CHALMERS

A Multiple-Landscape Model on Innovation

Master Thesis for the Erasmus Mundus Master in Complex Systems Science

ERICK MARTINS RATAMERO

Department of Energy & Environment
Physical Resource Theory
Chalmers University of Technology
Gothenburg, Sweden 2013

Master's Thesis 2013:1

Abstract

In this work, we present a new framework for innovation modelling. Deriving from models using fitness landscapes for that task, we extend this idea to using multiple landscapes to represent different facets and stylized facts about artifacts. By using different landscapes, we are able to keep modularity and ease of modelling, by representing different aspects by different landscapes and by combining them towards an evolutionary optimization approach. We propose a system with three different landscapes: environment, attribution and artifact, all of them existing not in a design space, but in a functional space. They are related, respectively, to how useful different tasks are, what do we expect from an artifact and how good that artifact is at performing different tasks. These three landscapes are, then, used to calculate a fitness value which is used as an input for an evolutionary algorithm. We have seen that this setup retains the basic evolutionary optimization approach from fitness landscape models, while also introducing new possibilities to be explored. It might capture some specific features like radical innovation better than those models, and it brings more flexibility and modularity into the realm of innovation modelling.

Acknowledgements

First, I would like to thank the support from Claes Andersson, my supervisor. It was great to work with the subject of innovation modelling, especially together with someone who could teach me so much about it.

I would also like to thank the support of Ph.D. candidate Petter Törnberg, for the explanations about some of the main concepts in this subject, and Prof. Kristian Lindgren, for gracefully agreeing to be my examiner.

Getting at this point in my life was not always easy, coming from where I came. It is something that I would never have succeeded in if it was not for the support from my family and friends, in every possible way. I will be always indebted to them.

Finally, many thanks to the European Commission, who awarded me an Erasmus Mundus Scholarship for the two years of my Master studies, without which this wonderful time getting to know the fascinating world of Complex Systems would not have been possible to me.

Erick Martins Ratamero, Gothenburg 01/7/13

Contents

1

Introduction

The influence of technological change in economics has been modelled and researched by evolutionary economists for quite a while [1] [2]. However, these works have not contributed to understanding the processes themselves to a satisfying extent. The recognition of innovation processes as complex phenomena and their treatment as such is relatively recent. By using tools from complexity theory, complex interaction structures, be it between components in a system or between agents in a network, can be modelled without incurring overparametrization.

As becomes clear, there is a myriad of possibilities to model innovation. From the group of models regarding innovation as a complex process, we can see a divide, as mentioned, between those focusing on firms and their interactions while adapting to changes in technology and those focusing on the essential qualities of the technologies [3]. This divide can also be seen as a differentiation between the cognition involved in the process and the artifacts[1] themselves. In this work, we will position ourselves more towards the latter, dealing with innovation in abstract terms without too much concern about the agents creating the artifacts.

Modelling innovation presents multiple challenges. This is a process that, by its own nature, deals with the unpredictable and the unforeseen. After all, it is the process through which new solutions for new or existing problems are generated. Its very essence is to change what is possible, to introduce previous impossible alternatives that could not be forecast. Therefore, it can be argued [4] that this is naturally a process of search and discovery.

[1] Throughout the text, the term ”artifact” is used. It is important to clarify that by ”artifact” we mean not only physical tools, but also non-physical sociotechnical systems, like work procedures or theoretical concepts, for example.

In fact, many of the existing models dealing with the nature of innovation use this approach of search and discovery. Modelling innovation as an evolutionary process, where there is a set of possible outcomes through which an agent walks towards the best, became very popular inside this field [5] [6] [7] [8] [9], because it intuitively captures one of the features we all recognize about innovation: it tends to improve things gradually.

By dealing with innovation as an evolutionary process, these models see it as a kind of optimization, where there is a set of possible designs, associated with numerical values representing how well these designs perform a certain task. We shall henceforth refer to this value as the fitness of a given design.

As a means of optimizing the choice from all possible outcomes, most of these models use some kind of evolutionary algorithm, a computational analogue to biological evolution. In this context, Darwinian phenotypes are represented by the design, whereas the fitness is a direct analogy to the biological concept of fitness. In general, the designs will be coded somehow to generate an equivalent to the Darwinian genotype. From there on, a process will be established to select and evolve genotypes which are associated with higher fitness values.

Of course, the use of evolutionary processes is necessary because innovation, when seen as an optimization process, is not trivial. A trivial solution could be solved by simple enumeration of all possible answers and posterior evaluation of each one of those; when dealing with all possible designs this is not possible, be it because of the computational cost associated with the evaluation step or simply due to the impossibility of enumerating all possible outcomes. Therefore, it is necessary to be more clever about it. The use of evolutionary processes supposes at least some degree of correlation between fitness values; we will assume this as a sensible assumption when dealing with innovation processes.

The mapping between the design space and the fitness values is commonly known as a fitness landscape. This is a concept borrowed from evolutionary biology, where it has been used to visualize the distribution of fitness values since Wright's seminal work in 1932 [10]. In general, high fitness values are associated with "good" solutions to the problem at hand, while low fitness values are mapped to "bad" solutions.

As with most evolutionary optimization problems, fitness landscapes in innovation modelling are generally smooth, and present many local maxima. This kind of landscape is known as a rugged fitness landscape. In this kind of landscape, optimization is normally achieved through a series of small improvements towards a local or global maximum, in a process of "hill climbing". Therefore, the evolutionary process occurs through a walk in the landscape, searching towards a maximum.

In summary, the field of modelling for innovation theory is quite vast and contains many different strains of work, concentrating on different scales, features and elements of this

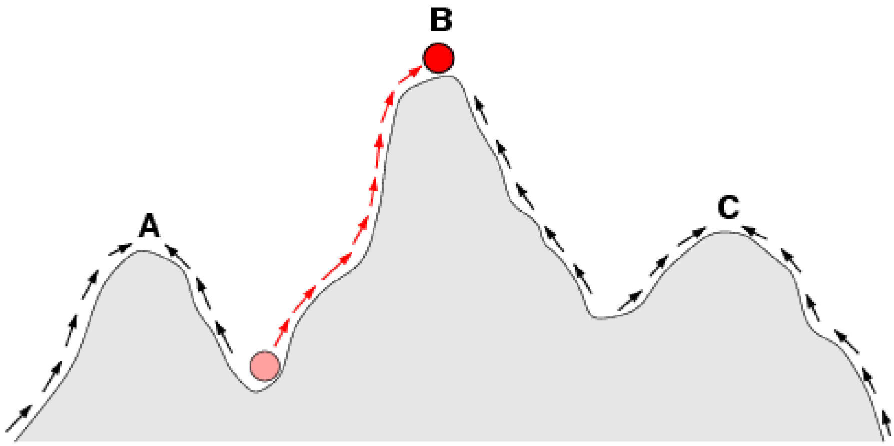

Figure 1.1: A two-dimensional fitness landscape. A one-dimensional design space is associated with fitness values, and the arrows indicate the directions of hill climbing towards maxima.

process. We are more concerned with modelling innovation in a micro level, not considering the interaction between agents in this field, or its social or economic influence, but dealing with it purely from an abstract dynamics level. In this area, evolutionary optimization is the tool of choice in most of the cases; the use of evolutionary algorithms pushes us towards fitness landscapes.

