Evolutionary Computation for Computer Music

Sjaak Jöbses

Bachelor Thesis

Institute of Sonology,
Royal Conservatory,
The Hague, NL

August 2011

Abstract

Nature's place in the arts and music has been firmly established. For many thousands of years, nature and life have provided, at least, ideas for musical or aesthetic utility. This utility is usually based on visual and physical properties of the natural object or phenomenon being observed. In recent decades the increase in computational power available for modeling actual natural processes, has provided the arts new means of exploring nature.

Electronic music has always been a field strongly influenced by systematic or model based approaches. In algorithmic music composition intra- as well as extra-musical concepts have been used to generate musical material. An example of the former being, Schoenberg's twelve tone technique and the latter, the use of dynamical systems and probabilistic functions by Xenakis. As is the case with many compositional models, each has its problems.

In search for emergent phenomena, evolutionary algorithmic techniques provide non-deterministic, extra-musical means to generate musical material. Evolutionary Computation is comprised of ideas from the fields of Evolutionary Biology, Computer Science and Complex Systems. The field of evolutionary art, or in a broader sense, Artificial Life for art is a young field. It also has received the necessary criticism often associated with technological art. Taking a wide perspective, commonalities among all these fields will be sought.

Outlined will be the relevant scientific ideas of Evolutionary Computation, in relation to algorithmic music composition. An overview of previous musical work and philosophical problems will be given. My own implementation of a virtual ecosystem will be described.

Acknowledgments

I owe many thanks to everybody at the Institute of Sonology for creating a tremendous learning experience in the past 4 years. The environment has always been such a one that it is not only the curriculum but also the students, events and neighboring departments that contribute to a pleasant atmosphere and interesting learning experience.

Specific thanks go to Paul Berg as my thesis adviser, for guiding me through a hectic last year and always providing critical comments which I've come to realize, build my own critical ability. Raviv Ganchrow for providing a highly in-depth exercise into sound and its multiple relations to space; the necessary philosophical, spatial, and artistic thought associated with it. Joel Ryan for providing guidance in many broadly oriented discussions about art, music and science. All the other teachers for sharing their experience and knowledge.

My dear friends for all the support, friendship and the opportunity to practice my passions alongside theirs. My family and girlfriend for all support.

Contents

1 Evolution, Complexity and Algorithms **1**

1.1 Evolutionary biology 2

1.1.1 Darwin and emerging ideas about Evolution 2

1.1.2 Genetics 4

1.2 Complex Adaptive Systems 6

1.2.1 Complexity 6

1.2.2 Neural Networks and Swarms 7

1.3 Complexity through Algorithms in Music 10

1.3.1 Chaos in Music 10

2 Evolutionary Computation **13**

2.1 What is computed evolution? 14

2.1.1 Genetic Algorithm 16

2.1.2 Genetic Programming 17

2.2 The fitness function 18

2.3 The representation problem 20

2.4 The multi-dimensional search space 21

2.4.1 The metaphoric search space 22

3 Applications and Problems in Music **24**

3.1 Evolutionary techniques in Music 24

3.1.1 Musical Organisms 26

3.1.2 Unconventional techniques 28

3.1.3 From musical interaction to audible traces 29

3.2 Towards virtual ecologies 31

3.2.1 Artificial Life for Music 34
3.3 Principles of Complex Adaptive Systems 36

4 Ecosystem approach and conclusions **39**
4.1 Creatures 39
4.1.1 Simulation Functionality 40
4.1.2 Sound Functionality 41
4.2 Conclusions 42

References **45**

Chapter 1

Evolution, Complexity and Algorithms

> I like to think
> (and the sooner the better!)
> of a cybernetic meadow
> where mammals and computers
> live together in mutually
> programming harmony
> like pure water
> touching clear sky.
>
> Richard Brautigan
> *All Watched Over by Machines of Loving Grace*

A short overview of evolutionary biology will follow ending with the frameworks of genetics as a medium in which evolutionary mechanisms can occur. Next, Complex Adaptive Systems (CAS) will be discussed. A field dealing with the modeling or simulation of complex phenomena found in for example, biology, economics and physics. Last, there will be some comments on the use of algorithms in music.

