or should not have. However they generally do not bring any predictive power.

Nominal emergence refers to a global property that cannot be a micro-property, like the total volume, colour, or temperature of an object. As the etymology suggests no additional assumption is imposed on the emergent notion. Nominal

2 From http://itp1.uni-stuttgart.de/en/arbeitsgruppen/?W=5&T=1, 01/03/07

emergence does not refine what are the expected properties for the different levels of investigation. Thus, a nominally emergent phenomenon in a given context might not be considered emergent in another, depending on these contexts particular assumptions. To illustrate the problem let me introduce the example of the colour "green", which might be associated to a range of wavelengths. But one might be interested in why the object emits these particular wavelengths (at the atomic excitation level for a LED, or through diffraction for a rainbow, etc.); or why "green" was associated to that particular range of wavelengths (which is related to the presence of receptors in human eyes, is green still green for colour-blind people?); or why we semantically associated various hues together in the same "green" concept (there might be cultural variants, so "green" is not a universally defined notion in terms of wavelengths). Nominal emergence just states the micro-macro relationship problem without hinting at the solution.

Basic emergence is defined by Aleš Kubík [KUBI03] as a "behavior reducible to agent-to-agent interactions without any evolutionary process involved. [...] The environment has no rules of behavior and is changed only by the actions of the agents. [...] Basic emergence then refers to a property of the system that can be produced by interactions of its agents (components) with each other and with the environment and cannot be produced by summing behaviors of individual agents in the environment". This definition is applicable only in contexts where "agent" and "environment" have a signification, and requires that we can somehow measure the behaviour of the agents as well as define the lack of evolution. In the context of [KUBI03] grammars are used for representing the agents and their interactions. It is certainly useful to compare explicitly what are the sum and the whole, but the definition would require some adaptation to be applicable to other contexts.

Dynamic and static emergence as introduced by Russ Abbott [ABBO06] refer to whether a temporal aspect is respectively necessary or not for the definition of emergence. For example, diamond and graphite exhibit different statically emergent properties of carbon, like the hardness property. Dynamic emergence is **stigmergetic** when it involves autonomous entities, with an autonomous entity defined "as a self-perpetuating region of reduced entropy that is implementing a dissipative structure's abstract design" [ABBO06] (see also the corresponding entries in Section 2.1.1). In addition, a requirement is introduced that the emergent phenomenon "may be understood in its own terms" and that "its understanding does not depend on knowing how it is implemented". This further restricts emergence to functionally irreducible cases with a formal higher-level system on these functions so they can be understood. But then, semantic closure have to be considered for how these functions and formal system relate together.

Syntactic and semantic emergence proposed by Howard Pattee [PATT95] are respectively concerned with the formal and functional aspects of an entity. Given a formal lower-level system, like a grammar, the syntactic emergence refers to how an entity defined at a higher-level of investigation appears in the lower level. The semantic emergence claim is that some function of the entity may not be described within the formal lower-level system. So as to illustrate the notion let me consider the dictionary example: It may be seen as a directed graph of words, each word pointing to some other words in its definition. Yet, the precise meaning of a word is not contained in the dictionary itself, but found only with respect to prior knowledge at the higher level, obtained by how the language is used in practice: If each word is replaced by a sequence number corresponding to the first occurrence of that word in the dictionary the formal directed graph remains the same, but the dictionary becomes completely useless to a human.

This leads to the notion of "semantic closure", that a higher-level of investigation is only completely defined by considering not only how the entities involved interact, but also by what meaning is associated to the interactions by an external observer or by the entities themselves. The controversy arises in both cases regarding the source of the attribution of the meaning. If it is given by the observer then it is a subjective property, not inherent to the system. Unless the observer is also part of the system, but this is equivalent to the second case that the meaning is given by some of the entities. But then, this introduces another philosophical debate as this assumes that a part of the system has the ability to attribute a "meaning" to another part of the system. Engaging in either debate is out of the scope of this dissertation.

More generally, a functionalist approach would use semantic closure to justify the irreducibility of some higher-level function. So, being semantically emergent is possibly simply the bottom-up equivalent of being functionally irreducible in a top-down context. Section 2.2 details the notions of reductionism in relation to causality, and gives possible reasons for the controversies. See also my personal opinion in Chapter 6 as to why I think this debate is mostly irrelevant and not particularly fruitful.

Weak, medium and strong emergence refer to what form of irreducibility and causal powers are attributed to the emergent entities over the lower-level from which they emerge. This is detailed in Section 2.2.

Emergence relative to a model does not consider emergence to be an intrinsic absolute property of a phenomenon, but that it can only be defined by considering this phenomenon with respect to an observer (which could be a formal model for example). Peter Cariani defines it as "a functional theory of emergence by giving an account of how new basic functions of the observer – measurements, computations,

and controls – can come into being" [CARI89]. The observer has predictive capabilities, a formalisation of the entities and their functions at the level with which it interacts. Emergence is associated to a divergence between the model formal predictions and what really happens. The case where new observables are necessary to represent new functions in the observer model is called **creative emergence**, otherwise this is **combinatorial emergence**[3]. Note that in this context an observer is really embedded in the system under investigation, as are humans making observations about the world. Which in turn gives another view on the notion of subjectivity, with the associated philosophical controversies.

Surprise of the observer has been proposed by Ronald *et al.* as a condition for emergence [RSC99]. The subject is highly controversial (see [KUBI03]), mainly because of different definitions of what "surprise" means. Arguments on the subject may be classified as to whether the observer is part of the system (surprise = difference from expectation = emergence relative to the observer internal model of the rest of the system) or whether the observer is independent of the system (in which case surprise and emergence are not properly defined within the system).

Computational emergence is an attribute applicable to other emergence concepts. It implies the existence of a formal system, that usually allows computation theory. Any emergence definition in this context will have the "computational emergence" attribute. This tells nothing about what properties the computations and formal aspects should have to be entitled "emergent" in the first place, and what other requirements the framework must respect. In particular, this attribute alone does not specify what forms of reducibility are considered, if any.

3 This short summary is far from fully rendering the works by Robert Rosen [ROSE98], Peter Cariani [CARI89], and others. I think I have captured the essence of the "emergence from a model" notion, but invite interested readers to refer to the material in [CARI89].

The "computational" attribute for emergence is used by people who propose that the universe is non-computable [CARI89], or that complexity is what cannot be simulated [ROSE98], so as to make the distinction with a **thermodynamic emergence** that could then be exhibited only by natural phenomena[4]. On the other side of the argument Digital Physics as proposed by Konrad Zuse [ZUSE69] and Edward Fredkin [FRED90] makes the distinction meaningless.

2.1.3 Quantifiable aspects of Complexity

Unlike the previous definitions and concepts, the notions in this section are not only descriptive, but do have some kind of predictive power. Hence they may form the basis for a quantitative theory of complex systems, however limited in scope this "reductionist" theory might look in a first time compared to the more elusive holistic concepts.

Entropy, whether the thermodynamics or the information-theoretic version of it (Cosma R. Shalizi gives a comparative argument in [SHAL04]), has been the subject of much attention. Since entropy is associated to disorder, the idea is that organisation (and the self- version) opposes entropy and therefore we shall be able to detect it when entropy reduces. See also the entry about self-organisation in 2.1.1 and the citation from Francis Heylighen [HEYL01] in that entry. When the probabilistic definition of entropy is used, then we can actually compute it. Prokopenko *et al.* [PBR06] present an information-theoretic approach of entropy and its relation with statistical complexity. In addition, "excess entropy" is defined by James P. Crutchfield and David P. Feldman [CF03] as "the intrinsic redundancy" of the system under investigation. Together with the statistical complexity measure C

4 Which is in my opinion a separate issue from the observer/model topic aforementioned.

(see the next entry below), the excess entropy E can be used to define Shalizi's "efficiency of prediction" indicator e = E/C [SHAL01]. In turn, this leads to a characterization of emergence: when the predictive efficiency is increased as a result of a transformation (the transformed entity can be predicted more efficiently than the original one). [SHAL01] gives an example on an ideal gas where thermodynamics emerges from the statistical mechanics.

Statistical Complexity measures the amount of information that is present in the past of a system, which is relevant to predicting its future. See the aforementioned [PBR06] and [SHAL06] for an introduction, [CRUT94] where James P. Crutchfield gives a link to the emergence issue, [SHAL01] for mathematics, and Frank B. Knight pioneer article on the topic [KNIG75] for the general notion on continuous systems. Statistical Complexity is the amount of information needed for optimal statistical prediction. The idea is that both well-ordered systems and highly random ones have a low complexity: The ordered systems state space usually comprises only a few states, and knowledge about these states is enough to predict the future. Random systems also require little knowledge of the past: for example, if the observed statistical distribution of events takes the form of a fixed repartition of future values, whatever the past value, then knowledge about the past is useless for predicting the future with maximal accuracy on average. Statistical complexity is thus a measure of how difficult it is to predict the future by monitoring the system past. It is defined as the amount of information present in the "causal states" of the system: the equivalence classes of system pasts that produce the same distribution of futures. A more detailed presentation is provided together with the algorithm in Section 5.1 for estimating the causal states. Statistical complexity was proposed as a measure of self-organisation [SHAL01]: A system is said to self-organise when its statistical complexity increases over time. The measure is an intrinsic property of

the system that can be computed from data [SS04]. I have extended the algorithm proposed in Shalizi *et al.* [SHRKM05] so data can now be provided on-line and the statistical complexity can be computed incrementally including for non-stationary systems. See also Section 5.1 for more details, and Section 4.2 for an application example.

Algorithmic complexity has been defined independently by Solomonoff, Kolmogorov and Chaitin (see [CHAI05] for an intuitive presentation of the concept). The idea is to use a universal Turing machine [TURI36] to describe an entity: this description then takes the form of a program. The algorithmic complexity of the entity is defined as the length of the shortest program that can produce the entity description. The problem is that this value is uncomputable and can only be approximated from above [CHAI74]. [CHAI74] also proves that "the great majority of the strings of length n are of complexity approximately n. These are the random strings of length n". In other words, the string itself is then its shortest description, and these form the vast majority of all strings. Unfortunately, this also includes descriptions of higher-level entities, and offers no discrimination between "emergent" or "trivial" ones. Moreover, in practice we may be more interested in approximate versions of a given entity and discard small variations as "noise" (see Section 2.4): When observing a phenomenon, we'd like to characterize not only the particular instance we're monitoring but also to generalise to all similar phenomena. In that case, algorithmic complexity may not be the best notion to use: it has no generalisation power to closely related entities that differ by a small variation.

The Edge of Chaos is an hypothetical region in parameter space between "order" on one side, and "chaos" on the other. The initial term comes from [LANG90], where Christopher G. Langton's λ parameter is hypothesised to reach a

high value when a cellular automaton has a potential for complex computations. This particular λ parameter interpretation was later refuted by Mitchell *et al.* [MHC93], but nevertheless, the idea that some indicators are low for both highly ordered and highly disordered systems, while high in-between, is a very useful one: It helps characterize systems that are in a way the most "complex", these where there is the most **diversity** in significantly different global behaviours.

Indeed, both totally ordered and totally random systems lack diversity, in the sense that it is not possible to distinguish statistically the states that are produced by the system: There are only a few distinct configurations possible in ordered systems, and totally random systems exhibit the same statistical distribution of behaviours whatever the initial conditions. Therefore, the Edge of Chaos hypothesis is also that systems need to exhibit a sufficient diversity so they can support advanced features like being able to compute. This argument was proposed by [LANG90] for cellular automata, and Stephen Wolfram devoted a whole controversial book [WOLF02] to the notion of cellular automata exhibiting complex behaviours. Though as aforementioned the indicator proposed by [LANG90] for detecting the edge of chaos was refuted by [MHC93], the idea remains and it is possible that other indicators could work better (including for cellular automata).

Andrew Wuensche [WUEN02] extends the notions of order and chaos to random boolean networks, which are automata on a graph structure instead of a regular lattice. The large-scale dynamical properties of both cellular automata and such networks are then studied and analysed, especially under perturbation. A balance between order and chaos is then specified as a condition for the network to exhibit a form of memory. The "memory capacity" defined by Natschläger *et al.* in [NBL04] precisely quantifies with an explicit measurement the "edge" region where the

system has maximum memory. [NBL04] also uses the idea that both random and ordered systems produce few indistinguishable final states, so in the context of neural networks[5] the system could be also described by its ability to separate initial configurations. This leads to the "NM-separation property", which was presented as indicative of high processing capabilities by Robert Legenstein and Wolfgang Maass [LM07A] with an explicit mention of the edge of chaos hypothesis.

The problem with these indicators is that they all define what [SHAL01] calls a "One-Humped Curve", where the maximum of the curve does not necessarily corresponds to a maximum in complexity. This last point is also detailed in my experiments in Chapter 4. The general Edge of Chaos claim is that some indicator related to complexity reaches a maximum between states that can be related to order and chaos, but one has yet to define what is meant by order, chaos, and complexity.

Scale-free relations are functionally defined by the presence of a few important elements with many less important ones, with a negative exponential relation between number and importance. The trade-off between importance (functional role) and number allows to scale the system by making it manageable as its size grows. Réka Albert and Albert-László Barabási [AB02] exhibit such scale-free relations in the domain of network graphs where a few nodes (ex: internet routers) allow efficient network traversal, by aggregating the many local sub-networks hierarchically.

I deliberately used a functional presentation because there is a controversy as to the exact mathematical relation and the meaning associated to the exponential

5 The "neural" networks are in the [NBL04] case actually networks of transfer functions, so they generalise the boolean networks for which the functions act on and produce boolean values.

decrease. The historical perspective provided by Michael Mitzenmacher [MITZ04] and Edoardo Milotti [MILO01] shows that power-laws and exponential relations are widely applied and ancient concepts[6]. A scale free relation is when the probability of finding the property of interest decreases according to some power-law or log-normal distribution as the scale increases. Such relations may be found for network graphs as aforementioned in [AB02] and in [WUEN02], but they also appear in finance, biology, chemistry, ecology, astronomy, and information theory (see [MITZ04]). When taken in the frequency domain, the "1/f noise" equivalent of the power-law appears in particular in electronics, sandpile models, and more inter-disciplinary fields (see [MILO01]). Such a wide range of applications makes them a good candidate for detecting cross-disciplinary universal phenomena, hence makes them a primary target for Complexity Theory. The risk is of course over-generalisation with no predictive power, and [MILO01] concludes by "Do we understand 1/f noise? My impression is that there is no real mystery behind 1/f noise, that there is no real universality and that in most cases the observed 1/f noises have been explained by beautiful and mostly ad hoc models".

Yet this is precisely what a functionalist point of view of emergence would appreciate: Irrespectively of the underlying elements, the functional property of being (relatively) insensitive to scaling remains, at least over the range of scales that matters for a specific problem, and whatever the mathematical model that is best suited for the description (power-law or log-normal). For example, efficient traversal through hubs in a network is a property that is interesting both locally (with a small group of nodes connected to a local hub) or globally (for reaching distant sites). Whether the distribution of connections below each node follows a

6 And so is the mathematical controversy, ex: between Simon and Mandelbrot as explained in [MITZ04]

specific mathematical form matters in this case much less than the property itself.

On the other hand, in the contexts where the power law is used to make predictions in the range corresponding to the tail of the distribution, an error in the formula can have drastic consequences. So the reductionist/functionalist debate strikes again, and [MITZ04] warns: "From a more pragmatic point of view, it might be reasonable to use whichever distribution makes it easier to obtain results. This runs the risk of being inaccurate; perhaps in some cases, the fact that power law distributions can have infinite mean and variance are salient features, and therefore substituting a lognormal distribution loses this important characteristic. Also, if one is attempting to predict future behavior based on current data, misrepresenting the tail of the distribution could have severe consequences". More generally the study of the tails of probability distributions and their decrease rate is the topic of the theory of "large deviations", for which Srinivasa Varadhan received the Abel prize in 2007. This is an important mathematical topic with consequences anywhere a predictive methodology is sought in the aforementioned disciplines exhibiting "power laws" (or related).

Finally, even a functionalist might be interested in the analysis of the differences between the mathematical forms. [AB02] proposes for example a generative model for the power-laws observed in networks, but perhaps other models are more statistically significant. The reason why a property is observed with a negative exponential-like relation in a given system, what led to this relation, might be interesting in order to better understand the function the property occupies in the system and its limitations. In particular [MITZ04] concludes "The fact that power law distributions arise for multiplicative models once the observation time is random or a lower boundary is put into effect, however, may suggest that power

laws are more robust models". Thus, conversely, analysing the form best suited to model a functional property behaviour might actually be indicative of the reasons why this property occurs. And as I explain in the next section "the reason why" something happens is a causality issue, which is precisely indicative of the functionalist/reductionist debate.

2.2 The tricky concept of causality

The concepts presented in the previous subsections give an image of a fragmented field, where controversies abound. This is precisely the case, and current attempts at creating a theory of emergent phenomena often end up having to define concepts that are specific to that attempt. Jaegwon Kim [KIM06] notes that "Emergence is very much a term of philosophical trade; it can pretty much mean whatever you want it to mean, the only condition being that you had better be reasonably clear about what you mean, and that your concept turns out to be something interesting and theoretically useful". Consequently, there are as many definitions as frameworks, and no real common theory.

Yet, many if not all emergence-related concepts in the previous section refer to some form of (or lack of) causation: If my informal introduction in Chapter 1 is correct, this is expected since "emergence" is precisely a term which is invoked when other explanations fail. Hence causation is the subject of controversies: If no reason can be given for "emergent" behaviours, why do they appear? The debate between reductionists and functionalists related by Jaegwon Kim [KIM99] revolves around the same idea. Jochen Fromm [FROM05] notes: "the cause is normally the unclear point in emergence" and proposes a taxonomy of emergence concepts based according to how they treat causality. Causality is also analysed as a major source

for historical debate by Peter A. Corning [CORN02], where finality (functional causality) is explained in detail in the section about synergism.

Hence, I think that a section on causality is necessary in this dissertation so as to clarify and try to understand possible reasons for controversies. Solving the problem of "why" certain phenomena appear in some context may thus very well put these phenomena outside the category of "emergence". Even if the rest of the dissertation concentrates on the more practical predictive testing of selected notions (see Chapter 3), I cannot ignore what is a major issue in the field. The motivation for this section is to clarify the concept of causality, so as to better understand the source of some controversies in the field of Complex Systems.

This section is however not strictly necessary for understanding the main parts of this dissertation in the next chapters. But it provides a background that allows to place my work in a larger context, especially in relation to the reductionism vs functionalism argument which is presented in the next sections. An alternative suggestion for the flow of this dissertation is to jump to Section 2.4 which contains some material and examples that justify the research methodology of Chapter 3, and come back to Sections 2.2 and 2.3 for more details if necessary.

A personal attempt at disambiguation for the main causality issues is now proposed, while remaining as neutral as possible in the reductionist/functionalist debate. A more engaged theoretical personal contribution and its consequences concerning simulations is presented in Sections 2.3 and 2.4.

2.2.1 Causality as a source of debate

In his review about the history and "re-discovering of emergence", Peter A. Corning [CORN02] traces back a major source of controversy to the notion of

causality. The two sides of the argument are presented as the "holists" on one side, and the "reductionists" on the other, with radically different perspectives on causality (which I'll detail below).

Mark Bedau in [BEDA03] states that "emergent properties without causal powers would be mere epiphenomena". Russ Abbott in [ABBO06] states: "In short, we define epiphenomenal and emergent to be synonyms", but then he puts the debate between "reductionists" and "functionalists", the later ones being equivalent to the aforementioned "holists".

In order to clarify the concept, I'll reuse in this document the classification that Emmeche *et al.* have outlined in [EKS00] by applying to the question of emergence the four Aristotelian concepts of causation:

Efficient causality is the notion that something implies, entails, or brings about something else.

Material causality is the notion that something is made of something else. Note that "matter", as in material, has the broader sense of "composition" here.

Formal causality is the structure or the form of something, like a house is defined by its architecture.

Functional causality, which replaces finality in Aristotelian terms: the role played by something (in relation to something else).

For example, discussions about an alarm clock may refer to the formal causality (the clock internal plan, why it works), functional causality (what the clock is used for, why it was built), material causality (the clock composition, why it exists at all), and efficient causality (the clock is the cause of the sound that is itself the cause of the observer waking up).

So, what are the two sides or the argument? Holists/functionalists are more concerned with the functional causality, whereas reductionists are more concerned with the material and formal causality. The clash often comes when the two camps refer to their favourite concept to explain something, thus bringing efficient causality in the balance. On the one side, the function of something is the ultimate source of why things happen, and on the other the explication comes from material and formal laws of operation.

Many apparent controversies end as soon as the notions of causality are refined. For example: "The whirlpool causes the water molecules to move in a restricted way" versus "Water molecules and heat processes amongst other things, are the cause of what we perceive and define as a whirlpool". In this case, the first statement would be an efficient causality (the restriction) between objects defined functionally and formally (the whirlpool and the water molecule movements). The second statement is about an object defined materially (the whirlpool). Natural language only is the source for a possible confusion: Applying the efficient causality of the first statement to the whirlpool of the second statement is meaningless and should be discarded as such (from the second point of view, the restrictions are part of the definition, not a consequence).

Is that all? Can causality and all controversies be solved by referring to this simple classification? Of course not, but it makes a good start. Further refinements could be made using notions like time dependency, what is required for objects to be comparable, probabilities, and more. Howard Pattee [PATT97] proposes that causation is a useful concept only when it identifies controllable events or actions. This is further extended by Fabio Boschetti and Randall Gray in [FG07], who propose a form of causation intermediate to the above four, as exemplified by: "The

flock will circumvent the obstacle. It thus appears that we were able to exert control on the behaviour of the flock; the flock appears to have causal power". The causation question then becomes identifying what "controllable" means, with the related philosophical issues that are out of scope of this dissertation.

Modern physics must deal with the quantum principle of no local reality and Bell's inequality violation, combined to the no communication principle, so as to avoid a time travel paradox in general relativity. The no local reality is in apparent contradiction with material causality, the no communication principle restricts efficient causality. Measurements may become important, since they can provide an objective source of investigation for material causality. But unfortunately, as mentioned by Howard Pattee in [PATT95], a measurement is only defined by the function of the measuring device: to provide a number, that is interpreted in the light of a theory. The theory then itself provides formal causality between the measurements, by way of its laws. As we see, the problem of causality is intrinsically linked to the problem of material objectivity.

2.2.2 Supervenience and identity

Supervenience is typically used to assume material causality while avoiding the issues related to other forms of causality. For example, saying that the mind supervenes on the body means that ultimately the body is the material source of the mind, without assuming anything as to how the mind may "emerge" from the body. A more precise definition of supervenience is given below in this section.

Supervenience is relevant for this dissertation because it represents a weak form of micro-to-macro relationship that still has useful consequences. Assuming supervenience allows to reason at a high level (for example on the movements of

billiard balls) and then to apply the result of this reasoning on the micro level through material causality (for example deducing that atoms have moved, even though the collision laws that apply to billiard balls do not apply directly to the atoms). This is what Russ Abbott calls "downward entailment" in [ABBO06], and which is outlined with more details in Section 2.2.4.

Downward entailment and supervenience are especially applicable to my examples concerning simulations in Section 2.4. These examples in turn introduce the research methodology proposed in Chapter 3. Hence, a section on supervenience helps understanding the causality issues that are relevant to this work. However the reader familiar with supervenience may safely skip this section.

Supervenience is concerned with a logical dependence between properties. Assuming properties A and B are defined, A supervenes on B means that each time entities differ with respect to property A, they also differ in property B. This means that no two entities may have the same B without having the same A. The difference is purely theoretical: whether we have the means of investigating this difference or not is out of topic for supervenience. The supervenience concept is also not concerned with "levels", just properties. These properties may be defined, or not, at different levels of investigation. A stronger version has also been proposed[7]: A property A strongly supervenes on a property B whenever each time it is possible to define properties A and B in a framework, no entity could differ in property A without also differing in property B, whatever the framework. This definition is detailed below.

As mentioned in the previous section, controversies occur when mixing different

7 For a more complete discussion on the various forms of supervenience see for example the Stanford Encyclopedia of Philosophy, accessible online at http://plato.stanford.edu/entries/supervenience

notions of causality. As an example let me consider a diamond made of carbon atoms. One could say that the diamond supervenes on the carbon atoms: when considering a particular, unique, set of atoms, one must also consider a particular, unique, diamond. No two diamonds may be made of the same atoms: this is material causality. On the other hand, if talking about formal causality, a diamond is generically "made of" a pattern of carbon atoms, and the atoms are all alike so we don't really care which specific atoms are used. A diamond is a specific pattern in carbon atom organisation, which distinguishes it from graphite: both could be defined with the exact same atoms, but their organisation is what matters. Therefore, the diamond also supervenes on the carbon atom organisation: two measurably different diamonds will have a different pattern, two exactly similar diamond, down to the atomic level, will have the same pattern.

But what does it mean to be "the same"? Equivalently, for the purpose of supervenience definition, what does it mean to be "different"? Is a reproduction "the same" as the original? The philosophical controversy arises when one chooses a different form of causality for the notion of identity, like the material and formal examples above. Digital objects are more concerned with the formal aspect, famous paintings with the material one, but what about the material reprint of an original digital artwork uniquely displayed for a specific exhibit? Then, there is also a functional (social) dimension to take into account. In some cases the material or formal identity does not matter as much as the functional identity. For example, when using a boat to escape a flood it doesn't matter whether the boat is made of wood or tin, or what form it has, so long as it floats. In this example "boat" has the functional identity "something that floats", irrespectively of the material or formal identity.

Compared to the weaker version that has just been explained, the stronger version of supervenience implicitly assumes we can define "the same" properties A and B across different frameworks. Of course, depending on the chosen perspectives for defining sameness in entities and sameness in properties, this stronger version may range from a tautology to a puzzling issue.

Without engaging in the controversy, that is assuming a particular definition for "sameness" has been given for a context, supervenience can then be used. However, in order to prove (or disprove) supervenience, one would need to derive an investigation tool that can precisely identify differences in the chosen properties. In the case of emergence between two levels of investigation, proving supervenience in practice would require measuring the exact state of the lower-level system. This is assuming such measurements do not themselves modify the lower-level state, as is the case in quantum physics. Without such a tool, the only remaining possibilities are to accept or reject supervenience as an axiomatic property of the system[8], or to build explicitly a system in which it holds. That last explicit system building scenario includes the case for deterministic computer simulations, so supervenience holds by definition for the examples in Section 2.4.

In any case supervenience does not help much for understanding the micro-macro relationship. The Wikipedia entry about supervenience[9] notes: "Supervenience has traditionally been used to describe relationships between sets of properties in a manner which does not imply a strong reductive relationship. [...] Supervenience allows one to hold that "high-level phenonema" (like those of economics, psychology, or aesthetics) depend, ultimately, on physics, without assuming that one can study those high-level phenomena using means appropriate

8 Which of course does not preclude its usage if it is assumed, see the downward entailment concept in Section 2.2.4.

9 http://en.wikipedia.org/wiki/Supervenience, version 21:10, 14 March 2007.

to physics". The next section deals with the micro-macro relationship, the problem of finding what is the cause of a given high level phenomenon. Section 2.2.4 deals with the strength of the reductive relationship that is mentioned in the Wikipedia citation.

2.2.3 Causal reductionism

This section deals with one of the major controversies: whether and perhaps more importantly how an "emergent" phenomenon is reducible or not to the lower-level elements and interactions from which it emerges. The different notions of causality that were previously introduced are analysed with respect to their relation to reductionism. This section thus deals with the bottom-up causal link. The next section deals with the other major controversy, related to the top-down causation.

Causal reductionism is the assumption that every phenomenon, whatever its level of investigation, ultimately have a cause, except possibly axiomatic properties which are postulated. If additionally a unique cause is assumed to have a unique effect then supervenience holds.

Depending of the causality perspective chosen, causal reductionism has different consequences. Material causal reductionism states in essence that whatever observed complex phenomena, they are always made of matter (in the broad compositional sense), be it an electron stream inside a computer or a magnetic field around the galaxy. Of course, material reductionists do not reject phenomena like consciousness or social constructs like flash mobs. It's just that stating that a brain and a crowd are made of atoms does not help much in understanding these phenomena, hence material reductionism may not be the best notion to use in these cases.

Formal causality gets around the problem by stating that brains and crowds additionally have an internal structure and governing laws that must be considered. Formal reductionism is then the assumption that such laws can always be found, that any higher-level effect is logically connected to the lower-level formal system. Unfortunately, most formal systems are known to be incomplete: No amount of formal causality may satisfyingly encompass all higher-level constructs. If, by analogy with [CHAI05], such intrinsically logically undecidable higher-level phenomena are the vast majority of all higher-level statements[10], then the question becomes whether these statements are really observable or not. Of course, it is still possible to postulate (whether this is true or not) that reality and all higher-level measurements are logically reducible. Therefore by definition any observer, part of that system, whatever its level of investigation, can only observe logically reducible statements. But even then, the observer may not be able to take advantage of the reduction in any efficient way: this would assume the observer has total knowledge of the underlying rules (which does not generally hold) and that it seeks perfect reconstruction, even for computationally incompressible statements (which is usually not what we want to do, see Section 2.4). Assuming formal reductionism or not is a matter of principles, and doesn't help much for practical investigations[11].

Functional causality offers another relation to reductionism. In this setup, higher-level phenomena are defined by their relation (function) with other higher-level phenomena and the environment at that level. Functional reductionism is a

10 Gregory Chaitin [CHAI05] gives a special attention to the case for real numbers in particular and to the limits of formal systems in general. The statement in the main text is not a citation, just a reformulation of what I think is a main idea in [CHAI05].

11 Unless one already has a working formula, in which case, of course, this paragraph doesn't apply. However taking advantage of a formula to relate the lower formal level to an observed higher level phenomenon would put that phenomenon outside the scope of some emergence definitions presented in Section 2.1.

contradiction, since the function is by definition only a higher-level construct. This is also the main holist approach: assume the irreducibility of the function to lower levels and consider only functional causes at the higher level. In that case, reductionism would take the form of assuming every phenomena has a function. But how is this function defined? It is not possible to isolate one part of the system and assign it a function independently of the rest of the system: Dependency loops are inherent to functional causality. An elaborated view on self-reference and an introduction to Howard Pattee's semantic closure concept can be found in [PATT95]. There may be dependency chains, which can be given some degree of functional causal power, but as soon as a loop is reached, the reduction argument breaks apart. For that reason, functional causality alone cannot meaningfully be associated with reductionism: there may be reductionism in a system, but then functional causality will not be the only causal relationship in that system. For the next parts of this dissertation, functional irreducibility refers to the first assumption that a function is only defined at a high level.

The fourth form of causality mentioned in Section 2.2.1 is the efficient one. Unfortunately, pure efficient causality also suffers from infinite regression. When given an efficient causality chain, one can always backtrack to the proximal cause, without end, so long as one stays purely in efficient causality. To break the chain, one requires another form of causality (such as material of formal). But then, the argument falls back to one of the previous points. Aristotle broke the argument at the other end of the chain, by stating the entities act according to their finality or purpose, which was relabelled the functional cause in Section 2.2.1. Once again, we're back to another form of causality. Reductionism is not a meaningful concept in pure efficient causality terms.

Are there other forms of causality one could apply reductionism to? Perhaps, but as previously discussed, these would certainly also come with their own lot of limitations. For example reductionism applied to the aforementioned notion of control also suffers from infinite regression (controlling the way to control the way... to control an effect). Amongst the four forms of causation presented in Section 2.2.1 the only one that is consistent with reductionism is the material one, and possibly the formal one too by construction or postulate. But as noticed, neither one helps much in understanding the emergence issue in general: emergent concepts are usually associated with a high-level functional definition. See also my summary and conclusion on this topic in Chapter 6.

2.2.4 Downward causation and the strength of emergence

Downward causation is the statement that some higher-level construct may exert causal power on the lower-level. This is the inverse problem as the one detailed in the previous section. The controversies are once again associated to what exactly one means by a causality relationship, as reviewed by Jaegwon Kim [KIM99]. As an illustration for a weak form of downward causation, or at least a downward explanation (see below), consider the case of a bonsai tree growing on a small rock. A bottom-up argument is to say that the reason for the small size of the tree is the lack of resources present in the small rock environment. But the reason is also top-down: the continuous care of a skilled gardener maintaining pleasant aesthetics. The aforementioned lack of resources makes no sense when considered at its own level, without the high-level goal of a pleasant aspect. This is a downward explanation for the lack of resources. Another example, of downward causation with

efficient consequences, would be the placebo effect if this effect exists[12]: when a patient is given sugar pills instead of active drugs and still reacts as if she/he would have received a real medicine. In this case the mind has a downward effect on the body.

Emmeche *et al.* [EKS00] distinguish between three types of downward causation: strong, medium, and weak. **Strong downward causation** is the mix of constitutive irreducibility and substantial realism of the higher level. The medium version is the combination of constitutive irreducibility, formal realism, and a refinement detailed below. The weak version is constitutive reductionism, formal realism, and a stronger version of the refinement.

Constitutive irreducibility is another way of saying that material reductionism alone is not enough: The building blocks that make up the higher level are assumed to involve a materially irreducible part. Substantial realism additionally claims that these new building blocks are matter as such, in the broad sense of a part of reality, that is "matter" at the higher levels is as valid as as "matter" at the lower levels. This amounts to the creation of new fundamental matter (broad sense) ex nihilo, and two identical low-level states could lead to distinct high-level ones thanks to the presence of new compositional matter at the high level. This contradicts supervenience of the emergent property on the low levels, as noted by Jaegwon Kim [KIM06]. When considering downward causation, the strong emergence requirements additionally state that new entities have material causal powers downward. Emmeche *et al.* [EKS00] give the example that strong emergence is like

12 The placebo effect is quite controversial, as exemplified by the heated argument between Asbjørn Hróbjartsson and Peter C. Gøtzsche [HG07] on one side, and Wampold *et al.* [WIM07] on the other. Pain treatment seems to be the domain with the least controversy, though even in this case the existence of a placebo effect is statistically hard to assert. In general the placebo effect, if any, strongly depends on the experimental conditions, as well as on what symptoms are treated.

considering that "the emergence of the cell as a living substance efficiently causes changes in the molecules, making them somehow specifically 'biological'". This form of emergence is usually only defined so as to be rejected, and Jaegwon Kim [KIM06] asks "whether it is a form of emergence at all".

Medium downward causation replaces the strong substantial realism requirement by a formal realism one, and adds another requirement detailed below. Formal realism does not mean formal reducibility, it means that unlike the previous case, the material component of the higher level may be formal elements of the lower one as opposed to material ones. Both requirements are generally accepted so long as these new entities are only considered from that level up, and so long as one defines "reality" from the higher-level entities point of view. For example, a simulation could be "real" for the agents in it, the "matter" the agents manipulate is real for them but only formally defined for the lower-level. The problem comes with the downward causation argument. At this point, no restriction has yet been put on what the agents are allowed to do on the lower level. The 'biological molecules' example given above for the strong version of downward causation could still have its formal counterpart with laws that are specific to the agents. For the medium version of the downward causation concept to be viable there must be some limitation that prevents the higher-level irreducible entities to modify, restrict, or more generally change in any way, the lower-level formal rules that lead to their existence. To use another example, so far, the mind could alter physical laws.

The term **strong emergence** usually[13] refers either to:

- The first creation of new compositional matter out of nothing.
- The lesser form of creation or modification of lower-level laws or effects.
- Both at the same time.

For the purpose of this dissertation, strong emergence is equated to considering either one of these properties.

Emmeche *et al.* [EKS00] refine the concept for medium downward causation by adding the requirement of an efficient causality restriction, which includes a temporal restriction. This requirement excludes from medium downward causation any change in the formal laws, together with any back-in-time change in initial conditions. The agents can no longer modify lower system laws in any efficient way, and the mind can no longer change the physical laws. In other words, the higher level entities can only constrain the domain of future possibilities of the system compared to past history, which is reminiscent of the cognitive domain notion as defined by Randall D. Beer [BEER04]. However, this does not preclude a unique lower-level state to coexist with several different higher-level entities, what Emmeche *et al.* [EKS00] call inverse supervenience.

One could argue that according to formal irreducibility, some phenomena are not logically reducible to lower-level rules and may thus be accepted, or not, with the same lower-level state. In the same way, it is possible to subscribe to the axiom of

13 But not always clearly. Mark Bedau [BEDA03] for example defines strong emergence when emergent properties are supervenient with irreducible causal powers. This formulation is confusing as it does not specify which causal powers and which forms of supervenience to consider. Some combinations are contradictory, but some others are valid, like in the downward entailment definition given further on in the main text. Jaegwon Kim precisely states that supervenience and irreducibility are two necessary but not sufficient conditions for emergence [KIM06], and notes that "how reducibility is to be understood in this context will require some discussion".

choice, or not, in ensemble theory. But then, such a phenomena cannot by definition have any downward causal power, which contradicts the downward causation concept.

Another interpretation is given by Emmeche *et al.* [EKS00]. By analogy with dynamical systems, the concept of boundary conditions is introduced. Medium downward causation would take the form of an influence of the higher-level concepts on the shape of the phase space, by changing some parameters, or by restricting the boundary to some region. Nevertheless, Emmeche *et al.* [EKS00] still do not fully clarify this inverse supervenience concept: their conclusion ends up with applying dynamical system rules only "in a somewhat metaphorical sense".

An additional potential issue with the insertion of that restriction for medium downward causation, is that it excludes some phenomena like the placebo effect[14]. This effect can be seen as a downward efficient causation from the mind on the body. Of course, the downward aspect depends on the perspective chosen for what is the mind, especially what kind of reductionism is assumed or not. In any case, Emmeche *et al.* [EKS00] do not pretend to solve all the controversies associated to downward causation; they propose an interpretation framework that admittedly does not cover all cases.

Jaegwon Kim [KIM99] proposes that the downward causation concept should be replaced by a **downward causal explanation** one: whether the explanation is given in terms of higher or lower level concepts. Jaegwon Kim [KIM99] concludes that while this may not be enough to "save real downward causation", "perhaps that is all we need or should care about".