However, there are shortcomings with this approach; a fitness landscape associates design possibilities with a single fitness value, and for a single artifact that means we are evaluating its merits for one task only. We know from our day-to-day experience that some of those artifacts are suitable for many different tasks; we also know that a single one of them can evolve to become many different artifcats suited to different tasks (the wheel is a good example here), and that different solutions to a single problem can eventually merge and become one. These are all stylized facts from innovation theory that, apparently, are not compatible with a simple fitness landscape.

Furthermore, innovation not always follow a gradual hill-climbing path. Many are the instances in history when a completely different solution to an existing problem was devised, with very little relation to the existing ones [11]. To propose models that can deal with these shortcomings, we aim to introduce a different way to use evolutionary optimization to model innovation: using multiple landscapes. We believe that, by introducing multiple layers of abstraction to this approach and changing the paradigm of how to model artifacts and their evolution, we can gain enough flexibility to encompass these different facets of innovation theory in a single framework.

The structure of this work is as follows: in section 2, we present briefly the motivation of this work, and the shortcomings perceived in similar innovation models. In section 3, we present the general concepts of the introduced model, as well as some specifics about the particular implementation developed to illustrate it. In section 4, we present some of the features included in our proposed model, and show some results from simulations of our implementation to complement the discussion. We also discuss briefly some possible features that could be implemented using this framework. Finally, in section 5, we discuss the achieved results and the merits of the work, while also pointing possible directions for future research using what was obtained here.

2

Motivation

As discussed, other models aim at capturing the essence of the abstract process of innovation through an evolutionary optimization approach. They succeed at this task in a limited way. What they manage to do is to capture gradual improvement over a specific fitness landscape, associated with a certain task or function. However, this is hardly the whole spectrum of possibilities associated with the innovation process.

A central point missing from single-fitness models using design spaces is an account of radical innovation. By using a dynamic of gradual improvement and "hill climbing", these models represent very well the incremental nature of innovation, improving slowly existing artifacts. However, they fail at showing the instances where a completely new solution for a problem emerges. The absence of the possibility of having a model contemplating both facets of innovation is quite limiting.

Another extension that we might want to add to this gradual, single-fitness approach is task differentiation. While there are merits in working with artifacts having one single function to unveil basic dynamics of innovation, artifacts might in reality clearly be useful for more than one function. There is more to the innovation processes than what emerges from a model dealing with only one task, and any model not considering multiple functions will not capture whatever comes from this possibility. By definition, a fitness landscape associates a single fitness value to each point in design space, so there is no plurality of fitness values in any case, and no multiplicity of tasks.

Also, the possibility of modelling processes of exaptation is absent in the current models using fitness landscapes, exactly due to the focus given to a single task. Exaptation is a process originally developed in evolutionary biology[12], but with a much more general applicability that has only begun being explored in the context of innovation [13] [14]

[15] [16]. In this process, a certain artifact is initially evolved towards a certain use, but later ends up being coopted to another role. It is clear that, without working in an environment with multiple functions, this is an innovation mechanism that cannot be present in a model.

Furthermore, fitness landscape models fail to capture the differences between different artifacts in their full sense. By representing them as a single point in a design space associated with a single fitness value, these models do not consider the utility of each of those artifacts for different tasks other than its "main" attribution, and that utility might be a defining factor for the adoption of one particular design in detriment to another. In addition to that, no mention is made of the expectations and attributions that an artifact carries, from design and from its performance in different scenarios. Finally, previous models do not consider the utility of the task performed as part of fitness, since it makes no sense to do that when there is only one task being evaluated.

From these considerations, we see that, while reducing an artifact to a point in a specific fitness landscape might unveil some interesting results, it hardly tells the whole story. Again, we might want to look into cases like that of the wheel, where a simple, primeval design evolved into highly specialized tools for very different tasks. A fitness landscape might capture quite well the development of one of the specialized tools, but it does not tell us anything about the many other artifacts that spun off that simple wheel. There is something missing if we aim to model this complex dynamics of multiple specialization.

Of course, if we only use a design space as the determinant for fitness values, we miss a degree of freedom for different tasks, which seems clearly needed if we want to model different artifacts specializing and interacting in different functions, as well as the performance of a single artifact in different tasks and its consequences. Therefore, it is necessary to introduce another paradigm, a functional one, to make these more complex (and hopefully more diverse and complete, without loss of generality) models possible.

By changing the paradigm, we aim at capturing different aspects of the innovation process, while not losing the evolutionary optimization approach of the fitness landscapes. While we want to keep being able to model gradual innovation, it might be interesting to have also radical innovation, with sudden changes between maxima. Furthermore, this change of paradigm might enable the existence of populations of artifacts specializing in different directions and becoming good at different tasks, and eventually the appearance of different artifacts for executing similar tasks. With the simple change of paradigm to functional landscapes, we see a myriad of opportunities arising, without losing the basic functionalities and features of the existing work in innovation modelling.

Of course, with the new possibilities comes a completely new approach. The treatment of evolutionary optimization will have to be altered, since the artifact is not a point in a design landscape any more, but a landscape itself; it will have different fitness values for different tasks. Therefore, talking about ”hill climbing” no longer makes sense. Our particular approach will be highlighted in the following sections.

3

Our Model

In the previous two sections, we have briefly presented the role of modelling in innovation theory, some of the ways of dealing with this subject and an important approach to it. We have positioned this work together with other models that use evolutionary optimization to model abstract facets of innovation, and we have presented the common features of this class of models.

However, we have also shown that these similar models capture only some of the many possible features of innovation as a process. Therefore, we want to present an alternative for modelling inside the realm of innovation. Our proposed alternative can represent different stylized facts not previously contemplated in most works, while retaining the basic functionalities presented to us by the fitness landscape class of models.

3.1 General Concepts

As presented in section 2, we want this new class of models to have a different paradigm, a different possibility for associating fitness values: a functional landscape. This way, fitness values are not only tied to the specific point in the design space represented by the artifact, but also to a point in the functional space. Through the use of a functional paradigm, a single artifact has one value associated to each point in this functional landscape, that is, it has different aptitudes for different tasks.

By implementing this new idea, we have functional landscapes associated with each artifact. These will represent how good that particular artifact is at each task inside this space of functions. Which kind of landscape that would be and what kind of correlations exist in it depend, of course, on the particular implementation of this framework; intuitively, however, it seems reasonable to consider very similar fitness values to be obtained by a given artifact at similar tasks. it means, then, that a correlation structure is in place.

By using functional landscapes for artifacts, most of what was presented in Section 2 as missing in the previous work can already be modelled. However, there is no guidance to the process of evolution here; the relative importance of tasks is being overlooked, even if it plays a big role in the process of innovation. Of course, we tend to focus on solving our biggest problems, and that provides the guidance necessary to this evolutionary process.