Obviously any dealing with the history of science suffers from biases and incomplete descriptions. I seek to give the reader a few fundamentals on which the rest of the concepts in this text can be put. I will outline the relations to music composition practice.

1.1 Evolutionary biology

The subject of biological evolution is an enormously wide one. It is a highly debated topic today, and probably has been so since the first humans had vague notions of it. I will briefly describe some scientific history but please note that there are many excellent references which will give the reader a more complete overview of this history. (Bowler, 2003)

The inherent interdisciplinary nature of evolutionary biology needs to be limited in the context of what might be musically useful. Specific information about the various studies of organisms can be avoided, as well as 'historic-' or an 'earth-science' field like geology.

Out of the many sub-disciplines of evolution an overview of topics relevant in our context is:

- **The essence of the causes** of biological evolution
- **Genetics** providing a protocol description for evolution
- **Artificial Life** in order to recreate biological phenomena

1.1.1 Darwin and emerging ideas about Evolution

As with many great scientific ideas, Charles Darwin was not the only person to coin the idea of the occurrence of evolution in natural life. Only a year before publication of "*On the Origin of Species*" (1859), Darwin received a manuscript from Alfred Russel Wallace, titled "*On the Tendency of Varieties to Depart Indefinitely from the Original Type*." In order to give neither of the gentlemen a priority their work was published around the same time.

Skipping ancient Chinese, Greek or Roman thought and greatly simplifying the history, the surfacing of evolutionary ideas in Western thought occurs during the enlightenment. George Louis Leclerc de Buffon already illustrates the similarity between species in, "*Historie Naturelle*" (1749). He stated the idea of ancestry and that the earth is much older than the Biblical 6000 years.