The **weak downward causation** version is not affected by the inverse

14 Though that may not be a problem for this particular example if, as explained in the introduction to this section, the placebo effect is non-existent.

supervenience problem. As in the previous examples, there are new irreducible higher-level constituents due to formal realism (the question of what form of irreducibility is explained below). Formal realism also precludes new material effects to appear from these entities at lower-level, as was the case in the strong version. But unlike the previous examples, the constitution or composition of these new higher level entities is assumed fully materially reducible: the matter of the higher level is made of lower-level matter. This eludes the problem of the inverse supervenience of the medium downward causation case: material supervenience holds for weak downward causation. As for the medium case, an additional requirement states that weak downward causation cannot be interpreted as any kind of efficient causation.

So, what can be the non-efficient downward causal power of a fully materially reducible effect on the lower-level? Emmeche *et al.* [EKS00] give an example in terms of attractors of a dynamical system. If some higher-level concept is identified with being in an attractor basin, arguably functionally irreducible, then the downward causation is associated to the fact that the lower-level variables can only take some values in that basin and not others. The higher-level notion has "restricted" the lower-level capacities, though in this case Jaegwon Kim [KIM99] would rather say this is just a downward causal explanation, as the restriction is inherent to the system. What's not clear in the argument by Emmeche *et al.* [EKS00] with respect to weak downward causation is the type of irreducibility it allows. Given that material reducibility is assumed by definition, and given the remarks of Section 2.2.3, we may assume that only a computational incompressibility and functional irreducibility is possible with weak downward causation as defined by Emmeche *et al.* [EKS00] (not a formal one). Then, downward causation takes the form of a restriction on the lower-level possibilities. The question is then the extent

of this restrictive power.

Weak emergence is defined by Mark Bedau [BEDA03] when a higher-level property is underivable except by a full simulation (no shortcut can be found). This framework assumes material and local formal reducibility: The higher-level phenomenon under consideration for weak emergence must be fully reducible to a set of micro-effects that is "local". Nothing is said or even implied for other macro-effects using micro-effects outside this local set (the system is perhaps not globally formally reducible). Weak emergence is then equivalent to computational incompressibility (see Gregory Chaitin [CHAI74]) over that local set. Weak emergence rules out the medium and strong versions of downward causation.

Russ Abbott defines in [ABBO06] another concept related to downward causation: **downward entailment**. Downward entailment is an effect that is defined in a framework "functionally irreducible" together with "materially and formally supervenient". Unlike the previous weak downward causation concept by Emmeche *et al.* [EKS00] the introduction of a requirement about no efficient downward causality is avoided because of supervenience, as is detailed below.

The combination functionally irreducible / formally supervenient is the one that makes the explanations in [ABBO06] confusing at times. However, there is no contradiction. Russ Abbott takes as an example the Turing machine implementation in the Game of Life. The function performed by a Turing machine is not logically deducible from the game of life rules alone: this requires higher-level concepts, the program and the machine itself. When considered solely as a precise arrangement of game of life cells these concepts make no sense. In other words, the function is irreducible, but the formal aspect is supervenient (different Turing machine states necessarily imply different cell configurations). The formal aspect of Turing

machines and all computability theory is not reducible to the formal rules of the Game of Life. But the reason is the functional irreducibility that comes in between, otherwise there would be no reason why formal higher-level abstractions should be considered independently of the formal lower-level ones.

Once this issue is clarified, downward entailment amounts to reasoning formally on the higher-level to infer lower-level properties, using negative logic and supervenience. In other words, thanks to the supervenience part of the definition, it is possible to reason on the functional part. What this means "is that billiard balls, gliders, Turing Machines, and their interactions can be defined in the abstract. We can reason about them as abstractions, and then through downward entailment we can apply the results of that reasoning to any implementation of those abstractions" (Russ Abbott, [ABBO06]). The first part is a functional interpretation (billiard ball), with its associated formal system (reasoning about). Then, thanks to supervenience (any implementation), the results of the higher-level formalism may be propagated to the lower-level.

For example, using Newtonian physics on the billiard ball will put constraints on its lower-level material and formal implementation. This is saying that since no two higher-level balls may have the same lower-level implementation, results about a higher-level ball must necessarily involve its unique implementation. As in the game of life example aforementioned, the higher and lower levels formal systems are disconnected: The Newtonian laws alone do not apply to individual atoms directly, other effects must be considered (the ball internal cohesion, etc.). Downward entailment, by assuming supervenience, is a way to reconnect the formal systems after they were disconnected by the functional irreducibility.

There are undoubtedly many other possible variations on the subject of

downward causation, and some are given by Emmeche *et al.* in [EKS00]. So long as the hypothesis are well-defined, the academic issue is then: Can we test, validate or refute these variations distinctive properties, or are they purely theoretical?

2.3 Personal discussion

Both Mark Bedeau [BEDA03] and Russ Abbott [ABBO06] insist on the fact there is no intermediate concept between strong emergence and causal reductionism. As was explained by the previous sections such a statement requires clarification as to what form of reductionism is considered. In these [BEDA03] and [ABBO06] cases, material and formal reductionisms are assumed (deriving from a simulation assumes formal reductionism, even if only locally), but not a functional one (that's the whole point of weak emergence). This section presents the case for "functionally and formally irreducible" together with "materially reducible". This precisely forms an intermediate concept, though as we'll see further on, not a particularly useful one in practice. However, by analogy with the presentation by Gregory Chaitin [CHAI05] such concept should in fact be the predominant possibility.

The problem is related in part to the incompleteness of formal systems, which is discussed in this section, and in part to what we really want to do with these formal systems, which is detailed in Section 2.4. Given a sufficiently complex underlying micro-level system, there exist macro-level statements which are not provable[15] (whether positively or negatively) using only this system micro-level framework. The question of why and when such formally unprovable statements are observed in practice is addressed in the next section.

15 In this section we're concerned with the limitations on weak emergence as defined by algorithmic incompressibility, which is precisely the framework in which Alan M. Turing [TURI36] notion of uncomputability has consequent implications as was demonstrated by Gregory Chaitin [CHAI74]. The relation with Gödel's theorem is provided in Section 11 of [TURI36].

These statements are stronger than Mark Bedeau's weak emergence [BEDA03], in the sense that any simulation of the macro-level effects would represent a logical "proof", hence these macro-level properties are not weakly emergent. Of course, material reducibility may very well still hold, depending on the physical definition chosen for "material". But our formal equations cannot explain all higher-level observations. This irreducibility problem is generic, fundamental, and cannot be ignored.

The above irreducible statements could at first glance seem to be related to strong emergence. However, they have some crucial properties:

1. Their only consequences are necessarily expressed in "higher-level" terms, whatever that means in a particular context. By definition, if such an irreducible phenomenon could have consequences on the level at which the corresponding statement is defined, then this would negate the unprovability. For example the halting problem does not have consequences on the automaton rules themselves. The only consequences in that case are on the higher-level of the "program" and its execution in time. Of course the boundary between levels may sometimes be unclear as commented in Section 2.1.1, and occasionally the original formal system may be expanded to new axioms. But then we're really considering another, different, system with it's own higher-level unprovable effects. As Russ Abbott points out [ABBO06], at the lower-level the fundamental forces and particles of physics are already irreducible phenomena we use as axioms for the lower level realism and formalism.

2. There is no practical way to distinguish between a logically irreducible effect at a higher level, and a logically reducible but computationally incompressible one. By analogy with the demonstration by Gregory Chaitin [CHAI74] or with the

seminal article by Alan Turing [TURI36], the problem of identifying a particular phenomenon as logically reducible or not is itself undecidable. As a proof sketch, let's consider that one could define an order for the different possible simulations by size, for example using the same binary coding as in [CHAI74], or the "enumeration of computable sequences" from [TURI36]. Then let's try all simulations one by one in order. If we find a simulation that produces[16] the phenomenon, fine, we've proved it is both logically reducible and computationally incompressible (we found the shorter version). Otherwise, there is no way to decide when to stop, there is the possibility a larger simulation produces the desired phenomenon: We can't decide on logical reducibility. Consequently, given a functionally defined higher-level phenomenon, there is no general way to distinguish whether it is formally reducible but incompressible or formally irreducible.

3. Given the difficulty to "revert" even simple deterministic chaotic dynamical systems to their initial conditions and evolution rules, exhibiting a logical reducibility for a given practical problem (and not a suitably designed scenario) may be computationally very complex. Not only is it impossible to distinguish between a theoretically logically reducible or not higher-level phenomena in general because this would be undecidable, but proving reducibility for the systems that are theoretically reducible is probably intractable on any real-sized problem.

What about strong emergence? Either a phenomenon is logically reducible to micro-effects, in which case it is a case of formal reductionism, not a case of strong emergence. Or it is logically irreducible, but then, the first point above is in essence a rejection of the strong downward causation hypothesis in that case. Therefore, the

16 The difference between the present case and [CHAI74] and [TURI36] is that the phenomenon under investigation is defined functionally at the higher-level, not formally from within the lower-level system. Provided we have a way to test whether the phenomenon is equal or not to a simulation result, then the suggested proof is essentially the same as in [CHAI74] and [TURI36].

combination of both logical irreducibility and strong downward causation is a contradiction: this rules out strong emergence.

What remains are logically irreducible phenomena that do not have any effect at micro-level, though they may still be reconnected to the lower-level by using supervenience as previously mentioned, which provides a form of downward causal explanation or entailment as Russ Abbott [ABBO06] puts it. But according to the second property above, these logically irreducible phenomena are not distinguishable in practice from weakly emergent ones: The problem of deciding whether a particular statement is logically irreducible, or logically reducible but computationally incompressible, is both theoretically undecidable in the general case and probably practically intractable for the exceptions anyway. So, this explains why previous works using the weak emergence concept still remain valid: Even if irreducible phenomena (logical or incompressible) would be much more frequent than reducible ones by analogy with [CHAI05] so we probably have already met some, we can't make the distinction in practice. And in particular for what Russ Abbott calls a "very complex autonomous, self-sustained entity, whose functional definition is linked to other such autonomous higher-level entities and their environment" [ABBO06].

Pure formal reductionism for all higher-level entities is insufficient in the general case due to incompleteness. Strong emergence was rejected. As mentioned in the introduction to this section, the only remaining possibility is the one that was dismissed by both Russ Abbott and Mark Bedau: an intermediate level between formal reductionism and strong emergence. Since it is undecidable whether an observed functionally defined entity could be formally reducible or not, that intermediate level both complements and is indistinguishable from weak

emergence. I call it **formally irreducible emergence**, for lack of a better term.

One may then legitimately ask whether formally irreducible emergence is at all observable in a computer simulation. The next section investigates why it is in fact observable, and why the formally reducible or not aspect of an observed entity is unimportant.

2.4 Implications for formal systems

2.4.1 Analysing the results of simulations

Weak emergence is not a very useful concept for "understanding" an emergent phenomenon in practice. Of course, assuming we could obtain a simulation equivalent to running the system itself, then possibly we could make predictions if that simulation can be made to run faster than real-time. This is certainly useful, and to a certain extent this is how we already use numerical simulations, especially in industrial contexts.

However, the full simulation tells nothing about understanding the higher-level phenomenon as such. Understanding involves abstracting notions and entities that we can relate together by reasonably concise statements (compared to the simulation) that still produce good approximations. Consider as an analogy saying that as long as one sticks to the exact wave functions of quantum physics then quantum physics apply and the object is reducible to waves. But in the case of considering a macro-level object (like a stone, a flower...) what we want is usually not to consider it as an intractable bunch of waves, but rather to find how it relates to other objects at its own level.

The task is thus to find simple relationships that describe entities and their

interactions with good accuracy, so our limited human minds may comprehend them. Steven Weinberg [WEIN02] says that Science is concerned with simple things. A corollary is the negation of the possibility for an objective definition of emergence that corresponds to our intuition. An advanced futuristic artificial intelligence or an hypothetical alien entity could very well label as trivial phenomena our brain structures have no chance to comprehend.

Let's now consider a computer simulation which can be made fully deterministic and reproducible. If one sticks to the exact observations obtained from that simulation they are surely formally reducible (though perhaps incompressible). But what we want is to find reasonably concise and precise approximations of the higher-level entities and their behaviours that are produced by the simulation. Does formal reducibility still holds in this case? Can the simulation produce observable and reproducible phenomena, functionally defined such that they are irreducible to the simulation program laws?

2.4.2 Examples

Let's consider for the purpose of this argument that the observer has total knowledge of the underlying rules, which does not generally hold if the observer is part of the system, but which is reasonable in the case of a programmer examining a computer simulation. Each observed statement is then perfectly logically reducible, though some statements are computationally incompressible (no shorter simulation can be found). If what we want to do is finding a shortcut, a concise and reasonably precise law that can describe the observation, then there is no guarantee that the approximation is itself formally reducible.

Example 1: Generalisation across simulation runs

Let's write a program that plots the Riemann Zeta function on the complex plane strip with real part between 0 and 1. We then observe that zeroes appear exactly on a straight line for all the simulation runs, and for as precise a result as we wish by setting the floating-point resolution. Can we generalise to all future runs?

Example 2: Generalisation by formalising higher-level laws

We are given a complex simulation in which some results always appear nearby a simple curve (parabola, line, exponential...), but there are small variations. As is usual in physics, let's consider these variations are noise and then derive a law with the curve to predict the coming up of new points with good accuracy[17]. The curve cannot then be directly related to the lower-level system: the formal reduction applies only to the exact points that were produced, including what was considered noise. And even without noise what was really obtained is only just a mathematical conjecture, as in Example 1, and there is no guarantee one could formally generalise to other simulation runs. But the higher-level shortcut is potentially useful, an "emergent" law from the simulation. We may consider the emergent law on its own level as an entity in itself, and use it there as we would for "physical" laws. One could then try to apply the scientific method on the higher-level as suggested by Jochen Fromm [FROM06], by defining experiments to get the limits of that law, check conditions whether it applies or not, etc. Without caring one way or another for the reducible to the lower-level or not aspect of the higher-level entities.

17 Many physical laws work this way: we build descriptive laws of motion, heat propagation, etc... that give a reasonable approximation of the corresponding high-level effects. Then part of what we call "noise" includes the variations against these imperfect approximations.

Example 3: Further types of generalisations

Let's assume that the result of executing a program with many individual parts (cellular automata, simulated ants, etc.) is that some of them agglomerate into entities with definite shapes (gliders, a hatchery in an ant colony, etc). The formal reduction argument applies only to the exact state and position of each individual part. As before it may be that the higher level shapes suggested by the individual points are mathematical conjectures, possibly unprovable. Moreover, and especially in the case of an ant colony, the shapes are possibly not exact, or with fuzzy boundaries (see also the controversies about autopoiesis in Section 2.1.1). Yet these would be considered emergent by many definitions, and once again, worth considering in themselves at a higher-level. But as before, they may very well then be formally irreducible. This is now a generalisation in space, not time. When combining both time and space, i.e. when deriving laws for the evolution of the above higher-level entities, then formally irreducible emergence may hold on both aspects. When additionally the simulation is non-deterministic (for example in some cases when using threads or network links, or physical random number generators) the "emergent" entities may very well still be observed. If they do, we'd have an even harder time trying to reduce them to the program formal rules.

Example 4: Methodological consequences

We are now given an artificial life simulation, in which we note that on average agents have a preference for doing one kind of action rather than another. Is this statement formally reducible? If we consider the exact runs that were observed and the exact tendency of the agents that was noted, and assuming the simulation is deterministic, yes. But if we want to generalise to other runs, we don't know. Perhaps there are some regions in parameter space where the simulation does not

produce this tendency, for example. My argument is that reducibility doesn't matter in this case, that it's much more fruitful to consider the higher-level in itself (the agents tendency) and apply it a practical approach. By using that "tendency law" and applying it the scientific method as suggested by Jochen Fromm in [FROM06], one may perhaps refine that "tendency law" and find the regions of parameter space where it does not work (if any), or just be satisfied with the law holding for all usual runs. By analogy the Newtonian laws hold for most everyday life situations, though relativistic effects are necessary to explain some observations (and actually may be useful in practice too, for computing GPS positions accurately for example), and neither of them has yet been formally related to the lower-level of particle physics.

2.4.3 Conclusion

The task for understanding a phenomenon, simulated or real, amounts to finding a reasonably precise and concise approximation for that phenomenon and its behaviour. Whether that phenomenon is formally reducible or not cannot be decided generally (see Section 2.3) and does not matter for practical purposes anyway (see above). The difference between formal irreducibility vs formal reducibility is that in the former case the simple shortcut description is necessarily approximative, rather that very often approximative for the formally reducible case (due to the predominance of incompressible statements, see Gregory Chaitin [CHAI74]).

More generally speaking, by considering the higher level entities in themselves (functionalist approach) and trying to formalise their relations directly at the higher level (reductionist approach), one does not need to care whether these relations and entities are "emergent", reducible, or in any way logically connected with the lower-level system, in order to produce satisfying results at the higher level.

Once again, this is reminiscent of what happens in physics: higher level prediction laws (like Newtonian physics) are convenient but imperfect shortcuts for the formal system of equations describing interactions at the nanoscale. The normal procedure is then to try to refine the observations so as to validate or invalidate these laws, potentially leading to the creation of new measuring devices, and so on, until we either improve the higher-level theory or find a better one for explaining the observations. This what Thomas S. Kuhn calls "normal science" [KUHN92]. At this point, downward entailment (see Section 2.2.4 and Russ Abbott's [ABBO06] presentation) may be a way to reconnect the higher-level formal laws with the lower-level system.

The main implication for formal systems and for simulations in particular is that even on a computer, it is possible to observe logically irreducible functionally defined phenomena, thus formally irreducible emergence as previously defined. This is a counter-argument to the formal (logical, causal) reducibility objection to computer simulations: Depending on its setup, a simulation may still be adapted for the study of complex systems and emergence, even the formally irreducible one.

The next chapter extends on this argument to consider what can actually been done in practice for complex systems.

Chapter 3: Research problem and methodology

If knowledge can create problems, it is not through ignorance that we can solve them.

Isaac Asimov

This chapter explains the approach chosen for the present work so as to deal with the problems that were reviewed in Chapter 2. The next section introduces the research problem, and why addressing it represents an advance of knowledge. Section 3.3 presents the design and methodological choices I made to reach the objective presented in Section 3.1.

3.1 Research problem: How to deal with complex systems

The main ideas related to complex systems presented in Chapter 2 depict a legitimate object of study, but the wide range of application domains they're supposedly applicable to makes it difficult to synthesise results into a consistent framework. As mentioned in 2.2, there are as many definitions as frameworks for complex systems, and no real commonly accepted theory. This dissertation is thus not a proposal to create yet another isolated framework.

Chapter 2 and most current works on emergence and complexity fall in the category of descriptive approaches, which I divide in two categories. The first category includes the globalist works like these of Russ Abbott [ABBO06], Stuart A. Kauffman [KAUF93], Peter Cariani [CARI89], and more references presented in Section 2.1, which are necessary to realise ontologies of emergence-related

concepts and their relations. The second category corresponds to frameworks in which the relations between the main notions and concepts may be formalised, like these of Cosma R. Shalizi [SHAL01], Aleš Kubík [KUBI03], etc.

Most current global approaches of the first category have descriptive, but little if any predictive value concerning the emergent phenomena. The frameworks proposal of the second category do have the quantifying aspects that are necessary to produce predictions, as explained in Section 2.1.3, but may lack the generality of the global approaches. Both are necessary if we hope to build a comprehensive theory of emergent phenomena and complexity. The problem is of course, how to reunite them.

Hopefully, we do actually have a way to handle the situation: the traditional scientific methodology. Putting the main global ideas to test and using these frameworks for predicting the outcome of future experiments is the classically accepted way of proceeding, what Thomas S. Kuhn calls "normal science" [KUHN62]: making observations, creating tools to verify these observations, emitting theories, refining observations, etc. I do not think that a new kind of Science [WOLF02] is needed at this point[18], but rather that it is necessary to apply the current Science in order to get a new kind of perspective. When that is done, and if this is possible, then, perhaps, the new knowledge gained will help us refine or redefine the way of proceeding, possibly by designing explicitly what are currently known as Complex Systems so as to reach specific objectives.

In the meantime there is no consensus that would lead to a theory for emergent phenomena. So, applying each framework individually and testing the main ideas in practice cannot currently be carried on with the goal of confronting a globally

18 Though [WOLF02] may be more a call for using computers as experimental tools, rather than a call for a clearly defined new methodology.

accepted "theory of complex systems" to reality. Yet the incremental refinements mentioned above are based on predictive experiments that help refine the main concepts, which brings a better comprehension of the phenomena involved and their relations, which in turn may eventually hopefully make these ideas become more widely accepted.

In this perspective the research issue for Complex Systems Science is at this point not so much a problem of gathering more notions and frameworks (there are already too many of these), but rather a problem of refining the current notions and making them more widely accepted through empirical and objective testing: reality has the final word. The methodology to achieve this goal in the present case gives:

1. Apply the concepts related to emergence on practical problems, and see how well they do in a predictive way.

2. Define more experiments with the express goal to refine an aspect or another of these emergence-related concepts, so as to push their limits, if necessary by creating new tools along the way[19].

In order to make progress in the understanding of what emergence and complexity related notions entail, the idea is thus to put them to test in a predictive way. By exhibiting cases for which these ideas work or not on concrete problems, the goal is twofold: 1. understanding the limits of these ideas, and 2. promoting the use of emergence-related concepts in practical problems. Indeed what is currently the norm in engineering (according to Julio M. Ottino [OTTI04]) does unfortunately not make use of all the promises of complex systems, like robustness to failures, self-repairing, etc., by analogy with biological systems for example. The use of

19 When the theory has matured enough we may need to actually build dedicated equipment, like satellites in astronomy and colliders in particle physics, but we have yet to see a well-defined Complex System / Emergence theory before reaching this stage.

emergence-related concepts on practical engineering issues would not only help in reaching point 1 above, but also promote the idea that global considerations are worth for practical problems too.

However this is not enough. If we want to progress on the formalisation of a theory of emergence we will also need better tools and methods of investigation. The mere fact of creating these tools may itself bring new advances, in addition to their usage. After all, the development of new theories is intrinsically linked to the ability to perform better observations. Famous examples include the development of Astronomy (as reviewed by Philippe Dutarte in [DUTA06]), Chemistry and Optics (see the examples given by Thomas S. Kuhn in [KUHN62]), and there is no reason to think that Complexity would escape this pattern. The current disorganised stage is precisely indicative of the need for collecting more evidence and creating new tools.

The research problem that drives the rest of this dissertation is thus the testing of complex systems ideas in a predictive way, together with the development of new methods and tools for asserting these ideas validity.

3.2 Design and methodological choices

The unrestricted study for the applicability of all notions relative to emergence would be actually a daunting task equivalent to creating a theory of Complexity, and which has resisted all attempts so far. Hence, this dissertation does not vainly propose to resolve that general problem. It is more realistic to opt for advances of knowledge with respect to some well-chosen and identified notions related to Complexity, and apply to them the classical incremental approach. Thus my work and this dissertation should be placed in the more general context of studies about Complexity, while remaining practical enough so as to allow at least some progress,

rather than vagueness that would not contribute at all.

The two directions mentioned in the previous section are considered: predictively applying complexity notions, and building more efficient analysis tools.

In particular, the Edge of Chaos hypothesis is a good candidate for investigation. Indeed, that hypothesis suggests increased system capabilities in a region of parameter space where the system behaviour is neither in the "order" nor in the "chaos" phase, but somewhere along a critical line in between. Trying to push a system toward that critical line, and monitoring quantitatively the effects produced on the system is a good application example of the methodology. Section 4.2 relates the experiments that are performed in this perspective.

Another notion I have put to test is what form of downward causation could be used in practice. The relation with the predictive methodology is realised through a control problem: Controlling the system by a form of downward causation toward a high-level goal makes the prediction that the system will indeed be tuned in that direction. Measuring the influence of this control then allows to verify if this prediction is correct. This is presented in Section 4.1, which exhibits a case where a globally defined target for the system (that has no meaning at the lower-level elements) has influenced the outcome of that system, precisely by means of modifying the lower-level parameters. Hence, this is a perhaps weak but nevertheless downward causation (or at least causal explanation): the system parameters are modified to realise and test a prediction that is defined only at the global level.

The edge of chaos hypothesis and the downward causation notions are thus the choices I have exploited. The next chapter details how each notion is dealt with, and how the results that were obtained help refine these concepts.

On the side of tools and techniques the aforementioned experiments all need some form of quantification of the system state. In order to achieve this quantification several indicators may be chosen, depending on the context. Each experiment thus explains and justifies the reasons why particular quantifiers are chosen. Yet this is not enough: Since I need to improve some state of art techniques to realise the experiments in Chapter 4, I might as well derive generically applicable and independent versions of the algorithms I created. This is a PhD dissertation in Computer Science after all! Therefore, Chapter 5 is dedicated to these algorithms and details in each case what were the needs and how the new algorithms that are presented there form an advance in their own, compared to previous techniques.

Chapter 6 summarises all these experiments and analyses what progress in knowledge was made throughout this dissertation.

Chapter 4: Applications

Omnibus ex nihil ducendis sufficit unum

...

Et ſi on étoit accoûtumé à aller par douze ou par ſeize, il y auroit encore plus d'avantage. *[LEIB03]*

Gottfried Wilhelm von Leibniz

This chapter presents the environments where selected complex systems and emergence ideas were applied and tested in the course of my PhD studies. According to the methodology explained in Chapter 3 the goal is to apply these ideas in practice in a predictive way, so as to test them, which in turn could refine the theory.

Section 4.1 includes a scenario introducing artificial life agents without giving them any fitness function (no instruction) while still maintaining some level of control on the system. The goal is to exhibit a practical and usable form of downward control, independently of whether it is called a downward causation or explanation (see Chapter 2). An article [BROD05A] I derived from this work was published in the proceedings of the CEC 2005 international conference.

Section 4.2 shows that using ideas of dynamical regime and synchronisation is fruitful in a spiking neural network context, in the form of a new learning rule: acting locally brings global capabilities. The work corresponding to this first part of Section 4.2 was published [BROD06A] at the IJCNN 2006 international IEEE conference. The complex systems hypothesis that the "Edge of Chaos" corresponds

to increased system capabilities is then put to test in practice, by monitoring the system state quantitatively. This second part of Section 4.2 thus complements and extends the first part, and it will appear [BROD07B] in the IJCNN 2007 proceedings.

4.1 Controlling a population of agents using global properties

The environment that is the topic of this section was designed to test complex systems ideas in an artificial life context. The following concepts were tested:

- That a Complex Adaptive System can exhibit "emergence" of some higher-level features. The adaptive aspect means that the agents in the artificial life setup possess an AI algorithm so they can react to their environment.
- A weak form of downward causation, where the higher-level entities can be used in some way so as to modify the long-term behaviour of the agents.
- How an open and dissipative system may generate complex behaviours.
- A quantitative approach for measuring different characteristics of the system at both the level of the agents and at the global system level.
- How control may be achieved by using these quantitative measurements.

In addition to these notions a visual qualitative feedback was used as a tool for exploring the system configurations, and for monitoring the results of the high-level control. Sections 4.1.1 and 4.1.2 present the environment. Section 4.1.3 explains how global control is reached in this environment. A discussion is provided in Section 4.1.4 about the results that were obtained, and Section 4.1.5 gathers implementation notes useful for reproducing similar experiments.

4.1.1 Presentation of the context

Most applications of genetic algorithms concern the optimisation of a predefined function. Parameters that are supposed influential are gathered into a set called a "genome" by analogy with biology, and a directed random search is performed to get the best parameters for the optimal function result. While this approach is very efficient in many cases, it is not satisfying for the purpose of exploring the capabilities of a Complex System with respect to emergent phenomena: A system that is designed toward optimising a function precisely discards or minimises the "distractions" that could hamper this objective. Yet, the capacity to simulate the natural evolution for the agents is both conceptually attractive and practically easy to implement. This study thus relies on a genetic algorithm in the larger sense of the term: it gives the agents the possibility to evolve, but without giving them a predefined goal to reach. There is no explicit fitness function in this system. This allows for opportunities to observe more complex and emergent phenomena, that are precisely the topic of this research.

Of course, the simulation itself comes with its own collection of assumptions, necessary to implement the system in practice. Therefore, while no explicit fitness function is given to the agents, it can be argued that the whole program contains an implicit criterion that will determine the success or failure of the agents at "reproduction"; that is, how well they can survive and transmit their (possibly slightly modified) parameters to other agents[20]. "The nature of the agents' AI, the nature of the environment, and their coupling via the agents sensory inputs and effectors [as well as] the physics of the world, and the way agents interact, define the conditions for survival and reproduction. Agents live as long as they can, and

20 As noted by Grogono *et al.* in [GCSYZ03] anthropomorphism is both a plague and a useful tool when dealing with artificial life systems.

reproduce if they both wish to and can do so" [BROD05A].

In this system however determining with precision what the implicit fitness entails would be exceedingly hard: The coupling between the agents and their environment is defined in such a way that chance[21] has a non-negligible role in determining whether an agent will even survive, let alone reproduce. The implicit fitness of the agents is thus most probably formally an irreducible component: Running the simulation is the only way to know whether and when an agent would reproduce.

How is the control issue handled, then, if there is no fitness function to act on? The solution is to monitor the behaviour of the system at a higher level than that of the agents. Synthetic indicators of the system global behaviour are quantified. They are then used to define a goal for the system itself, which in turn allows to direct changes in the system parameters that are supposed the most influential. This is a form of downward causation: the agents possibilities and behaviour are constrained by quantities that are not defined at the level of the agents themselves. Notions like "the number of prey/predator population cycles" are quantified precisely and used to direct the system toward a goal with a measurable influence.

The next section presents the world physics and features, the way agents interact, and more generally the simulation parameters. Section 4.1.3 explains how the global control was achieved and the associated experiments. The results and the interest of this environment are discussed in Section 4.1.4. Section 4.1.5 consists in a few implementation notes.

21 In the form of a random number generator, the respectable and widely appreciated Mersenne Twister by Makoto Matsumoto and Takuji Nishimura [MN98].

4.1.2 The world

The setup

The world is three-dimensional, delimited by a ground, and cyclic along the X and Y dimensions. Figure 1 shows the bare world terrain:

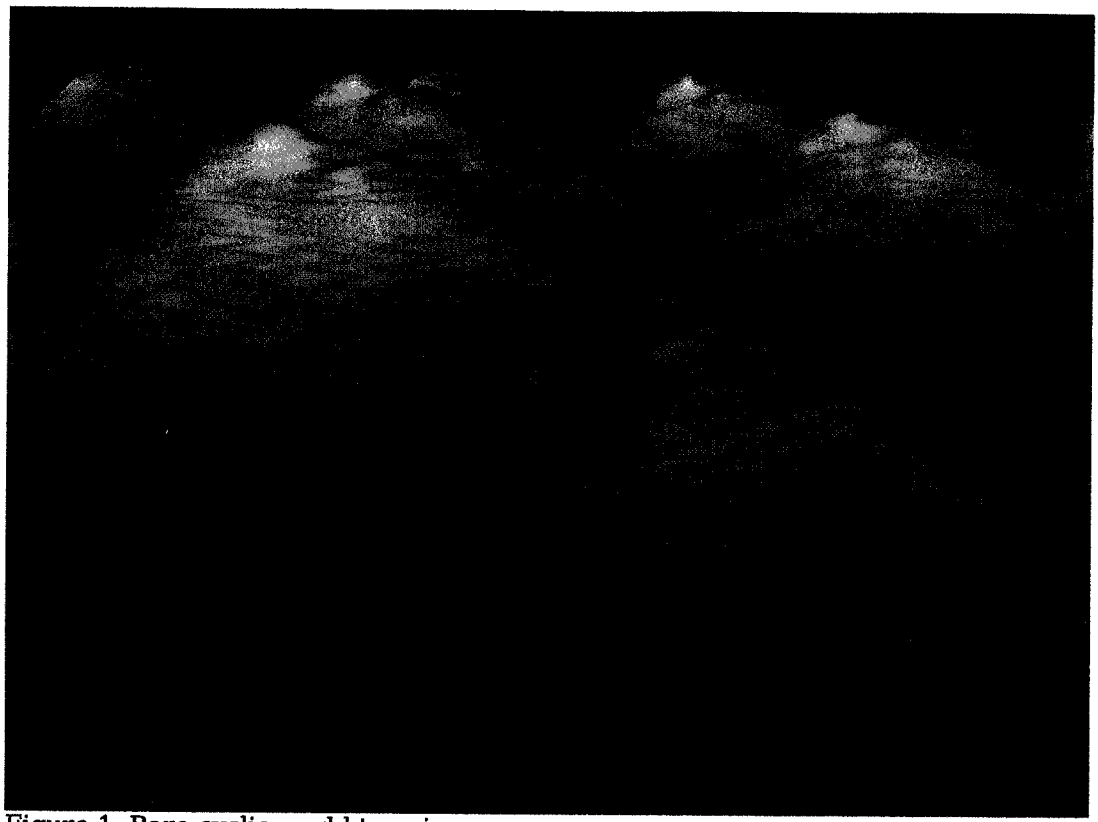


Figure 1: Bare cyclic world terrain

Details about the terrain generation are given in [BROD04] and are not duplicated here. In short, a spatially consistent noise is generated by applying smoothed inverse wavelet transforms, and then used as a cyclic height field. This bumpy surface makes the agents evolve on a really three-dimensional environment even if they only move on the ground. Obstacles (visible in Figure 3) are also scattered randomly on the terrain.

A physics engine simulates gravity and all the agents movements. Energy is also provided consistently with the simulation space and time units. It is the driving resource for all the dynamical aspects of the simulation.

The system is open and dissipative (see Section 2.1.1) with respect to this energy source:

- It is open because energy is continuously provided in the form of some "grass" on the floor, simulated as a density of available energy per surface, and "growing" at a constant rate up to a maximum. That energy is then collected by the agents in a way described below.

- It is dissipative because agents need to spend some energy in order to perform any action: moving, reproducing, fighting, etc. Even maintenance takes energy: the agents energy reserve decreases at a fixed rate and they "die" when they have no more energy.

In addition to "grazing", the agents may also "hunt" so as to gain the energy they require: they can chase after other agents and fight. A winning predator will then be able to convert the prey mass into an equivalent of energy (the "digestion" is perfect) up to a maximum capacity. Both grazing and hunting are parametrised by the agents internal AI, defined below in the corresponding subsection.

Prey-predator relations are fixed for the whole simulation. Agents belong to a species and may hunt on other species. For the experiment in [BROD05A] the five-species predation graph is cyclic and this cyclic property is robust to the extinction of up to two species, as shown in Figure 2.

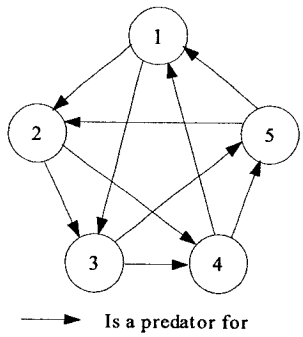


Figure 2: Predation diagram

Five species are defined in this experiment, numbered in order. Each of them is a predator for the next two species in a cyclic way.
If one or two species go extinct the remaining three or four may still form a cyclic predation graph.

Nothing distinguishes the species apart from this predation graph. The agents initial parameters are drawn from the same statistical distribution whatever their species. One of the measures defined in Section 4.1.3 consists precisely in a quantification of the diversity between species that results from the simulation (and which is expected due to separate evolution).

The agents and the AI

The agents are embodied as mobile objects reacting to steering forces. This model is inspired from a similar setup by Craig Reynolds [REYN99]. The agent body is simulated as a point object for the physics engine, and a radius is added for collision detection. Newtonian physics act on the point, the agent current velocity may only be changed by applying a steering force vector. These steering forces are classified into two categories:

- External effects, on which the agent has no influence. For example gravity and friction. These forces are applied whatever the agent does at each integration step of the physics engine.

- The agent's own steering decisions resulting from the AI algorithm described below. The agent must spend the necessary amount of energy to be able to apply the desired forces for the desired time.

There is a maximal amount of energy that can be spent in a given time as well as a maximum steering force amplitude. These respectively simulate the facts an agent can do only so much effort and that it has limited physical capabilities to realise this effort. The external influences are of course not affected by these limits. As a side effect gravity-assisted trajectories are possible: an agent sliding down a hill may reach a higher acceleration that it would normally be able to reach. In practice the AI is not sufficiently developed to take this effect into account.

A rotational momentum hack was introduced in addition to this model, since a point-object would otherwise be unaffected by rotations. Together with the steering forces, this means that the mass of an agent directly influences how much energy needs to be spent to change direction.

In addition to the point position and the scalar velocity value an attitude quaternion stores the agent orientation in space. The implementation notes in Section 4.1.5 give details on how the quaternion is additionally used for preserving a local vertical.

As in [REYN99][22], steering forces are then provided to perform basic actions. These are in the present case:

- Seeking, which returns a force to apply in order to reach a fixed point in space.
- Fleeing, which is the opposite of seeking: getting away from a fixed point as fast as possible.
- Avoiding, which can be applied to both fixed and mobile targets. It returns a force to apply in order to avoid a collision with the target. The current agent's

22 Though not with the same implementation choices, in particular for collision avoidance and for wandering.

trajectory if no force is applied is computed (straight line projection in the future at constant speed), together with the same estimation for the target based on its externally visible position, direction and speed. Objects radius are also taken into account. A null vector is returned if the trajectories do not cross. Otherwise, a vector perpendicular to the current agent's direction is returned, oriented away from the estimated collision point along the trajectory.

- Pursuit which is the same as Seeking but for mobile objects. This is used by predators for hunting preys.

- Evasion which is the opposite of Pursuit, and used for the opposite reason.

- Wander, which helps to explore the world by returning a random vector. The idea is to base the statistical distribution of returned forces so that the current direction will be the most probable outcome of applying the force: the null vector implying no change is the most probable outcome, large changes are the least probable. The rationale is to avoid Brownian motion-like trajectories and to ensure a kind of smoothness in the direction changes.