To represent this bias towards more useful tasks, it is necessary to, somehow, give weights for different points in our functional space, prioritizing the most important functions and making the least important ones fade into the background. We are talking, here, about a kind of "mask" for our introduced landscape; that can be thought of as an additional landscape, to be layered together with the one presented before.

Therefore, we have a second landscape, representing the general fitness of each task; we will call it environment landscape, since it describes a general perception of relative utility that should be common for all artifacts. A change in this landscape is equivalent to a fundamental change in the intrinsic utilities of different tasks, and therefore, a change in the general environment. As with the first landscape, it is reasonable to think of very similar tasks having very similar utilities; this should be a guiding point for implementing this specific feature, indicating some correlation structure to exist.

However, evaluating an artifact's fitness over the whole functional space seems to make no sense, given that no one would expect it to be good at all possible functions. Again, we need to assign values to each point in the functional space, not indicating how good that certain artifact is for a task, but showing how suited it is expected to be at the function at hand. Once more, we can present this a landscape. Note that this is also tied to each artifact, as opposed to the environment one. We will call this an attribution landscape, since it represents how we attribute tasks to a certain artifact and expect it to perform at them.

In summary, we have three different landscapes: one associated with the artifact and its performance, one with the expectations related to that artifact and one indicating how important different tasks are. There is a fourth landscape implied here, associating the design space to the final fitness values. This landscape, however, depends on the other three from now on. As a single fitness value will finally be associated to an artifact at each point of the functional space, it is still possible to see the whole model as a single, high-dimensional, landscape; we have opted for simplicity and modularity when separating different factors into different entities.

As it is clear by now, the idea of "hill climbing" in its simplicity disappears in this framework. There is no more walk in a single landscape and a simple search for maxima.

An artifact is not represented by a single point any more, but by two separate landscapes. As a thought experiment, let us consider the problem of a single artifact with a fixed attribution in a fixed environment for a moment and analyse its implications in this new mindset.

Considering all landscapes to be normalized, maximum fitness will be obtained when the artifact landscape conforms exactly to the environment inside the area of attribution (i.e. where the attribution landscape is non-zero). That represents an artifact doing exactly what is expected from it, with more focus on important tasks. Then, our problem is not a walk in a landscape as in the previous models, but a change of landscapes, having this maximum fitness situation as final target.

In a scenario of evolutionary optimization, we will need to have artifact landscapes that can be mutated and evolved towards a maximum of fitness. For fixed attribution and environment, this would be our equivalent to a "hill climb". It is important to emphasize again that these are fundamentally different paradigms to represent the same stylized fact: the previous models represent optimization as a landscape walk in design space, and our framework does the same through a landscape change in functional space.

This is a general concept for modelling innovation; it is not concerned with specifics about the format of the landscapes, nor the evolutionary algorithm to be used, nor the dimensionality of the problem. The details are up to the implementation and, while some basic concepts should be kept, they should not affect the capability of modelling the features mentioned here.

The way in which we aim to model some of the features presented as missing in section 2 should be clearer by now. As an example, an artifact landscape that starts to optimize in two different areas of the functional landscape and that, at a certain point, becomes two different artifacts, with their own landscapes and with attributions split between them, is an example of how to implement specialization.

Finally, most of what was possible in simple fitness landscape models continues to be here. We have seen that a scenario without change in environment and attribution landscapes is equivalent to a simple "hill climbing" situation in previous works. Also, it is possible to have a relatively simple situation of a changing environment, something that was also possible in a fitness landscape model. It only means that the artifact would be climbing an ever-changing hill, instead of a static one. It is also possible to model competition between similar artifacts; it is feasible but not trivial to do that while using previous models, but here it is merely a matter of changing landscapes according to the other members of the population. It might even be interesting to introduce a competition landscape, with the negative (or positive) effects on fitness brought by fierce competition with the remaining artifacts.

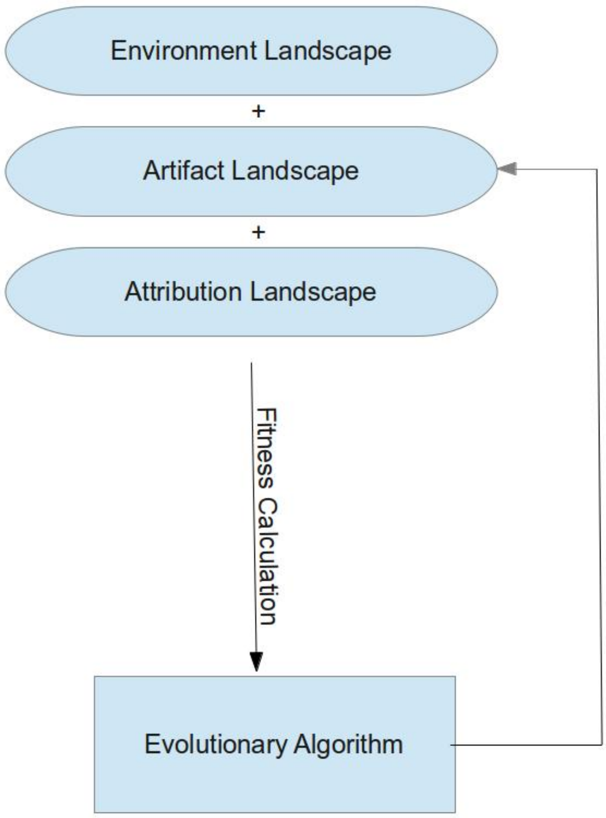

Figure 3.1: Through three layers of abstraction, we achieve a single fitness value for each artifact, while keeping conceptualization easy and good modularity between components of the model.

With that, we finish our presentation of the general concepts of this framework. We will now proceed to showing our sample implementation of this concept.

3.2 Particular Implementation

After we have established the baseline concepts for our modelling framework, it would be interesting to have a proper implementation of the ideas presented, as a proof of concept to show this paradigm can actually produce the features we wished to see and that it is a feasible set of principles for modelling innovation. Therefore, we have decided to prototype the basic ideas of this class of models and run some numerical simulations of it, to see the possible outcomes.

As this is no more than a proof of concept, some rigour was sacrificed for ease of visualization. All landscapes were implemented as three-dimensional, meaning we have a two-dimensional functional space where one-dimensional fitness values are mapped. This was a choice to make the results and features we get from the numerical results easier to see.

With the dimensionality of our landscapes defined, we need now to define how exactly to generate these landscapes. We would like for all of the landscapes to present some sort of correlation structure, so that similar points in the functional space have similar values for fitness.

It seems desirable that the function-fitting that will occur through the evolutionary algorithm does not attain a perfect solution. That would stall the algorithm and it would not bring anything new. It would be equivalent to having a fitness landscape with a single maximum: after some optimization, we would attain the global maximum and that would be the "end" of the algorithm. Instead, since we will basically be fitting the artifact landscape to the environment, these two landscapes will be generated through vastly different processes.