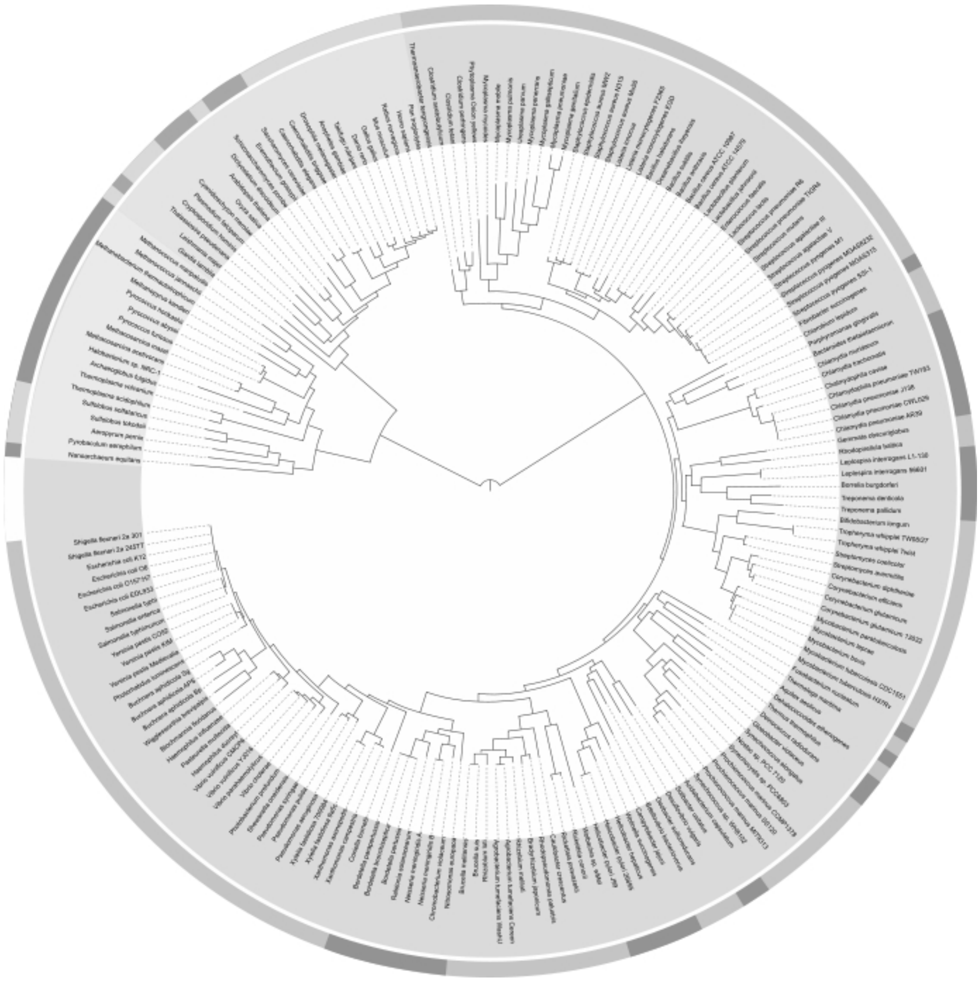

Figure 1.1: Phylogenetic tree of life

In 1809 Jean-Baptiste Lamarck publishes his book, "*Philosophie Zoologique*". An influential idea was the concept of "Inheritance of acquired characteristics". Lamarckian inheritance is somewhat of an umbrella term, but the main idea into which it translates is the fact that offspring acquire characteristics (or traits) from their parents. Another concept was his "Use and disuse", in which organisms would lose their characteristics that they would not require (and vice versa). According to Lamarck, life was still spontaneously generated and common ancestry was not among his ideas. The medium through which evolutionary processes occur was still to be understood.

Modern day intricacies of evolutionary theory are still being debated. But central ideas have emerged:

1. **The living world is not constant.** Evolution occurs, there seems to be a scientific consensus.

2. **Evolutionary change has a branching pattern.** Species descend from remote ancestors.

3. **New species form when a population splits into isolated fragments.** The famous example being, Darwin's observations of the finches on different Galapagos Islands.

4. **Evolutionary change is gradual.** Organisms that differ dramatically from their parents are often not able to survive in the same environment

5. **The mechanism of adaptive change is natural selection.** Organisms with favorable traits will be more likely to survive; due to competition for resources in the environment. Competition and variable traits being important mechanisms.

The word *mechanism* implies a mechanical look at the process of evolution. For our purposes, the usage of evolutionary ideas in a musical context, this is a necessary simplification. Continuing the mechanistic view for a moment, one might think there is an appearance of a 'designed outcome' due to the actual functioning of the process evolution. The emergence of order over time might be the strongest suggestion in this 'design'-direction. The emergence of order is what a composer, among other goals, might seek for during composition.

1.1.2 Genetics

Explanations for evolutionary mechanisms of heredity and variation can be found in the field of genetics. Important in this context are the high level function and structure provided by discrete units, called *genes*.

The work of Gregor Mendel (1822) laid out the first ideas in genetics in the paper, "*Versuche Uber Pflanzenhybriden*" (1865). Mendel's work concerned the functioning of hereditary information. He set out to find how variation and adaptation where organized in this medium of the genome. Note that the terms like genes and genome were not yet used at that time. His results were opposed to the ideas of Lamarckian inheritance, specifically the idea that offspring acquired the traits of their parents.

Through many long experiments with pea plants, he found that each *trait* of an organism could be assigned a discrete value, a so called *allele*. Each trait is encoded in a gene pair, for example pea color, can be yellow or green. A trait can be either dominant or recessive. Mendel's *Law of Segregation* states that a parent randomly passes one of the genes in a gene pair, to it's offspring. Whichever gene is dominant determines how the trait will express itself. The *Law of Independent Assortment* claims that the gene pairs are selected and transmitted independently of other gene pairs.

During the 1940's and 50's a lot of research went into DNA, *DeoxyriboNucleic Acid*, and established was the idea that this macromolecule was the carrier of genetic information.

During the 1960's and 70's scientists where challenging the *The Modern Synthesis*. This synthesis was, among others, mainly based on the work of Darwin and Mendel. One of the assumptions being challenged was the fact that evolution occurred gradually. The morphology of organisms simply couldn't change very rapidly (over the course of a few generations), according to the way genetic mutation and evolution functioned. According to Stephen Jay Gould and Niles Eldredge, there was clear evidence in the fossil records that evolution could make jumps. This evidence was called the *punctuated equilibria*. Referring to the the fossil records that showed long periods of gradual change and sudden spikes of activity. Historical contingencies as well as biological constraints might have a role as important as natural selection.