- Changing speed to a given target. A force is returned either in the current direction if a speed increase is desired, or the opposite. The result is a positive or negative pure acceleration that takes into account the current mass of the agent. If the resulting force is within the maximal acceleration range, the agent reaches the desired speed at a fixed future time (equivalent to the AI integration step, see the implementation notes in Section 4.1.5). Otherwise another call to this function is necessary at the next step, and so on.

No extra data is needed for the computation of these steering forces that the agent would not normally have access to. They can therefore be seen as useful subroutines for the AI.

The first task of the AI is to combine the various steering forces so as to react to the current situation. For example, in the presence of a predator and an obstacle, the AI must take into account both evasion and obstacle avoidance. Of course, trying to do both at the same time risks achieving neither one correctly: the agent would both collide with the obstacle and be an easy prey in this case. The forces are first weighted to give a larger influence to the closest objects, assumed to be the most immediate threats, using an inverse squared distance scheme. They are then additionally weighted by genetic parameters, introduced in the next subsection, such that some agents would preferentially favour fleeing and other collision avoidance in this example.

The second task of the AI is to find energy. Without sufficient energy the first task cannot even be accomplished. Energy must be collected for self-maintenance as well. The agent may choose to graze or to hunt. The decision to do either one is parametrised by the immediate observables (like the local grass concentration or the presence of preys) as well as by genetic parameters.

When an agent has enough mass and energy the AI may also try to reproduce. Conditioning the reproduction to energy gain is a way to direct the evolution so it produces agents that are capable of gaining energy. However nothing is specified as to how that may be done. The mating and offspring generation conditions are described in the genetic algorithm section below. The AI criteria for reproduction are chosen according to the number of visible predators, the amount of energy present on the floor, and more genetic weights.

The basis for the algorithm of the agents' AI is thus:

- IF not enough energy THEN hunt or graze
- ELSE IF good quiet place THEN spawn

- ELSE seek good quiet place

The genetic algorithm and the AI parameters

Agents additionally have the possibility to evolve by exchanging genetic material with other agents to produce offspring.

No energy is introduced during the reproduction operation: the only exclusive source of energy for the whole system is the grass. Consequently an agent needs to metabolise energy into some equivalent mass, up to a maximum physical limit (agent bodies are finite), if it wants to spawn: Part of this metabolised mass is in fact the new offspring. Some energy is also passed on to the newborn. Spawning may be performed only if the agent speed is below some threshold. As a result of all these properties spawning is a dangerous operation: the agent looses mass and energy that it will need to regain later on, and speed that makes it vulnerable to predator attacks. The newborns are similarly potentially fragile. Possible evolutionary strategies are thus: take the time to collect a large amount of energy and mass before creating a very robust child, or spawn fragile newborns at a much higher rate. No particular strategy is imposed on the agents, evolution may explore the whole range of possibilities.

There is no synchronization or cyclic phase during which the agents reproduce. There is no tournament selection for "best" individuals. Agents mate when they can do so and when the AI estimates this is a good time and place. Agents live as long as they can, together with their offspring.

Generally speaking there are two options for defining the genetic algorithm:

- Monoparental reproduction & mutation, like bacteria division.
- Crossover and mutation, which implies that agents must find some genetic

material for the crossover, coming from another agent.

Unfortunately the AI is not advanced enough so the agents would choose their partners according to a self-organised criterion (like for example a pretty blue plume on the head). This is a possible extension to this work, in addition to other extensions described in Chapter 6. At the time when the experiment was conducted however I was left with two choices: agents meeting each other on an explicit criterion, or chance collision.

The first choice poses the problem of defining this criterion, which goes against the policy chosen so far to let as much freedom as possible for the evolution.

The second choice interferes badly with the steering force approach: A genetic algorithm relying on chance collisions to reproduce ends up favouring AI that collide... which we don't want. Such a scenario was still considered initially, with the result that agents rapidly gathered into massive swarms where they could reproduce. This lead to overcrowded parts of the world which agents could hardly escape, especially when the population in each swarm reached a critical size where the input rate of energy was not sufficient for the survival of the whole colony. Another solution was thus necessary, that let the agents a greater freedom of movement.

A way to maximize agents interactions without relying on collision or other explicit rendez-vous points is to do like plants, fishes, mushrooms... This leads to the concept of a spore cloud[23].

When an agent gains enough energy the AI may thus decide to emit a spore in the environment. This action costs itself some energy. Then, later on, when an agent wants to reproduce, it can check in the environment for available spores (that are

23 Thanks to Nawwaf Kharma for pointing this idea out.

not its own, agents are not auto-fertile in this simulation).

The trick is to implement the spores as a volumetric concentration rather than as separate objects, so the spore/agent "collision" detection does not need to be simulated. The probability of fecundation is given by the total concentration of all compatible spores. The precise spore which is chosen is selected with a probability proportional to its current concentration, relatively to competing spores in the same volume. Note that this reinforces the asynchronous nature of this genetic algorithm: an agent may choose a spore emitted some time ago by another agent which is now dead.

Simulating the diffusion of spore clouds required adding an integrator to the physics engine. The revised engine integrates the classical diffusion equation $\frac{dc}{dt} = k\left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} + \frac{\partial^2 c}{\partial z^2}\right)$ with c the concentration and k a diffusion speed constant.

In addition, the spores efficiency decreases with time and sterile spores are removed. This is modelled simply by giving the spores a half-life time, like radioactive elements. Spores concentrations are thus decayed at each integration step. Together with a minimum concentration below which no fecundation is considered possible, this allows to keep the amount of spores in the environment within manageable limits.

Now that agents can find a compatible genetic material for crossover from within the environment, influential parameters for the agent survival may be modelled as genes. In this experiment the genotype corresponds to AI parameters. The agents phenotype is thus their behaviour, resulting from applying the AI with these parameters.

The "genes" are classified into categories corresponding to the different actions of the basic AI algorithm presented in the previous subsection:

– When to decide the agent has enough energy or not before going back to hunting or grazing. Evolution may produce individuals that collect a maximum amount of energy before reproducing, possibly enough to spawn multiple children at the same time. Or on the contrary evolution may produce individuals that reproduce as soon as they can.

– Whether to hunt or graze. The agent has access to the local grass concentration, the positions and apparent speeds (in the agent's own local coordinates) of the neighbour objects, and their nature (prey, obstacle, other agent from the same species). The agent's reserve of energy is taken into account as well. A weighted combination of these parameters is performed by the AI to decide whether to go hunting or grazing. Some of the genes are these weights.

– How to graze. The agent may decide to first look for spots which exhibit a large amount of energy before sitting there grazing everything and moving again. Or, it may decide to wander around and take whatever grass there is. This is a strategic choice to solve, a compromise between the amount of energy to risk spending looking for the best grazing places, and the energy that an agent might expect to gain once there.

– How to hunt. In the steering force model, what to do when there are multiple preys and obstacles. The forces may be contradictory and as mentioned above applying all of them at the same time risks achieving neither of the initial objectives. The weight given to each force in such a situation is determined by genetic parameters. In addition, a predator needs to engage some amount of energy in the fight. If that energy is more than what the prey has to fight back, then the predator wins. Otherwise the prey wins the fight and it escapes, with

the result that both the prey and the predator have lost some energy. For the predator, engaging more energy increases the probability of winning the fight, but also the amount lost if the prey escapes. This decision is controlled by genetic parameters: evolution may lead to more or less aggressive predators.

- Seeking a good place to reproduce also involves steering forces combinations, which are determined by gene weights. The criterion for what represents a good place is based on the number of visible predators, preys, obstacles, and agents of the same species, as well as the amount of available grass at this place, and of course if compatible spores are found. The combination of all these values is weighted once again by genes.

- Finally, when the agent spawns, it needs to pass on part of its mass and energy to its offspring. As mentioned above this decision influences the rate at which children may be produced. Genetic parameters thus influence whether agents will spawn fragile offspring very fast or whether they take the time to create more robust children.

Figure 3 shows a snapshot of the application at the end of a run. This is the colour version of the image shown in [BROD05A]. Species are numbered from 1 to 5 as specified by the colour order, so as to form the predation graph mentioned in figure 2: Blue hunts Red and Green, Red hunts Green and Magenta, etc.

Figure 3: Snapshot of the simulation at the end of a run.

White patches of floor show regions where all the grass was collected. Spore clouds fill the valleys. Some obstacles are also visible (white boulders and trees, but they differ only by their visual appearance and are simulated the same way). The population evolution for each species is overlaid on the picture. In this run Cyan went extinct rapidly, but Green enjoyed a long lasting calm period while the other species were fighting each other in prey/predator cycles. A closer inspection of the simulation replay shown that the topology of the terrain had a high influence: The plateau between the hills on the left forms a kind of sanctuary. This is also the explanation why Green enjoyed a calm period for a long time, albeit with a low population count: So long as no Blue or Red predator enters the plateau, Green

benefits from the protection of the heights. However there is only so much food available there and this limits the Green total population size. Some Green agents get down the hill from time to time due to this overcrowding, but they don't survive long in the valley that is occupied by the two Green predators. At some point a Blue agent discovers a wonderful reservoir of preys, and the Green population is quickly exterminated. The battle for the valley rages on for a few more population cycles, until Red definitely takes the lead. The simulation stopped when reaching a threshold of 2000 agents for one of the species population count. Preliminary experiments were conducted to set this number: all of them suggested that past 2000 individuals the species become the dominant one. 2000 was also below the maximum population size that can be sustained by the energy influx into the system in all cases. This maximum sustainable population limit depends on the amount of grass, hence on the terrain topology, hence on the random seed.

A similar situation was observed in all runs: an evolution of the population toward prey-predator cycles, then a final explosion of one species population. The plot 4 is taken from [BROD05A] and shows the population dynamics for a given set of environmental parameters and three distinct random seeds.

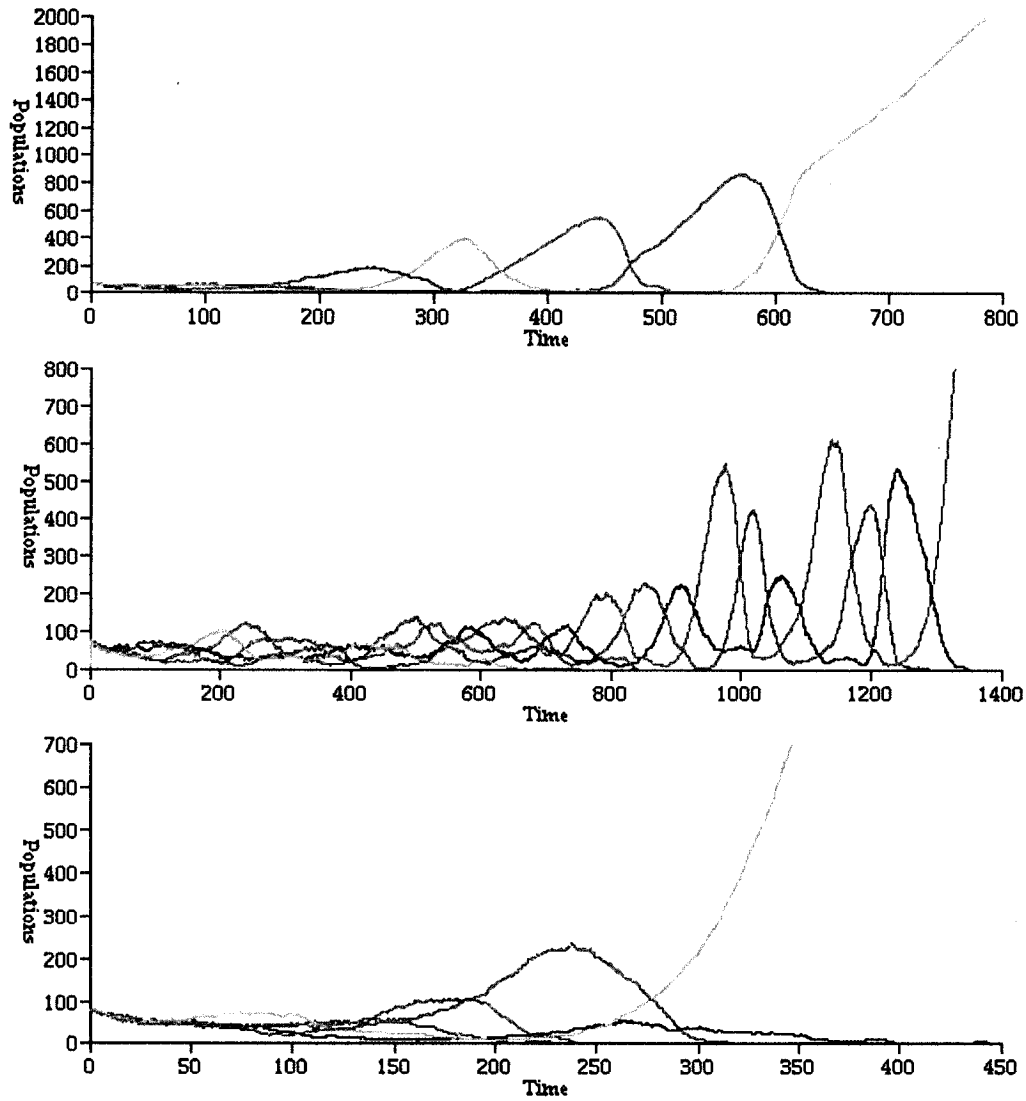


Figure 4: Population dynamics variability depending on the random seed

As previously mentioned chance has a high influence on the simulation outcome. This is not surprising given the examples where the topology of the randomly generated terrain has an important functional role (sanctuary effect and total quantity of available grass). With the same environmental parameters, figure 4 exhibits three runs with marked differences. The scale of the plots was chosen to

highlight the population cycles. As an indication the last red bump observed in the third graph is roughly the same size as the first red bump in the first graph, about 200 agents over 150 time steps. Some features common to all the runs may be extracted: there is an initial transient phase during which no species clearly takes the lead on the others. Then, the prey/predator cycles take place, followed by a final explosion. In the first run we also notice an inflexion of the Cyan population graph when its last Red preys are going extinct: Cyan has to rely on grazing and this doesn't bring as much energy as hunting, hence the Cyan individuals reproduce at a reduced rate.

Some characteristics like the number of prey/predator cycles are notions not present at the agent level, that can be said to "emerge" from the simulation. Yet, with an appropriate definition, they can be measured precisely and automatically: In this experiment a population cycle is defined when there is an increase of population by more than a threshold, followed by an decrease of population by more than the same threshold. The threshold is chosen to be the initial population size of 100 agents, so a first cycle can be counted only when the population of a species has doubled or more. With this definition there are 4 cycles in the first graph of Figure 4, 14 cycles in the second, and only 1 in the third. We now have a precisely defined and quantifiable "higher-level" notion to monitor.

The next subsection examines other variables that were monitored in this experiment, as well as environmental parameters. A method for controlling the system using the higher-level notions is proposed: a form of downward causation and control is thus exhibited on a concrete example.

4.1.3 Global control

In order to reach global control of the simulation there needs to be variables to act on. The number of population cycles presented in the previous subsection is one global notion that is defined and quantified at the level of the whole simulation, but which is not present at the level of the agents. This section first introduces other such notions, and then details how they are used to control the system toward an objective.

Measures of the system higher-level behaviour

Since all runs end with one species dominating the world, the time before that final state can be monitored and averaged over a batch of runs with the same environmental parameters. Ideally, for a stable ecosystem, no species would take over the world and thus this time variable would be infinite. Since no stable ecosystem was observed long transients are the fallback option: According to the Edge of Chaos hypothesis (see Chapter 2), long transients are generally associated to systems with the richest dynamics. Therefore, the goal here is to maximize the time before a species dominates the world.

Conversely a stable and diverse ecosystem would require that not too many species go extinct: the average number of extinctions for a batch of runs with the same environmental parameters is computed. The average time to extinction is monitored as well. The same reasoning as before applies and the goal is also to maximize this variable.

Both time measures are only crude indicators of the system potential for generating and maintaining diversity. A direct measure is more appropriate. The total genetic material diversity could be chosen, but unfortunately there may be

"neutral networks" (as defined for example by Shackleton *et al.* [SSE00]) of agents with the same phenotype (effective behaviour in the present case) and different genotypes. I thus chose to concentrate on genes that have a direct and easily interpretable influence: the AI weights that are used to determine whether an agent preferentially hunts or graze to collect energy. This is a behavioural characterisation that relies on anthropomorphism, yet it can be precisely defined in this setup. It does not completely eliminate the risk of neutral networks since some agents phenotypic behaviours may be equally well suited for the agent's survival. Yet we're now measuring diversity in the phenotype space, not the genotype one.

It is expected than the separate evolution of the species generates an interspecies variance when measuring the hunter/grazer indicator. What is needed is an assertion of the potential of the environmental parameters to generate or maintain this variance: The goal is to seek for sets of parameters that on average lead to more species differentiation.

A first problem is to define what constitutes a generation. Since reproduction is asynchronous it is possible for an individual to mate by choosing a spore emitted by potentially any previous generation. The agents present at the beginning of the run are given the generation number 0. This number is incremented when an agent spawns, so its offspring is naturally defined as the next generation.

Now that generations are defined the result of evolution may be asserted. The interspecies hunter/grazer behaviour variance is measured at the beginning and the end of a run, for the last generation in each species comprising more than 30 individuals. The ratio of these variances indicates how much agents from different species specialised into grazing or hunting. As previously mentioned this ratio is more than one due to separate evolutions. For some runs it remains low, between 1

and 2. But some other runs exhibited ratios of 15 or more. As for the population cycle indicator, this diversity measure is a global property that is not defined at the level of the agents, even if it uses directly some agent's genetic (lower-level) properties in its definition.

The number of generations per time unit is the spawning frequency, an indicator of how fast a species generates new individuals. As aforementioned fast spawning fragile agents is a choice, and generating a few but more robust children is another. The spawning frequency is thus an indicator of this evolutionary strategic choice. However in each run a species has taken over the world, and at this point the nature of the evolutionary pressure changes: there are no predators any more, and since the agents don't age and die, generating more children would mean more competition for the limited food in the finite world. If the simulation was run for a longer time, perhaps the agents would even stop reproducing and just graze the grass as it grows, leading to a potential equilibrium point where no evolution happens any more. This hypothetical possibility was not observed as I chose to stop the run when entering the last species world domination phase, at a 2000 individuals threshold. I however decided to discard the species that survived at the end of the run in the counting of the spawning frequency due to the change of evolutionary pressure aforementioned.

The last two global indicators that were considered in this experiment rely on the life time of the initial agents. As previously mentioned these agents can coexist with there own descendants. Yet they may also die, and the species turnover gives a global and measurable indicator. The initial agents life time mean and standard deviation are computed and the deviation/mean ratio is considered. It is small when initial generation agents die nearly all at the same time, for example during a

prey/predator cycle. This ratio is thus called the life time span of the initial agents. The mean life time of the agents, compared to the species own time before extinction, is also a crude indicator for how fast the first generation lets the place to the others.

Retrospectively these last two life time indicators do not bring as much insight into the system as the other ones. The results presented in [BROD05A] and reproduced in the next subsection show the greatest amount of "no significant variation" for the life time based indicators.

Nevertheless, now that we have quantifiable global measures on the system that are precisely defined, the next step is to try to relate them to the lower-level parameters of the simulation. The goal is to test if a form of "downward causation" could be used in practice, or whether this notion remains theoretical.

Lower-level parameters

Some simulation parameters may have a higher influence on the global measures than others. If the "open/dissipative" hypothesis for complexity holds (see Chapter 2), then the system unique source of energy influx, the grass growth rate, should be very influential on the simulation. Similarly the maximum grass density limits the total amount of energy that is potentially available and should also be an important parameter. Continuing on this energy approach, since the agents may metabolise energy into mass, which is then released during digestion when a predator has successfully caught a prey, the energy to mass conversion ratio is a good candidate for inclusion: it directly sets what is the value of a unit of mass in terms of energy storage.

The agents limitations on mass are logically the next parameters to consider. In

addition to their interpretation in terms of energy reserve for the system, the maximum mass an agent can reach and the minimum one necessary to create a "viable" offspring are limits that directly influence how many children an agent can spawn in one row. These parameters are thus expected to influence the evolutionary possibilities offered to the system. Similarly the maximum energy an agent can store is added to the influential parameters set.

Other genetic algorithm considerations lead to the inclusion of the reproduction costs: what energy gain is necessary so the agent is allowed to emit spores, and the spore emission cost itself.

Finally, the agent self-maintenance cost is also included as it sets how fast an agent dies if it cannot gain new energy.

Environmental influence over the global indicators

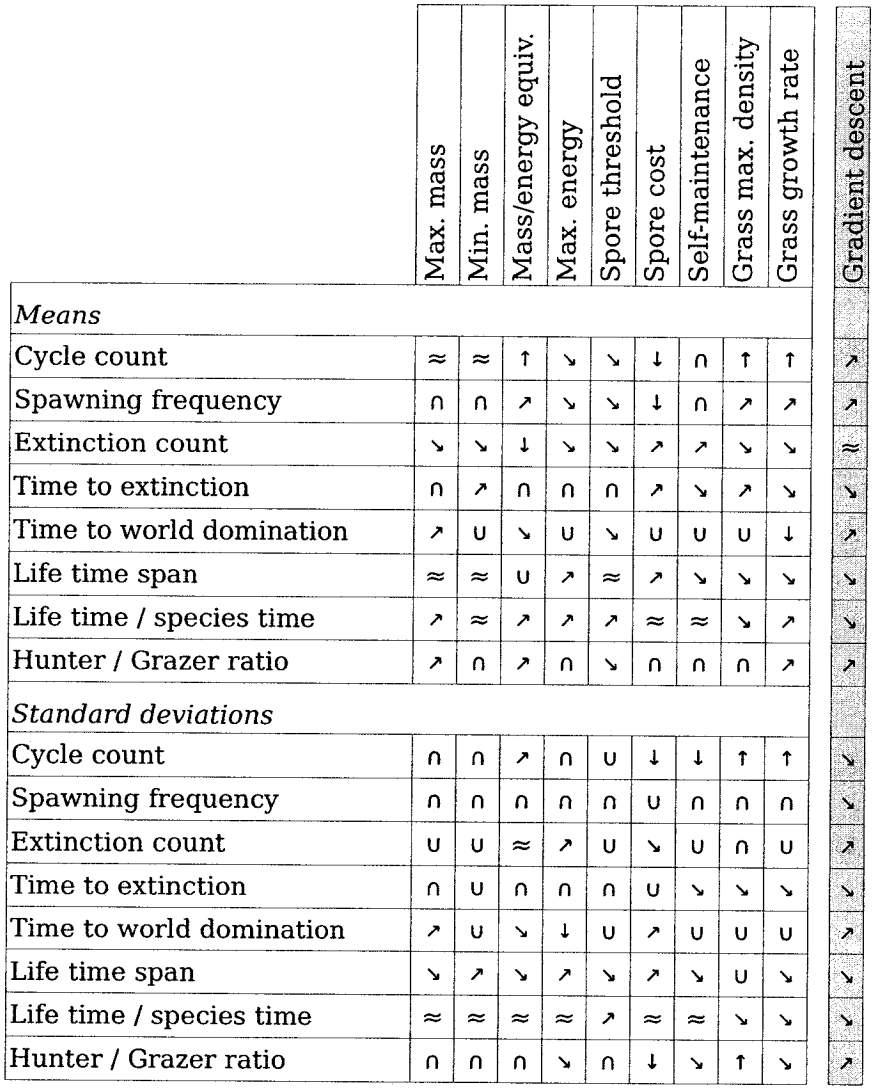
Visual inspection was the primary qualitative tool for estimating which environments were the most interesting, with complex dynamics and possibilities. I thus directed a random search by visual inspection and changing some parameters until I was satisfied with the environment. Once a good environment was found, I then fixed the corresponding set of simulation parameters and initiated a quantitative study with the global objective measurements.

There are height global indicators defined in this experiment, presented in the corresponding previous subsection. Nine simulation parameters were considered influential. Conceptually, each higher-level observable can take a range of values that depends on the lower-level parameters. In the simplest case, it takes only one value, and this defines a landscape of this global observable for each simulation parameter. In the usual case each higher-level observable may take a range of

values restricted in a complex way by the lower-level parameter values. Global control ideally avoids having to deal with this complex dependency.

One approach would be to work directly on probability distributions for the values taken by the global indicators. While this would allow to better capture the global indicator/simulation parameter dependence relations, this would also imply monitoring the changes in these probability distributions incurred by modifying the lower-level parameters. A simpler approach was chosen for [BROD05A], by modelling the higher-level observable distributions simply by their mean and variance. While this is crude, it has nonetheless worked in the present case.

The variations of the height global indicators mean and variance provoked by modifying the nine parameters are then analysed. Three batches of runs are performed for each parameter: at the chosen value and with positive and negative changes. The result is provided in a synthetic presentation in Figure 5. It can be seen as a qualitative snapshot of the indicators landscape in environmental parameter space, around the chosen point of interest. The last greyed column is the result of the gradient descent experiment described in the next subsection.



	Max. mass	Min. mass	Mass/energy equiv.	Max. energy	Spore threshold	Spore cost	Self-maintenance	Grass max. density	Grass growth rate	Gradient descent
Means										
Cycle count	≈	≈	↑	↘	↘	↓	∩	↑	↑	↗
Spawning frequency	∩	∩	↗	↘	↘	↓	∩	↗	↗	↗
Extinction count	↘	↘	↓	↘	↘	↗	↗	↘	↘	≈
Time to extinction	∩	↗	∩	∩	∩	↗	↘	↗	↘	↘
Time to world domination	↗	∪	↘	∪	↘	∪	∪	∪	↓	↗
Life time span	≈	≈	∪	↗	≈	↗	↘	↘	↘	↘
Life time / species time	↗	≈	↗	↗	↗	≈	≈	↘	↗	↘
Hunter / Grazer ratio	↗	∩	↗	∩	↘	∩	∩	∩	↗	↗
Standard deviations										
Cycle count	∩	∩	↗	∩	∪	↓	↓	↑	↑	↘
Spawning frequency	∩	∩	∩	∩	∩	∪	∩	∩	∩	↘
Extinction count	∪	∪	≈	↗	∪	↘	∪	∩	∪	↗
Time to extinction	∩	∪	∩	∩	∩	∪	↘	↘	↘	↘
Time to world domination	↗	∪	↘	↓	∪	↗	∪	∪	∪	↗
Life time span	↘	↗	↘	↗	↘	↗	↘	∪	↘	↘
Life time / species time	≈	≈	≈	≈	↗	≈	≈	↘	↘	↘
Hunter / Grazer ratio	∩	∩	∩	↘	∩	↓	↘	↑	↘	↗

Legend:

- ↗ Direct variation
- ↘ Inverse variation
- ↑ Strong direct variation
- ↓ Strong inverse variation
- ≈ No significant variation
- ∩ A maximum was observed at the initial point
- ∪ A minimum was observed at the initial point

Figure 5: Variations of the global indicators with respect to the simulation parameters.

Many results correspond to the expectations. For example when the self-maintenance cost is increased there are more and faster extinctions, and when the spore emission cost is increased there are less generations.

Energy considerations are more interesting. It was not immediate for example that increasing the amount of energy would cause more prey/predator population cycles. Indeed, more energy could lead to less incentives for hunting compared to grazing. Yet, both the grazer/hunter behaviour diversity and the number of population cycles increases, despite the fact that it takes less time for a species to dominate the world. Hence there are more cycles in a shorter time, which is also reflected by the shorter life span of individuals from the first generation. Consistently with observing more cycles there is an increased spawning frequency, together with a greater hunter/grazer behavioural diversity. All this fits the open/dissipative considerations: more energy to dissipate corresponds to more and faster diversity.

The maximum mass parameter results are more difficult to interpret, but worth mentioning. This parameter was actually the most influential for increasing the hunter/grazer diversity, yet it has no influence on the number of population cycles. The previously mentioned consequence of more mass – in terms of maximum amount of children an agent can spawn in one go – may have a role, but does not in itself provides a sufficient explanation. More mass also means more energy gained by a predator when digesting a prey, so possibly there is an additional evolutionary incentive to specialise toward hunting.

The global indicators standard deviations are also indicated on Figure 5 but they do not provide much insight. Most of the cases are either an extrema at the point of investigation, or no significant variation. This is suggestive that the landscapes

defined by the standard deviations are jagged and that the deviations are too erratic to serve as reliable indicators.

Global control

Now that a (admittedly crude) mapping between the simulation parameters and the global indicators has been established it is possible to use it the other way around.

I propose to define a target objective to reach, expressed only with global notions that are not defined at the level of the agents. If the system can be directed, or controlled, to reach that objective, then a weak form of downward causation has been reached: The possibilities of the agents have been constrained at the agent lower level, by notions that cannot be expressed at this level.

The environmental influence map that was previously computed may now serve to reach this objective. This map can be seen as a snapshot of a high-dimensional parameter landscape, around the initial point. By following the gradients in the directions of interest it shall be possible to reach at least a local optima.

In the present case the chosen goal was to increase the potential of the environment for sustaining stable population dynamics: Ideally no species would end up dominating the world while there would still be prey/predator cycles. It is not known whether a point in environmental parameter space exists where the simulation leads to such a stable ecosystem. Perhaps there are large regions where this happens, and perhaps the simulation does not contain the (unknown) conditions that would be necessary for unending prey/predator cycles. In either case, following the gradient leads us closer to the objective.

The subsection introducing the global measures has presented the time

indicators in terms of transient phenomena, for the ideal case where an infinite time corresponds to the desired goal (no final world domination). The target now is thus to increase these time indicators. More population cycles are also desired to match the longer times: we ideally want a dynamic system, not a frozen one. More diversity would be interesting as well.

Analysis of the Figure 5 shows that the worst factor for the time it takes for a species to take over the world is precisely the energy input into the system, the grass growth rate. This parameter is thus reduced, despite having beneficial effects on other global measures (population count especially). On the other hand the maximum grass density is increased: The current value corresponds to a minimum in the time to world domination, increasing it has a strong positive influence on the number of population cycles, and on the average time before a species goes extinct.

The maximum energy an agent can store is decreased due to its negative influence on the number of cycles, and because the current value is a minimum for the time to world domination so changing it either way is beneficial. Spore costs are also at minima so should be changed: they are decreased given their negative influence on the number of population cycles.

The self-maintenance cost has opposite effects on the desired features so it is kept unchanged. The minimum and maximum mass limits are increased because of their influence on the time to world domination, and because they could reduce the number of extinctions. The mass value in terms of energy units is decreased for similar reasons.

The results of this gradient "descent" step are provided on the last column of the array in Figure 5. The time before world domination has increased, together with the number of cycles, the spawning frequency and the behavioural diversity. Species

extinction happen sooner on average, but then, the total number of extinctions didn't change. The result of this gradient descent step is thus a success.

The whole study could be conducted again with the new environment: First, batches of runs to assert the shape of the global measures landscape at the new point. Then, a second gradient descent step, and so on.

In practice this would have required too much computational power[24] for little added benefit: This experiment has already shown that controlling a system by modifying its lower-level parameters according to a goal defined only with global notions is possible, therefore realising a weak form of downward causation as previously explained.

4.1.4 Discussion and perspectives

An anonymous reviewer commented for the initially submitted version of [BROD05A] that "the discussion here is too focused on the problem of emergence". I therefore adapted the paper perspective so it concentrates more on evolutionary computation problems like the ability to maintain diversity without a fitness function. The present dissertation has reverted the situation, exposing more of the issues about emergence and less concerning genetic algorithms. Yet, it would not be complete without mentioning these genetic algorithms considerations.

One problem encountered during the initial design was indeed to find a suitable engine for maintaining diversity. Several candidates were proposed in the literature: Mutualism was identified by Pachepsky *et al.* as promoting diversity and stability in

24 30 of runs per batch for each parameter distinct value makes it 810 runs for building the gradient map of Figure 5. Running the preliminary experiments on a Pentium IV machine was feasible, but building the map required running the experiments on many machines in parallel (30 machines allows to run a month of computation in one day), and the material possibly to perform these parallel experiments was not always available.

[PTJ02], Jef Huisman and Franz J. Weissing analyse that Competition leads to unpredictability in [HW01], and the role of Predation is reviewed by Chase *et al.* in [CAGDal02]. In this study the simplest solution was to produce a competition for access to the energy and a possibility for predation. This justifies the whole grazing/hunting scenario. Similarly, spore emission is performed on energy gain and thus the genetic algorithm should logically end up favouring agents that are somehow successful in gaining energy, fuelling the competition.

However the results of this experiment are better supportive that an energy *increase* is the source of the complex behaviours, rather than a *decrease* that would correspond to more competition. The energy influx was decreased in the gradient descent experiment because it had a too detrimental effect on the time needed for a species to take over the world, but all other indicators suggested that the energy was in this case a source of diversity: positive effect on the number of population cycles, on the spawning frequency, and on the behavioural hunter/grazer diversity. The maximum grass density exhibit other similar properties and was even increased for the gradient descent.

Hence, at least in this simulation, competition is not the main driving factor for a diverse evolution. Feeding more energy in this open and dissipative system, on the other hand, does generate the desired diversity.

By analogy with turbulent physical systems it would be interesting to push this energy consideration further. Indeed physical systems exhibit different phases according to the amount of energy that is fed in the system (like a saucepan on the oven): quiescent heat transfer, convection rolls, and turbulence. An extension to this experiment would be to push the limit of the "open/dissipative" analogy in this virtual world and check whether these stages occur as well or not.

As an illustration of Sections 2.3 and 2.4, the global measurements realised for this study are all logically reducible to the simulation rules, even if they are possibly computationally irreducible (the only way to get them is probably to run the whole simulation). I observed that in all the runs a species ends up dominating the world, but as aforementioned perhaps there are environmental setups where this does not happen. Had I tried to fit an exponential curve to the population cycles exhibited in the Figure 4, and used the exponent as a global measure for how fast the population explodes, then each estimated exponent would also be both formally reducible, and a good quantifiable global indicator. The correlation coefficient obtained during the exponential fitting would also be formally reducible. Yet, there would still be no proof that the population generally follows an exponential divergence across multiple runs. I could for example have collected as much correlation coefficients as desired over many runs, all that would have done is increase (or possibly decrease) the confidence for the existence or not of a higher-level general exponential law for the population cycles, not prove it[25]. In other words, the exponential law is a higher-level postulate, irrespectively of the simulation rules. Yet, the exponent indicator for the population divergence speed would be quantifiable, and a good candidate for inclusion this study main experiment.

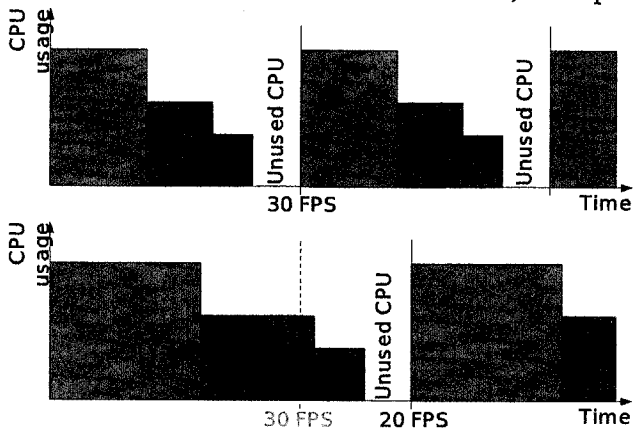
This is a good example of the observation that whether or not emergence is reducible does not matter much in practice. Trying to apply the main ideas like downward causation, energy considerations, diversity, etc. in a practical way and see how well they work on concrete examples is much more fruitful: This is one of the main points of this dissertation.

---

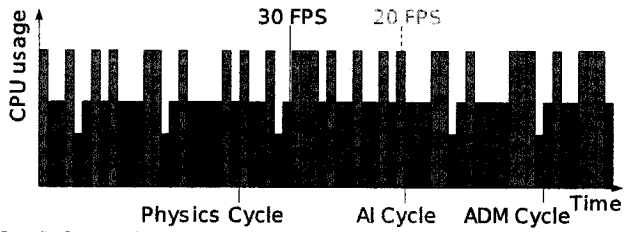
25 The exponential is just postulated as an approximation. Even trying one by one all valid bit patterns for all floating-points that may be used for the environmental parameters, the order of 2^{64*9} possibilities, would just produce a final value for how well the approximation holds, but not change the way it is defined.

4.1.5 Implementation notes

Some implementation notes were already given in [BROD04]. In particular, the simulation of the world relies on a discrete event scheduler. Events are generated cyclically for each agent's AI, for the physics integration, and for diffusing the spore clouds. The main advantage is the distribution of the events in time, allowing for a graceful degradation in case the simulation is overloaded, as depicted on Figure 6.



Frame drop effect when waiting for the next vertical refresh rate divisor. In this example, a vertical rate of 60Hz leads to 30 frames per second. If the computations can't fit in one frame, then the frame length is increased to the next divisor at 20 frames per second. Not only does this result in a drastic drop in frame rate, but CPU is still wasted.



Using an event scheduler allows to spread the events in time, with separate cycle lengths. If all agents can complete their physics during one frame, the visual result will still be smooth. Otherwise, only those agents which could not complete their physics won't have moved during this frame. This is a graceful degradation case, and may even be unnoticeable if the number of agents skipping a frame is low enough. Moreover, these agents will then move before the others on the next frame, so the degradation is statistically diluted over all agents in time. AI and administrative tasks don't have to be applied as often as physics, and have little if any immediate visual effect, so can tolerate an even worse fate. Finally, no CPU is wasted waiting for the next frame, provided the scheduling overhead is negligible compared to the average execution time of the events.

Figure 6: Distributing events with a scheduler results in smoother simulations

The discrete event scheduler separates the simulation time unit from the real time. This allows in particular to pause and restart the simulation for visual inspection, as well as slowing it down or speeding it up if the computational resources are sufficient. As the events are posted always in the same order the simulation is completely deterministic: it may be run in batch mode as fast as possible, without graphics, and later on replayed to investigate the most interesting runs. These abilities have been invaluable investigation tools, especially during the initial exploratory experiments.