As a first process, we have taken a famous approach to the generation of rugged landscapes in innovation modelling: the NK-model. This is a model initially used to represent biological systems, as with many other ideas now used in innovation theory. The NK-model will be discussed in depth later on; for now, we will limit ourselves to saying that it generates rugged landscapes through epistatic relations, relying on correlations instead of analytical functions for that. We will use this model for generating our environment landscape.

Since the environmental landscape is not being generated through analytical functions, our artifact landscapes were defined as a set of $f : \mathbb{R}^2 \Rightarrow \mathbb{R}$ functions. However, as these functions will have to evolve through an evolutionary algorithm, it is necessary for them to have a genotype-like representation that can easily be mutated, at least. Thus, we will be using Postfix strings, also to be discussed in depth further in this work.

Finally, our attribution landscapes are not much more than simple masks, defining areas where to evaluate an artifact. As these attributions will have to be changed, divided and combined, they need to defined in a simple way, for ease of computation. Furthermore, they not only define an area, but also give different weights for different tasks. We have decided to have two-dimensional Gaussian functions filling these requirements. Our attribution landscapes, then, are of the form:

$$f(x,y) = \sum_{i=1}^{N} w_i e^{-(a_i(x-x_i)^2 + 2b_i(x-x_i)(y-y_i) + c_i(y-y_i)^2)} \tag{3.1}$$

The value N indicates how many different areas will be part of that artifacts' particular attribution. Then, the values x_i and y_i indicate the maximum value of each single area of attribution in the functional landscape; w_i gives different weights to the different areas, so that not all areas are equally important for a given artifact. Finally, a_i, b_i and c_i define the variance structure of the two-dimensional Gaussians.

As we are using normalization for all landscapes, an artifact with multiple attributions tends not to be as good in a single task as a specialized one with a single attribution. On the other hand, multiple-attribution artifacts can achieve rather high fitness values by specializing in many different areas at the same time, and it would be expected that, besides being prime targets for division of attributions, they would be rather more resilient to sudden changes in environmental conditions.

The final fitness is calculated by taking the difference between the artifact and the environment landscapes, weighted by the attribution. This way, letting $a(x,y)$ be our artifact landscape, $v(x,y)$ our environment landscape and the previously presented $f(x,y)$ the attribution landscape, our final fitness value F would be defined by

$$F = \iint (f.a - f.v)^2 dxdy \tag{3.2}$$

The fitness measure is, then, a weighted Euclidean-like distance between the artifact and environment landscapes, with the attribution acting as the weights.

This measure will be used in our evolutionary algorithm. There is little concern to have this algorithm as optimal or computationally efficient as possible, given that this implementation is mainly a proof of concept. With that in mind, a very simple approach was chosen. Instead of dealing with a complete genetic algorithm, our program generates a population of artifacts and, for each generation, mutates each member of the population a number of times. These mutations are evaluated, and the one with the highest fitness is passed on to the next generation.

In summary, our implementation uses NK-landscapes for environment, Postfix strings for artifact landscapes, two-dimensional Gaussian functions as attribution. It, then, calculates a weighted distance between environment and artifact, using the attribution as weights. Our algorithm generates an environment and a population of artifacts with their own landscapes, and at each generation mutates each member of the population a number of times, evaluates the mutations and, for each member, chooses the best-performing mutant to be passed to the next generation. It is a relatively simple approach to the basic concepts presented in the previous sections. Some more discussion on the way the NK-model and the Postfix strings were implemented follow.

3.2.1 NK-landscapes

The NK-model [17] for fitness landscapes is quite popular in innovation theory [7] [18]. This is a model for generating rugged landscapes based on the epistatic relations between elements. It can go from a completely decomposable architecture, with no epistasis, to a fully conected architecture, where each and every component affects all the others. This kind of landscape is constructed in such a way that all components are affected by the same number K of other elements.

The name of the model, therefore, comes from the simple way it is defined: there is a total of N elements, and each of them is affected by K others. a landscape and its relative ruggedness is defined, then, by only two values. Each point in an NK-landscape is represented by a string with N elements, and its fitness F is simply the mean fitness of its components w_i:

$$F = \frac{1}{N}\sum_{i=1}^{N} w_i \tag{3.3}$$

The fitness value of each w_i depends on the value of the string in the position i and in K other positions. For each possible combination of those values, a random number is assigned. For binary strings, a total of 2^{K+1} random values is, then, to be drawn. This is a strong argument for having binary strings; strings in bigger bases would quickly require too many random values to be drawn, even for moderate values of K.

Of course, a binary string of length N is well-suited to represent a N-dimensional hypercube, but not a two-dimensional plane. It is necessary, therefore, to have a suitable mapping of the two-dimensional coordinate values in functional space into binary strings that can be used in the NK-model.

Given a certain resolution for the quantization of functional space, it is, of course, possible to convert the quantized values of coordinates to binary strings. Then, a simple

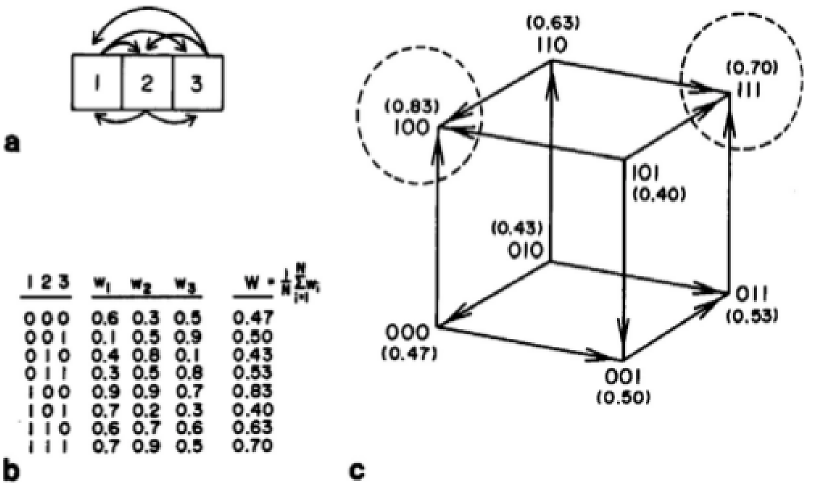

Figure 3.2: Example of a N=3, K=2 NK-landscape. Here, N=3 entails a three-dimensional cube, and K=2 entails the draw of 8 values. Figure from Frenken, 2006. [3]

concatenation of both coordinates would give us a unique binary string associated with each point. However, doing that introduces an unforeseen problem: adjacent points in the functional landscape might end up with very different strings associated to them. This mapping from two-dimensional plane to N-dimensional hypercube is not smooth.

The solution for this problem is to code the binary strings using Gray coding [19]. This is a code generated recursively: for generating the Gray code for n digits, we use the list of codes for $n-1$ digits, reverse it and concatenate with the original. Finally, we prefix the original list with a 0 and the reversed list with a 1. This code has the property of changing only one single bit between successive entries; it generates a much smoother mapping, and it was the chosen approach for binary coding in this work.