Rotation arithmetic is implemented purely in geometrical terms, mapping one original attitude quaternion to a final one, which avoids the need for trigonometric functions. In addition, since there are an infinite number of possible axis/angle combinations for mapping one direction to the other, it is possible to choose one of them with an additional desired property. The shortest angle is a usual candidate, but this choice would imply the undesirable side effect of modifying the agent's local "upward" basis vector widely. A more appropriate choice here is the axis/angle combination that maximally preserves the local "up" vector. By using the ground normal as the vertical, this allows the agents to smoothly follow the floor curvature as they move.

More details and the random seeds used for the experiments are available together with the well commented source code. It is available as free-libre software (GNU-GPL) on my web page at http://nicolas.brodu.free.fr and in an independent third-party version control system at https://gna.org/.

Thanks also goes to the Open Scene Graph team for providing a high-quality graphics library for managing a three-dimensional scene. See the project site at http://www.openscenegraph.com.

4.2 A test for the Edge of Chaos hypothesis

This section reports results that were published in [BROD06A] and accepted for publication in [BROD07B]. As for the previous section, the system is modified locally, and the effect of these changes are monitored globally. Unlike the previous section, control is reached by using local notions. So, while the previous work was concerned with the applicability of a form of downward causation, the present work uses a more conventional approach of "engineering" the local interactions in the system in order to bring desired global changes. The goal is once more to apply some notions associated with complexity (see Chapter 2), and especially to put the Edge of Chaos hypothesis to test, as well as to define a quantitative approach to monitor how the system behaves in practice.

Some tools were necessary to reach this objective and had to be created for the occasion: An algorithm for computing the multifractal spectrum of a time series and another one for computing the statistical complexity of a system, both taking in account the addition and removal of on-line data in an incremental way. These algorithms are presented respectively in Sections 5.2 and 5.1, but the choices and necessities that lead to their creation are detailed below.

The notions that are tested in this section are:

- The concept of a critical line between order and randomness where a system benefits from increased capabilities, also known as the Edge of Chaos hypothesis.
- A quantification for what is the state of the system with respect to this critical line, if it exists.
- Synchronisation and incompatible constraints are used to drive the system

toward this critical line. The synchronisation acts locally on the local elements dynamical state, while incompatible constraints refer to structural contradictions.

The next section presents the background and context for this work, the environment in which the above notions are tested.

4.2.1 Background on recurrent neural networks and Liquid State Machines

Liquid State Machines (LSM) were introduced by Maass *et al.* in [MNM02] and [MNM03] as a way to harvest the combinatorial power of a recurrent neural network. Input data are provided, which the recurrent network processes and stores as a "reservoir" of non-linear transforms. The neuron activities are then fed to a weighted sum classifier which is given the task to find input transforms that answers a given problem. [MNM03] gives examples for computing polynomial combinations on the inputs as well as spike coincidence detection.

More precisely, a LSM comprises the elements depicted in Figure 7:

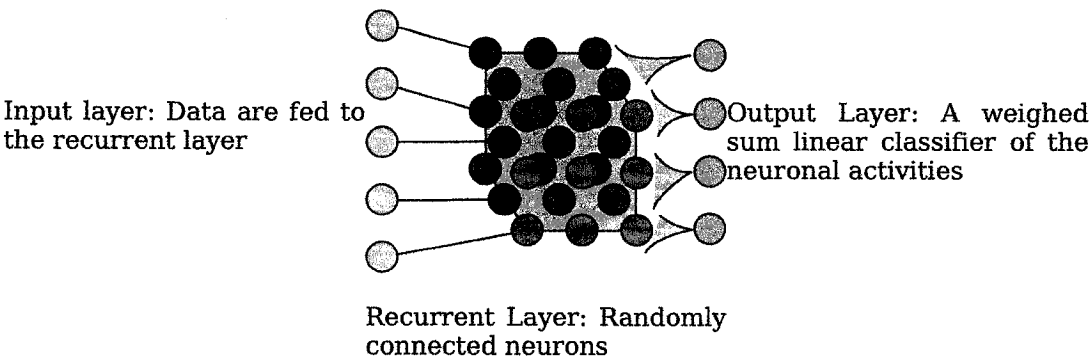


Figure 7: Liquid State Machine architecture

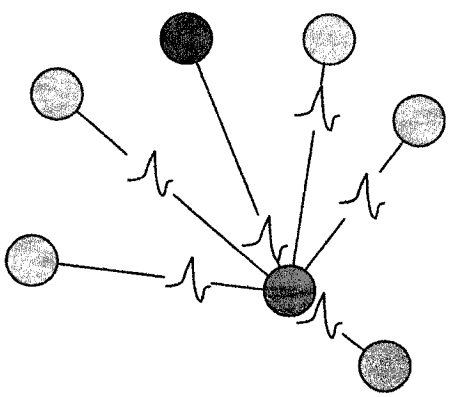
Herbert Jaeger [JAEG01] introduces Echo States Network, which are the independently discovered equivalent of the LSM but with continuous instead of

spiking neurons. Both approaches use the same "reservoir computing" method:

- The inputs are transformed in many different and intricate ways by the recurrent layer.
- The output linear classifier draws upon the diversity generated within the recurrent layer so as to approximate a desired function.

Both LSM and Echo State Networks have their own domains of applications. LSM were initially derived from biological considerations, and are well suited for the on-line processing of spike trains as they come (Maass *et al.* [MNM03] give an example for speech analysis). Echo State Networks were initially derived from engineering considerations and are well adapted for the task (Herbert Jaeger and Harald Haas [JH04] provide an application to wireless communication).

The specificity of the LSM is to use spiking neurons as the elements of the recurrent layer reservoir, instead of continuous transfer functions. Many types of spiking neurons exist (see the review by Wulfram Gerstner and Werner M. Kistler [GK02]), ranging from detailed models of biological neurons to less realistic but more efficient representations. A spiking neuron task can be schematically represented as in Figure 8.



A spiking neuron accumulates the energy it receives in the form of spikes (short impulses of energy). If a threshold is reached the neuron itself sends a spike. Otherwise, the stored energy decays progressively, until another spike is received.

Biologically motivated models also introduce a minimal delay between spike emissions. When the neuron emits a spike it enters a refractory period during which it temporarily ignores further spikes, until it becomes ready again.

There is also a transmission delay between the emission of a spike and its reception by the destination neuron.

Figure 8: Schematic representation of a working spiking neuron.

Liquid State Machine were defined in [MSN02] as computing devices "without stable state". In fact they are also an open and dissipative environment that can perform computations:

- Dissipative because the energy stored by the neurons (membrane potential in the biological case) decays with time after a spike a received.
- Open, because of the assumption extra energy is available so the neurons may emit spikes, which are by definition energy impulses higher than rest state, and because input spikes are continuously provided to the network.

The recurrent loops of the "reservoir" layer also are naturally subject to positive feedback. Taken together with the dissipation, these form what Robert Legenstein and Wolfgang Maass call a "fading memory" effect [LM07A]: Some spikes may be sustained by the recurrent loops and fade long after the inputs that generated these spikes are gone. Yet they should fade after some time, or the system becomes chaotic and it is not possible to relate the outputs to the inputs. Herbert Jaeger's "echo state" property in [JAEG01] is the equivalent requirement for continuous instead of spiking neurons.

The LSM is thus an environment where long transients and complex dependencies take place: a prime target for testing the Edge of Chaos hypothesis.

The "reservoir computing" approach has actually been already analysed in terms of the Edge of Chaos. Natschläger *et al.* ([NBL04], [BN04]) use boolean gates instead of spiking neurons as the "reservoir", and introduces a notion of separation (the NM-separation, see Section 4.2.4) that defines a good indicator for the system state position with respect to the critical line. Stefan Bornholdt and Torsten Röhl [BR03] modify the network topology in a way reminiscent of the Hebbian rule that is used below, in order to investigate the self-organization properties of the network.

[JAEG01] introduces the notion of Memory Capacity for Echo States Network. High performance with respect this Memory Capacity in a boolean network using the reservoir computing approach is shown to happen near the critical line in [BN04]. [LM07A] explicitly mentions that "the requirement that the network is operating in the ordered phase is important in these models, although it is usually described with a different terminology". So the Edge of Chaos is desirable but should not be crossed.

The Edge of Chaos hypothesis was thus already well mentioned in a "reservoir computing" context. Yet, in all these previous references, the recurrent layer in kept untouched and training occurs only on the output classifier. Quantitative indicators are measured on the unmodified recurrent layer, and a posteriori postulated to validate the Edge of Chaos interpretation.

The present work both introduces training into the recurrent layer, and tests its influence with quantitative indicators. The Edge of Chaos hypothesis is used predictively to define the experiments: interpreting the results of the training then validates or invalidates the hypothesis.

David Norton and Dan Ventura [NV06] also applied Hebbian training on the recurrent layer and monitored its effect using the separation indicator. The work in [BROD07B] for the IJCNN07 conference and which is also presented below thus offered an occasion to reproduce a similar separation experiment at little cost, as I already had all the necessary setup in place. As it happens, that extra experiment confirms their finding independently. The approach which I presented in [BROD06A] and which is extended here also gives an interpretation for [NV06] precisely in terms of order and of chaos considerations.

The question is now: How to apply learning on the recurrent layer in addition to

the linear classifier output? A first answer is given by Song *et al.* [SMA00] in the form of a version of the Hebbian learning rule adapted to spiking neurons. The next section presents this learning mechanism, which I then generalise into a whole family of learning rules using order and chaos considerations. Section 4.2.3 proposes to test this new family on one example that relies on multifractal spectrum estimation (see Sections 4.2.3 and 5.2). The idea is to test the generalisation that was proposed, by using a completely unrelated information from the one used by the Hebbian rule, so as to check whether learning still occurs: In that case, the principles that were extracted matter more than their implementation. Section 4.2.4 logically introduces the quantitative indicators that allow to check whether and how learning occurs. Sections 4.2.5 and 4.2.6 present the experiments and their results. Section 4.2.7 finally concludes and discusses the interest of this study in the larger framework of the dissertation.

4.2.2 Learning mechanisms for the recurrent layer

The Hebbian learning rule adaptation for spiking neurons proposed by Song *et al.* [SMA00] can be applied directly to the recurrent layer of the Liquid State Machine. It works by monitoring the activation patterns of linked neurons (see figure 8), and increasing this link weight when a timely pattern is observed.

More precisely, given a neuron N and an afferent neuron A, respectively in blue and red in Figure 8, [SMA00] proposes to monitor the time difference Δt between the reception of the spike emitted by A and the next one emitted by N. When a neuron A emits a spike that does not trigger N, it will thus have $\Delta t \neq 0$. This may happen either because N needs several more spikes to reach its threshold ($\Delta t > 0$), or because the spike emitted by A arrived during a refractory period ($\Delta t < 0$)[26].

26 [SMA00] uses the opposite convention for the sign of Δt.

[SMA00] then formulates reinforcement learning for links that are perfectly synchronised: The connection weight between A and N is updated by an amount depending on Δt, and maximal for Δt = 0. [SMA00] also notes that "causality is a key element" and that connection updates should favour the incoming spikes that could have a causal relationship with the outgoing one. Hence connection weights are also decreased for spikes that arrive too late (Δt <0), in particular during the refractory period. The formula for updating the A-N connection weight by a given amount F is given by:

$$F(\Delta t)=\begin{cases} A_p \exp(-\Delta t/\tau_p), \Delta t \geq 0 \\ -A_n \exp(\Delta t/\tau_n), \Delta t < 0 \end{cases}$$

with A_n, τ_n constants for the negative case and A_p, τ_p their positive equivalents.

[SMA00] interprets this as a form of "competitive learning" between the connection weights, in the sense that only one A neuron may be perfectly synchronised for a given emission by N. The rule applied in the experiments below use a proportional gain instead of an absolute addition, but the principle remains the same:

$$G(\Delta t)=\begin{cases} 1+R \exp(-\Delta t/\tau), \Delta t \geq 0 \\ 1/(1+R \exp(\Delta t/\tau)), \Delta t < 0 \end{cases}$$

Formula 1

[SMA00] also presents the Hebbian rule in terms that correspond to synchronisation and incompatible constraints. I have proposed to generalise these principles to a whole family of potential learning rules in [BROD06A]. The reasoning goes like this:

- In the [SMA00] formula the relevant parameter for synchronisation is assumed to be Δt. [SMA00] also notes that synchronisation results in faster communications, less latency for the transmission of the signals throughout the network. Yet, [SMA00] also insists that intrinsic competitive learning is important and should be achieved without external regulatory mechanism.

- When interpreted in the light of the Edge of Chaos hypothesis, the synchronisation is a factor of global order. Competition is a factor of chaos.

- In this example competition is structural: Incompatible constraints are implicitly defined by the nature of the neuron connections and processing. Structural constraints are also called "frustration" by Hugues Bersini and Pierre Sener [BS02] in a neural network context and generate what they call "frustrated chaos".

- Synchronisation on the other hand is the mean, the parameter on which we can act, so as to reach the critical line goal in the Edge of Chaos hypothesis. Synchronisation acts on the regime (or dynamical state) of the system local elements (the neuron links in the [SMA00] Hebbian learning case). Yet its effect is interpreted by global edge of chaos considerations. This is a form of local control.

I thus proposed in [BROD06A] the following approach to test whether these notions could be generalised:

a. Choose and identify a local measure that can render some dynamical property of a system element state. That measure should be expressed in "low-level" terms only.

b. Identify a significant target for this measure in terms of synchronisation. The way to reach that target defines the learning rule.

c. Check that there are incompatible constraints which prevent all the elements from reaching the target at the same time.

Then such a rule can be viewed as a slider between global order and chaos. The way to control this slider is provided by acting on the synchronisation / incompatible

constraints properties of the local measure. If the Edge of Chaos hypothesis is correct learning will occur, in the form of increased system performances.

A sanity check on the Song *et al.* [SMA00] rule presented above finds that:

a. Δt is indeed a low-level measure for quantifying the dynamical property "linked neurons are spiking together".

b. Synchronisation takes the form of the target goal $0 \leq \Delta t < \varepsilon$ with ε as small as possible, with the perfect case $\varepsilon=0$. Updating the connection weights as suggested in either of the above formula improves the synchronisation: A connection weight directly determines the energy accumulated by reception of a spike. If it is increased (resp. decreased) the received spike causes the accumulation of a larger (resp. lesser) amount of energy (for excitatory spikes). Hence increasing the weight for $\Delta t>0$ augments the probability that the corresponding spike will trigger the threshold next time, which corresponds to the ideal $\Delta t = 0$. The equivalent inverse argument holds for $\Delta t<0$.

c. The structural constraints of a refractory period, together with a triggering threshold higher than the amount of energy brought by a single spike, effectively prevent all Δt from reaching 0 simultaneously.

The hypothesis to test is now whether these principles hold, and not only in a descriptive way (for currently observed rules), but in a predictive way (for a new one). Of course many examples of such rules would be needed to assert some confidence in the hypothesis and possibly create a theory of learning in recurrent neural networks. But once again we find here an illustration of the main argument of this dissertation: We need to apply the Complex Systems ideas and test them concretely, in order to see what works and not, get the limits of these ideas, etc. If the aforementioned hypothesis for creating new learning rules holds, even for a

single rule, then a practical demonstration of local control for a complex system has been achieved in an efficient way, using concepts from complexity theory predictively rather than descriptively. As I explain in Section 4.2.7, this application has also achieved the side effect of refining the applicability or not of the Edge of Chaos hypothesis.

4.2.3 Proposal for a new learning rule using multifractal analysis

As a test for the learning rule creation methodology defined in the previous section, I decided to build a new learning rule that can be interpreted only by using chaos/order considerations. It is also based on a different information than the one used by the Hebbian rule proposed by Song *et al.* [SMA00], which excludes the possibility that the rules would be in fact equivalent. If the rule works, it will be hard to find another reason why it does so, apart from the order and chaos, synchronisation and incompatible constraints, Edge of Chaos interpretation.

The first step in the proposed methodology is to find an appropriate measurable indicator of a low-level dynamic property. Timing information across neuron connection links was taken by the Hebbian rule, so I had to choose something else. The most immediate indicator is the spiking frequency of a single neuron. This is the common notion (see the review by Wulfram Gerstner and Werner M. Kistler [GK02]) of coding the spike rate, but it is unsuitable in the present case: We need an estimator for the neuron dynamical regime, and averaging the spike timings as a frequency throws away all the sequence information present in the spiking time series.

The multifractal spectrum (See Section 5.2) of the inter-spike time series, on the

other hand, preserves some information about the way spikes are received. Previous experiments in Biology (loss of multifractality during heart failures as shown by Ivanov *et al.* [IAGHal99]), Finance (Benoit Mandelbrot's multifractal model of asset returns [MBF97]), Environmental Research (see the review by Tchiguirinskaia *et al.* [TLMWD00]), Image and signal processing (Evelyne Lutton and Jacques Lévy-Véhel [LV07] combine both multifractal and evolutionary computations), and Physics (where multifractal analysis was developed, see the presentation by Muzy *et al.* [MBA93]) amongst other domains show that multifractal estimation is well adapted to characterise the irregularity properties of time series.

In fact multifractal analysis gives some information that is complementary to the mean/variance based statistics and thus the spike rate. It characterizes how "frequent" different kinds of irregularities are present in the data (see Section 5.2 for details). Multifractal analysis also allows to distinguish between different kinds of noise: Brownian motion has a characteristic monofractal spectrum with H=h(q)=0.5 for all exponents q. Integrated pink noise gives H=1.0.

So, not only multifractal analysis is a good choice because it provides a quantitative aspect of some properties of the time series, but these properties were not captured and used by the Hebbian rule. Moreover, the multifractal spectrum computation is based only on a single neuron dynamics irrespectively of the afferent and efferent links, which further enhance the difference with Hebbian learning.

This rules out the possibility that the new rule would be equivalent to the Song *et al.* [SMA00] Hebbian one. Yet, multifractal spectrum estimation was at the time not readily available for the needs of this problem: it must be very efficient since it is applied for all neurons, and it must support adding new data incrementally when spikes are emitted. I therefore improved the wavelet-based method that Manimaran

et al. describe in [MPP05] and created an efficient algorithm that fits the present experiment needs. This new algorithm is described in Section 5.2 together with more information concerning what the multifractal spectrum entails.

Now that a measure has been chosen that can render some dynamical property of the neurons, the second step in the proposed methodology is to identify a synchronisation target, so as to define a learning rule. Multifractal is concerned with the scaling properties of the data fluctuations (see [MPP05] and Section 5.2). In this case, synchronisation means that neurons have the same spectrum. Updating the weights as before goes in the sense of more synchronisation: An increased probability of reaching the threshold on reception of a spike means more correlation between the neurons spike timings, hence more correlation between the inter-spike time series. More correlation between the series means closer spectra.

The third point is to check there are incompatible constraints that restrict how elements can reach the target goal. It would be possible in theory that all neurons exhibit the same spectrum, for example if their inter-spike timing sequences are not discernible from white noise. However, when we act on a connection weight to increase correlation between two neuron spectra, this also means we are decreasing the correlation for the other afferent links that are not modified. Applying the suggested rule thus carries its own source of frustration.

In mathematical terms let $h_N(q)$ be the spectrum for a neuron N and $h_A(q)$ the one for a neuron A afferent to N, when they can be reliably estimated (see Section 5.2). The synchronisation term I proposed above can be translated to a simple sum of squares between the spectra: $s=\sum_q (h_A(q)-h_N(q))^2$. Thus s = 0 becomes the synchronisation target. The idea of an exponential decrease for the weight update with respect to the synchronisation target works well for the Hebbian rule, so it is

reconducted.

The learning rule is thus, in terms of the gain G that is applied to a connection weight: Formula 2

- $G(s) = 1 + R \exp(-Cs)$ when both spectra estimations are reliable, with R and C constant parameters.
- $G(s) = 1 / (1 + R)$ when $h_N(q)$ is reliable but $h_A(q)$ is not.
- No change when $h_N(q)$ is itself unreliable.

The last two points handle the failure case when a spectrum cannot be reliably estimated. In this case, the neuron is considered unstable and cannot be synchronised. The connection weight coming from an afferent unstable neuron A is thus decreased, it is considered detrimental to the synchronisation of the efferent neuron N. When a neuron is itself unstable it cannot be synchronised with any of the afferent neurons and the connections are left unchanged.

If the Edge of Chaos hypothesis holds the exact mathematical formulation for the new rule shouldn't matter much. The mere use of synchronisation to cancel incompatible constraints should push the system toward the critical line and we should observe an increase in system performance, irrespectively of the rule details. At this point the fact that such considerations match the Hebbian rule only has a descriptive value. The methodology proposed and the creation of this particular new learning rule allow in addition to test the predictive power of the edge of chaos hypothesis: The multifractal rule is biologically irrelevant (unless proven otherwise!) and I chose the above formula for s and for updating the rule arbitrarily. I cannot think of another reason why the rule would work as intended, except by invoking the Edge of Chaos hypothesis as well as the corresponding synchronisation / frustration

interpretation.

4.2.4 Quantitative indicators for the effects of applying a learning rule

The task at this point is to monitor the effects of applying a learning rule to the recurrent layer. This task is independent from the learning rule itself. Once quantifiable indicators indicate the evolution of recurrent layer properties, they can can be used for: 1. Gaining insight as to why learning is effective. 2. Checking whether the multifractal rule works as intended, by comparison with the effects of the Hebbian rule.

"Quantifying the effect of learning on recurrent spiking neurons" is also the title of [BROD07B]. This section and the next ones expand on the work presented in this paper and in [BROD06A].

Separation

A first indicator that was previously mentioned has been introduced by Maass *et al.* in [MNM02]: The separation property. The idea is to check how well the recurrent layer can produce measurably different responses for different inputs. The separation property is intrinsically linked to order/chaos considerations: If the system is too static, many inputs are mapped together in a small number of states, their corresponding trajectories in the system state are merged and information is lost. This corresponds to low separation. On the other hand, when the network is too random, the response it gives to different inputs is not statistically significantly different, and it is not possible to distinguish between the inputs.

Several definitions of separation exist. The one that is used in [BROD07B] extends the definition by Eric Goodman and Dan Ventura [GV06], so as to replicate

the experiment realised by David Norton and Dan Ventura [NV06] independently: Monitoring the effect of the Hebbian learning with the separation indicator. Other definitions in related works are proposed at the end of this subsection.

The separation is defined as the ability to produce measurably different outputs for different inputs. But how are these differences asserted, especially by using only recurrent layer properties?

The direct input values are not available from within the recurrent layer, but the network state that results from the application of these inputs may be monitored instead. [NV06] and [GV06] use regular sampling, and note for each neuron whether it has spiked or not during the sample time interval. This leads to a boolean vector representation of the network state. Yet, the linear classifier of the output layer works on the activity of the neurons, their spiking rate. Since the target is to measure the ability to separate inputs, then it is more consistent to work in the data space that is effective for the linear recogniser. I therefore defined in [BROD07B] the state of the network as a vector of neuron activities, rather than a vector of boolean values.

The second part of the definition concerns the measure of the difference in the outputs. That part is unfortunately not defined in terms of the recurrent layer properties, but depends on a specific input/output mapping. For another problem it is possible that the same numerical input values are associated to different output classes. Knowing this limitation I still decided to apply separation in the [GV06] form, partly because I wanted to replicate independently [NV06], and also because the extension that I propose below would have been too computationally expensive.

For each output class j, the network state vectors S_i for the inputs i that are mapped to j are averaged into a class centre $C_j = \frac{1}{N}\sum_{i=1}^{N} S_i$. The average distance

between the class centres is then defined as the separation. Yet, while [GV06] introduces the concept this way, their formula also includes the null difference between a class centre and itself. [BROD07B] only counts the N(N-1)/2 differences between the N output classes, and defines separation as:

$$Sep = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \|C_i - C_j\| \qquad \text{Formula 3}$$

As noted above this definition depends on the specific input/output mapping task that is considered in the experiment. Ideally one would like to assert the effect of applying a learning rule on the recurrent layer generally, for all input/output tasks. The [GV06] definition (and its variation above) could thus be extended by integration over the whole space of possible input/output mappings. But as aforementioned this would make it impractical to apply for the current experiment. Yet, this is a possible direction for future work, especially considering it would be comparable to the global average approach definition by [NBL04].

Natschläger *et al.* [NBL04] introduces the Network Mediated (NM) separation, based on Hamming distance mean field considerations. Informally, NM-separation measures the average difference in network states that results from applying different inputs, but correct the term by subtracting the effect that would be caused by changing an input by any other. Hence it measures the ability of the network to discern particular inputs and it is not affected by the general tendency of the dynamical system trajectories to diverge. NM-separation defined as the average of this ability for all inputs is therefore not specific to a single input-output mapping. However, NM-separation was concretely applied in [NBL04] to a "reservoir computing" architecture using Random Boolean nodes instead of spiking neurons. The application of NM-separation to LSM is not immediate, and perhaps the extension I suggested above for the separation definition of [GV06] would be

simpler to implement: Integration of the above equation over the whole input/output mappings space could be approximated by large-scale Monte-Carlo sampling methods. A possible experiment would be to compare the result of this sampling with the NM-separation (in systems where it was applied), both in terms of behaviour with respect to the critical line and in terms of implementation complexity.

The "linear separation" capability of the network as defined by Robert Legenstein and Wolfgang Maass [LM07B] is related to but more general than the separation definition by Eric Goodman and Dan Ventura [GV06]. [LM07B] measures the state of the network as a N-dimensional vector of neuron activities instead of boolean values, as above, for M inputs. The mapping to specific outputs is avoided: The rank R of the N×M resulting matrix is called the linear separation property. R guarantees that this number of inputs may be separated linearly. This is interesting for a specific set of inputs, but an extension to the space of all possible inputs would still be necessary to assert the general effect of learning on the recurrent layer[27]. [LM07B] also performs an estimation of the generalisation capabilities of the network by measuring its response to small noise variations from the inputs. Both these measures could also be used as an extension to the current work.

While separation is a widely used and applicable notion in the LSM field, it remains so far not entirely satisfying for the needs of the current problem: Asserting the effect of the learning rule based solely on the basis of the recurrent layer properties, irrespectively of any input / output mapping task. Time has thus come for implementing another Complex Systems notion: the statistical complexity of the recurrent neurons layer.

27 [LM07B] does not apply learning to the recurrent layer so is not concerned with this problem.

Statistical complexity

Unlike separation, statistical complexity is an intrinsic property of a system that can be estimated based only on observations made on this system. When it is applied to a neuron behaviour, statistical complexity thus reflects only properties of that neuron behaviour. And similarly when it is applied to all the neurons in the recurrent layer.

A short presentation of statistical complexity was provided in Chapter 2, together with references. A longer introduction for the needs of this section is presented below. The need for an incremental algorithm for the estimation of statistical complexity is also detailed here, as was the need for an incremental estimator of multifractal spectra. The algorithm itself is presented in Section 5.1.

Statistical complexity measures the amount of information that is present in the past of a system, which is relevant to predicting its future. In the context of a neuron this means that some information present in the past spike timings[28] could be used to predict the future spike timings, while some other information won't help. For example, it may be that the neuron is locked into spiking at the highest frequency possible (it spikes just after the refractory period), for various reasons (it continuously receives many spikes, it participates in a short feedback loop, etc). In that case observing the past of the system always gives the same cyclic pattern, and predicting when a spike will be emitted again can be done with good accuracy.

Yet, the information present in the cyclic pattern is low: there is only one kind of behaviour and the neuron is locked into it. At the other extreme, if the neuron apparently spikes in a random way, then knowing when it has previously spiked is not very informative either: predicting the occurrence of a new spike with a fixed

28 In fact, the timings of all causally linked entities. This point is omitted for the purpose of this introduction, but it is detailed later on.

distribution (ex: Poisson) whatever the previous patterns may be the best we can do. In either of these two extreme cases the amount of information present in the past of the system which is relevant to predicting its future is low.

On the other hand, if the neuron spikes with sometimes cyclic activities, sometimes apparently chaotic ones, and sometimes keeps silent for an extended period of time, then it may be better modelled with a finite state machine with transition probabilities estimated from the observed data. In this case, the statistical complexity, the amount of information needed to encode the finite state machine, will be higher than in the previous two cases. If the number of states increase, for example because the neuron sometimes emits spikes at a second different fixed frequency in addition to the first one, then more information is needed to encode the state machine.

This example introduces two notions: 1. The notion that the statistical complexity is low for both ordered systems and disordered ones, while high for systems in between. It is thus a prime target for the current experiment. 2. The notion of state machines based on the system past, used to predict its future. This is actually how the statistical complexity is mathematically defined, see Cosma R. Shalizi's dissertation [SHAL01].

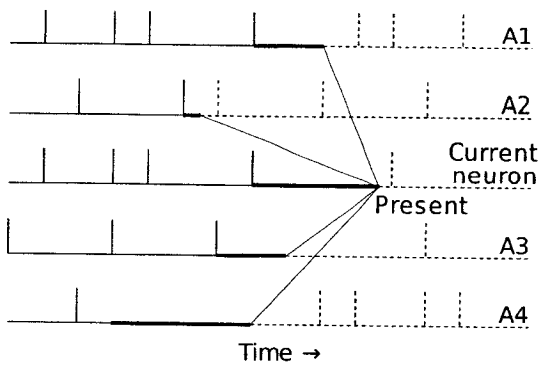
An algorithm for the practical computation of statistical complexity on discrete time series is found in [SHAL01], detailed by Cosma Rohilla Shalizi and Kristina Lisa Shalizi in [SS04], and extended by Shalizi *et al.* in [SHRKM05]. Yet, these algorithms compute the complexity of a static data set, assumed to be measured from a stationary time series, and they do not take in account the addition and removal of data. But the present case is different: The mere fact that a learning rule is applied makes the system non-stationary and this has to be taken into account. [SHAL01]

shows that in this case the statistical complexity is a function of time, which fits well the current needs: monitoring the evolution of the recurrent layer properties while learning.

A new algorithm for the computation of (slowly varying) time series is thus needed. The assumption here is that the stationary approximation holds for the neurons behaviour at least over some immediate past moving window, the time before the learning rule modifications become too important, so that we may still collect data and build consistent probability distributions. When a new observation (a new spike) is added, that window is shifted and expired data, if any, must be removed. The algorithm that handles both addition and removal of any observation is presented in Section 5.1.

Application of statistical complexity and thus of this algorithm requires that we define a "past" and a "future" for the neurons. The spike timings of each neuron are collected and kept for as long as necessary to create these past and future. Yet this is not enough. In the context of statistical complexity the past (resp. future) actually refers to a past (resp. future) light cone (see Section 5.1): the state of all entities that could be causally linked to the present state of the neuron. Indeed, for the prediction quantification to be completely inclusive of all information that could be possibly collected from the past about the future, then any observable bit counts.

But what does a light cone mean for a neuron? In the present case entities of interest include the afferent (resp. efferent) neurons. Spikes emitted by the these neurons may affect the current one only past the transmission delay (see Figure 8). Figure 9 shows these spikes in action (cited from [BROD07B]).



This represents the past light cone construction for a given neuron, the future light cones may be derived by symmetry. A1 to A4 are afferent neurons, the diagonal lines represent the transmission delays in time. Dashed spikes have no influence on the current neuron state, though some of them have occurred in the past. Thick lines represent the time to the last spike before the delay point. The case of A4 is presented in the main text: the maximum time has been reached.

Figure 9: Light-cone construction for spiking neurons.

Figure 9 shows how the light cones are defined and calls for some explanations. For each related entity, only the last (resp. next) spike is taken into account, starting from the delay point, as indicated on figure 9. The current neuron is also included in the definition, with a transmission delay of 0, so as to make a consistent framework for all involved entities. Distance-two entities are not taken into account, but theoretically they should be if considering an infinite light cone. The information that would be gained by including spikes that are received by the afferent neurons corresponds to considering the membrane potential (energy state) of each afferent neuron, instead of just the spikes it emits. Yet these spikes are emitted when the afferent neuron reaches its threshold, so already contain information about that neuron energy state. In practice it is assumed that the afferent neuron spike timing information is representative enough of its state, so we can discard level-2 relations.

The justification for retaining only the last spike received from each entity is that previous spikes from the same entity are assumed to have a negligible influence compared to the last one: the amount of energy that was gained by the current neuron when receiving the last spike is assumed to be much greater than what remains from the previous spike from that entity. This is not generally a valid assumption: when two spikes are received in a short time from the same entity, the

energy gained from the first has decayed but may still be enough for the second to trigger the current neuron threshold, for example. In practice however there is a refractory period that limits the maximum spike rate of an afferent entity A so the energy gained from the first spike has decayed already. In addition, spikes from other entities may come in that period and would have a much larger influence when the second spike comes from A.

Another assumption is that past a certain maximum time, spikes that are too old have a negligible influence. This is justified by the decaying property of the neuron membrane potential. Figure 9, A4, exhibits such a case.

Finally, the algorithm for the statistical complexity estimation can only process discrete data, and thus the time intervals are discretised: this implies the assumption that the exact timing is subject to noise and that small variations within the retained quantisation minimal period do not matter.

Taken together all these assumptions allow to keep the experiments within manageable computational power limits, without making unreasonable compromises.

Each neuron maintains its own "present" time, which lags on the simulation time by at most a fixed constant. The spike timings are consigned for each neuron. When a new spike is emitted, the light cones of the current, afferent and efferent neurons are affected. They are thus updated, which produces pairs of past/future observations for each of the involved neurons. These new data are fed incrementally to the complexity analysers. When the simulation time is advanced, too old spikes are discarded as aforementioned, which results in the removal of the old past/future pairs. Each neuron thus maintains the best possible up to date estimate for its statistical complexity.

Ideally, the light cones should be defined for the whole network, not just individual neurons. Some information may indeed be duplicated across neurons, so the sum of all the neurons complexities is not the same as the complexity for the whole network. However, defining light cones for the whole network hits practical applicability problems. The immediate implementation of the definition would be to define discretised states of the network, like the boolean vector representation previously mentioned in the separation property subsection, and then collect enough of these vectors to make past/future light cones. Unfortunately this straightforward implementation would hit a combinatorial limit: with only a few tens of neurons (say 64), a few vectors for the past/future cones (say 3 for each=6), and representing the neuron states with bits instead of a finer discretisation for their spike rate, there would be already $2^{64*6} = 2^{384}$ possible past/future mappings. Unless the network behaviour is very repetitive, maintaining statistical distributions about past/future mappings won't be easy. The other solution would be of course to find a better way to handle the problem than this straightforward application, but this is another research topic altogether.

So, for the current study, the sum of the neuron complexities is both within reach and also significant: it reflects (to a constant) the average complexity of a neuron. This measure may be monitored while applying a learning rule, to see how the neurons react on average. The next section describes the experimental setup for the simulation. Results for both the separation and the statistical complexity indicators are provided in Section 4.2.6.

4.2.5 Experiments

Experiments were conducted with two objectives: 1. Monitoring the effect of

applying a learning rule to the recurrent layer. This objective is reached by comparing the evolution of network properties when learning is applied, to their evolution when no learning is applied to the recurrent layer. 2. Asserting whether the Multifractal learning rule works as intended, with the reasoning in Section 4.2.3 explaining the implications for the Edge of Chaos hypothesis. This is achieved both by the first objective (if the rule works) and by comparing the effect of the Hebbian learning rule with the effect of the Multifractal learning rule for a finer analysis.

Three experiments are thus defined:

- The Basic version consists in training only the output layer, the linear recogniser of the LSM architecture (see figure 7). This is the usual operational mode of the LSM. It provides some capabilities for learning an input/output task without touching the recurrent layer.

- The Hebbian version consists in replicating exactly the same experiment (including the random seed) as the Basic version, but applying the Hebbian learning rule to the recurrent layer. Thus, both the recurrent layer and the output layer have their own training facilities.

- The Multifractal version, which consists of replicating exactly the same experiment again but replacing the Hebbian learning rule by the Multifractal one.

A first batch of experiments was conducted for [BROD06A], in which only the learning performance is monitored, because the separation and statistical complexity indicators were not available at the time. However the separation and the statistical complexity measures are computed for all three experiments in [BROD07B]. In the Basic version the recurrent layer is not modified, so both indicators are expected to be constant throughout the experiment. This basic

constant level provides both a sanity check for the algorithms, as well as a value to compare with in the other two experiments.

The data classification task

In these experiments I used a simple task: Classifying a data set (artificial or real) into two output categories. The goal is to measure the effect of applying a learning rule, not to prove that the LSM is able to perform complex computations (that was already the topic discussed by Maass *et al.* in [MNM02] and [MNM03] in particular).

A LSM operates on spike trains, not on continuous values. The data set is thus composed of (spike train, output class) mappings. However, except for specially adapted data, classification problems are usually not provided in the form of spike trains, but rather as vectors of real values (for example Lutz Prechelt's Proben1 [PREC94] data set which is used below). So, a way must first be found to encode a continuous value into a series of spike.

I designed a simple "population coding" scheme (by analogy with the definitions from Wulfram Gerstner and Werner M. Kistler [GK02]) for this purpose. In this scheme, each component of the input data vector is considered as a channel of information. A group of receptor neurons is connected to this data channel. All such groups form the input layer of the LSM.

Each receptor neuron produces a cyclic spike pattern with a fixed frequency linearly dependent on the data value. Hence, close values in data space are mapped to close variations in frequency. More exactly, given a data value v, the receptor neuron simply accumulates the fixed constant $\alpha v+\beta$ at each time increment δt, with α and β parameters specific to that neuron (independent from v). Therefore the

integral of that constant value is a linearly increasing amount of energy. When this energy reaches a threshold the neuron spikes, at a fixed frequency that depends on v. By fixing the threshold to an arbitrary value is is possible to set α and β so as to specify a minimum and a maximum frequency response. Biological significance is not needed in the current experiment, and this model works well in practice.

Each neuron from the same group has distinct frequency responses (different α and β). Hence, the group pattern is a combination of frequencies that is still linearly dependent on the data values, but more complex than a single cyclic pattern. The linear property is important in the case of noisy data recognition (used in the artificial data test set) so the network can generalise the training examples to nearby values.