In summary, we will be generating an NK-landscape to represent our environment. This is done through converting our two-dimensional functional space to binary strings using Gray coding. By mapping our coordinate values to binary through this specific code and concatenating both coordinate values, we get an unique binary string, whose length N depends on the resolution of the quantization made in the space of functions.

Finally, the fitness values associated with each binary string are calculated. We start by drawing 2^{K+1} random values. These are generally drawn from the unit interval $[0, 1]$. Then, for each binary string, we use equation 3 to derive the fitness values. For each element w_i, one of the 2^{K+1} values is taken depending on the binary value of that element and its K epistatic relations. Finally, we take the mean of these values to obtain the final fitness of that string.

3.2.2 Postfix strings

For our artifact landscapes, we wanted to have a way to generate functions $f : \mathbb{R}^2 \Rightarrow \mathbb{R}$ from analytic expressions, as opposed to the epistatic approach of the NK-model. The objective was to avoid a trivial way for fitting one to the other perfectly and, therefore, to remove the existence of an absorbing state during the evolutionary optimization process.

However, one extra requirement for these functions is that they should be defined in a way an evolutionary algorithm could operate over, and by which optimization could easily occur through mutations in the functions. Therefore, it would be necessary for the functions to have some kind of genomic representation, where operators like mutation could be applied with a certain ease.

Of course, a trivial representation of analytic functions is the way we are used to representing them on writing, with mathematical operators marking the operations to be executed and parentheses marking the precedence relations. That would be the ideal in terms of ease of use and visualization. However, such expressions are not very easy to generate computationally, and mutating them is not trivial.

Then, it is necessary to look for an alternative way to present analytic functions that can be easily treated inside an evolutionary algorithm. That comes in the form of postfix strings. These are strings containing operators and operands, just like a "normal", infix string would. The difference comes in the notation, especially in the way the precedence of operators is shown.

An infix string, the kind we are normally accustomed to seeing, positions the operators between the operands (as, for example, in $2 + 2$). This notation presents some problems. The precedence of operators is not obvious without external rules for defining them, as it can be seen from the example $1 + 2 \times 3 + 4$, where different results are obtained from different orders of execution. Only by using an external rule (in this case, multiplication taking precedence over addition) we can solve it unambiguously.

On the other hand, a postfix string positions the operators immediately after their operands. With this notation, there is no ambiguity at all: precedence is position-based, not rule-based. Coming back to our $1 + 2 \times 3 + 4$ example, it would be solved as in the infix case (multiplication taking precedence) when written as $123 \times +4+$, and it would be solved as $(1 + 2) \times (3 + 4)$ if written as $12 + 34 + \times$, for example.

As we can see, creating, mutating and solving postfix expressions is quite simple in computational terms, since we do not need to worry about external indicators of precedence

5 **1 2 +** 4 * + 3 -
5 **3 4 *** + 3 -
5 12 + 3 -
17 3 -
14

Figure 3.3: Example of a step-by-step resolution of a simple postfix string, with scalar operands.

like parentheses. The expressions are simple concatenations of operands and operators, and they can always be solved unambiguously. As long as a single value is returned at the end and protected operators are used (preventing divisions by zero, for instance), these expressions are very simple to be treated.

Until now, we have only dealt with scalar operands. However, the basic idea of the expressions and how they are constructed, mutated and solved can be trivially extended for matrix operands. In fact, this is what is being done in this implementation: since our target is to have functions $f : \mathbb{R}^2 \Rightarrow \mathbb{R}$, we want to have not a single scalar value at the end, but a matrix with values for every point in a quantized $\mathbb{R}^2$ plane.

Artifact landscapes are, thus, generated as a postfix expression. These expressions will be of a given length, and will always be checked to be well-formed, with the correct number of operators to output a single result at the end. It will be a string composed of operands and operators, and this string will be mutated for the evolutionary algorithm and then solved to obtain values in the form of a matrix. This matrix gives us a value for each point in the $\mathbb{R}^2$ plane. These values will, finally, be combined with the environment and attribution landscapes to calculate a fitness value.

4

Results and Features

After presenting the general framework developed for innovation modelling, its differences to other models in this area and the details about the proof-of-concept implementation developed in the scope of this work, we now proceed to show the results of some brief simulations using our particular implementation. Again, these results are aimed at showcasing the possibilities contained inside this framework, and not an end in themselves.

Firstly, the basic feature of evolutionary optimization will be shown here. This framework retains this capability of evolving towards a maximum in a static situation where attributions and the environment do not change over time. While a very simple result, it is a basic building block for any model about innovation and, therefore, an important feature to retain in any framework proposing to be complete.

Then, we aim to present some of the features that are not obvious (or even not possible) in a pure, single-fitness landscape approach. These are basically results related to interactions between different artifacts and dynamics of generation and recombination of those elements. In our implementation, we introduced the possibilities of specialization, with a single artifact generating two separate ones, and generalization, with two separate artifacts being combined to generate a single one.

Finally, we will quickly discuss some of the results envisaged to be possible by using this framework, but that have not been implemented in our proof of concept. These are more complex dynamics, dealing with dynamic environments, radical innovation and innovation cascades, for example.

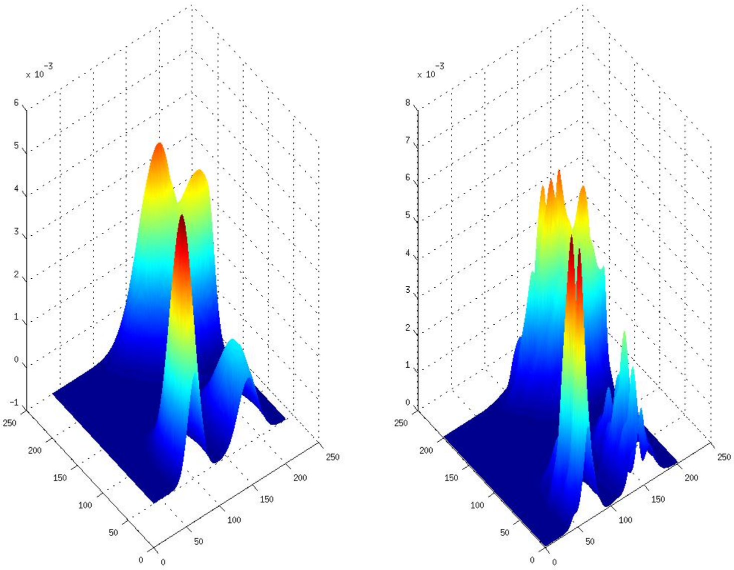

Figure 4.1: Weighted artifact landscape adapting to weighted environment landscape.

4.1 Evolution and adaptation

As presented in section 2, most fitness landscape models rely on a basic dynamic of "hill-climbing" to represent evolutionary optimization. There is a walk in the design space, always trying to reach a local or global optimum of fitness. This is a very simple representation of gradual innovation, but an useful one in most situations. It is, thus, necessary for any framework in innovation modelling to retain this basic dynamic.

The way "hill-climbing" is obtained in our framework was briefly discussed: for a static environment and attribution, a single artifact will be able to wander through its design space (that space being all the possible combinations of operands and operators for postfix strings) towards maxima of fitness; alternatively, this can be seen as the function represented by a postfix string fitting more and more the environment inside the attribution areas.

On Figure 4.1 we can see an example of a real simulation run with our implementation. On the left, we have an artifact landscape already weighted by its attribution, and on the right we have the general environment landscape weighted by that artifact's attribution. The more these two plots look similar, the higher the fitness of that artifact is.

In this figure, we have the result of a simulation of one of the artifacts after 100 generations. At each generation, the artifact is mutated 10 times (the first mutation being itself), and the best of these 10 mutations is passed to the next generation.

We can see, at Figure 4.1, that after only 100 generations we already have an artifact landscape that conforms quite well to the environment, when weighted by the attribution. That means this artifact is already quite close to a maximum of fitness. In fact, this model retains the basic dynamics of evolutionary optimization necessary for innovation modelling.

4.2 Specialization

Some of the features that are normally not present in evolutionary optimization models of innovation will now be presented. Firstly, we want to show the possibility of having artifacts becoming specialized. From a "father", two "children" will be generated, with different attributions.

By modelling this process, together with generalization (to be presented in the next section), we want to see the emergence of dynamics in terms of recombination of artifacts. By creating the possibilities of splitting and merging artifacts, the number and specificities of them are not determined by the initial conditions any more, but by the dynamics of the model itself.

On Figure 4.2, we can see an artifact that can perform two different tasks very well. It has, thus, two separate areas of attribution. On the left, we have the artifact landscape weighted by the attribution, and on the right the environment landscape weighted by the attribution. This is, then, a candidate for splitting.

This is, then, what happens. The algorithm detects that artifact as a target for specialization. It, then, copies the genome from the father, meaning both new artifacts will initially have the same genome. Finally, it splits the attribution areas into two, giving one of the peaks to each of the children.

The condition to consider an artifact as candidate for splitting is, basically, to have two peaks of similar magnitude in the artifact landscape, meaning it is quite suited to two specific tasks. Then, when splitting it, we force each of the children to be attributed one of the tasks from the father.

As we can see from Figure 4.3, that one artifact from Figure 4.2 was divided into two children. From both right sides, we immediately see the splitting of attribution areas: now, each of the children is being evaluated in different areas of the environment landscape. Of course, this is a process that could also be modelled in a gradual fashion, with

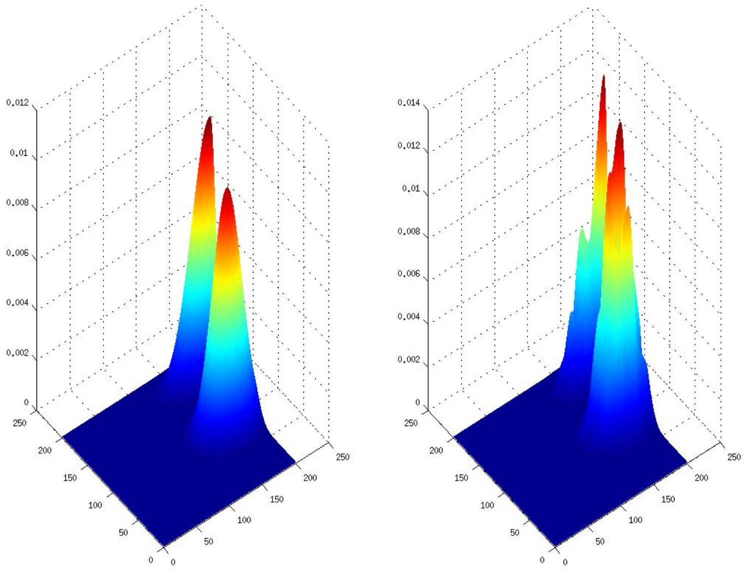

Figure 4.2: Artifact with two areas of attribution.

the two new artifacts drifting apart in terms of attribution over time.

From the left side plots of Figure 4.3, it is not immediately obvious that the genome is exactly the same for both children. Here, we need to remember that we are seeing the artifact landscapes weighted by attribution; what makes the left side plots so different from each other are the attribution areas, not the genomes. Of course, these children will only be identical right after specialization; with completely different attribution areas, they will start evolving in different directions, and their genomes will drift away from each other more and more.

4.3 Generalization

Here, we will present a feature that is, in a sense, the mirrored version of specialization: generalization. While in the last section there was a single artifact originating two different ones, here we will do the vice-versa: two fathers will originate a single child, combining both attribution landscapes.

We can see on Figure 4.4 the two artifacts to be merged for this example. They are chosen by our algorithm for having similar artifact landscapes and attribution areas. This indicates that they are expected to perform similar tasks, and they perform these tasks at a similar level.