Both artificial data and real data sets were used in the experiments, in the form of (vector of continuous data in [0..1], output class). The continuous values are then fed to the receptor neurons which convert them into spike trains. The parameters used for this transformation are given for each experiment.

The Basic experiment setup

The LSM architecture features the "fading memory" property, as presented in Section 4.2.1. This means that when exposing a data instance to the network, some sustained internal loops may subsist from the previous instance, unless the exposition period is long enough so this "memory" has "faded". Data instances are thus presented one by one (in random order) with a long enough exposition duration, so the internal neuron activities correspond to the current instance. The exposition time is also an important factor for the application of the multifactal and the Hebbian learning rules, as discussed in the next subsections.

Once all data in the training set have been presented, the output linear classifier is provided a set of (activity signal S_i, output class j). The S_i are the neuron activity vectors defined in Section 4.2.4, j is either -1 or +1 for a boolean classification experiment.

The task for the linear classifier is to map the S_i into j. This can be done easily with linear least square estimation techniques, but the application of the learning rules to the recurrent layer in the other experiments changes the optimum combination for the weights. A simple gradient descent algorithm is thus used in [BROD06A] for its online adaptability, despite the known instability risks and suboptimal learning performances: The experiments concentrate on the effect of applying learning rules to recurrent layer, so a simple output layer algorithm is sufficient for the present needs. [BROD07B] uses the direct least square estimation of the weights from the activities monitored during the epoch: This is correct for the base version since the connection weights between the recurrent neurons don't change. However a direct least square estimation runs the risk of suboptimal results when the learning rules are applied during the epoch, since the weights are estimated only at the end of the epoch with all collected activity signals. On the other hand the gradient descent instability is then avoided, and at least in the case of the base experiment optimal weights are produced for each epoch.

For the gradient-based experiment [BROD06A] the weighted combination result R is first computed: $R=\sum_{i=1}^{N} w_i S_i$. The error function that is optimised is a classical half sum of square: $E=\frac{1}{2}(R-j)^2$. The gradient descent is then a straightforward application: the weight w_i is updated by the amount $\Delta w_i = -r\frac{\partial E}{\partial w_i}$, with r the learning rate.

One step of gradient descent is performed after each exposition duration. An

epoch completes when all data in the training set have been presented in random order. The classification error is then monitored by computing R for each data, and comparing the sign of R with the real +1 or -1 class of the data. The ratio between the number of matches and the total number of data instances gives the classification error. Training stops after a predetermined number of epochs.

For the least-square estimation in [BROD07B] the network activity vector is collected for each data instance j together with the corresponding class target +1 or -1. This defines a matrix S_{ij} of activities and a vector C_j of targets. At the end of each epoch the Least-Square solution for the equation SW=C is computed to estimate the weights W_i (that is, the W_i minimise $||SW-C||^2$).

The Hebbian experiment setup

Applying the Hebbian learning rule given by Formula 1 directly after each spike results in practice in too fast fluctuations of the neuron connection weights: learning is too sensitive to transient effects and does not give good results. In practice for this study statistics about Δt are collected over the whole exposition duration for each link. The average Δt are then fed to Formula 1. This is a plausible approach: neurons now synchronise with other neurons that give consistent information on average, so learning occurs over a larger time scale than that of the immediate signal transmissions. Learning is applied at the end of the exposition period, just after the global linear recogniser is updated as in the basic version.

Another problem was observed: some weights become irrelevantly large compared to the energy threshold for the spike emission, and in some other cases the total sum of the weights is either arbitrary large or low. Though one of the motivations explained by Song *et al.* [SMA00] is to avoid the use externally imposed

constraints – competitive learning is presented as an attempt to solve this problem – the repeated application of either Formula 1 or the original formula in [SMA00] results nonetheless in an unbounded increase of some connections, at least in the artificial LSM context. Imposing a bound on the influence of one spike in terms of energy is not only plausible, but also a necessity in our case. Similarly the sum of the afferent connection weights is regulated: this does not prevent a weight from dominating all others, but it was found to improve the performance in this study experiments. These bounds are of course also reconducted in the multifractal version for consistency.

The multifractal experiment setup

Each neuron maintains a multifractal spectrum analyser. When the neuron spikes the time elapsed since the last spike is fed to the the analyser. The problem is that the effective result of this new data is not immediately available. Indeed, the multifractal spectrum estimation relies on a time-frequency decomposition: it is subject to the fundamental delay necessary to capture the low frequency components of the signal. More generally it is not possible to capture the frequency decomposition of the signal instantaneously. The more precise the multifractal estimation, and the longer the delay before getting the result.

Applying the same setup as for the Hebbian and the basic versions hopefully solves this problem. Learning occurs only at the end of the exposition period, so as long as this data exposition duration is long enough, the analyser provides results matching the current data. Additionally, the obsolete internal data in the analyser corresponding to the previous instance have been discarded and replaced by new data corresponding to the current one.

Monitoring the Separation

The separation as defined in Formula 3 simplifies into a simple Euclidean norm between the class centre vectors when there are only two output classes.

At the end of each data exposition period the neuron activities are already computed for training the Basic classifier, so the network state vector S is readily available. This activity vector is then added to the one that is maintained for the output class of the currently exposed data.

At the end of an epoch each class vector is averaged by the number of instances mapped to that class, and the separation is computed as in formula 3.

Monitoring the Statistical complexity

As mentioned in Section 4.2.4 the incremental algorithm needs to handle both the addition and the removal of data. Addition is performed as previously described: each time a neuron spikes, for that neuron and all its afferent and efferent connections. But removal is rather arbitrary and is only necessary to respect the stationary approximation assumption. On the one hand waiting too long risks to give meaningless results, but on the other if data is removed too soon the algorithm may not have to time to converge. Section 5.1 shows that convergence is very fast in the case of cellular automata, but these systems benefit from two factors: 1. The light cones are simpler in the case of cellular automata and there are less possible time/future combinations than in the present case. 2. Data from multiple cells are collected and fed to a unique analyser since the cells are stateless and all execute the same rule. In the present case each neuron is a single isolated system, with its own properties like for example the connection topology or the link to an input receptor. Therefore it is not immediate whether the algorithm will converge in a manageable time or not.

Preliminary tests were conducted on the basic experiment: Since the recurrent layer is not modified, the data may be added (and not removed) for as long as necessary to observe a stabilisation of the sum of all neuron complexity values. Some of the experiments stabilised before an epoch completed, but many others required more time. Increasing the receptor neuron frequencies provided the solution: More spikes are generated for the same amount of presented data. Thanks to this trick the analyser now converges below one epoch. I decided to remove old data as soon as possible: any (past, future) light cone pair observation that is older than an epoch is discarded by the concerned analysers when a neuron spikes.

Data are thus added and removed incrementally, on each spike, thanks to the algorithm in Section 5.1. The total sum of the neuron statistical complexities is then computed at the end of each epoch, together with the separation.

4.2.6 Results

Effects of the learning rules on the training and test errors

Two sets of experiments were first conducted for [BROD06A]. The first set monitors the error E of the output linear recogniser on artificial data. The second concerns applying the same methodology on real data. In each case the three versions are run with the same random seed. The basic version corresponds to the original LSM setup where the recurrent layer is not modified: It provides the basis for judging the effect of the other two experiments. The Hebbian and the multifractal rules are then respectively applied in the other two runs with the same random seed.

20 training and 20 testing instances are produced for the artificial data. Each instance is a real-valued vector of 10 data channels. 5 receptor neurons form the

reception group for each channel. The receptors α and β are calculated so as to set the minimum and maximum spiking frequencies to respectively 1 Hz and 50 Hz. Group diversity is then generated by multiplying the α and β of each receptor by a factor drawn from a normal distribution centred on 1 with a variance of 0.1. The training and test sets are built by uniformly drawing a random number in [0..1] as the base data for a channel for each output class, and adding a normally distributed random noise with mean zero and variance 4.10^{-3} for each instance. Instances are split evenly between output classes.

Testing is performed on 10 epochs so as to get an average between different instance presentation orders. 50 training epochs are monitored. The learning rate for the output layer linear recogniser gradient descent is set to 0.01. Too large a value was observed to make the gradient descent unstable, but 0.01 worked in all the runs. The R parameters for the Hebbian and Multifractal rules, in the Formula respectively 1 and 2, were set to 0.1. The C parameter in formula 2 was set to 0.1 as well, according to exploratory preliminary experiments. The parameter τ in Formula 1 was set to 20 ms according to the value proposed by Song *et al.* in [SMA00]. The exposition duration of the data is set to 1 second of simulated time. The Liquid State Machine was built as a cube of 6×6×6 neurons with the same parameters as the reference given by Maass *et al.* [MNM03].

The test error is null in many of the runs on the artificial data, which means the LSM could learn the task and generalise well. As the objective is to monitor the influence of applying a learning rule on the recurrent layer, I then focused on monitoring the training phase. Figure 10 shows the evolution of the training error E for all three experiments, in two particular runs chosen for exhibiting a marked difference between the Hebbian and the multifractal rules.

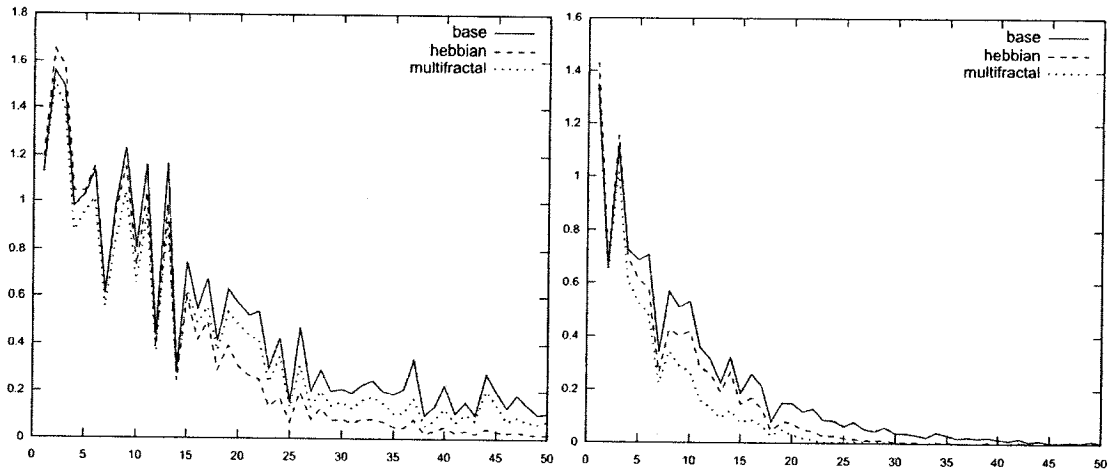
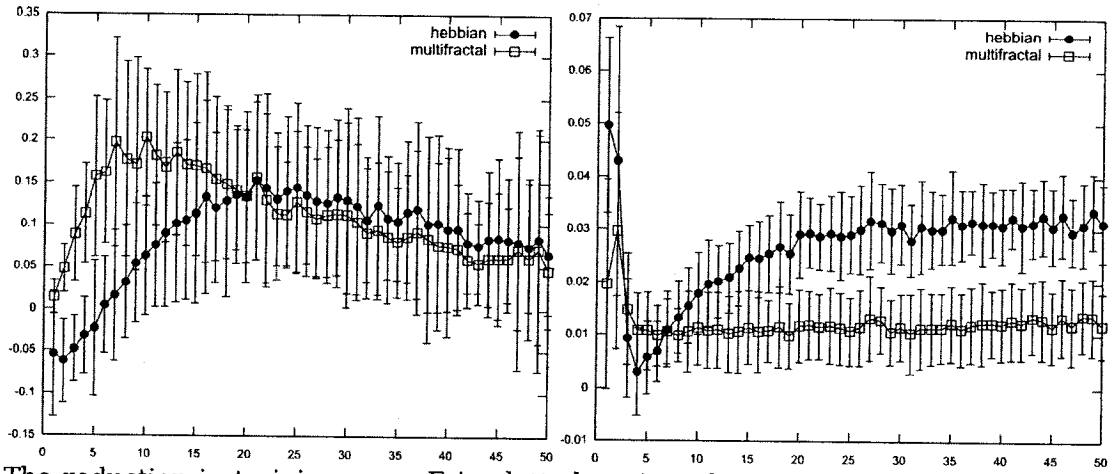


Figure 10: Training error vs number of epochs, random seeds 5 and 16.

Both the Hebbian and the multifractal rules have boosted the learning process compared to the base experiment. In some of the runs the multifractal rule works best (Fig. 10 right), while in some others the Hebbian rules does (Fig. 10 left). Most of the runs do not exhibit as much difference. In some of the runs one rule works better than the other for some epochs, then the reverse happens for the next epochs, then reversion again, till the end of the training phase. A better insight on the effect of these rules is provided by the separation and the statistical complexity indicators (see the next section), but these were not available at the time of [BROD06A].

In order to highlight the differences between the learning rules a comparison with the base experiment was performed: Since the effect of applying the rules on the recurrent layer is to boost the global learning process, then statistics may be collected about this gain on multiple runs. The results are provided in Figure 11 in the form of the average error reduction (with standard deviation bars) for each rule, on both the artificial data experiment (left) and the real data experiment (right) that is described below.



The reduction in training error E is plotted against the number of epochs, one the artificial data (left), and on the real data (right) described below in the main text.
Figure 11: Learning performance boost on artificial and real data

On the artificial data the multifractal rule tends to produce a better improvement during the first training epochs, while the Hebbian rule gets a slight advantage at the end. One hypothesis would be that the multifractal rule better captures the dynamics so it is more effective at the beginning of the training, while the Hebbian rule that monitors spike coincidence acts on a more volatile criterion and it needs more time to be effective. On the other hand the connection reinforcement is more significant for the Hebbian rule than the multifractal (in terms of their synchronisation criteria) so the end of the result would be more "stable" for the Hebbian rule. The volatility of the spike coincidence would also explain why the Hebbian rule produce a negative gain (loss of performance) on average at the beginning, possibly its effect perturb the output classifier gradient descent too much. These are just possible explanations for the results, but the corresponding hypothesis cannot be validated with only these experiments. The statistical complexity especially provides a different interpretation. Whatever the reason for the initial difference in the rules in the artificial case, the most important point here is that the multifractal rule works. As aforementioned this demonstrates a concrete use of the Edge of Chaos hypothesis as a predictive tool. The discussion

in the next section will expand on this point.

Another experiment was conducted in [BROD06A] on real data provided by Lutz Prechelt's Proben1 data set [PREC94]. The cancer1 task[29] consists in classifying cells as cancerous or not based on their visual aspect. Observations are normalised and provided as 9 real values between 0 and 1, and there are two output classes: This matches the current scenario. The same parameters as above for the receptor neurons, LSM, and learning rules were reused for another batch of runs, but using the real data instead of the artificial one.

The cancer1 problem initially contains 525 training instances and 174 test instances. This is respectively about 26 and 9 times more than for the artificial case, which posed computational resources issues. The data set was thus reduced to 100 training and 100 testing instances. This unfortunately changes the nature of the problem so the results of this study cannot be compared precisely with the results of the Proben1 experiments: testing 100 prediction errors from 100 learned instances is not the same as testing 174 prediction errors with 525 training instances.

30 batches of runs provided an average and variance for the evolution of the training error as well as the testing performance, for all three experiments. As before the training error reduction was monitored during all 50 epochs. The result is provided by the right plot of figure 11. Both learning algorithms have converged quickly and do not provide further improvement on the learning error. Despite the fact that the multifractal rule converges to a lower gain in the training error, the generalisation capabilities are not affected: the prediction (classification) error is similar for both rules, as shown on table 1:

29 This data set is the result of the work from O. L. Mangasarian and W. H. Wolberg. See also "Cancer diagnosis via linear programming", SIAM News, Volume 23, Number 5, September 1990, pp 1 & 18. This reference and more are included in [PREC94] and also provided together with the 'cancer' data set files.

	Base experiment	Hebbian rule	Multifractal rule
Average classification error	6.20 %	5.70 %	5.73 %
Standard deviation	1.71 %	1.34 %	1.58 %

Table 1: Performance comparison for predicting real data

On some runs applying the learning rules did not produce an increase in global performance. However this may partially be attributed to the inefficiency of the gradient descent for the output layer. Nevertheless, on average over the 30 runs, both learning rules boosted the performance of the LSM with similar results.

In [BROD07B] I did not use gradient descent but rather the direct least-square estimation of the output layer weights, as explained in Section 4.2.5. The results of [BROD07B] which are reproduced below are thus not subject to the gradient descent inherent instability. However, since learning rules are applied throughout the epochs and the activities collected at the same time, the computation of the weights at the end of the epoch uses activity signals from the beginning of the epoch that may not correspond to the current signals that would be produced at the end of the epoch (except for the base experiment). Since the learning rates for the multifractal and Hebbian rules are low, these changes are supposed small enough so the least-square estimate still produces results that are not too far from the optimum. Figure 12 shows a great variability in the residual training error, as compared to the general shape of Figure 10. Note also that the parameters used for this experiment differ from the parameters used for Figure 10, especially the size of the network and the usage of real data, for reasons explained in the next section: only the general shape is comparable, not the numerical values.

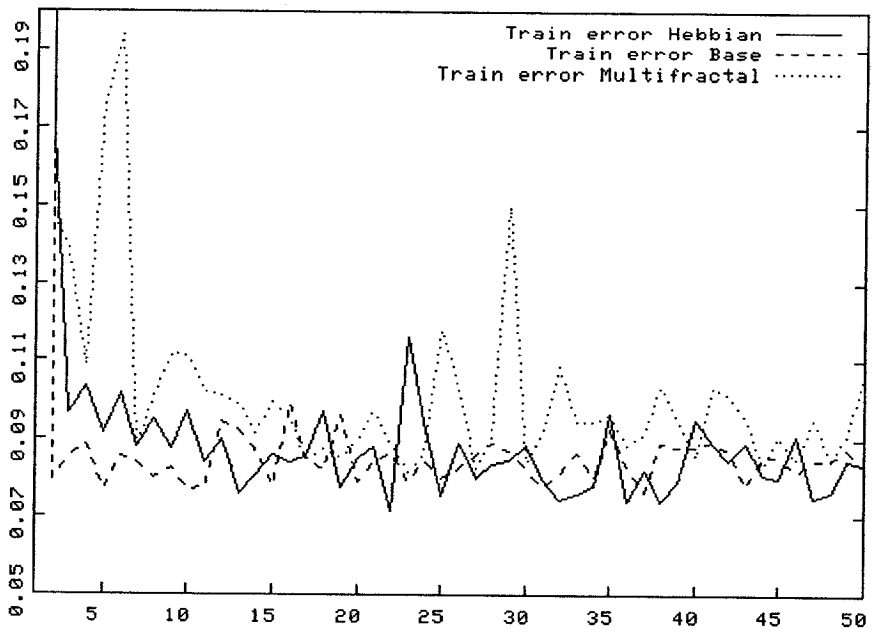


Figure 12: Training error vs number of epochs using least-square weights estimates

Nevertheless the weights are much closer to the optimum now compared to the slow convergence of the gradient descent, as shown by the immediate drop at epoch 1 right to the residual noise: Figure 10 takes all 50 epochs before convergence with the artificial data, and Figure 11 shows about 20 epochs necessary for convergence of the gradient descent on the real data. In fact Figure 13 shows that the convergence of the network state actually also takes about 20 epochs in the present case for the multifractal learning rule and 30 epochs for the Hebbian one, but this is not visible just with the training error: Other indicators are necessary, which is the topic of the next subsection.

The testing errors for the new configuration of [BROD07B] are averaged over 15 testing epochs for 30 batches of runs, and they are provided on table 2:

	Base experiment	Hebbian rule	Multifractal rule
Average classification error	4.9 %	5.3 %	6.7 %
Standard deviation	3.7 %	3.8 %	5.3 %

Table 2: Performance comparison for predicting real data on a reduced setup

This configuration uses a reduced-sized network compared to the one in [BROD06A]. The real data set is also different, as explained in the next section. As an indication for the significance of the numerical values in this table, a linear classifier (weighted combinations of the input data) produces a classification error of 23.3% on the testing data set. This is directly comparable with the results in table 2, as the performance that the output layer would reach alone without the recurrent layer. Hence, even the reduced-size network still brought non-linear power to the output layer classifier.

The results for both rules and especially the multifractal one are probably simply explained by overfitting, as is supported by the separation monitoring in Figure 13 in the next subsection. The larger network used in the experiment for table 1 did not suffer from overfitting. One possible explanation is that small networks are more sensitive since there are less states to synchronise than in large networks. More experiments would be necessary to assert this hypothesis validity and extent. In any case the testing errors are only a crude indicator for monitoring the effects of the learning rules, and the residual training errors are much too erratic, so other indicators become necessary. This is the topic of the next subsection.

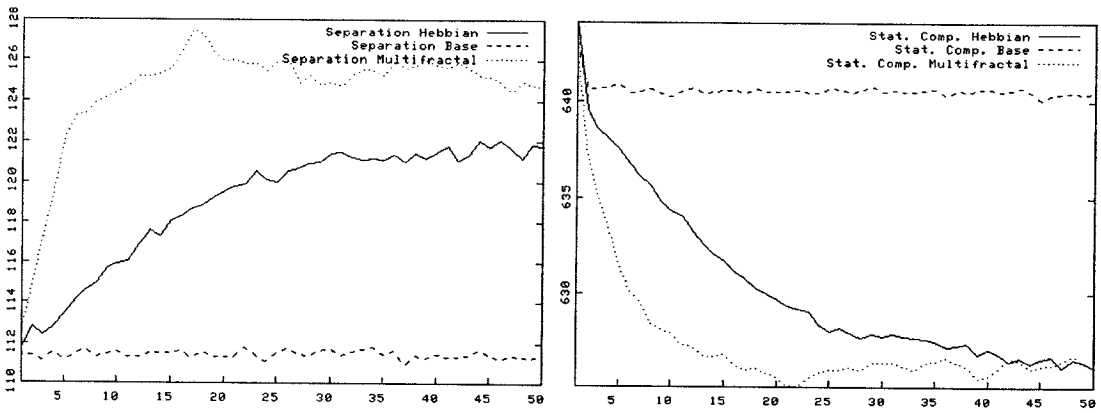
Quantification of the effects of applying the learning rules

The separation and statistical complexity were applied as described in 4.2.5 for the set of experiments reported in [BROD07B] and mentioned at the end of the previous subsection. Unfortunately applying these indicators is costly, so for the same simulated time, the experiments now take much more real time to complete. The simplest solution was to reduce the size of the LSM to a 4×4×4 cube. This is detrimental to the "reservoir" approach in the sense that less basic transforms on the inputs are available for the output linear classifier, and so was reported by

Maass *et al.* [MNM03] as decreasing the LSM performance. However this section is not concerned at all with the linear output classifier: both separation and statistical complexity are computed using state information present in the recurrent layer, irrespectively of the global classifier performance in the output layer. Moreover, the previous subsection has reported testing errors much below the raw linear classification possibilities (4.9% instead of 23.3% for the base experiment), so the "reservoir" is still large enough to produce basic transformations on the inputs in the present case. Other changes in parameters are the aforementioned increase in the receptor neurons spiking frequency to a minimum of 30 Hz and a maximum of 90 Hz, using 3 receptors per channel to match the new LSM size, and a 0.07 learning rate for the Hebbian and multifractal rules. The gradient descent was replaced by a direct least square estimation, but the effects of this change are only visible for the output layer and were discussed in the previous subsection.

The data set is the real data provided by the cancer1 task of Lutz Prechelt's Proben1 benchmark [PREC94], but reduced to 40 training and 30 testing instances without duplicates.

Average results over a batch of 30 runs for the separation (left) and statistical complexity (right) evolution while learning are shown in Figure 13. The Separation is given as a rate of spikes per second. The Statistical Complexity is provided as the number of bits necessary to encode the information about the causal state machines for all neurons.



Units are explained in the main text: A spike frequency for the separation, and a number of bits for the statistical complexity. The indicators are plotted vs. the number of training epochs.

Figure 13: Evolution of the Separation and the Statistical Complexity indicators when a learning rule is applied to the recurrent layer of a Liquid State Machine.

Separation is the ability of the network to distinguish between the output classes. As a result of applying the learning rules, the two output classes have been separated by 122 and 125 Hz. Given that the base version gives a separation of 111 Hz without modification to the recurrent layer, the capabilities of the output linear classifier to distinguish the inputs belonging to the two classes has increased by 9.9% and 12.6% respectively for the Hebbian and multifractal rules. It is interesting to note that applying the multifractal rule leads to a better separation than the Hebbian one, so a smaller training error since separation is computed on the training set. On the other hand both rules are overfitting as shown by Table 2 in the previous section, and the greater separation matches the greater overfitting.

David Norton and Dan Ventura [NV06] conducted some experiments where Hebbian learning was shown to increase separation for non-random inputs. The real data set in this study has well defined categories, not random ones. The result of the present experiments not only confirms a finding from [NV06] – separation increases as a result of applying the Hebbian rule – but also provides a framework for explaining why it does so: Incompatible constraints in the neurons spike timings

have been smoothed out by synchronisation, so the information contained in the input spikes is less destroyed by the chaotic response to the incompatible constraints, hence the output layer can better extract this information back. Note that this interpretation relies on the remarks of Section 4.2.2 and the generalisation method for creating new rules, which is not a full theory but an hypothesis that was only confirmed so far by the fact the multifractal rule works as intended.

The statistical complexity indicator provides another insight into the system. The amount of information necessary to optimally predict the neurons behaviour has decreased by about 14 bits for the whole network for both rules. What this means is that the neurons have become more predictable as the effect of applying the learning rules: their reactions to the input spike patterns is more determined than before applying the learning rule. That learning results in more specialised neural circuitry may perhaps sound logical to a biologist. In the present context the observed reduction in neuron complexities is also consistent with the fact that rules bring synchronisation between the neuron states. In any case this experiment brings a quantitative measure for the behaviour simplification of the neurons as a result of learning.

In the context of the edge of chaos considerations of Section 4.2.3 this means that the system was shifted away from maximum complexity, which is supposed to also correspond to a critical line. Hence the reduction of complexity contradicts the general Edge of Chaos claim that the system would benefit in this state from longer transients and more complexity, associated usually to better processing capabilities (i.e. separation in our case), leading to better performances, etc. Since both measures (separation and complexity) were obtained from the same system, and since one increases while the other decreases, then that system cannot be pushed

toward a region where both would be maximum. This is unexpected, especially since the multifractal rule works nicely here again, and it was precisely designed to validate the edge of chaos hypothesis if it worked. The next section discusses these results and provides a possible explanation.

4.2.7 Discussion

The previous experiment shows that at least the separation and the statistical complexity indicators cannot be interpreted both at the same time as a system drift toward a single critical region. Yet, the multifractal rule was expressly designed so it can hardly be interpreted apart from order and chaos considerations, and it not only gives similar performance results as the Hebbian rule, but it also reacts in the same way with respect to both separation and statistical complexity.

What are the consequences for the Edge of Chaos hypothesis? If it is interpreted as a concept that there is some critical region where a system gains all kinds of properties, then that region has received a serious challenge concerning its extent. It could still be that the LSM system started in the present experiments from a saddle-point in a multidimensional space where the effect of the learning rules would lead to an increase in one indicator and a decrease in another, and that the Edge of Chaos region is further away where both properties would be maximal. Yet, I do not think this is the correct explanation.

More probably there are subsets of system parameter space which correspond to each individual indicator extrema. One such subset may be defined as the one for which the system reaches a good generalisation performance (like a low prediction error). Another one is defined by an indicator (like the separation) where the system has better processing capabilities on the training set. Another subset corresponds to

long transient effects (for example Stefan Bornholdt and Torsten Röhl [BR03] introduce the notion of average attractor lengths in a boolean network setup). Another subset corresponds to maximum difficulty of predicting what will happen next based on past observation (the statistical complexity). And so on, for other properties measured by other indicators.

Each of these subsets may also define a boundary between (possibly disconnected) regions with lower indicator values, that can be interpreted by order and chaos considerations: the separation is low for both highly static and highly random systems, the statistical complexity as well, etc., and each indicator peaks within a maximum-valued set between low-valued regions. It may also be that some of these sets overlap, that there would be "islands" where some of the properties simultaneously achieve a high value. But, and this is the trick, there would be no general "critical line" between a general "order" and a general "chaos". Ironically, as often with emergence-related issues, each individual element (the indicators in the present case) may exhibit properties (bumps between low-valued areas) that are defined for each element, but that make no sense as a whole.

What about the multifractal rule then? The general methodology proposed in Section 4.2.2 explicitly starts by "a. Choose and identify a local measure that can render some dynamical property of a system element state". With respect to that measure only, the Edge of Chaos concept may very well still hold. This would explain that working with synchronisation and incompatible constraints considerations is effective, and why the multifractal rule works at all. In this case, the system is pushed toward a "critical" region that depends on the chosen learning rule, different for each rule. There is no guarantee however than this region is in any way related to either the one for maximum separation or the one for maximum statistical

complexity. In the case of both the Hebbian and the multifractal rules it happens that the system was shifted toward a region that also corresponds to better learning performance on the training set (separation), and even more so for the multifractal rule. But on the other hand for that rule there is a case where generalisation performance is not optimal (see table 2), so the synchronisation/incompatible constraints "critical region" corresponding to the multifractal rule differs from the network generalisation optimal region.

The general methodology for creating learning rule should thus be manipulated with caution. It was designed by generalising the principles found in the Hebbian rule, and checked for consistency with that rule. It was shown to also work on another example, in the sense that it produced a rule that generally behaves like the Hebbian one and succeeds in improving the network performance. Yet, it is possible that some indicators that can be interpreted in terms of synchronisation and incompatible constraints will not lead to an "edge" that correspond to a maximum in terms of processing capabilities (either training or testing performances).

As a conclusion two future research lines would extend this study well:

- Creating many other rules and testing which work or not. This would help refine the criteria of the rule design methodology, and possibly help understanding what are the common features to all of these rules that make them work or not. This is a neural network based approach, that would perhaps lead to understanding why and how learning works on recurrent spiking neurons.

- Research more criteria that quantify a system in terms of order and chaos and related concepts. The general notion of the "edge of chaos" could then be refined into "islands of critical regions", where "critical" means that several

desirable properties peak together. Identifying what concepts usually peak together would be of great interest for predicting a system behaviour.

Once more the approach and the experiments presented in this section are an illustration of the main theme of this dissertation: trying to apply ideas related to complex systems in practice, with quantifiable indicators of the system state, and using these ideas in a predictive way, is a necessity to make progress in the understanding of complex systems. The tools that are created along the way are also useful contributions in their own, as I now present in Chapter 5.

Chapter 5: Algorithms

Once you compress out all the redundancy from anything meaningful, the result necessarily looks meaningless, even though it isn't, not at all, it's just jam-packed with meaning! *[CHAI05]*

Gregory Chaitin

This chapter presents the algorithmic tools and methods that were created in the course of the main projects described in Chapter 4. These algorithms were created because the available state-of-art tools did not fill the needs of the projects. They are detailed in this chapter because they represent valuable contributions in their own. Care was especially taken to make these algorithms self-contained: They are readily available as Free software, without attached strings, no dependence on external tools or environments. They are also very focused: one problem, one solution. My intent is to make these algorithms not only useful contributions, but readily applicable tool boxes in other Complex Systems contexts.

Section 5.1 presents an incremental estimator for the statistical complexity of a time series. It can be applied with live data, so new data can be added and obsolete data can be discarded. This is especially useful in the case of non-stationary systems where the properties that are quantified change over time. This algorithm is generic and widely applicable, and brings new capabilities compared to the state of art by Shalizi *et al.* [SHRKM05]. This algorithm was successfully exploited in order to quantify the effect of applying learning rules on a recurrent spiking neural network (see Section 4.2). A short presentation of this algorithm is also included as part of

[BROD07B] for the IJCNN 2007 conference.

The next algorithm in Section 5.2 computes the multifractal spectrum of a time series with a new implementation: An incremental version of, and an extension to, the discrete wavelet analysis technique by Manimaran *et al.* [MPP05]. This is the algorithm that is used in Section 4.2.3 as an illustration for creating a new learning rule for a spiking neural network. More details about the algorithm and its advantage over competing techniques are presented in Section 5.2 and in [BROD05B].

The third algorithm is a spin-off from the project in 4.1, with added value but no link to the main emergence issue. It is included because it forms an advance over the state of art and because it can be useful in particular for the study of multi-agent systems: An algorithm for performing nearest neighbours queries in a dynamically changing environment, where the agents are mobile. The main motivation behind the spin-off was to increase the performance of multi-agent simulations, and this algorithm would have been used in the environment introduced in Section 4.1 if it was available at the time. The article describing the neighbourhood-finding algorithm [BROD06B] has been accepted conditionally to some changes in the Journal of Graphic Tools.

Finally, Section 5.4 concludes this chapter by summarising the main features of each project that is presented, and give directions for possible future works.

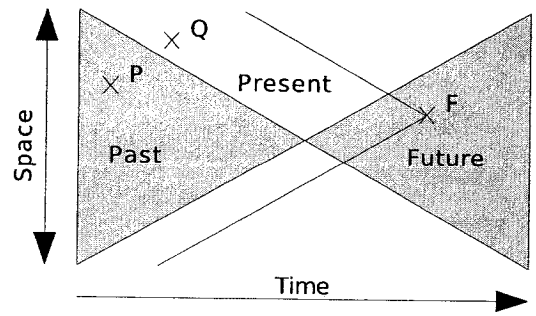
5.1 Incremental statistical complexity computation

5.1.1 Introduction

Statistical Complexity was introduced in Chapter 4 using an example in a neural network context. This section provides more details and introduces the notions that are necessary for understanding the algorithm presented in the next section.

Let's consider the following general prediction problem: Given the full knowledge of all the past observed states of a given system and all of its causally linked entities, to what extent can we predict that system's future observations? In other words, using a physics terminology: Given the observed past light-cone of a system, how well can we predict the future light-cone? In this context, a past (resp. future) light cone would theoretically extend to minus (resp. plus) infinity in time: it would include the state of the present entity in a system and all other entities that could have a causal influence on (resp. be causally influenced by) the present state. An example is provided in Section 5.1.3 for cellular automata.

Figure 14 (from [BROD07B]) depicts the relations between the past and future light cones and shows that the question is not as immediate as it seems: Some points in the future light cone (ex: F) are causally dependent on points (ex: Q) that cannot be known from within the system since they are not in the past light-cone (all points P).



The past light-cone of a system comprises all observations P that could possibly be made on that system and all the entities that could have an influence on it. Similarly the future cone refers to all entities that could be influenced by the system present state. The problem is that some states in the future cone, like F, depend on information that is outside the past light-cone, like what happens at Q.

Figure 14: Relations between the past and future light cones of a system

Information theory provides a measure for quantifying how difficult is the problem: the amount of information that is present in the points P and which is necessary to predict the points F. Or equivalently, how badly the information that is contained in the points Q is missing for predicting the points F.

More formally, a given past cone c may lead to possibly many future cones f. The probability distribution p(f|c) then captures how well we can predict possible futures when we observe a given past.

Conversely when two pasts c_1 and c_2 lead to the same probability distribution over futures it is not possible to distinguish between them, whatever the observations that are done on the system from that point onward. For the present and future of the system, for all purposes, these pasts are thus equivalent. The equivalence classes $\varepsilon(c) = \{x: p(f|x)=p(f|c)\}$ are called the Causal States of the system (see Cosma R. Shalizi's dissertation [SHAL01]).

Knowing in which causal state class the system is at a given time allows to predict the future with maximal accuracy: Intuitively, subdividing the classes would be useless since this brings no additional information on the future distribution. The causal states are also the minimal information that is needed to predict the future: all possible system states are taken into account by the light-cones so any other predictor would duplicate information already present in the causal states. Proofs and a precise mathematical definition for statistical complexity and these properties are described in [SHAL01].

To sum up, $\varepsilon(c)$ and its associated probability distribution give the best prediction for the future of the system, given the observed past c, with the minimal possible amount of information. The original problem presented in Figure 14 can now be reformulated as "how much information is contained in c (the points P),

which is relevant for computing $\varepsilon(c)$ (the best we can do about the points F)". This is mathematically the mutual information $I[c;\varepsilon(c)]$. As shown by Shalizi *et al.* [SHRKM05] this quantity is simply the entropy $H[\varepsilon] = -\Sigma_{\varepsilon} p(\varepsilon) \log_2(p(\varepsilon))$ in the case of discrete systems: the amount of information needed to encode the causal states. This $H[\varepsilon]$ is the statistical complexity of the system.

The algorithm that is presented in [SHRKM05] and earlier in [SHAL03] also allows to compute the self-information $I[\varepsilon] = -\log_2(p(\varepsilon))$ of each single state ε. For a given system state defined by the cone c, $I[\varepsilon(c)]$ defines the local complexity field at that point c: the number of bits that are necessary to represent the system causal state ε. These values may then optionally be scaled between 0 and 255 to produce grey-scale images, as the ones presented in Section 5.1.3 below.

The causal states and the statistical complexity of the system can be estimated from data and form a purely objective measure on that system. When the system is constant there is only one state, hence a null complexity. Similarly when the system apparently "ignores" the pasts and produces the same distribution of futures whatever the observed past (like noise), then there is also only one such distribution (for example uniform) and again one state, so a null complexity. The statistical complexity reaches a maximum for systems that behave in intricate ways but are not totally disordered. Hence it was a good choice for testing the Edge of Chaos hypothesis in Chapter 4.

Yet, while the algorithm in [SHRKM05] is adapted for the estimation of statistical complexity for fixed data sets, it handles neither the addition of new data as they become available, nor the removal of expired data in cases where the system is not stationary. These were precisely the two properties that were necessary for the application of the algorithm to the problem discussed in Chapter 4.

The next section presents the algorithm from [SHRKM05] and explains what changes were made to it. The implementation details are also presented there, with pseudo-code. Section 5.1.3 tests the algorithm on concrete cellular automata cases. Section 5.1.4 concludes the discussion on this algorithm.