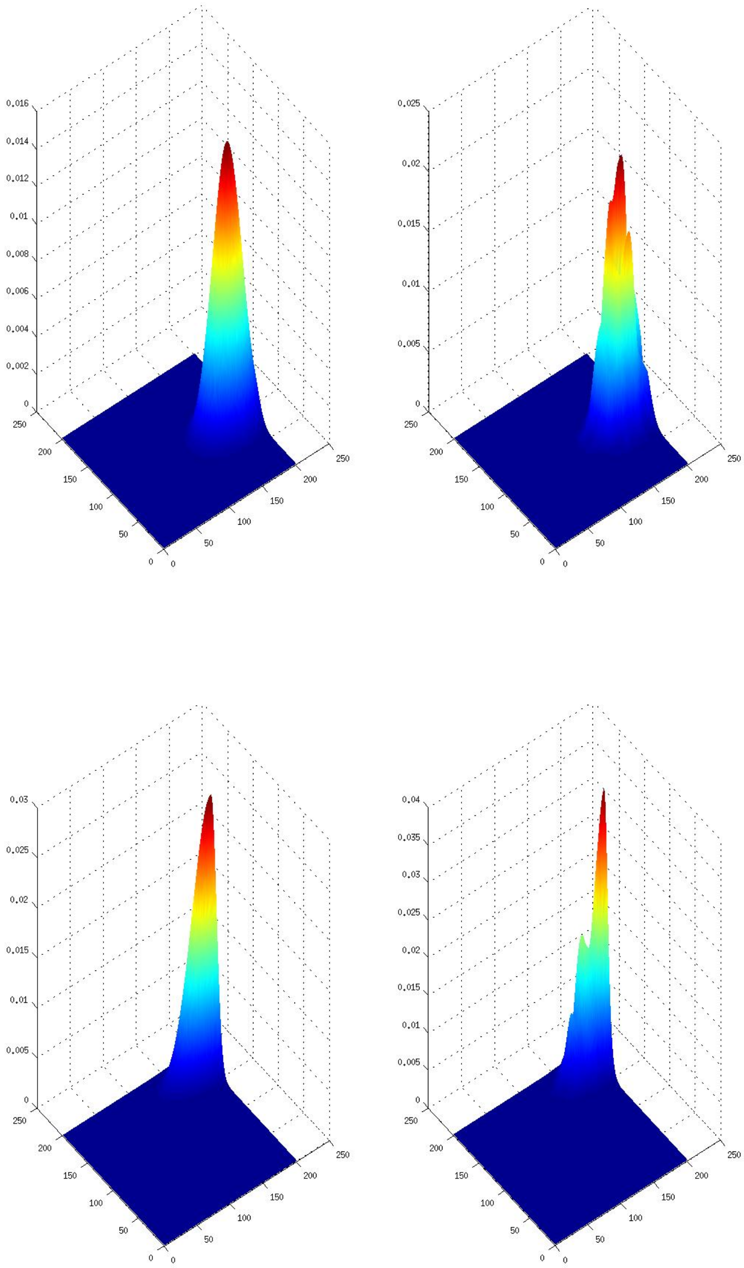

Figure 4.3: Two artifacts with separate attribution areas.

Here, our particular implementation is limited: there is no trivial way to merge genomes and keep the generated phenotypes similar to the pre-merged versions. Due to that limitation, the child simply takes the genome of one of the fathers, at random.

Then, we see the result of the merging on Figure 4.5. The artifact landscape on the left is exactly that of the first of the two fathers, and the attribution is now changed to become a composite of both attributions from before. This can be seen more easily on the right, where the environment landscape weighted by the attribution is clearly different from both plots at Figure 4.4, albeit quite similar to them.

4.4 Possible extensions

Since the implementation done for the modelling framework was merely a proof of concept with limited scope, not all possibilities envisaged were converted into working code for running simulations. The multiple-landscape approach opens up a myriad of possibilities, and there was not enough time to actually go forward with each of them and generate numerical outputs representing the concepts and features foreseen.

One of the examples of the working ideas that were not included into our code was the possibility of having the environment landscape to change gradually with time. This is certainly an option that makes sense intuitively, since our perceived utilities for different tasks evolve with time, and the evolution of artifacts should in some way accommodate this ever-changing nature of the utility function associated with the functional space.

With a changing environment, radical innovation might ensue. It seems possible that new maxima might emerge inside a given attribution area. It is, then, certainly possible to have an artifact jumping suddenly from being adapted to one maximum to another one. That would be an interesting account of radical innovation, and it would not interfere with the basic dynamics of gradual innovation in place.

Furthermore, in the same fashion we can have a dynamic environment, we can also have dynamic attribution landscapes. This is implemented to some extent with specialization and generalization; but it can certainly be extended for gradual drifts from area to area. With that, phenomena like innovation cascades, where there is a feedback process between the innovations and the landscape, could also be modelled.

Finally, there is close to no interaction between different artifacts in our code at the moment. Except for the specialization and generalization processes, different artifacts evolve in parallel without any crossover between them, and no concern whatsoever for

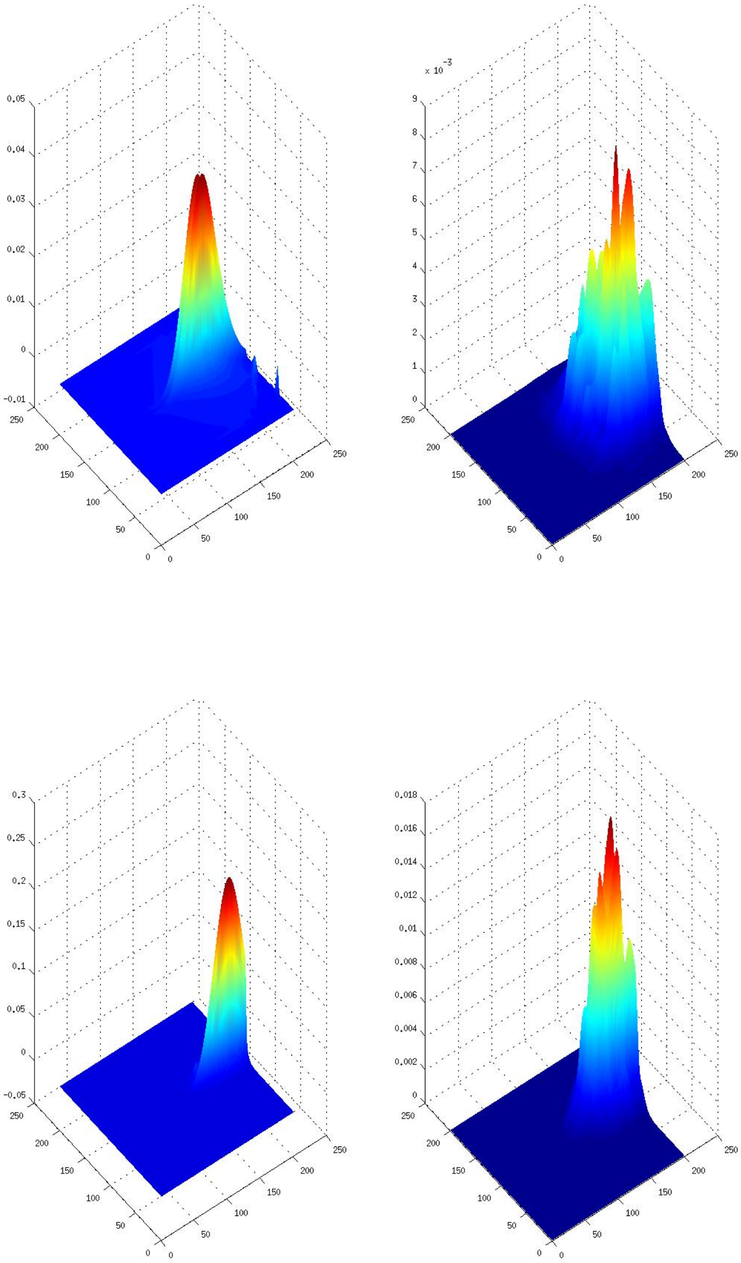

Figure 4.4: Two artifacts with separate attribution areas before generalization.