5.1.2 Incremental statistical complexity estimation

The algorithm which is presented by Shalizi *et al.* in [SHRKM05] assumes the availability of all observed system pasts beforehand, and does not support reclustering when new observations become available. These observations may either refine the estimated future distributions for each past, or bring in new previously unobserved pasts. Both problems are addressed in this section.

The algorithm presented in [SHRKM05] and a more detailed version of it in [SHAL03] assumes the series is conditionally stationary: We can then combine past/future observation pairs made at different times so as to estimate the probability distributions. The algorithm that is presented in this section requires this assumption as well. However, if the system is (slowly) varying, the stationary approximation may be postulated for the recent past. The expired data should then be discarded: When the system is non-stationary the Complexity evolves with time [SHAL01]. The risk run by making the conditionally stationary assumption on a slowly varying system is that our distribution estimates are slightly wrong, but so long as the system does not change too fast these estimates will still be below the Chi-Square test precision used for detecting similar distributions (and that test cannot be too precise anyway otherwise there is a false negative risk of not recognising similar distribution estimates computed from a finite number of samples).

Figure 15 presents the current algorithm from [SHRKM05] in pseudo-code, rewritten so as to include comments given in the main text of [SHRKM05], and after checking the source code provided with that article[30]. The notations used are $d=\{(f,n_f)\}$ for a distribution over futures, as the number of times n_f each future f in this set was observed. An observation is for this algorithm a pair $u=(c,d_u)$ associating a past cone c to a future distribution d_u. An estimated causal state is a pair s=(U, d) associating a set $U=\{u\}$ of past cones to a distribution d_s of futures, so that d_s is the sum[31] $d_s = \Sigma_U d_u$ of each individual d_u.

Input: All observations in a set U_{obs}
Output: A set S of estimated states

let $u \in U_{obs}$ # u is an observation from U_{obs}
$S := \{ (\{u\}, d_u) \}$ # initialise the set S of states
$U_{obs} \leftarrow U_{obs} \setminus \{u\}$ # remove u from U_{obs}
for each $u \in U_{obs}$ in random order
  for each state $s \in S$, if $\chi^2(d_u, d_s) < \alpha$ # Perform a Chi-Square test
    $s \leftarrow (U_s \cup u, d_s + d_u)$ # insert u into s
    loop to the next u # consider only the first match
  $S \leftarrow S \cup \{ (\{u\}, d_u) \}$ # no match ⇒ create a new state

Figure 15: Pseudo-code for the existing algorithm

I reproduced this algorithm with a straightforward implementation and it did not produce as good results as I expected. The algorithm converges in the limit of an infinite number of observations to the correct states (see [SHAL03]), but behaves poorly for handling the inevitable errors when estimating the future distributions with finitely many samples.

As an example let's consider three different past light cones c1, c2, c3, each observed with estimated probabilities over future cones p1, p2, p3. The following

30 The project providing this code is Cimula, http://cimula.sourceforge.net/
31 The text in [SHRKM05] and [SHAL03] mentions the average distribution. Since here I maintain the future counts instead of the future frequencies, the average distribution is obtained by summing the individual $d=\{(f,n_f)\}$.

possible scenario may happen: First, c1 is put into a new state s1. Then c2 is presented and p2 does not match p1, so c2 is also put into a new state s2. c3 then arrives, and matches both p1 and p2 with the Chi-Square test, despite the fact p1 was too far from p2. c3 is put in state s1, so s1 now has probability distribution (p1+p3)/2. At this point, c2 would also match s1 but it is still classified into c2. Since observations are never reconsidered there will be a spurious duplicate state s2 that actually matches s1.

This situation may happen because p1, p2 and p3 are imperfect estimations from the data of the true probability distributions q1, q2, q3. Assume that in this case there really is only one state. In the limit of an infinite number of collected observations the algorithm would work because p1, p2, and p3 would then converge to q1,q2,q3 and be equal as they refer to the same state. With finitely many samples, it is possible that the algorithm in [SHRKM05] produces spurious states, resulting in higher complexity estimates. Cosma R. Shalizi explains [SHAL03] the conditions for which the algorithm works correctly.

The second type of error is classification into a unique state because p1 and p2 match, whereas q1 and q2 would differ. Randomizing the order as proposed in [SHRKM05] may help because if other pasts are merged with p1 before p2 is presented, the average s1 cluster distribution may have been shifted away from p1 and p2 may not match any more. Randomizing makes this error less likely to happen but does not suppress it completely.

The new algorithm I propose is described in pseudo-code in Figure 16. The notations are the same as before.

The first difference is in the arguments: new observations are provided and older ones may be removed, and the algorithm may be called with partial

information. As data becomes available the algorithm may be called again to update the estimates with the latest information. This makes the algorithm invaluable for getting early estimates and monitoring a live system.

Compared to the previous algorithm the arguments are now the updates: for each past cone, a count (positive or negative) of the future cones that were observed with this past. Past/future associations may be removed when they are too old and we need to check the evolution of the complexity values. This allows to feed the observations incrementally as they become available and as they expire, and always get the most up to date complexity estimates.

Inputs: New observations in a set U_{obs}
A current set of states S, possibly empty
Output: The set S updated with U_{obs}

for each $u \in U_{obs}$ in random order

```
    if a state s ∈ S contains v=(c_u,d_v) ∈ U_s      # see Note 1
        s ← (U_s \ v, d_s - d_v)                    # remove v from s
        if s = Ø
            S ← S \ s
        else
            call merge({s})
        d_u ← d_u + d_v                             # update u, see Note 2
        if d_u = Ø                                  # all futures were removed
            loop to the next u

    M := Ø
    for each state s ∈ S,  if χ²(d_u, d_s) < α
        M ← M ∪ {s}

    if M = Ø
        S ← S ∪ { ({u}, d_u) }                      # no match ⇒ create a new state
    else
        let s ∈ M                                   # choose one s from M
        s ← ( U_s ∪ u, d_s+d_u )                    # insert u into s
        call merge(M)

subroutine merge(M):
    let s ∈ M                                       # s is a state from M
```

```
for each state t ∈ M, t≠s
        s ← ( U_s ∪ U_t, d_s+d_t )          # merge t with s
        S ← S \ t                           # remove t from S

M ← Ø
for each state t ∈ S, t≠s,  if χ²(d_t, d_s) < α
        M ← M ∪ {s}

if M ≠ Ø call merge(M)
```

Note 1: This can be implemented either with a map c→s or by looping through the states.
Note 2: Since only previously observed (past,future) mappings may be removed, d_u contains only positive counts at this point if it is not empty.

Figure 16: Pseudo-code for the new incremental algorithm

There cannot be any matching duplicate states after a call to the new algorithm by construction: The merge routine is called whenever a state is modified including by itself when necessary. The recursion eventually terminates if all states are merged, or when no states match. So, the merge routine guarantees that all states differ when it returns. New states are created only if a cone doesn't match any existing state. The first type of error made by the previous algorithm is thus avoided.

The downside is a potential increase of the second type of error: that some cones are misclassified into the same estimated state whereas their true causal state differ. However, thanks to the re-clustering in the first part of the loop, a mismatching cone would first be removed from its current cluster when it is observed again. It is then reclassified by taking into account the latest observations. So, the probability of a misclassification decreases each time the cone is observed, becoming null in the limit of infinitely many observations.

The reference implementation provided on my web site[32] introduces another

32 See http://nicolas.brodu.free.fr/en/programmation/mesincom/index.html

refinement: tracking the cluster corresponding to each given observation, even during the merges. When the algorithm returns, the user is then provided with these clusters in the same sequential order as the observations were fed to the algorithm. This would be equivalent to first calling the pseudo-code presented here, and then looking up the state corresponding to each provided past observation.

The usage is very simple:

1. Define what a light cone means for a given system.

2. Collect (past/future) light cone pairs by observing the system.

3. When enough observations are made, feed them to the algorithm. This may be after each observed pair (fully incremental) or at any other time, trading granularity for speed (less calls mean less re-clustering).

4. The clusters, hence by extension the complexity estimates, are now available for each observation that was provided, in that order.

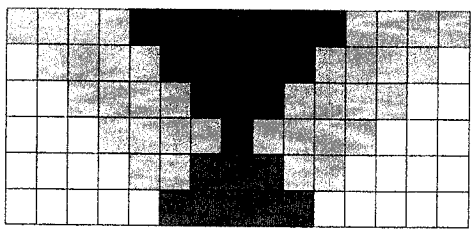
The application of statistical complexity to detect patterns in cellular automata was proposed by Shalizi *et al.* in [SHRKM05] and these experiments are reproduced independently with the new algorithm in the next section. This allows to monitor how well the incremental algorithm converge to the same values as provided by the non-incremental one.

5.1.3 Analysis of cellular automata

Shalizi *et al.* [SHRKM05] propose monitoring the local statistical complexity for the cluster corresponding to each cell state in a cellular automaton. These experiments are reproduced here so as to: 1. Show how to apply the algorithm on a simple example. 2. Compare how the incremental version can converge to the states

found by the non-incremental version[33].

The first point in the previous section for applying the algorithm is to define what the light-cones are for the target system. In the case of cellular automata, a past (resp. future) light-cone consists in all the previous (resp. next) cell states that could influence (resp. be influenced by) the current cell state. Figure 17 shows the light cones for an elementary cellular automaton.



Time goes from top to down on this elementary cellular automaton grid. The present cell is coloured in green. A past light cone of size 4 (counting the present) is shown in blue. The future cone of size 3 is shown in red. All the grey cells would be necessary to compute the future cone.

Figure 17: Past and future light cones in an elementary cellular automaton.

In the case of elementary cellular automata the update rule is memoryless and applied for all points the same way. Boundary effects would have to be taken into account, but the use of a cyclic world makes each cell equivalent. Past/future cone pairs may thus be collected across all points to compute the statistical complexity of the whole automaton.

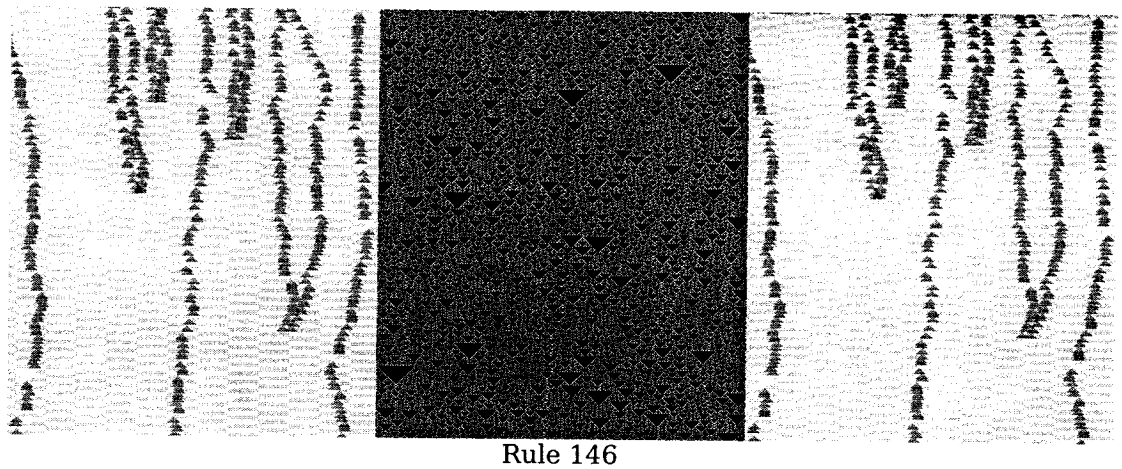
Figure 18 shows the algorithm in action on rules 146, 54, and 110[34], when feeding the observations one by one (right) and when feeding all observations at the same time (left). Only a third of the size of the cyclic world is shown. The result is in these cases a quick convergence of the incremental version to the states given by the non-incremental version. In each case there are "particles", definite patterns in the cell states that interact with each other. Compared to the raw cellular automaton binary values in the middle, the complexity field highlights these

33 My own algorithm is called in both cases, but with different ways of handling data. See also Figure 18.

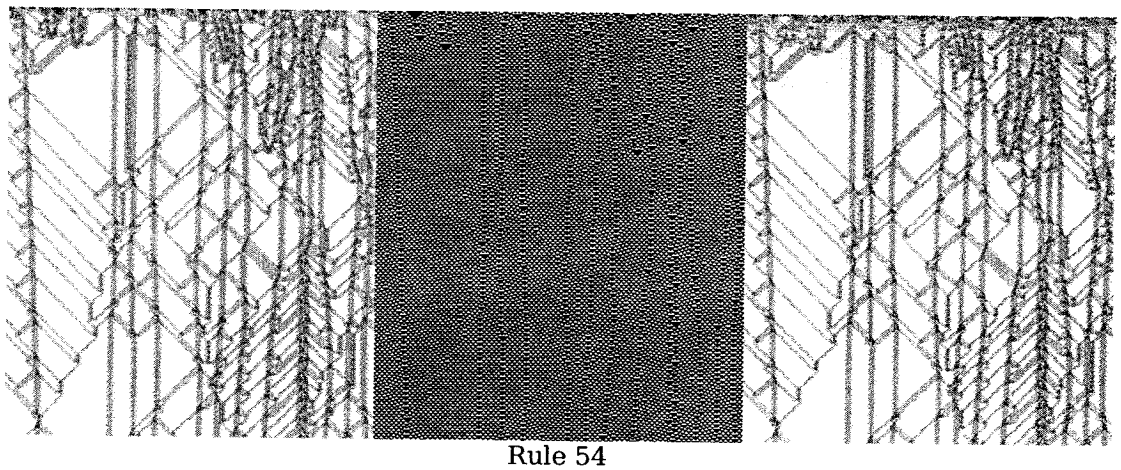
34 The convention for numbering the rules is the one by Stephen Wolfram [WOLF02].

patterns. The regions where particles interact exhibit the most complexity (black spots in Figure 18). Such regions are especially visible on the rule 110 example.

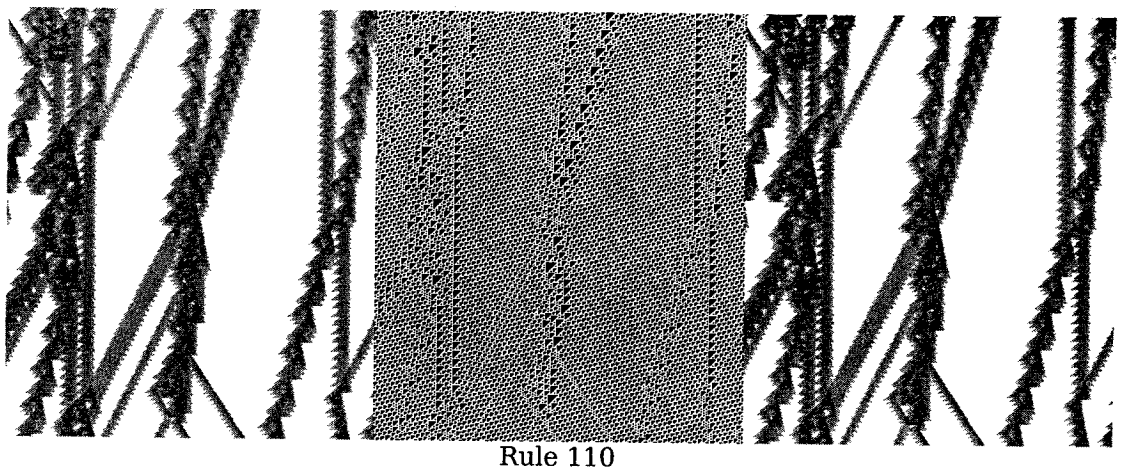
These images are directly comparable with the ones provided by [SHRKM05]. The same conclusion applies: Statistical complexity is a good filter for detecting patterns, especially since it is applied exactly the same way to the different systems and needs no further specialisation.



Rule 146



Rule 54



Rule 110

Left: All observations were collected before computing the local complexity field.
Middle: The raw cellular automaton.
Right: The observations were fed incrementally one by one to the algorithm.

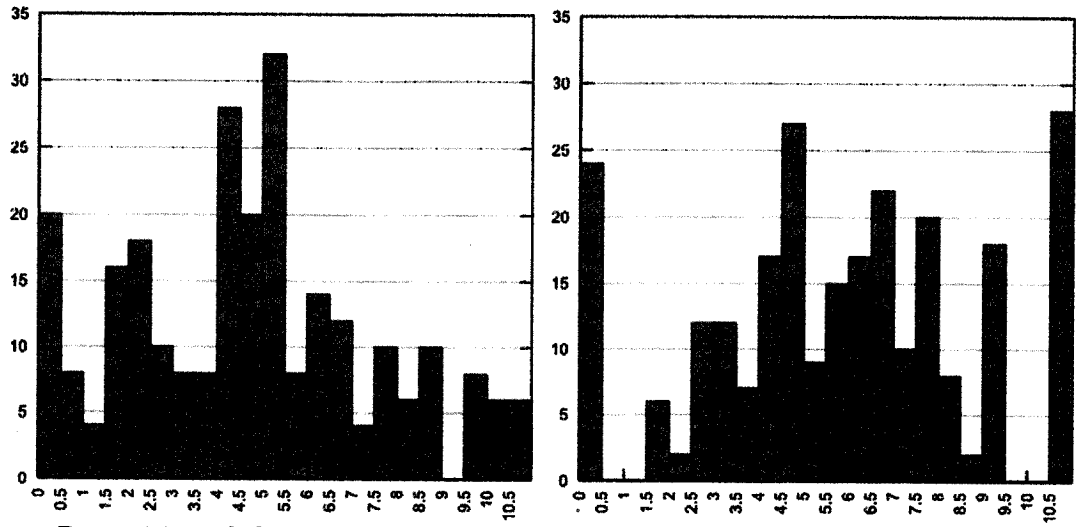
Figure 18: Local statistical complexity fields for some cellular automata.

A second experiment was conducted for this study. The Edge of Chaos hypothesis is that complexity reaches a higher value between order and chaos. The general claim from Stephen Wolfram book [WOLF02] is that only a small class of cellular automata generate complexity. A systematic study of what complexity values are obtained by monitoring the elementary cellular automata thus seems an interesting experiment to perform.

An exhaustive computation of all possible elementary cellular automata states was carried on for past cones of depth 6 and future cones of depth 4. A similar analysis as the one for Figure 17 shows that 23 bits are needed to fully determine the future light cones, whereas the past light cones have a support of 11 bits. There are thus 2^{23} different configurations, with possibly as much as 2^{11} estimated states in this setup. Since the knowledge of the oldest row in the past light cone fully determines the next rows, it is expected that the complexity value is at most the past cone support size. In Figure 17 the support size was 7 bits, in the present experiment the past light cones of depth 6 make the maximum complexity value 11 bits.

Since the setup for the present experiment is to compute the complexities on the exhaustive list of all possible states, transient states that are never revisited by each automaton are also included. A second batch of runs is thus performed where a cyclic world is initialised with 100000 random bits, and run for 500 steps to remove the transient states. The next 1000 states are then used to collect past/future cones so as to build the complexity estimates. If there were no transient states this experiment would then collect 10^8 observations on a $2^{23} = 8.39\ 10^6$ total space, so an average of about 12 observations per possible configuration. When the transients are removed this number is expected to grow.

The repartitions of the complexities obtained with and without transients for all of the 256 elementary cellular automata are shown in Figure 19.



Repartition of elementary cellular automata complexities when including all possible initial states

Repartition of elementary cellular automata complexities after suppression of the transient states

Legend: Bins are created on the horizontal axis for complexity values in bits. The vertical axis unit is the number of rules that fall in each bin.

Figure 19: Repartition of elementary cellular automata complexities

An increase of "complex" rules is observed when the transient states are not taken into account. This point is explained below. There are also more null complexity rules, which reflects the fact many automata converge to a fixed or repetitive pattern, like all 0 or all 1.

Some rules like 90 or 150 have an additive property [WOLF02]: the final state is the same as if summing the evolution of each initial bit separately. 2k+2f-1 initial random bits could possibly influence the k^{th} step, for a future depth f. What this means is that the information present in the limited-size past cone (a few steps) is unable to render the influence of all these bits, unless the additive rule highly squashes some bits (like rule 0). The estimated complexity of these rules is generally high in the second experiment (11 bits for rules 90 and 150). In these cases the first

experiment (exhaustive search) renders the best view, as it takes into account all possible initial bits that fully determines the system at any step. Rules 90 and 150 produce an estimated complexity of 0 bits in the first experiment! Other additive rules exhibit less extreme difference between both experiments. It may be that a similar effect for other non-additive rules, dependence on more and more initial bits, could explain the apparent complexity increase in the second experiment compared to the first, though it is not the only explanation.

Perhaps there are not enough samples so the distributions are badly estimated hence would not match. Reducing the size of the cones to a past depth of 4 and future of 3 gives a total number of possible mapping of 2^{15}, hence producing an average of about 3000 samples per configuration if there were no transient (and more otherwise). The corresponding experiment was conducted, but unfortunately the size reduction did not solve the issue: There is still an increase of apparently complex rules in this case. The hypothesis there are not enough samples is thus not validated by this experiment. Capturing more samples by increasing the world size and number of steps in the original experiment could possibly be done as well, but at the expense of a larger computational time for little added benefit.

Another hypothesis would be that the past/future light cones of size 6/4 are not large enough to capture the whole diversity of the system. The time dependencies of the automaton "particle" constructs are too large and not captured by the 6/4 cones limited scope. Considering the transients is then equivalent to adding noise and make the distributions look similar, hence reduces complexity. Removing the transients reveals that the 6/4 cones are not deep enough and there is a distinct distribution for each cone, hence maximum complexity. The problem is that increasing the size of the cones dramatically increases the computation time,

especially considering a sufficient number of samples needs to be collected, which makes larger experiments impractical to perform. Moreover, in the limit case of infinite time-dependency on the past, changing the light-cone to any finite size wouldn't help anyway.

A final hypothesis is that there really are different distributions of non-transient futures for each observed non-transient past. This would also be compatible with a maximum complexity whatever the cone size.

When the maximum (11 bits here) is reached this means all the bits are necessary to encode the states, hence there is one state per distinct past, hence each observed non-transient past has a distinct probability distribution over futures. This visually corresponds to rules that are apparently chaotic, with few distinguishable patterns. However this does not mean these rules are complex in other definitions of the term. Rule 90 for example leads to an estimated maximum of 11 bits, as aforementioned, while it has 0 bit complexity in the exhaustive experiment. On the contrary rules like 146, 54, and 110 in Figure 18 exhibit intermediate complexities of respectively 6.1, 7, and 5.8. Yet rule 110 in particular has proven universal computational power [WOLF02] and in this respect it is more "complex" than rule 30, used as a random number generator in [WOLF02]. But rule 30 gives a complexity value of 11 bits in the second experiment[35], more than the 5.8 of rule 110. The exhaustive search in this case is once again closer to our intuition: 10 bits for rule 110 and 6.8 bits for rule 30.

So, to sum up, we have two sometimes contradictory pieces of information:

35 Maximum complexity might seem to contradict the usage of rule 30 as a random number generator. But when the observations are limited to a single column, and trying to predict the next bits on that column, then experiments gave a complexity of 0. This confirms that any configuration of future bits on the same column are equally distributed with respect to the past bits on that column.

- The exhaustive complexity of the system, including transient states.

- The apparent complexity of the system: This is the complexity that corresponds to what we can actually measure by observing the system while it is running.

It would seem that only the second is of any practical use when monitoring a process that has been active for some time, or a physical phenomenon. Especially since the transient causal states can only be visited with non-null probability for a limited time, then are never visited again when moving to the recurrent causal states. Yet, the first experiment has given twice a complexity behaviour that better matches computational interpretations: for the additive rules and for the universal 110 vs. random 30 rules. In this case, perhaps there are better notions of complexity to use than the statistical one.

5.1.4 Conclusion

Statistical complexity is a powerful notion for quantifying the intrinsic difficulty of predicting a system from its past observations: null complexities correspond to systems that can be predicted with a fixed distribution of futures, while higher complexities correspond to systems with multiple states and each state with a different distribution. Statistical complexity can be estimated based purely on data observations without prior knowledge and thus form an objective measure on the system, though as shown by the example in the previous section, finite-size effects may interfere with the estimations.

The main advantages of the new algorithm compared to the state of art are: 1. The possibility to get early estimates as soon as data is available and update these estimates on the fly when more data become available. 2. The possibility to remove

expired data in the case of a non-stationary system. This new algorithm is thus a useful contribution in its own.

Both these features were used for the application of this algorithm to the main project presented in Chapter 4. But the algorithm is not limited to this project: the reference implementation is applicable to any user-defined light cone type and independent from any external library. This algorithm may thus be efficiently reused by other projects.

5.2 Implementation of a real-time multifractal analyser

5.2.1 Introduction

Multifractal analysis is a kind of statistics that operates on the regularities of a time series. It can be used to gain insight on the self-similar properties of a signal, in the form of a condensed "spectrum". In the context of the experiment in Section 4.2.3, the multifractal spectrum is used as a way to quantify the spiking properties of a neuron. An introduction to the theory is presented in this section so as to explain the interest of the incremental algorithm, with pointers given for further details.

Let's assume that we want to assert how smooth is a time series x(t). We might try to investigate how x(t) behaves nearby each point t_0. If x is continuous there will be no "jump" in the series. It might be additionally differentiable, etc., up to some order N and not N+1. Thus x(t) might be approximated around t_0 by a polynomial of degree N. If we could somehow monitor how frequent are the different N degrees over the whole series, this would already provide a global measure of irregularity. But we can do better.

Let's now remove this "polynomial trend" from the signal around each t_0: a polynomial $p(t-t_0)$ of order N is subtracted from x(t). Let's call what remains $r(t-t_0)=|x(t)-p(t-t_0)|$. If we now fit an exponential with parameters α and β to r, this gives $r(t-t_0)\simeq\beta e^{\alpha|t-t_0|}$ around t_0. Assuming this can be done[36], the largest α for which this polynomial p and exponential fitting exist is the local Hölder exponent of x(t) at t_0. It is then guaranteed that x(t) is differentiable up to the floor(α)=⌊α⌋, but α

36 It is possible that the series is itself a regular C^∞ function for some values of t_0 and the analysis will fail in these cases. Multifractal analysis thus works best for functions that are irregular almost everywhere, or conveniently for many real and noisy data sets.

also gives an additional information: how fast what remains diverges, the "strength" of the irregularity. As before quantifying how frequent is each α over the whole series gives an idea about that series global irregularity, and now also the strength information. Rudolf H. Riedi and Jacques Lévy Véhel [RV97] present the precise mathematical definition how to do this, which relies on the Hausdorff dimensions D_α of the sets of t_0 with given α values.

This definition is perhaps mathematically the most elegant, but it is not always the most useful for computing the multifractal spectrum in practice. Mandelbrot *et al.* [MBF97] introduce multifractal analysis using the notion of statistical self-similarity: relations between the distribution of a random variable X(t) and the same variable at a different scale: X(ct), with c a scale constant. In particular, if there is only one α value for the series, then the singularities are all alike and scaling t by c smooths them down. Thus X(ct) is distributed like $c^H X(t)$, with H the constant Hurst exponent. When there are several α values the relation becomes more complicated: X(ct) is distributed like $c^{H(c)}X(t)$, but H is now a random variable related to how "frequent" is each α. This is especially interesting from a practical point of view, because we now have an indirect way for quantifying the irregularities using statistics on the time series, rather than going through the approximations of each D_α. [MBF97] then presents how to do this using the moments of the random variable X(t): Monitoring the scaling properties of $E(|X(t)|^q)$ with respect to t and q allows to get back a constant c for each q as well as the multifractal spectrum, as the exponents of t: $E(|X(t)|^q)=c(q)t^{\tau(q)+1}$, with τ(q) related to the dimension D_α. See also the presentation [MBA93] by Muzzy *et al.* for how τ(q) and D_α are related.

This brings the computation of the spectrum a step closer to a practical algorithm: Computing moments on the data set is much more accessible than

computing irregularity strengths and Hausdorff dimensions. Yet a way must still be found to estimate the exponential relations for $t^{\tau(q)+1}$ on average over the whole series efficiently. Another piece of the puzzle comes from [MBA93] as the idea of using a wavelet decomposition of the series. Indeed, some wavelets have the "N-vanishing moments" property that their convolution with a polynomial of degree less than N is null. Therefore they make a prime target for the computation of the α values directly: The polynomials have already been removed and what remains, the $r(t-t_0)$ above, can be the target of an exponential fitting. How to relate the exponential fitting for $E(|X(t)|^q)$ to the one for the wavelet coefficients obtained by applying the wavelet filter over the whole series is explained in [MBA93].

The task is now computationally much simpler: moments, wavelet decompositions and a few exponential fittings are easy to perform. Yet [MBA93] uses a continuous wavelet transform, which is still somewhat cumbersome to apply. Variants using discrete wavelet decompositions exist, as well as other methods that directly fit polynomials to parts of the signal instead of wavelets. These methods are reviewed and compared by Oświęcimka *et al.* [OKD06], with precision and applicability to non-stationarity series considerations. A solution combining the advantages of the polynomial fitting approach with discrete wavelets is introduced by Manimaran *et al.* [MPP05]. The idea of this method is to interpret the local polynomial removal as a high-pass filter, removing the low "trend" frequency of the component, and expressing the relations between a signal x(t) and a scaled version of itself x(ct) as fluctuations of the higher frequency component in the signal when the low-frequencies are removed. This precisely can be done directly by discrete wavelet filtering, achieving a similar effect as the polynomial fitting but at lower cost. [MPP06] extends the argument to non-stationary series.

Yet, these methods apply to static data sets and are thus not well suited for the needs of my experiments in Section 4.2.3. In this setup one multifractal analyser is maintained for each neuron in the network and a new data value is added at each spike. Efficiency considerations exclude recomputing the spectrum from scratch at each spike: an incremental method is necessary to update the spectrum with new data with minimal recomputations. I thus extended the discrete wavelet method from [MPP05] so as to make it incremental, which was not a trivial extension: Discrete wavelet transforms are computed on dyadic frames, power-of-two partitions of data, and shifting the framing to include a more recent value is not possible without recomputing all decomposition levels. Another solution is thus proposed in the next section: maintaining the dyadic frames in parallel and maximally sharing the information between frames. A neat side-effect advantage is that in the end, the multifractal estimates may be averaged over all the frames to gain more precision.

The new algorithm which is presented here is thus not only faster but also more precise and robust than the state of art from [MPP05], and it fits the needs of real-time incremental updates of a multifractal spectrum.

5.2.2 Updating discrete wavelet transforms incrementally

As mentioned in the introduction the first difficulty is to maintain the wavelet decomposition information when new data are added.

A discrete wavelet transform processes the data into a low-frequency (averaged) version and high-frequency details, both at half the original resolution. The averaged data are then decomposed again, recursively, up to L levels of

decomposition. Figure 20 shows the relation between the successive low-frequency data for a wavelet with a filter support of 6 elements. The numbers indicate what data indices are needed to compute each element of the higher levels of decomposition. The high-frequency details require data indices in a similar way.

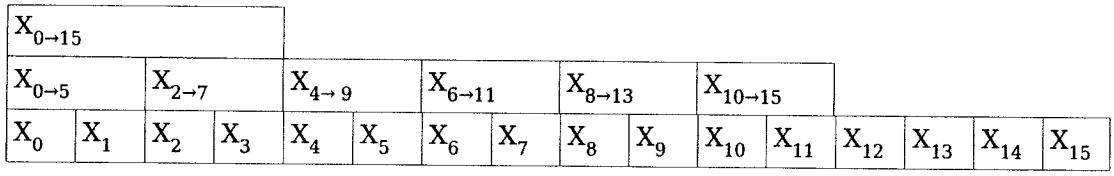


Figure 20: Relations between low-frequency levels of decomposition

Figure 20 also outlines the dyadic framing scheme that partitions the data. In this example, data X_0 and X_2 are aligned to the level 1 elements, X_0 and X_4 are aligned to the level 2 elements, etc. Since a level L+1 corresponds to half the resolution of level L, 2 elements of level L are paired at the same spatial location in one element of level L+1.

But if a new value X_{16} arrives to the analyser it cannot be paired with X_{15} since this one is already paired with X_{14}. This problem does not happen for static data sets but poses a real issue for dynamic data. One could consider the alternative pairing shown in figure 21, but then, all previous computations are lost: the data indices that are necessary change and wavelet filters need to be applied recursively to all levels again.

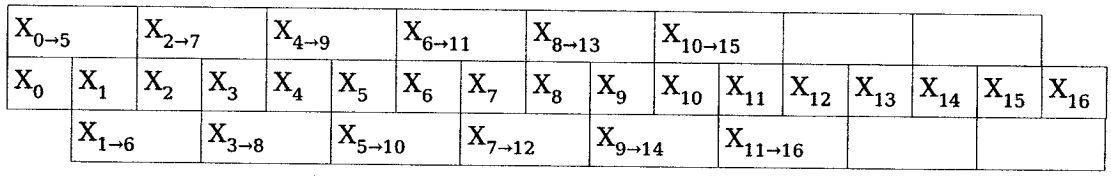
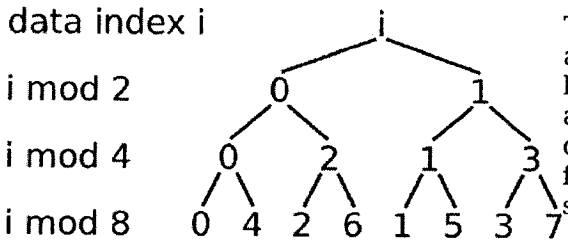


Figure 21: Alternative pairing schemes for taking new data into account

It is possible to maintain both frames at the same time and switch from one to the other as new data arrive. This is manageable when there is only one level of decomposition, but becomes exponentially problematic: the number of alternative

framings is 2^L with L the total number of levels.

The solution comes from careful examination of figure 20. X_0 and X_4 are aligned to the level-2 elements in this case, but another framing exists where the level-2 elements are aligned with X_2 and X_6. These two framing share the computation for the level-1 element $X_{2\to7}$. By extension, any higher level using the level-2 $X_{0\to15}$ or $X_{2\to17}$ also share the level-1 $X_{2\to7}$. The number of alternative framing is 2^L but fortunately the size of the elements at level L is also 2^L, which means that elements are exponentially shared. Therefore the total amount of memory that is necessary to store all alternative framings is O(L) and remains manageable. Figure 22 shows how the data indices are shared for the higher levels of decomposition.



The data indices are split between even and odd pairing groups, as in Figure 21 but recursively. Each node in this tree is an element of the corresponding level of decomposition. It is shared by all the frames indicated as the leaves of the subtree for that node.

Figure 22: Sharing the computations between the dyadic frames

[BROD05B] describes exactly which indices are necessary for the decomposition of the data and its reconstruction by wavelet transforms, as well as a formula for calculating the size of the data that is necessary to compute a given level with respect to the wavelet filter length.

The algorithm for updating the wavelet transform incrementally is now simple:

1. Start from data level 0.
2. Switch to the alternative frame for the current level (if it is not level-0) and integrate the new data value.
3. Stop if the maximum level is reached. Otherwise produce the low and

high frequency data using the wavelet filter.

4. Loop to 2. using the low-frequency result as new value for the next level.

The result of this algorithm is an always up-to-date wavelet decomposition that is also obtained optimally in terms of previous computation reuse. After 2^L updates a previous frame is reused and points 2-4 above only compute the changes brought by the latest data. Any intermediate result that could be shared between different frames has been shared. The updated discrete wavelet transform of the time series is obtained in constant-time, whatever the data size, with only O(L) wavelet filter applications.

5.2.3 Updating the multifractal spectrum in constant-time

The method mentioned in introduction for computing the multifractal spectrum from the wavelet decomposition relies on extracting the data fluctuations at a given level. The moments of the fluctuations are then fitted with an exponential: the value of the exponent estimates for each moment form the multifractal spectrum.

The first step is thus to extract the fluctuations, according to the method described by Manimaran *et al.* [MPP05] and which is adapted in the present case for incremental updates. The global "trend" of the data at each scale should first be obtained. Noting the level λ, the "trend" $D(\lambda)$ is computed by inverse wavelet transformation, from the level λ down, but without the details. Thus $D(\lambda)$ has the same resolution as the original data $D(0)$ and their difference forms the desired fluctuations $F(\lambda) = D(0) - D(\lambda)$ of the data at the scale $s = 2^{\lambda}$.

As is apparent in Figure 20 the highest levels of decomposition corresponding to the latest data are not available. This is a fundamental problem for all time-

frequency analysis techniques: It is not possible to instantaneously get a frequency decomposition from the signal, and the lower the frequency (the higher the level λ in this case) the larger the time delay. This problem is accounted for in Section 4.2.5 and translates for this algorithm into a "lag" between the data that is fed to the analyser and the data the spectrum relates to. The exact lag time depends on the size of the wavelet filter and the number of levels. Assuming the reconstruction and decomposition wavelet filters are the same size w, then the lag is given in [BROD05B] by $(2^L-1)(w-2)$. For the current problem of determining the moments of the fluctuations the lag means that only R data can be reconstructed (perfectly or without details), so the fluctuations are computed on these R data only.

The power-sum p(λ,q) is first computed from the fluctuations: $p(\lambda,q) = \sum_{k=1}^{R} |F(\lambda)_k|^q$ where $F(\lambda)_k$ is the fluctuation obtained from level λ, taken at the data index k. The power-mean $f(\lambda,q)=\left(\frac{1}{R}p(\lambda,q)\right)^{1/q}$ is then fitted exponentially to the scale to get the spectrum: $f(\lambda,q)\propto s^{h(q)}$. Both points are detailed in the next subsections.

Updating the power-sum p(λ,q) in constant-time

The problem is that when a new data value arrives (after the overall aforementioned lag) it cannot be integrated directly as a new term into the power sum. As previously explained it is necessary to modify the current dyadic frame if we want to be able to integrate the data into the wavelet decomposition. The fluctuations are thus computed on different frames, and so are their power-sum. It is not correct to add terms from different frames as the information they contain partially overlaps.

The solution is to keep the current power sum together with the decomposition

level elements and share them as in Figure 22. This way, when the same dyadic frame comes again after 2^L new data values, the correct power sum is ready for update. An immediate benefit of this scheme is that only the last term of the power sums $p(\lambda,q)$ has to be computed for each level λ, and added to the current total.

Data removal also has to be considered since we're dealing with finite memory. Moreover, removing old data as soon as possible allows to deal with non-stationary time series, where we assume the stationary approximation holds over the last data values only. In this algorithm, the size of the higher level buffer determines the data extent in the past that needs to be kept, as explained in [BROD05B].

A problem with the current scheme is that subtracting the terms corresponding to the obsolete data in the power sum cannot be performed at the time the data is removed. Indeed, as Figure 20 suggests, recomputing the oldest term of the power sum is not possible since the data that would be necessary for this computation has already been removed at the higher levels. The solution is simple: At every data addition, once the power sum is updated with the new data, the last term for the current dyadic frame is computed since the old-but-not-yet-expired data value is available at this point. This term is then subtracted from the power sum just before storage together with the shared elements as aforementioned. The next time the same frame comes again after 2^L updates the expired data is gone, but so is the corresponding term form the power sum.

Another trick is explained in [BROD05B] to further increase the performances. Since wavelet filtering is a linear combination of the data values, so are combinations of filters. Both direct and inverse wavelet transforms are filter operations. But the data trends array $D(\lambda)$ is computed from the low-frequency component at level λ, without including the high-frequency details. The $D(\lambda)$ values

are therefore linear combinations of the low-frequency elements of level λ. Since the level λ-1 low-frequency elements can be reconstructed from both low and high frequency elements of level λ by the inverse wavelet transformation, this means D(λ) can be expressed as D(λ-1) plus a linear combination of the high-frequency components of level λ. And since F(λ) = D(λ) – D(0) and F(λ-1) = F(λ-1) – D(0), it turns out there exist a relation in the form $F(\lambda)=F(\lambda-1)+\sum_{k=0}^{m-1} c_k d(\lambda)_{i+k}$. The unknown parameters in this relation are the size m of the filter that links F(λ) and F(λ-1), c_k the coefficients of that filter, and i the first index of the details d(λ) to apply that filter to. [BROD05B] explains how to precompute all these parameters: F(λ) can now be obtained from F(λ-1) at run-time at very low cost.

To sum up, updating the L power-sums p(λ,q) for each q can be done in a single O(L) pass:

1. Initialise F(0) = D(0) – D(0) = 0 by definition.
2. For each λ from level 1 and up to the maximum level L

   a) Compute $F_{old}(\lambda)$ using $F_{old}(\lambda\text{-}1)$ and the details at level λ, according to a precomputed filter as previously mentioned. Similarly compute $F_{new}(\lambda)$ from $F_{new}(\lambda\text{-}1)$. Old and new refer to the first and last values (in FIFO order) that were added to the data buffer.

   b) Update the power sum p(λ,q) by adding $|F_{new}(\lambda)|^q$ and keep this intermediate result as the final one.

   c) Subtract $|F_{old}(\lambda)|^q$ from p(λ,q) before storing it back together with the level λ elements, for sharing as in Figure 22.

The result of this operation is a set of L power-sums for each q, computed in O(L) constant time irrespectively of the data size, ready for an exponential fitting.

Exponential fitting to get the multifractal spectrum

Now that the L power-sums p(λ,q) are computed, for each q, the goal is to find the h(q) such that $f(\lambda,q)\propto s^{h(q)}$.

Taking the logarithm of the previous expression gives $\log_2 f(\lambda,q)\approx h(q)\lambda+C_1$, since $s=2^{\lambda}$, and with C_1 a constant.

Using the definition of $f(\lambda,q)$, the expression becomes $\log_2 p(\lambda,q)\approx q\,h(q)\lambda+C_2$ with $C_2 = q\,C_1 + \log_2 R$ another constant that can be precomputed for each q.

A weighted least square fitting is then performed to estimate the best h(q): $\sum_{\lambda=1}^{L} w_\lambda\left|\log_2 p(\lambda,q)-(q\,h(q)\lambda+C_2)\right|^2$ is minimized. The weights are necessary to counter the biased influence of the highest terms. Usually the w_λ are taken to match the exponential values (p(λ,q) in our case) but this would make the weights data-dependent. A faster run-time solution is to use a weighting scheme w_λ that can be precomputed and which relies only on λ. I suggest in [BROD05B] using white noise as a neutral reference, so no bias is introduced in the spectrum estimation for possible long-term or short-term dependence of the data (these data dependence notions are introduced by Jones *et al.* [JLM96]). Using white noise gives the weighting scheme $w_\lambda=\sqrt{2^\lambda}$. This scheme is used by default in the provided implementation but can be overridden by the user if need be. In practice for the experiments that were conducted in [BROD05B] the weighting scheme only has an influence in the second decimal place for the h(q) estimates for most cases.

The solution to the least square fitting is provided in [BROD05B]. It is in the form $h(q)=\sum_{\lambda=1}^{L} C(q)\log_2 p(\lambda,q)$ with C(q) a constant that does not depend on the data and which can therefore be precomputed. Step 2b of the algorithm sketched in the previous subsection now includes the addition of the $C(q)\log_2 p(\lambda,q)$ terms to the h(q). When reaching the highest level L, the h(q) are ready for all q. Given the

fact that on x86 based personal computers (and perhaps other FPUs) there exists a "x.log$_2$y" assembly instruction, the h(q) can be computed in a single O(L) pass very efficiently.

The correlation coefficient for the exponential fitting can be computed as efficiently. It can be used to detect cases where there is no exponential relation. As mentioned in the introduction the analysis may fail for some series. The correlation coefficient thus provides a way to detect at run time when the analysis fails. This error condition is detected and recovered from in the main project, Section 4.2.3.

This concludes the presentation of the spectrum estimation. As a new data value is fed to the analyser, and an expired one removed, the h(q) spectrum is maintained in constant O(L) time, independently of the data size.

5.2.4 Conclusion

The algorithm presented in this document is based on the discrete wavelet method for estimating a multifractal spectrum that was introduced by Manimaran *et al.* in [MPP05]. Yet, for the main project in Section 4.2, I needed a way to characterise the multifractal properties of a time series efficiently and incrementally. I thus decided to extend the previous algorithm in this direction.

The new algorithm was primarily designed toward real-time incremental updates of a multifractal spectrum. Yet, some of the improvements introduced in this document would be beneficial to other wavelet-based techniques, like the trick for computing F(λ) from F(λ-1) that could have been used as well with the static [MPP05] technique. One final feature makes the new algorithm better than [MPP05] even for the analysis of static data sets: averaging over all dyadic frames. Indeed, the method presented here computes successively the multifractal spectrum on all

the possible dyadic frames. Averaging the result over all the frames thus results in more precision and numerical stability: the estimation is not subject any more on the (implicit) bias of the other method of choosing only one particular frame. The new algorithm is thus not only more efficient than the previous one, but also more precise. As shown in [BROD05B] it is also robust enough for most applications: the numerical drift that occurs when adding many floating-point numbers is not an issue.

Therefore, the algorithm presented here represents a useful contribution in itself. It is generic, and could be applied to many domains.

Finally, the reference C++ source code of the algorithm is provided as Free software on my web page[37] in the hope it will be useful. This implementation is standalone and does not require the use of additional libraries. It includes all the improvements and options that are presented in [BROD05B] and in this dissertation.

37 http://nicolas.brodu.free.fr/en/programmation/incremfa/index.html

5.3 An algorithm for finding the neighbours of a given mobile agent

This algorithm initially started as a side project to the environment presented in 4.1. One of the limitations for including a large number of agents in the simulation is the cost of finding the neighbours of each agent (for example to find preys). Finding the neighbours of a given point is a general problem that has roots in density estimation and pattern classification (see the references given by Friedman *et al.* [FBF77]) and that is currently still a major topic of investigations, especially for geographic location systems (see the description by Sankaranarayanan *et al.* [SAS05]). However these applications usually assume a fixed data set, and the problem then becomes how to pre-organise this set in order to make the finding of the nearest data instances to a query point efficient at run time.

This approach does not correspond to the present case. The current problem is to deal with mobile agents that have no predefined positions. The task is to find, at run-time, the K nearest or all the neighbours of an agent within a given range, for each agent. Additionally updating the agents position must be very efficient since this operation is done at a higher frequency than the queries themselves: For example a physics engine integrates the agents trajectories more often than the agents AI needs the neighbourhood information.

The naive algorithm of looping over all other agents, for each agent, so as to find the closest ones, is computationally $O(N^2)$ and thus not suited for a real-time simulation. Classical static data set preprocessing methods are reviewed by Hanan Samet [SAME95] and they generally use a tree-based data structure to hold the pre-organised data, but these don't work well either in the present case: Changing the agents' positions degrades the tree properties, which then has to be rebuilt, and this

is a costly operation (and more so than to the constant-time update presented below in any case). Maintaining the K-nearest neighbours list for mobile agents is still feasible with tree-based techniques, as in the algorithm proposed by Li *et al.* [LYH04], though in this case the trajectories of each object within each other object's local basis needs to be computed. Even then tree updates are still unavoidable and certainly not constant-time. Piotr Indyk and Rajeev Motwani [IM98] present approximate nearest neighbours finding techniques that are adapted to some large-dimensional database problems, but they are not suitable in the present case either since we need the exact neighbours for the AI.

In fact, since the agents presented in Section 4.1 are simulated with principles similar to these introduced by Craig Reynolds [REYN00], the solution presented there also match the present needs: The simple bin-lattice spatial subdivision method, which allows for constant-time updates, insertions and deletions of points, and amortised query time often close to O(1) for each agent.

The next section thus describes the bin-lattice algorithm and explains how it handles the nearest neighbours finding problem for moving objects. The main idea of the new algorithm is then presented, in the form of an improvement over this bin-lattice method. The algorithm is then detailed in Section 5.3.2, followed by how the main idea may be further optimised in Section 5.3.3. Section 5.3.4 proposes performance comparisons that show the advantage and disadvantages of the new algorithm in various situations, especially for the case of finding only the k nearest neighbours. Section 5.3.5 concludes and recapitulates the main strengths and weaknesses of the new algorithm.

Though not directly related to the main emergence issue, this algorithm is presented in this dissertation as it represents an advance compared to the state of

art. The article [BROD06B] detailing this algorithm with further benchmarking was accepted conditionally to some changes in the Journal of Graphics Tools.

5.3.1 The bin-lattice method and how to improve it

The idea behind the bin-lattice method is simple: discretise the space over the region of interest into multiple cells and maintain a list of agents present in each cell. When the neighbourhood query is performed only the cells that are within range need to be processed, and all the agents that are in distant cells are not even considered. Each cell thus acts as a spatial database of agents. The best case is a request below one cell size and that cell is empty or contains only the best match: the algorithm is thus O(1) for one request. The worse case is when all agents are within a single cell, in which case all of them are considered and the algorithm degrades to the O(N) case for one request. In general there are only a few agents in each cell: Hopefully the algorithm can be tuned to a particular environment by selecting an adequate cell size. A finer discretisation means less agents per cell on average, at the expense of more memory consumption and the need to process more cells for the same query distance.

Despite its simplicity this solution is well adapted to dynamic situations. When an object moves the structure is updated in constant time. If the object remains in the same cell no change is necessary. If the object moves to another cell it is removed from the current cell double-linked list of objects in O(1), and so is inserting it into the new cell list. Thus this solution answers the question of maintaining a guaranteed constant-time level of performance as objects are moving, unlike tree based solutions.

The problem now becomes the identification of which cells to keep and which

cells to discard. This is where the new algorithm comes into play. Current state of art [REYN00] is to simply perform a triple-loop within the minimum and maximum reachable cells along each dimension. This actually corresponds to running through a N1 norm cube. However queries are usually specified using the N2 Euclidean distance, thus require only searching through a sphere volume which is exponentially smaller than the N1 cube as the dimension increase: The 2-dimensional circle is about 79% the size of its bounding square, the 3-dimensional sphere is about 52% the volume of the bounding cube, the ratio goes down to 31% in four dimension, and less than 1% in dimensions nine and above. What this means is that potentially many cells are uselessly considered by the N-dimensional loop, and not accessing these cells results in a net benefit (the alternative solution is to test the distance of each cell at run-time, with the associated test cost).

Figure 23 shows a situation for three dimensions, where 68 cells out of 343 could be saved by considering only these which intersect the query sphere.

While attractive this solution poses essentially two problems: 1. How to identify the cells of the cube that fall completely outside the sphere. 2. How to do it efficiently, so the induced costs don't cancel the potential benefits of the operation. The next section describes the first point, Section 5.3.3 the second.

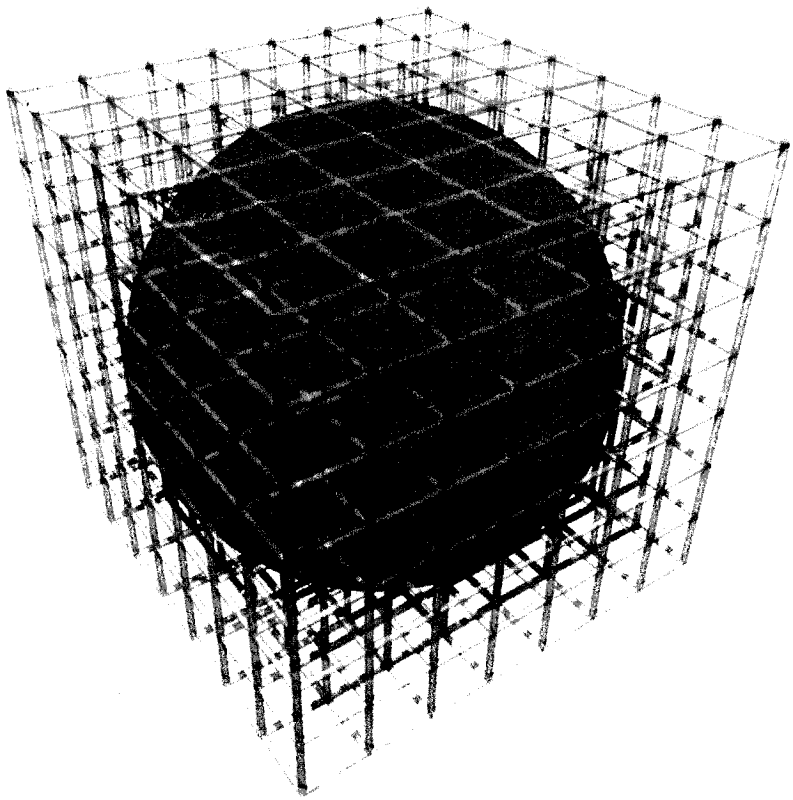


Figure 23: Query sphere intersection with its bounding cube.

5.3.2 Proposed Solution

The new algorithm handles finding the cells that intersect the query sphere in two steps:

1. Looking up into a pre-computed table what is the list of all cells intersecting a generic query sphere centred on 0, from centre to sphere edge.

2. Translating this list to the real query centre. Care is also taken to handle cells that would fall outside the region of interest, as well as not duplicating entries in a cyclic world.

The next two subsections detail each of these two steps.

Building an index table for the intersecting cells

The problem can be formalised in this way: If C is the query centre, and d is the maximum distance to look for neighbours, the objects X that must be found by the algorithm are these for which $x \leq d$, with $x = \|X-C\|$ the distance between C and X. A cell that intersects the query sphere is a cell L such that $\exists P \in L$ a point within that cell such that $\|P-C\| \leq d$. Conversely, all points P such that $\|P-C\| > d$ cannot be in the query sphere. All cells L such that $\forall P \in L \; \|P-C\| > d$ should be rejected.

The first step previously mentioned is presented in this subsection and consists in building a pre-computed table that is independent from C. Let's assume for now C is within the region of interest. In that case, C belongs to a cell L. Without knowing the exact location of C within L, it is still possible to reject all cells E that are too far away: If $\forall P \in E, \; \forall C \in L, \; \|P-C\| > d$, then the cell E is rejected. By looking at the minimum squared distance between cells it is thus possible to pre-exclude some cells. Figure 24 shows the minimum between cells in two dimensions, but the process can be extended to higher dimensions easily.

18	13	10	9	9	9	10	13	18
13	8	5	4	4	4	5	8	13
10	5	2	1	1	1	2	5	10
9	4	1	0	0	0	1	4	9
9	4	1	0	0	0	1	4	9
9	4	1	0	0	0	1	4	9
10	5	2	1	1	1	2	5	10
13	8	5	4	4	4	5	8	13
18	13	10	9	9	9	10	13	18

Consider that the query centre is located in the greyed cell in the middle of this array. Points in that cell may be at minimum distance 0 from the query centre if P=C.
Points in the cells surrounding the centre one may mathematically be at ε minimal distance from C, with ε depending on the floating point precision of the algorithm implementation. For the present purpose equating ε to 0 just increases the risk (with very low probability) that the cell is uselessly included, which does not affect correctness (just performance).
Minimal squared distances to other cells are given in this array, with examples for $5 = 1^2+2^2$ and $13 = 3^2+2^2$.

Figure 24: Minimum squared distances between cells

A similar computation for three dimensions indicates there are 27 cells at distance 0, 54 at squared distance $d^2=1$, 36 at $d^2=2$, 8 at $d^2=3$, 54 at $d^2=4$, etc.

Each cell is then represented by an offset from the centre cell. This process is related to the translation of the query sphere to the query centre and it is detailed in the next section. For the current explanation needs, each cell offset specifies the X/Y/Z coordinates of the translation from the centre cell, in cell units.

Since all squared distances are integers the cells may now be sorted by increasing distance from the query centre into a linear distance-indexed array: First all offsets for cells at $d^2=0$, then offsets for cells at $d^2=1$, etc.

At run time the user specifies a query distance d. All there is to do is compute the integer $n = \text{floor}(d^2) = \lfloor d^2 \rfloor$, such that $n \leq d^2 < n+1$. As previously explained all cells that are at a distance strictly above d are rejected. Therefore, all cells at n+1 or more are rejected, and only cells up to and including $n = \lfloor d^2 \rfloor$ need to be considered. At run-time, all there is to do is truncate d^2 and fetch the corresponding entries in the offset array: Since this array is sorted by distance as previously explained, running it up to $n = \lfloor d^2 \rfloor$ gives all the offsets of the cells that intersect the query sphere.

Some d^2 values are missing from the offset array because they do not correspond to an exact squared distance, for example 7. These distances may still be reached by truncating floating-point values, but by construction they do not bring any new cell: Missing d^2 entries are thus treated as the largest entry present in the array that is below d^2 (for example 6 instead of 7 in three dimensions, or 5 in two dimensions).

Figure 25 recapitulates the algorithm so far:

```
At precomputation time:
    L := the cell at the origin (0,0,0)
    A := []                                          #Array of sets of offsets
    for each cell S in the region of interest
        d := inf {x: ∀ P ∈ S, ∀ C ∈ L,‖P − C‖=x}     # See Figure 24
        A[d²] ← A[d²] ∪ {offset(S)}                  # d² in cell units, integer
    G := []                                          # Global offset array
    D := []                                          # Distance array
    i := 0
    for n := 0 to max(d²)
        if A[n] = Ø                                  # Missing d², see main text
            D[n] ← D[n-1]                            # A[0] ≠ Ø by construction
        else
            for each offset f ∈ A[n]
                G[i] ← f
                i ← i + 1
            D[n] ← i                                 # global offsets for d² > n
At run-time:
    n := ⌊d²⌋                                        # Truncate query distance
    for each 0 ≤ i < D[n]                            # All cells from centre to edge
        f := G[i]                                    # Offset of the cell
        S := translate(C, f)                         # Real cell S at that offset from C
        process(S)                                   # See the next sections
```

Figure 25: Building and using the global offset array

This pseudo-code forms the base of the algorithm only, important refinements are introduced in Section 5.3.3: Optimisations.

As is apparent in the run-time section of Figure 25, the new algorithm has replaced the triple-loop of the bin-lattice method for running through the N1 norm cube, by a single loop running through all cells intersecting the query sphere. Therefore the cells that are within the cube but outside the sphere are not even accessed, which promises reduced costs. However the new algorithm loop must go through an array indirection to get the offsets, as well as a translation to get the real cells from the offsets and the query centre. The next subsection details how the translation is performed, and Sections 5.3.3 and 5.3.4 deal with the costs considerations for the new algorithm.

Translating the indexed sphere to the query centre

The previous subsection has introduced the notion that each cell in the region of interest is given an index. This subsection details hows this index is computed, and how it is used for translating the generic distance-indexed offset array into the real query sphere centred on any given location.

The new algorithm assumes that the region of interest where the objects evolve is discretised into power-of-two numbers of cubic cells along each dimension. This limitation could possibly be waived in an extension to this algorithm, but it is however usually acceptable and brings non-negligible optimisations. The user may additionally always enlarge the region of interest to the next power of two if the cell size is kept constant, or change the cell size to match the region size if so desired.

Let B_X, B_Y, and B_Z be the numbers of bits that correspond to the 2-exponents along the X, Y, Z dimensions respectively. For example, for B_X=6, B_Y=5 and B_Z=3, there are respectively 64, 32, and 8 cells along each dimension. The remainder of this subsection refers to coordinates in these cell units. The actual locations of the cells themselves in the simulation are defined by the user with both the cell size and the position of the origin cell (which may be negative in world coordinates).

With this convention each cell in the region of interest may be given an absolute linear index encoded on $B_X+B_Y+B_Z$ bits. For example, the cell at position X=37, Y=12, and Z=3 is given the index 011_01100_100101, written in binary as Z_Y_X (underscores added for clarity). This packed binary index is essentially used to access the cells that are stored in a large contiguous memory array.

An unpacked format is also introduced to ease arithmetic operations on indexes, and forms the basis of the aforementioned offset representation. The Y component is shifted by $B_X+B_Y+B_Z$ and zeros are used to fill the gaps. The previous example

would be written as 01100_000000_011_00000_100101. The main advantage of this format is that it allows simultaneous (parallel) operations on all three components in a single machine register[38] without fear of overflows: Suppose for example that an offset is given as -11X+3Y+2Z, then the following two operations (addition, mask) are enough to realise the translation:

	01100_000000_011_00000_100101	Centre cell at (37, 12, 3)
+	00011_000000_010_00000_110101	Offset = (-11, 3, 2)
=	01111_000000_101_00001_011010	Note the overflow here
AND	11111_000000_111_00000_111111	Mask out the overflow bits
=	01111_000000_101_00000_011010	Unpacked result =(26, 15, 5)

Packing this result allows to give the final index of the cell in the large memory array with a shift to put Y back in place, a binary OR, and a final mask:

»	00000_000000_000_01111_000000	Shifted version of the result
OR	01111_000000_101_00000_011010	OR'd with the result itself
=	01111_000000_101_01111_011010	
AND	00000_000000_111_11111_111111	
=	00000_000000_101_01111_011010	Packed format for (26, 15, 5)

So, all components were translated to the query centre using the generic sphere indexed offsets in two elementary operations (+, AND), and packed to get back the offset of the cell in memory with three more elementary operations (shift, OR, AND). This is a form of Single-instruction, multiple data Within A Register (SWAR), a

38 In this example 25 bits are used by the unpacked format, so it can be implemented on a 32-bits architecture. Larger regions of interest may require 64-bits machines.

general programming technique presented in Randall J. Fisher's dissertation [FISH03].

The previous explanation works well only if the world is cyclic along all three components, due to the masking operations. Cyclic worlds are precisely desired for some simulations (for example in Grogono *et al.* [GCSYZ03]), but sometimes only along some dimensions, as the main project presented in Section 4.1 is cyclic only in X and Y. For non-cyclic regions of interest the above presentation fails: The offset for X in this example was interpreted as –11 but could as well be interpreted as 53 on the 6 bits two-complement arithmetic. For a cyclic world this doesn't matter but for a non-cyclic one both values are distinct and need to be represented. Indeed, while the cells themselves are always attributed positive coordinates in the region of interest, the offsets from the sphere centre may be negative.

The solution is simple: Encode each non-cyclic component using an extra bit, so as to allow for offsets with full-range precision in both negative and positive domains. When an overflow is still observed this means the cell falls outside the region of interest. In this case, the solution is simply to set a flag and ignore these offsets, and later process a unique "outside" region cell, which is conveniently attributed the global index ($1 \ll B_X+B_Y+B_Z$) so as to preserve the linear nature of the memory array storing all the cells.

Thanks to the unpacked offset format it is now possible to efficiently realise the translation from the generic distance-indexed query sphere to the real query centre. The global offset array G presented in the pseudo-code of Figure 25 actually is an array of unpacked offsets, using machine-register size units of memory.

5.3.3 Optimisations

The initial idea of running through only the cells that intersect the query sphere was so far implemented efficiently, but unfortunately it is not yet optimal and there still are extra cells that are uselessly included. To realise how this is possible let's consider what information has not yet been exploited: the sub-cell location of the query centre within its cell.

The problem is that the table-building process described in Figure 24 considers the minimal possible distance within points in two distinct cells. Yet if the centre is located at distance x from the border within its cell, it is at distance 1-x from the other side. Hence in Figure 24 most distances are actually too conservative estimates for a given situation: the right and left cells from the greyed middle one in Figure 24 cannot be simultaneously at distance 0 from the query centre. The original bin-lattice algorithm did not suffer from this problem since it computed the minimum and maximum cells along each dimension: for example, it considers between 1 and 27 cells for query distances below one cell size in three dimensions. But since in the present case the exact location of the centre is not known at precomputation time both sides were conservatively assigned to 0-distance: all 27 cells would be unconditionally processed.

The solution is to build separate pre-computed distance tables for each possible situation corresponding to locations of the query centre within its cell. The correct table is then selected at run-time with minimal and constant cost, and then the list of offsets for that particular situation is handled as before without additional cost. Figure 26, cited from [BROD06B], shows the relations between the cells and the query distance in two dimensions.

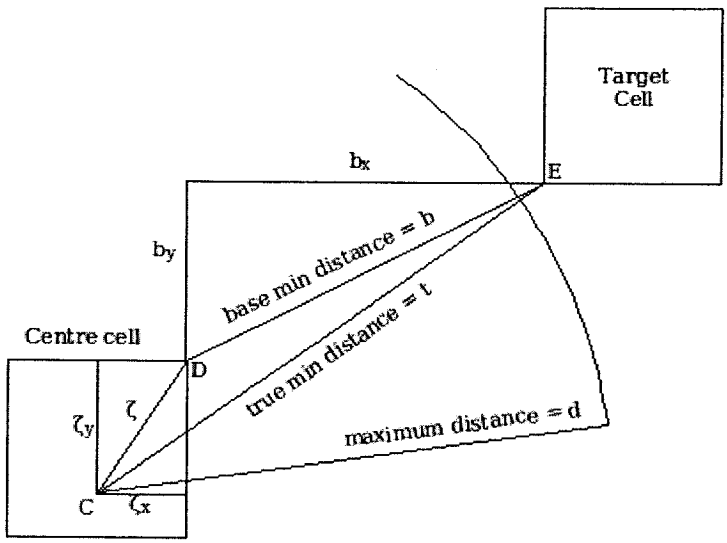


Figure 26: Distance considerations for building the specialised offset tables.

What Figure 26 highlights is that considering the sub-cell distance from the query centre to the cell edge along each dimension, ζ_x and ζ_y in this example, allows to reject cells at a base distance b (computed as before) that is smaller than the query distance d. Considering relations in the triangle CDE allows to express conditions for the rejection of the target cell with ζ_x, ζ_y, b_x, b_y and d. These conditions are examined in [BROD06B] so as to extract the parts that may be precomputed, which then form the basis for building the aforementioned specialised offset tables. The part that may not be precomputed may still be turned into a runtime test for the rejection of the whole cell: The analysis for the costs of that test compared to the benefits of rejecting the cell (thus not having to process all agents in that cell) is described in [BROD06B]. The costs depend in particular on the target cell spatial location (diagonal or along the main axis directions) and the gains on the average cell load in number of objects (which determines the benefits of rejecting the cell).

Thanks to this mechanism the new algorithm is also able to process less cells than the bin-lattice one even for query distances below one cell size. To realise how

this is possible consider the diagonal cells marked "0" around the grey centre cell in Figure 24: By considering the sub-cell location of the centre in a similar way as in Figure 26 these diagonal cells may also be pre-excluded by the specialised tables, whereas the original bin-lattice algorithm using the N1 norm would always include them. Thus, for query distances below one cell unit, the "sphere" intersection with its "bounding cube" may perhaps not be an adapted description anymore, but the principle remains the same.

Another obvious optimisation is to unconditionally process all objects when the query sphere covers the whole world: The brute-force method has reduced setup costs and would be more efficient in that case. A less obvious but still potentially interesting and related optimisation is possible in the case when the region of interest is non-cyclic. In that case when the query sphere extends outside the main region potentially many offsets are actually redundant pointers to the "outside" region. Once again the simpler bin-lattice with min-max along each dimension does not suffer from this problem, so a solution has to be found to maintain the new algorithm performance comparable or better than the simpler bin-lattice. An exhaustive solution considering all possible intersection possibilities between the sphere and the region of interest could perhaps be theoretically possible, by defining more specialised offset tables, but in practice the sheer amount of combinations make this solution intractable. A sub-optimal yet efficient solution is to consider the bounding parallelepiped for the intersection of the query sphere and the main region, which is what the bin-lattice actually does. This parallelepiped contains cells outside range, as the N1 cube does, but nevertheless, its volume may be lower than the total sphere volume due to part of the sphere being outside the region of interest. Therefore comparing the volume of the sphere with the volume of the parallelepiped allows to choose at run-time with a simple test which of these

methods is the most efficient. The worse case is the reversion to the bin-lattice behaviour, the parallelepiped, and so the new algorithm remains within the bin-lattice performance. Hopefully there are cases where the sphere falls outside the main region by only a few cells and thus running through the sphere still brings in less cells than running through the parallelepiped. In these cases the new algorithm thus once again considers less cells than the bin-lattice one.

Finally another optimisation is covered in [BROD06B]: Maintaining a list of non-empty cells. As for the parallelepiped, when the size of this non-empty list is lower than the size of the sphere, then it may be more interesting to run through the list than through the sphere.

In any case the amortised costs for processing a single agent depend on the chosen method: indexed offsets of the sphere cells, non-empty list, brute-force, or bin-lattice. For example in the case of the non-empty list we are trading a run-time test for rejection because a cell is empty (when using the sphere) by a run-time test for rejection because the cell is too far (when using the list). The bin-lattice method clips the query to the region of interest by construction, whereas the sphere method must handle the "outside" case for each processed cell in non-cyclic worlds. And so on for other run-time costs. Comparing the respective volumes processed by each technique is in these conditions not the fairest criterion for selecting the best method at run-time: For example the brute-force consideration of all objects has basically no overhead, so it may be more efficient even when the sphere does not completely cover the whole region of interest. The reference implementation (see Section 5.3.5) thus introduces user-definable weighting factors for selecting each method: The volumes are still used as the basis for the comparison (more exactly, the number of memory access) but they are first scaled up or down by the weighting

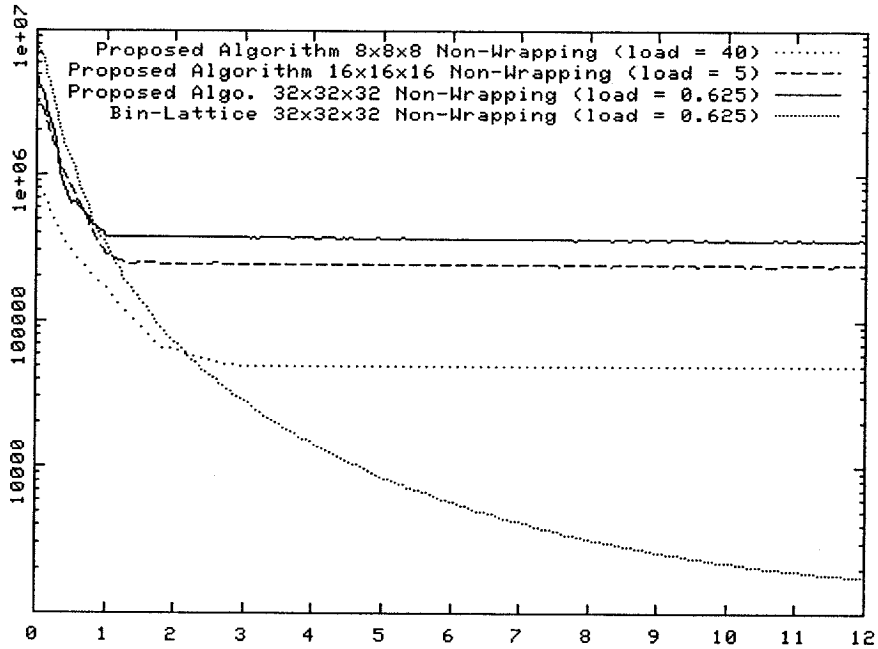
factors. Thanks to this final trick the user may get the best out of the algorithm, by tuning these factors according to preliminary benchmarks for representative cases of a real-world application.

5.3.4 K-Nearest neighbours and benchmarking

The previous sections deal with the finding of all objects within a given range, but typically some applications will only use the K nearest objects.

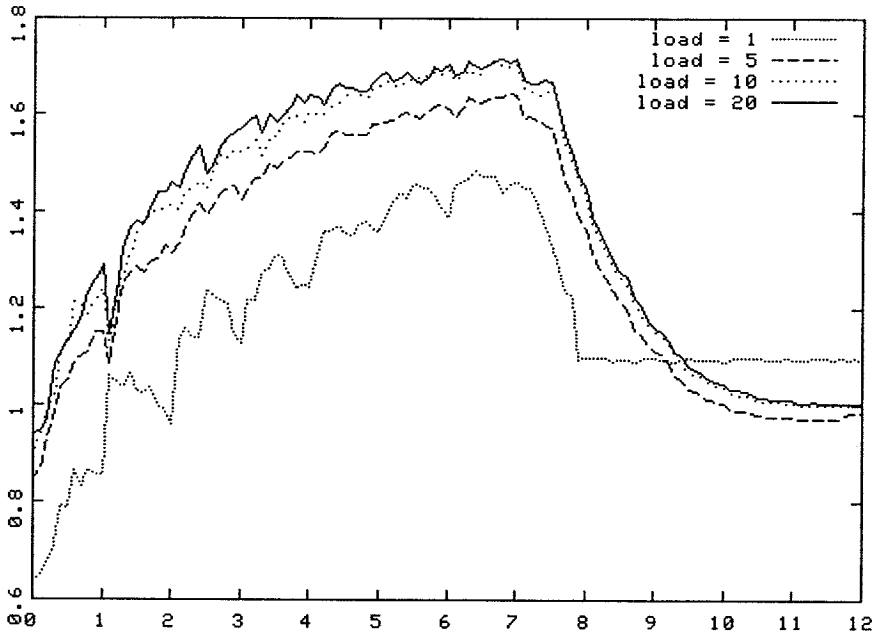
In this case, running through the distance-indexed offsets from query centre to sphere edge presents an advantage: The ability to prematurely stop when the K objects have been found, without running through the whole volume of the sphere. This is a possibility that is not present in the other methods described in the previous section (non-empty list, etc.) and which is also a new feature compared to the original bin-lattice algorithm running through the N1 cube.

Figure 27 presents the behaviour of the new algorithm for solving the K-nearest neighbours problem in different situations, for the same total number of objects. The discretisation size is important due to differences in the run-time costs: processing more cells with less agents in each cell, or processing less cells with more agents in each cell. A bin-lattice implementation is also presented and can be compared to the algorithm best case.



Legend: Number of queries vs query distance in cell units
Figure 27: K-Nearest neighbours performances comparison

The premature stopping capability of the new algorithm is clearly visible on this graph, with a constant performance as the query distance grows. Figure 27 also presents typical behaviours of the new algorithm that are detailed in [BROD06B]: Sensitivity to the load average and to the query distance. In particular for small distances below one cell the increased run-time costs of the new algorithm do not always compensate the simplicity of the bin-lattice one. Yet even this fact needs careful consideration, and some situations exhibit clear gains even for distances below one cell, due to the fact the new algorithm may reject some of the cells that are considered by the bin-lattice algorithm as previously explained. This is apparent on Figure 28, especially for high load averages.



Legend:
Horizontal axis: query distance in cell units
Vertical axis: Performance ratio between the new algorithm and the bin-lattice one.
Figure 28: All-neighbours queries performance ratio over bin-lattice

Figure 28 measures the performance gain for the new algorithm over the bin-lattice one, in a 16×16×16 cyclic world: A ratio value x in this plot means the new algorithm runs at x times the speed of the bin-lattice one.

The cyclic nature of the world is apparent Figure 28 when the query distance approaches half the world size: since it is cyclic, the sphere wraps around unconditionally at (16-1)/2 = 7.5 cells and the remaining uncovered portion of the world decreases rapidly. Figure 28 also exhibits the effect of the aforementioned optimisations that are not applicable to the k-nearest-neighbours case, like the non-empty list optimisation that is apparent for the low load average of 1 object per cell.

Finally, these benchmarks and [BROD06B] do not tune the weighting factors previously mentioned. A real-world application would benefit from this feature, for example by using the possibility to revert to the bin-lattice algorithm for small

distances: Again, even for distances below one cell unit the new algorithm may be faster so the user is invited to check whether reverting to bin-lattice for her/his particular application is relevant or not. For K-nearest neighbours in particular tuning parameters allows to get the best of the reference implementation.

5.3.5 Conclusion

While it is not directly linked to the main issue of analysing complex systems this new algorithm addresses a common issue for many simulations in the domain: Finding the neighbours of mobile agents.

The main strength of this new algorithm is the reduced, near-optimal, number of cells that are considered during the neighbourhood query. This makes it a particularly well-adapted choice for situations with high load averages, where the cost of processing a cell is maximal. But this also makes the algorithm well-adapted compared to the bin-lattice one for processing long-distances queries, for example when all objects within line of sight should be returned (for example when messages are passed between agents). Another strength is the ability to prematurely stop as soon as the K nearest neighbours are found, if K was specified as the maximal number of neighbours to return.