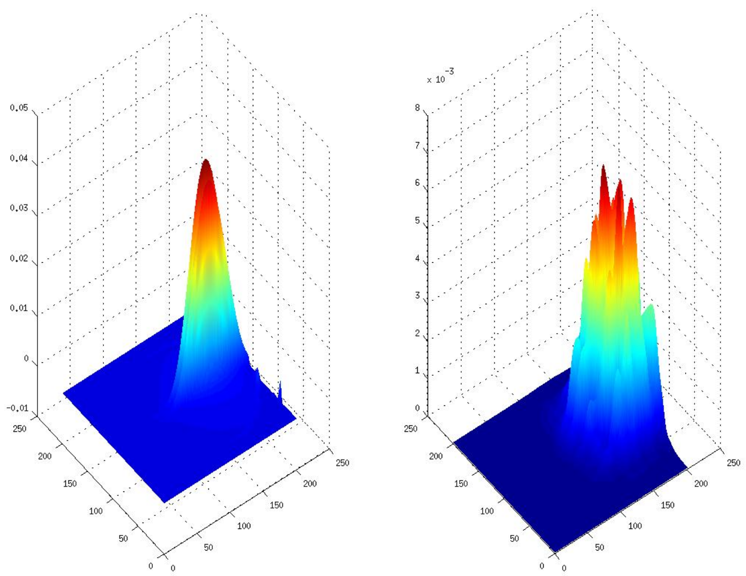

Figure 4.5: Artifact with both areas of attribution and taken the genome of one of the pre-generalized artifacts.

what the other members of the population are doing in terms of evolution. A possibility with this framework would be, for example, to have different artifacts whose attributions and area of specialization are competing closely, and make that competition the cause of a detrimental effect in each other's fitness values.

5

Conclusion

In this work, we present a new framework for innovation modelling. Deriving from models using fitness landscapes for that task, we extend this idea to using multiple landscapes to represent different facets and stylized facts about artifacts. By doing that, we are able to keep modularity and ease of modelling, by representing different aspects separately and combining them towards an evolutionary optimization approach.

We propose a system with three different landscapes: environment, attribution and artifact, all of them existing not in a design space, but in a functional space. They are related, respectively, to how useful different tasks are, what do we expect from an artifact and how good that artifact is at performing different tasks. In this context, evolution is not seen as "hill climbing" as in the previous models, but as fitting functions. With the extra flexibility we propose, different artifacts might have very high fitness values based on their performance in completely different tasks.

Our particular implementation, designed as a proof of concept, uses three-dimensional landscapes for ease of visualization. The environment landscape is based on the NK-model, while the artifact landscape uses postfix strings as a form of generating analytic functions in a genome-related fashion. The attribution landscapes are composed of two-dimensional Gaussian functions. These three landscapes are, then, used to calculate a fitness value measured as the inverse of the total weighted difference between artifact and environment landscapes, where the weights are given by the attribution. This fitness value is, finally, used as an input for a simple evolutionary algorithm, which changes the artifact landscape.

We have seen that this setup retains the basic evolutionary optimization approach from fitness landscape models. For a static environment and attribution, there is still some sort of ”hill climbing”, but done in a different way. The basic functionality of previous models is maintained. Moreover, new functionalities can easily be added: the possibility of forking a single artifact to generate two separate ones is the first new feature introduced. The opposite process, through which two artifacts performing very similar tasks merge into one, is the next logical step.

Some other possible features in this model are briefly discussed. The possibility of making both environment and attributions dynamic opens up the way to model feedback processes and innovation cascades, for example. Moreover, it is possible to introduce a larger degree of interaction between different artifacts, by having them compete for specific areas of the functional space, for example.

Finally, we posit that this framework, while retaining the basic functionalities of fitness landscape models, also introduces new possibilities to be explored. It might capture some specific features like radical innovation better than those models, and it brings more flexibility and modularity into the realm of innovation modelling.

5.1 Future Work

This is a framework which presents many possibilities for continued development. First of all, it is necessary to emphasize once more the nature of this work: it is not aimed at presenting a polished model and final results, but rather to showcase a different set of mind to model innovation and briefly explore some of the possibilities it enables. By its own nature, this is a not a finished work by any means.

There is a lot of room for investigation inside the proposed guidelines from this work. For example, the possibilities for different ways of generating landscapes, their advantages and shortcomings, and the suitability of different methods for each of the landscapes presented here are left open. An extensive survey into this realm would be an interesting extension to what is presented in this paper.

Finally, this work has presented a rather general view of innovation, without focusing on specific features or areas. While this is the ideal way of presenting a framework that can be suited to many different specificities, it lacks particular results or validation. Ideally, more work by implementing these ideas in innovation modelling with a focus on particularities, be it on certain specific features or on subsets of the domain of innovation, is needed to get this modelling template to a more mature stage of development.

Bibliography

[1] J. C. Fisher, R. H. Pry, A simple substitution model of technological change, Technological forecasting and social change 3 (1972) 75–88.

[2] P. A. David, Technical choice innovation and economic growth: essays on American and British experience in the nineteenth century, Cambridge University Press Cambridge, 1975.

[3] K. Frenken, Technological innovation and complexity theory, Economics of Innovation and New Technology 15 (2) (2006) 137–155.

[4] G. Dosi, et al., The nature of the innovative process, Technical change and economic theory 2 (1988) 590–607.

[5] L. Leydesdorff, The triple helix: an evolutionary model of innovations, Research Policy 29 (2) (2000) 243–255.

[6] S. Kauffman, J. Lobo, W. G. Macready, Optimal search on a technology landscape, Journal of Economic Behavior & Organization 43 (2) (2000) 141–166.

[7] E. Almirall, R. Casadesus-Masanell, Open versus closed innovation: a model of discovery and divergence, Academy of management review 35 (1) (2010) 27–47.

[8] L. Barnett, Evolutionary search on fitness landscapes with neutral networks, Ph.D. thesis, Citeseer (2003).

[9] T. Ma, Y. Nakamori, Agent-based modeling on technological innovation as an evolutionary process, European Journal of Operational Research 166 (3) (2005) 741–755.

[10] S. Wright, The roles of mutation, inbreeding, crossbreeding and selection in evolution, in: Proceedings of the sixth international congress on genetics, Vol. 1, 1932, pp. 356–366.

[11] M. Tushman, D. Nadler, et al., Organizing for innovation, California management review 28 (3) (1986) 74–92.

[12] S. J. Gould, E. S. Vrba, Exaptation-a missing term in the science of form, Paleobiology (1982) 4–15.

[13] G. Bonifati, 'more is different', exaptation and uncertainty: three foundational concepts for a complexity theory of innovation, Economics of Innovation and New Technology 19 (8) (2010) 743–760.

[14] N. Dew, S. D. Sarasvathy, S. Venkataraman, The economic implications of exaptation, Journal of Evolutionary Economics 14 (1) (2004) 69–84.

[15] M. Villani, S. Bonacini, D. Ferrari, R. Serra, D. Lane, An agent-based model of exaptive processes, European Management Review 4 (3) (2007) 141–151.

[16] D. Lane, R. Maxfield, D. Read, S. van der Leeuw, From population to organization thinking, in: Complexity Perspectives in Innovation and Social Change, Springer, 2009, pp. 11–42.

[17] S. Kauffman, S. Levin, Towards a general theory of adaptive walks on rugged landscapes, Journal of theoretical Biology 128 (1) (1987) 11–45.

[18] K. Frenken, A fitness landscape approach to technological complexity, modularity, and vertical disintegration, Structural Change and Economic Dynamics 17 (3) (2006) 288–305.

[19] K. E. Mathias, L. D. Whitley, Transforming the search space with gray coding, in: Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on, IEEE, 1994, pp. 513–518.