The main weakness of this algorithm is its increased complexity, which manifests itself mainly through additional run-time costs at various stages of the algorithm. This means that despite the new algorithm processing less cells than the bin-lattice one, it may still be slower on occasion. Hopefully there is always the possibility to revert to the bin-lattice in these worst cases, which is handled by the reference implementation.

The reference implementation is available on my Web site[39]. It is provided in C++ with all the tuning parameters that are introduced in the previous sections. The new algorithm and this implementation are not as simple as the Bin-Lattice one, but may bring non-negligible performance gains (for example up to 1.7 times faster in Figure 28). Moreover the reference implementation is readily usable with a simple interface, so all the complexity of the algorithm is hidden from the user application.

This new algorithm thus represents a valuable contribution in itself.

39 Available at http://nicolas.brodu.free.fr/en/programmation/neighand/index.html

5.4 Conclusion

This chapter has presented three algorithms that were created in the course of investigating complex systems.

All three algorithms presented in this chapter represent an advance in their own domain. The first brings the notion of statistical complexity to systems that are monitored on-line, by allowing for continuous updates of the complexity estimations. This is especially useful for systems where these complexity values are expected to evolve over time, so monitoring this evolution can serve for validating or not ideas from Complexity Theory. Chapter 4 has provided a usage example for this algorithm in a context where the "Edge of Chaos" hypothesis is tested predictively.

The second algorithm also brings incremental monitoring capabilities to experiments. The multifractal spectrum that is evaluated reports the regularity properties of a time series, and as mentioned in Chapter 4 this technique has been applied already to many domains: biology [IAGHRSS99], finance [MBF97], environmental research [TLMWD00], signal and image processing [LV07], physics [MBA93], etc. The ability to check efficiently how this spectrum evolves over time thus has a wide possible range of applications. The algorithm presented in Section 5.2 has been explicitly designed to be generic and standalone, so it is not restricted to the experiment presented in Section 4.2.3 and it may be reused directly in other applications.

The third algorithm also addresses a generic problem with a great re-usability potential. The original motivation for performing efficient neighbourhood queries in a multi-agent context is directly transposable in any problem where mobile objects evolve in a multi-dimensional space. Though the reference implementation only deals with three dimensions the method proposed in 5.3 can be extended to other

dimensions as well. As a directly usable toolbox the implementation proposes a variety of cases to handle cyclic and non-cyclic regions of interest, and does not impose conditions on the type of the objects that are the subject of the queries. Though not directly related to the main emergence and complexity issues this algorithm may thus prove useful in many other real-time simulations involving complex interactions between agents.

The main argument of this dissertation is that understanding complex systems at this point goes through the predictive testing of the ideas from complexity theory. Yet, in order to realise practical experiments so as to confront these ideas to reality, there is a need for investigation techniques that can monitor some aspect of the system during its evolution. The first two algorithms presented in this chapter fall in this category. As noted by Thomas S. Kuhn [KUHN62], in the history of science the means that are necessary to realise some experiments – the tools and techniques that need to be put in place – contribute themselves to the advances in the corresponding domains. In the specific field of Computer Science these tools take the form of new algorithms and techniques, solving problems that were previously not always easy to implement or to run in a reasonable time. While the tools presented in this chapter do not bring great theoretical advances, they nevertheless constitute useful practical extensions to the general state of art.

Chapter 6: Conclusion and proposals

The only way of discovering the limits of the possible is to venture a little way past them into the impossible.

Any sufficiently advanced technology is indistinguishable from magic.

The truth, as always, will be far stranger.

Sir Arthur Charles Clarke

This chapter summarises and recapitulates in Section 6.1 the major contributions of this work, both theoretical and practical, and it offers a personal view on these findings. Possible research that would extend well this work is then presented in Section 6.2, as well as a conclusion for this dissertation.

6.1 Contributions and retrospective

6.1.1 Theoretical

The theoretical contributions of this work come from:

- The direct analysis of reductionism in simulations, presented in Sections 2.3 and 2.4.

- The view offered on the Edge of Chaos hypothesis as the result of the experiments in Section 4.2.7.

- The demonstration of an effective weak form of downward causation in Section 4.1.

Retrospectively I still think that the reductionism vs functionalism debate is not a particularly fruitful one. I back this claim up by the facts in Sections 2.3 and 2.4:

1. Even formal systems may produce phenomena that are better described at a higher level by more concise and not necessarily provable laws of operation.

2. Purely functionalist arguments form a self-referential loop in the higher-level system (see Section 2.2.3) and therefore a debate as to their nature is better investigated by the dynamics of feedback loops and causality networks, for example, than by trying to reduce them to a lower-level of operation.

3. Only a reductionist approach, whatever the "level" of the observations, can offer quantitative predictions. Indeed, a necessary condition to produce numbers is to make use of a formal system in which they may be interpreted, hence a form of reduction.

Hence both aspects are intrinsically necessary for the comprehension of complex systems: The functions that are identified may provide informal explanations in a first time, but as soon as a deeper understanding is sought in this particular complex system under investigation, formalising these functionally defined behaviours is necessary so as to get quantitative feedback. In turn this forms another "level" of laws to which the observed phenomena may be "reduced" to. Then, as was proposed by Jochen Fromm [FROM06] in particular, the only methodology we know for investigating complex systems that has consistently proved efficient is the scientific one, which applied in this context translates to considering this system as a micro-universe in itself, with its own laws of operation, etc., so as to formalise theories within this micro-world, test them, refine them and so on. Such an approach certainly seems to be successful already for some systems like the Game of Life, where a community of enthusiasts has created a bestiary of

existing entities, what's known about interaction rules, and more[40]. In a sense, the discovery of Turing equivalence for the game of life could not have been possible without first thinking in terms of entities like the gliders and their interactions. So, both functionalism and reductionism are in my opinion necessary to make any progress on complex systems. Which extends my previous remark, that fighting over whether a phenomenon is reducible or not is probably irrelevant for practical purposes.

The second point in this summary concerns the Edge of Chaos hypothesis. The feedback I got from the experiments in Chapter 4 go against a "strong" form of this notion. In this particular case at least, there is no global "critical line" for the system where it gains all kinds of desirable properties. Rather, what this experiment suggests is that each quantitative view we get on the system might peak at some maximum between generally "ordered" and generally "disordered" regions, whatever that means for this particular indicator, but:

1. There is no guarantee that each indicator peaks in a single maximum, unless proven otherwise for that indicator.

2. At least in the present example the maxima in training set classification performance (separation) and in another indicator (complexity) do not peak together.

Hence I propose the "weak" form of the Edge of Chaos hypothesis, which states that: "There exist quantitative indicators which peak in regions between ordered and disordered state spaces, but these indicators do not necessarily peak together". In particular, performance, computational power, length of transients, etc., are not necessarily correlated and neither they are with other quantitative indicators of

40 See [ABBO06] from which this example is cited and also http://entropymine.com/jason/life/ for further investigations.

order and chaos. The experiment in Chapter 4 backs this claim, in the sense that I've exhibited different indicators, both peaking between order and chaos, but with different maxima and opposite evolution while measured on the same system. That this system evolves toward a critical region is supported by the fact the multifractal learning rule works, which can hardly be interpreted outside the Edge of Chaos hypothesis by design. That this rule precisely reduces the complexity of the system proves there is at least a quantitative indicator that peaks outside the critical region associated with the multifractal rule. Hence, I conclude on this topic by proposing that general "order" and "chaos" considerations are useful, but that the general notion of an edge in between should be revisited. In particular, an extension to this work would be to identify what aspects (informational, functional, shape of relation graph, mathematical form, etc.) there are in common to the indicators that usually peak together, hence forming "critical islands". This would require a large study over multiple systems, interdisciplinary, in which a maximum number of indicators would be applied in practice. Then, analysis of this study would be necessary to extract the commonalities of the correlated indicators, possibly ending up in better notions for formalising the Edge of Chaos hypothesis.

The third point mentioned at the beginning of this section is the demonstration of an effective weak form of downward causation, or at least of downward causal explanation (see Section 2.2.4). The system presented in Section 4.1 was successfully directed toward a goal that is defined using notions, like the number of population cycles, that are defined neither at the spatial nor at the time scale of the individual elements (the agents) comprising the system. Yet the modifications performed on the system to reach that goal directly constrain the possibilities of the agents, through for example the self-sustaining cost or the energy that is fed into the system. This is thus a case where low-level elements possibilities were

constrained by the use of a higher-level notion. Moreover, the downward causation was used predictively: The modifications were performed in order to reach an objective, and the system measurably reacted as planned. Thus, there is at least of weak form of downward causation that may be used for practical purposes, and Section 4.1 proposes a method for doing so.

6.1.2 Practical

The practical contributions of this work come from:

- The establishment of a new algorithm for estimating the Statistical Complexity of a system incrementally, updated as data values are added or discarded. This is especially helpful for slowly varying systems when the complexity values evolve over time.

- The creation of an algorithm for computing on-line the multifractal spectrum of a time series, giving the latest up-to-date estimate in constant time as data values are fed incrementally. The new multifractal algorithm also offers more precision compared to similar methods based on wavelet decomposition, due to the fact it considers and can average over all possible dyadic frames of decomposition of the data, as is explained in Section 5.2.

- The contribution of a valuable algorithm for finding the mobile neighbours of agents in a multidimensional environment, with application to both cyclic and non-cyclic three dimensional worlds. This algorithm is generic and may be reused in many contexts, and in particular for multi-agents simulations and for some graphics applications.

The statistical complexity algorithm is the most closely related to the general theme of complex systems amongst the three algorithms. The main challenge in

creating this algorithm lied in its implementation, which had to remain efficient despite frequent re-clustering: The writing of a specialised incomplete gamma function implementation for the needs of this algorithm is only one part of the solution, there is also locality of reference to consider, etc. Closely behind in terms of challenge was to ensure that the algorithm gives meaningful results as soon as possible, which lead to the convergence considerations presented in Section 5.1 about the need for merging clusters. This feature was not present in the previous algorithm implementation by Shalizi *et al.* [SHRKM05]. Replicating that implementation was the first step and offered only moderate difficulty, thanks to the publicly available source code they provided (in Object Caml) that complements the pseudo-code from [SHRKM05].

This leads me to emphasise the importance of providing the source code for the experiments and the algorithms as Free Software, at least for research purposes, and in the cases where an algorithm implementation is possible in addition to pseudo-code[41]. Free software offers the recipient the liberty to use, modify, distribute, and distribute modifications as well, and the FSF[42] licenses in particular ensure that these freedoms will remain in future derived works. Depriving a researcher of any of these liberties is contrary to the Science part in "Computer Science": Without use there is no experimental replication possible, without modification there is no possibility for an incremental research methodology, and without freedom to distribute both original or modified versions there is no possibility for independent research[43].

41 Theoretical works where no implementation is currently possible obviously cannot fall in this category.
42 See the Free Software Fundation site at http://www.fsf.org for more details about what freedoms I mean by Free software.
43 The risk would otherwise be that replication is allowed only for people selected by the original researcher, with the inevitable interference on integrity.

The second point answers basically the same needs as the first – incremental updates – but in a different context. Multifractal spectrum estimation can be done with different algorithms (see Section 5.2), and depending on the context some are better adapted than others. However discrete wavelet transforms are fast, and the precision obtained [BROD05B] with the incremental version of the algorithm that is presented in Section 5.2 is both sufficient for the needs of the experiment in Section 4.2, and finer than that of the previous algorithm using wavelets by Manimaran *et al.* [MPP05] in any case. As before the algorithm that I present in Section 5.2 is publicly available (which is not the case for [MPP05]) as Free Software, and directly applicable in a user application as it is self-contained.

Finally the last algorithm is also a useful contribution in its own as it answers a recurrent problem in a new and efficient way. The tight bound for the query sphere for finding the nearest neighbours of mobile objects is nearly optimal at pre-computation time, and can be made so with an additional run-time test as shown in Section 5.3. It is adapted to such situations as the experiment in Section 4.1 and the simulation framework proposed by Craig Reynolds [REYN99] from which part of that experiment was inspired. The new algorithm extends the bin-lattice method presented in [REYN00]. It is not so well suited however for the processing of large static data sets where tree-based techniques would probably be more efficient. However for dynamically updated data sets, like the position in space of mobile agents in a simulation, tree-based techniques become costly and my proposed algorithm works efficiently, generally more so than [REYN00]. The cases where the additional complexity is not worth its overhead can be easily reverted to the simpler bin-lattice method, which is clearly detailed in [BROD06B] and also together with the reference implementation. Hence, the new implementation answers the problem and lets the user tune the algorithm depending on the target application needs with

great flexibility.

6.2 Future work

6.2.1 Extensions to the projects presented in this dissertation

The methodology proposed in Chapter 3 is simply the application of concepts from Complexity Theory (or embryo of, if any), but in a predictive rather than purely descriptive way. This methodology has lead to the advances presented in the previous section, and there is no reason why it wouldn't work for other concepts and frameworks as well. In this section I thus comment on possible future extensions to the works in this dissertation, that would either complement or continue the projects that were presented.

One of the extensions that I think would enhance [BROD05A] is the introduction of both communication and artificial intelligence for the agents. This would be an interesting experiment in terms of artificial life, perhaps to be related to the more ambitious objectives of Gilbert *et al.* [GBBCal06]. In the present context the extension proposed above would offer an opportunity to try the formalisation/ prediction methodology for complex systems depicted in Section 6.1. Indeed, the analysis of an enhanced environment like the one suggested in Section 4.1 could go well beyond the definition of higher level effects like prey/predator population cycles. After all, the expressed desire from [BROD05A] for long-term stable population dynamics has not been yet reached. A full-scale attempt at identifying and formalising global entities and their relations in systems like an enhanced version of Section 4.1 would be well worth the effort. It would demonstrate that the methodology proposed by Jochen Fromm [FROM06] and which is mentioned for the

game of life example in Section 6.1 is generalisable to other complex systems as well.

The neural networks experiments conducted in Section 4.2 may be extended in two ways. The first is the design of new learning rules to check the limits for the learning rule creation methodology I proposed, and which is based on synchronisation and incompatible constraints. This would present an interest in the neural networks research domain, possibly some of these rules could produce even better results than the current ones. The second way of extending these experiments would be to introduce other quantitative indicators that peak in a region between order and chaos. Actually, the larger research trail mentioned in Section 6.1 of investigating which features are common to the indicators that peak together, could include the neural network experiments as one of the frameworks to consider.

On the algorithms side, the statistical complexity algorithm could be extended so it would also compute the transition probabilities between the causal states, hence forming the estimation for the complete ε-machine. This would make the algorithm closer to the one in Shalizi *et al.* [SSC03] while retaining both the light-cone perspective from Shalizi *et al.* [SHRKM05] and the general applicability of the implementation I propose to any user-defined type. What's necessary to make this work is to find an API for the user to specify in a simple form the transitions between light cones, so these can in turn be exploited for computing the transitions between their clusters (the estimated causal states).

A parallel version of the algorithm would be welcome as well. In the current version and in [SHRKM05] the observed pasts that are collected are processed in random order, but that's just a trick to damp out the fact we're processing them

sequentially whereas ideally they should be processed in parallel (no past has precedence on another[44]). Besides the immediate scaling gain that would result from a parallel implementation in a multi-CPU system, the parallel version would also more closely match the fact that the estimated clusters are not ordered. This parallel extension is of course not really useful in case of a completely incremental use, with observations fed one by one. However the concurrent feeding of new observations would certainly be welcome as well.

The multifractal spectrum estimation algorithm could be extended to support different wavelet families. In the current version, only wavelets for which the analysis and reconstruction filters are the same length are usable (such as the Daubechies wavelet family). However the incremental computation method described in Section 5.2 and [BROD05B] does not impose such restrictions theoretically and an implementation that allows other wavelet basis is certainly possible.

The neighbourhood finding algorithm was specifically implemented for three-dimensional worlds, both cyclic and non-cyclic. However the algorithm that is presented in Section 5.3 is applicable to any dimension. As the ratio between the query sphere and its bounding cube decreases exponentially with dimension, this algorithm could potentially bring corresponding gains. Its main limitation in higher dimensions would be memory consumption, necessary to store all the indexed tables. Yet, this memory/cpu consumption trade-off is a recurrent theme in computing, so the new algorithm may very well propose one side of this compromise where necessary. Thus, extension to higher dimensions is a possible future work

44 This is not incompatible with the fact that observations are, in fact, collected sequentially. The same past cones are observed multiple time in this sequence, with different futures, which is the basis of building the future cone distributions. The estimated causal states – the clusters of past cones – are not ordered.

that would complement well the current implementation. Another improvement to this algorithm would be the use of vector instructions for the distance computations, which could possibly lead to non-negligible performance gains.

6.2.2 Extensions to this dissertation and final note

More generally all the present work could be extended to test other notions than the Edge of Chaos and downward causation in a predictive way. A candidate that I have considered is the concept of feedback loops, but the corresponding experiments are not yet finalised and I consider them for future extensions to this work for this reason. Moreover, feedback loops is a very general concept that would require a dedicated multi-disciplinary approach in itself. As for the Edge of Chaos indicators broad study I proposed above, there would be a need for quantifying some state property of multiple systems and monitoring the effect of feedback loops predictively.

In any case, the practical investigation of complex systems is necessary if we want to progress on defining the applicability limits of the main notions related to the emergence issue. The tools we devise to achieve this goal are not only necessary but also may prove worthy contributions in themselves. The predictive testing and incremental methodology that is at the heart of research in other disciplines is also well suited for the "complex" systems, which, by analogy to the citations at the heading of this chapter, would then become a full-fledged theory and perhaps not so "complex" any more.

Bibliography

[AB02] Réka Albert, Albert-László Barabási (2002), *Statistical mechanics of complex networks*. Review of Modern Physics 74(1):47-97.

[ABBO06] Russ Abbott (2006), *Emergence explained: Abstractions: Getting epiphenomena to do real work: Essays and Commentaries*. In Complexity, Volume 12, Issue 1.

[AGRE03] Philip E. Agre (2003), *Hierarchy and History in Simon's "Architecture of Complexity"*. Journal of the Learning Sciences 12(3)

[ASHB56] William Ross Ashby (1956), *An Introduction to Cybernetics*. Chapman & Hall, London.

[BEDA03] Mark Bedau (2003), *Downward causation and autonomy in weak emergence*. Principia 6: 5-50.

[BEER04] Randall D. Beer (2004), *Autopoiesis and Cognition in the Game of Life*. Artificial Life 10(3):309-326.

[BN04] Nils Bertschinger, Thomas Natschläger (2004). *Real-time computation at the edge of chaos in recurrent neural networks*. Neural Computation, 16(7):1413-1436.

[BS02] Hugues Bersini, Pierre Sener (2002), *The connections between the frustrated chaos and the intermittency chaos in small Hopfield networks*. In Neural Networks 15(10):1197-1204.

[BR03] Stefan Bornholdt, Torsten Röhl (2003), *Self-organized critical neural networks*. Physical Review E 67:066118.

[BROD04] Nicolas Brodu (2004), *Crogai project report: Using AI with steering behaviors to model crowds*. Unpublished, available at http://nicolas.brodu.free.fr (checked 05/03/07)

[BROD05A] Nicolas Brodu (2005), *Environmental fitness for sustained population*

dynamics. Proc. IEEE Congress on Evolutionary Computation (CEC) 1:343-350.

[BROD05B] Nicolas Brodu (2005), *Real-time update of multi-fractal analysis on dynamic time series using incremental discrete wavelet transforms*. Unpublished preprint, submitted for review to the Fractals journal.

[BROD06A] Nicolas Brodu (2006), *Learning using Dynamical Regime Identification and Synchronization*. Proc. IEEE International Joint Conference on Neural Networks (IJCNN) pp270- 276.

[BROD06B] Nicolas Brodu (2006), *Spherical Indexing for Neighborhood queries*. Unpublished preprint, conditionally accepted with some changes to the Journal of Graphic Tools.

[BROD07A] Nicolas Brodu (2007), *A framework for the emergence of intra-species mutual recognition patterns*. Unpublished preprint, a one-page abstract will be presented to the International ACM Genetic and Evolutionary Computation Conference (GECCO).

[BROD07B] Nicolas Brodu (2007), *Quantifying the effect of learning on recurrent spiking neurons*. Accepted for publication to the IEEE International Joint Conference on Neural Networks (IJCNN).

[CAGDal02] Jonathan M. Chase, Peter A. Abrams, James P. Grover, Sebastian Dieh, Peter Chesson, Robert D. Holt, Shane A. Richards, Roger M. Nisbet, Ted J. Case (2002). *The interaction between predation and competition: a review and synthesis*. Ecology Letters 5:302–315.

[CARI89] Peter Cariani (1989), *On the design of devices with emergent semantic functions*. PhD dissertation. http://homepage.mac.com/cariani/CarianiWebsite/Cariani89.pdf

[CF03] James P. Crutchfield, David P. Feldman (2003), *Regularities Unseen, Randomness Observed: Levels of Entropy Convergence*. Chaos, 15: 25-54

[CHAI74] Gregory Chaitin (1974), *Information-Theoretic Limitations of Formal Systems.* Journal of the ACM, 21(3):403-424.

[CHAI05] Gregory Chaitin (2005), *Meta Math! : The Quest for Omega.* Pantheon Books Ed.

[CORN02] Peter A. Corning (2002), *The re-emergence of "emergence": A venerable concept in search of a theory.* Complexity, 7(6):18-30.

[CRUT94] James P. Crutchfield (1994), *The Calculi of Emergence: Computation, Dynamics, and Induction.* Physica D 75:11-54.

[DUTA06] Philippe Dutarte (2006), *Les instruments de l'astronomie ancienne.* Vuibert Ed.

[EKS00] Claus Emmeche, Simo Køppe, Frederik Stjernfelt (2000) *Levels, Emergence, and Three Versions of Downward Causation.* In Downward Causation, Minds, Bodies and Matter, p13-34, Århus: Aarhus University Press.

[ES70] Eigen M. and P. Schuster (1979), *The Hypercycle: A principle of natural self-organization.* Springer Ed.

[ETXE04] Arantza Etxeberria (2004), *Autopoiesis and natural drift: genetic information, reproduction, and evolution revisited.* Artificial Life 10(3):347-360.

[FBF77] Jerome H. Friedman, Jon Louis Bentley, Raphael Ari Finkel (1977), *An algorithm for finding best matches in logarithmic expected time.* In ACM transactions of mathematical software 3(3):209-226.

[FG07] Fabio Boschetti, Randall Gray (2007), *A Turing test for Emergence.* In M. Prokopenko (ed.), Advances in Applied Self-organizing Systems, Springer-Verlag, London, UK, 2007.

[FISH03] Randall J. Fisher (2003), *General Purpose SIMD within a register: Parallel processing on consumer microprocessors.* PhD Dissertation.

[FRED90] Edward Fredkin (1990) *Digital Mechanics*. Physica D, 254-270 http://www.digitalphilosophy.org/dm_paper.htm

[FROM05] Jochen Fromm (2005), *Types and Forms of emergence*. Unpublished preprint, arXiv.org:nlin.AO/0506028

[FROM06] Jochen Fromm (2006), *On Engineering and Emergence*. Unpublished preprint, arXiv.org:nlin.AO/0601002

[GBBCal06] Nigel Gilbert, Matthijs den Besten, Akos Bontovics, Bart G.W. Craenen, Federico Divina, A.E. Eiben, Robert Griffioen, Gyorgy Hévízi, Andras Lõrincz, Ben Paechter, Stephan Schuster, Martijn C. Schut, Christian Tzolov, Paul Vogt, Lu Yang (2006), *Emerging Artificial Societies Through Learning*. Journal of Artificial Societies and Social Simulation 9(2).

[GCSYZ03] Peter Grogono, GuoRong Chen, JunFeng Song, Tao Yang, Lei Zhao (2003), *Laws and Life*, 7th IASTED Conference on Artificial Intelligence and Soft Computing (ASC).

[GERS07] Carlos Gershenson (2007), *Design and Control of Self-organizing Systems*. PhD Dissertation (currently as draft, 03 march 07).

[GK02] Wulfram Gerstner, Werner M. Kistler (2002), *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press.

[GLEI87] James Gleick (1987), *Chaos: Making a New Science*. Viking Penguin.

[GV06] Eric Goodman, Dan Ventura (2006), *Spatiotemporal pattern recognition via liquid state machines*. Proc. IEEE International Joint Conference on Neural Networks (IJCNN) pp 7579-7584.

[HEYL01] Francis Heylighen (2001), *The Science of Self-Organization and Adaptivity*. In: L. D. Kiel (ed.) Knowledge Management, Organizational Intelligence and Learning, and Complexity.

[HG07] Asbjørn Hróbjartsson, Peter C. Gøtzsche, (2007), *Powerful spin in the conclusion of Wampold et al.'s re-analysis of placebo versus no-treatment*

trials despite similar results as in original review. Journal of Clinical Psychology 63(4):373-377.

[HOLL98] John Holland (1998), *Emergence: From Chaos to Order.* Addison-Wesley Ed.

[HW01] Jef Huisman, Franz J. Weissing (2001), *Fundamental Unpredictability in Multispecies Competition.* The American Naturalist, 157(5).

[HW73] H. Haken and M. Wagner (1973) *Synergetics - Towards a New Discipline.* Cooperative Phenomena ed., Springer Verlag.

[IAGHal99] Plamen C. Ivanov, Luís A. Nunes Amaral, Ary L. Goldberger, Shlomo Havlin, Michael G. Rosenblum, Zbigniew Struzik, H. Eugene Stanley (1999), *Multifractality in healthy heartbeat dynamics.* Nature 399:461-465.

[IM98] Piotr Indyk, Rajeev Motwani (1998), *Approximate nearest neighbors: towards removing the curse of dimensionality.* Proceedings of the 30th ACM Symposium on the Theory of Computing, pp 604-613.

[JAEG01] Herbert Jaeger (2001), *Short term memory in echo state networks.* GMD Report 152, German National Research Center for Information Technology.

[JH04] Herbert Jaeger, Harald Haas (2004), *Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication.* In Science, April, pp. 78-80.

[JLM96] C.L. Jones, G.T. Lonergan, D.E. Mainwaring (1996), *Wavelet Packet Computation of the Hurst Exponent.* Journal of Physics A 29:2509-2527.

[KAUF93] Stuart A. Kauffman (1993), *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press.

[KAUF95] Stuart A. Kauffman (1995), *The Third Culture: Beyond the Scientific Revolution, Chap. 20.* By John Brockman, Simon & Schuster Ed.

[KAUF00] Stuart A. Kauffman (2000), *Investigations,* Oxford University Press.

[KIM99] Jaegwon Kim (1999), *Making Sense of Emergence.* Philosophical Studies

95:3-36.

[KIM06] Jaegwon Kim (2006), *Emergence: Core ideas and issues*. Synthese 151:547–559.

[KNIG75] Frank B. Knight (1975), *A Predictive View of Continuous Time Processes*. The annals of Probability, 3(4):573-596.

[KPN05] Alexander S. Klyubin, Daniel Polani, Chrystopher L. Nehaniv (2005), *Empowerment: A Universal Agent-Centric Measure of Control*. Proc. IEEE Congress on Evolutionary Computation (CEC) 1:128-135.

[KUBI03] Aleš Kubík (2003) *Toward a Formalization of Emergence*. Artificial Life 9: 41-65.

[KUHN62] Thomas S. Kuhn (1962), *The Structure of Scientific Revolutions*. University of Chicago Press. Reprinted in Champs, Flammarion.

[LANG90] Christopher G. Langton (1990), *Computation at the edge of chaos*. Physica D, 42.

[LAUG06] Robert B. Laughlin (2006), *A Different Universe: Reinventing Physics from the Bottom Down*. Basic Books ed.

[LEIB03] Gottfried Wilhelm von Leibniz (1703), *Explication de l'arithmétique binaire*. Des Sciences, pp85-89. Available online at Académie des sciences, Paris, ads-00104781.

[LM07A] Robert Legenstein and Wolfgang Maass (2007), *What makes a dynamical system computationally powerful?* In New Directions in Statistical Signal Processing: From Systems to Brain, pp127-154, MIT Press.

[LM07B] Robert Legenstein and Wolfgang Maass (2007), *Edge of chaos and prediction of computational performance for neural microcircuit models*. To appear in Neural Networks.

[LYH04] Yifan Li, Jiong Yang, Jiawei Han (2004), *Continuous K-nearest neighbor search for moving objects*. IEEE International conference on Scientific and

Statistical Database Management, pp 123-126.

[LV07] Evelyne Lutton, Jacques Lévy Véhel (2007), *Evolutionary multifractal signal/image denoising*. In Evolutionary Computer vision, EURASIP Book Series.

[MAND82] Benoit Mandelbrot (1982), *The Fractal Geometry of Nature*. W. H. Freeman & Co.

[MBA93] J. F. Muzy, E. Bacry, A. Arneodo (1993), *Multifractal formalism for fractal signals: The structure-function approach versus the wavelet-transform modulus-maxima method*. In Physical Review E 47(2):875-884.

[MBF97] Benoit Mandelbrot, Adlai Fisher, Laurent Calvet (1997), *A multifractal model of asset returns*. Cowles Fundation discussion paper number 1164.

[MHC93] Melanie Mitchell, Peter T. Hraber, James P. Crutchfield (1993), *Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations*. In Complex Systems, 7:89-130.

[MILO01] Edoardo Milotti (2001), *1/f noise: a pedagogical review*. Invited paper at the 2ndo. Encuentro del Grupo Latinoamericano de Emisión Acústica y 1ro. Iberoamericano, E-GLEA-2, Buenos Aires (Argentina). Preprint at arXiv.org:physics/0204033.

[MITZ04] Michael Mitzenmacher (2004), *A Brief History of Generative Models for Power Law and Lognormal Distributions*. Internet Mathematics 1(2):226-251.

[MN98] Makoto Matsumoto, Takuji Nishimura (1998), *Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator*. ACM Transactions on Modeling and Computer Simulation (TOMACS) 8(1):3-30.

[MNM02] Wolfgang Maass, Thomas Natschläger, Henry Markram (2002), *Real-time computing without stable states: A new framework for neural computation based on perturbations*, Neural Computation 14(11):2531-2560.

[MNM03] Wolfgang Maass, Thomas Natschläger, Henry Markram (2003), *Computational models for generic cortical microcircuits*. In Computational Neuroscience: A Comprehensive Approach, J. Feng, Ed. pp 575-605.

[MPP05] P. Manimaran, Prasanta K. Panigrahi, and Jitendra C. Parikh (2005), *Wavelet analysis and scaling properties of time series*. Physical Review E, 72:046120.

[MPP06] P. Manimaran, Prasanta K. Panigrahi, and Jitendra C. Parikh (2006), *Multiresolution analysis of fluctuations in non-stationary time series.* Unpublished preprint arXiv:nlin.CD/0601065v1.

[MULL04] Barry McMullin (2004), *Thirty Years of Computational Autopoiesis: A review*. In Artificial Life 10:277-295.

[NBL04] Thomas Natschläger, N. Bertschinger, R. Legenstein (2004), *At the edge of chaos: real-time computations and self-organized criticality in recurrent neural networks*. Proc. Advances in Neural Information Processing Systems.

[NP77] Grégoire Nicolis and Ilya Prigogine (1977), *Self-Organization in Nonequilibrium Systems*. John Wiley Ed. New York.

[NV06] David Norton, Dan Ventura (2006), *Preparing more effective Liquid State Machines using Hebbian learning*, IEEE International Joint Conference on Neural Networks (IJCNN) Proc. pp 8359-8364.

[OKD06] Paweł Oświęcimka, Jarosław Kwapień, Stanisław Drożdż (2006), *Wavelet versus Detrended Fluctuation Analysis of multifractal structures*. Physical Review E 74:016103.

[OTTI04] Julio M. Ottino (2004), *Engineering Complex Systems*. Nature 427:399.

[PATT95] Howard Pattee (1995) *Evolving self-reference: matter, symbols, and semantic closure*. Communication and Cognition - Artificial Intelligence, 12 (1-2), 9-28.

[PATT97] Howard Pattee (1997), *Causation, Control, and the Evolution of*

Complexity. Downward Causation. Ed. P.B. Anderson, P.V Christiansen, C. Emmeche, and N.O. Finnemann. In Press (according to http://www.ssie.binghamton.edu/faculty_pattee.html, 04/03/2007).

[PBR06] Mikhail Prokopenko, Fabio Boschetti, Alex J. Ryan (2006), *An Information-Theoretic Primer on Complexity, Self-Organisation and Emergence*. Unpublished, submitted to Advances in Complex Systems.

[PRAT94] Terry Pratchett (1994), *Interesting Times*. Victor Gollancz publisher.

[PREC94] Lutz Prechelt (1994), *Proben1, a set of neural network benchmark problems and benchmarking rules*. Technical Report 21/94, Fakultät für Informatik, Karlsruhe University.

[PTJ02] Elizaveta Pachepsky, Tim Taylor, Stephen Jones (2002), *Mutualism promotes diversity and stability in a simple artificial ecosystem*. Artificial Life 8:5-24.

[REYN99] Craig Reynolds (1999), *Steering Behaviors For Autonomous Characters*. Proc. Game Developers Conference, pp 763-782.

[REYN00] Craig Reynolds (2000), *Interaction with Groups of Autonomous Characters*, Game Developers Conference, pp 449-460.

[ROSE98] Robert Rosen (1998), *Essays on Life Itself*. Columbia University Press, New York.

[RSC99] Edmund M.A. Ronald, Moshe Sipper, Mathieu S. Capcarrère (1999), *Design, observation, surprise! A test of emergence*. Artificial Life 5(3):225-239.

[RYNA06] Alex J. Ryna (2006), *Emergence is coupled to scope, not level*. Unpublished preprint, arXiv.org:nlin.AO/0609011.

[RV97] Rudolf H. Riedi, Jacques Lévy Véhel (1997), *Multifractal Properties of TCP traffic: a numerical study*. INRIA research report 3129.

[SAME95] Hanan Samet (1995), *Spatial data structures*. Modern Database Systems: The Object Model, Interoperability, and Beyond, W. Kim, Ed., Addison-

Wesley/ACM Press, pp 361-385.

[SAS05] Jagan Sankaranarayanan, Houman Alborzi, Hanan Samet (2005), *Efficient Query Processing on Spatial Networks*. 13th ACM International Symposium on Advances in Geographic Information Systems.

[SHAL01] Cosma Rohilla Shalizi (2001), *Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata*. PhD dissertation.

[SHAL03] Cosma Rohilla Shalizi (2003), *Optimal Nonlinear Prediction of Random Fields on Networks*. Discrete Mathematics and Theoretical Computer Science, AB(DMCS):11-30.

[SHAL04] Cosma Rohilla Shalizi (2004), *The Backwards Arrow of Time of the Coherently Bayesian Statistical Mechanic*. Unpublished, arXiv.org:cond-mat/0410063

[SHAL06] Cosma Rohilla Shalizi (2006), *Methods and techniques of complex systems science: An overview*. Chapter 1, pp. 33-114, Thomas S. Deisboeck and J. Yasha Kresh, Complex Systems Science in Biomedicine, in press.

[SHRKM05] C. R. Shalizi, R. Haslinger, J-B. Rouquier, K. L. Klinkner, C. Moore (2005), *Automatic Filters for the Detection of Coherent Structure in Spatiotemporal Systems*. Physical Review E, 73(3):036104.

[SIMO69] Herb A. Simon (1969), *The architecture of complexity*. In The Sciences of the Artificial pp 192-229. Cambridge, MIT Press.

[SMA00] Sen Song, Kenneth D. Miller, Larry F. Abbott, *Competitive Hebbian learning through spike-timing-dependent synaptic plasticity*. Nature Neuroscience 3(9):919-926.

[SS95] John Maynard Smith and Eörs Szathmáry (1995), *The Major Transitions in Evolution*. Oxford University Press

[SSE00] Mark Shackleton, Rob Shipman, Marc Ebner (2000), *An investigation of redundant genotype-phenotype mappings and their role in evolutionary*

search. Proc. IEEE international Congress on Evolutionary Computation 1:493-500.

[SSC03] Cosma Rohilla Shalizi, Kristina Lisa Shalizi, James P. Crutchfield (2003), *An Algorithm for Pattern Discovery in Time Series*. Journal of Machine Learning Research 7.

[SS04] Cosma Rohilla Shalizi, Kristina Lisa Shalizi (2004), *Blind construction of optimal nonlinear recursive predictors for discrete sequences*, Proceedings of the 20th ACM International conference on uncertainty in artificial intelligence pp504-511.

[TLMWD00] I. Tchiguirinskaia, S. Lu, F. J. Molz, T. M. Williams, D. Lavallée (2000), *Multifractal versus monofractal analysis of wetland topography*. In Stochastic Environmental Research and Risk Assessment 14:8-32.

[TURI36] A. M. Turing (1936), *On computable numbers, with an application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society, 2(42):230-265. *A correction*. Proceedings of the London Mathematical Society, 2(43):544-546.

[VMU74] Francisco Varela, Humberto Maturana, R. Uribe (1974), *Autopoiesis: the organization of living systems, its characterization and a model*. Biosystems 5:187-196.

[WEIN02] Steven Weinberg (2002), *Is the universe a computer?* New York Review of Books 49(16), also available at http://www.nybooks.com/articles/15762.

[WIM07] Bruce E. Wampold, Zac E. Imel, Takuya Minami, (2007) *The story of placebo effects in medicine: Evidence in context*. Journal of Clinical Psychology 63(4):379-390.

[WUEN02] Andrew Wuensche (2002), *Basins of Attraction in Network Dynamics: A Conceptual Framework for Biomolecular Networks*". In Modularity in Development and Evolution, Ed. G.Schlosser and G.P.Wagner. Chicago

University Press 2004, pp 288-311.

[WOLF02] Stephen Wolfram (2002), *A New Kind of Science*. Wolfram Media Ed.

[ZUSE69] Konrad Zuse, Rechnender Raum, Friedrich Vieweg & Sohn, Braunschweig (1969) *Calculating Space (English Translation)*. MIT Technical Translation AZT-70-164-GEMIT, MIT (Proj. MAC), Cambridge, Mass. 02139, Feb. 1970.