The mechanisms and causes of evolution are necessary to understand for a composer wanting to utilize these concepts in music. Genetics provides a medium through which evolution operates, enabling reasonable modelling of the mechanisms.

1.2 Complex Adaptive Systems

The sole concept of biological evolution provides one half of something that might become musically useful. The concept of entities evolving and adapting to their environment is a compelling one. Even when the analogies to musical entities are still vague. The other half will be the environment these entities actually reside in and all the complex dynamic behavior that is associated with it.

At some point in history, reductionism begun failing in explaining complex phenomena and systems in nature. The science of Complexity or Complex Adaptive Systems has been emerging with a holistic view of these phenomena and systems. It is a highly interdisciplinary science, especially relevant to our later discussion about ecosystems.

1.2.1 Complexity

Assuming nature, and so the process of evolution which takes place in it, is a complex system, we need a definition of what such a system might actually be. Adding a term from evolution, we can ask; What is a complex adaptive system? Mitchell describes a complex system as,

> "a system in which large networks of components with no central control and simple rules of operation give rise to complex collective behavior, sophisticated information processing, and adaptation via learning or evolution" (Mitchell, 2009, p. 13)

Adding two often used terms she also gives this alternative definition,

> "a system that exhibits nontrivial emergent and self-organizing behaviors"

Especially in the context of a virtual ecosystem or artificial life, these definitions seem to hint at the potential such a system might posses. Adaptation through learning or evolution, might be processes a composer is very familiar with in terms of a possible struggle with musical material. Maintaining the musical context for a moment, notions like emergence and self-organization are much harder to place within some sort of compositional activity. Ways of interpreting these notions will become clear in chapter 3.

Goldstein describes emergence as,

> "the arising of novel and coherent structures, patterns and properties during the process of self-organization in complex systems" (Goldstein, 1999, p. 42)

At least *structures, patterns* and *properties* are words that can be also be found in music composition vocabulary.

Complexity science historically stems, among others, from the field of Cybernetics. This field concerns itself with structure of regulatory systems. It's most famous (and probably superficial) idea is that of the feedback loop. Dynamical systems can be described as closed loops where every action will cause a reaction within the system. In the book, "*Cybernetics: Or Control and Communication in the Animal and the Machine*" (1948), Norbert Wiener lays the foundations for concepts like feedback, stability, the black box, information and noise in the context of Cybernetics.

1.2.2 Neural Networks and Swarms

Through some examples of complex phenomena it might become apparent how these might relate to evolutionary computation, or especially evolutionary mechanisms that take place within an artificial environment. When we take the vantage point of a ecosystem, a neural network (within an organism) might be seen as a low level complex system. Collective animal behavior can be seen as phenomenon emerging on a high level.

Neural Networks are a popular example of a complex phenomenon under study. Described as follows,

> "The study of neural networks is the study of information processing in networks of elementary numerical processors. In some

> cases these networks are endowed with a certain degree of biological realism and the goal is to build models that account for neurobiological data. In other cases abstract networks are studied and the goal is to develop a computational theory of highly parallel, distributed information-processing systems." (Wilson R., 1999, p. 597)

Information processing is achieved through a network of nodes, connected by links. Each of these nodes has *weighted* inputs and an output. The information received on the input will be processed by a so called *activation function*. This function is dependent on the weighted values of the input and can be roughly compared with a transfer function. A network is typically ordered in layers. The Artificial Neural Network (ANN) is, among others, defined by these parameters:

- The interconnection pattern of neurons
- The learning process for updating the weights of the neurons
- The activation function

ANN are typical machine learning algorithms that can be applied to large data sets for function approximation, classification or other data processing methods. A famous musical example utilizing the direct activity of a ANN is David Tudor's work "*Neural Synthesis Nos. 6-9*" (Tudor, 1995). He uses custom build electronics that incorporated an experimental Intel chip that simulated individual neurons in order to generate audio. The electronics was called the *Box*. It contained 64 non-linear amplifiers with programmable connections between all of them. 16 of the 64 'neurons' had *RC-circuits* attached to them, which made them into an audio oscillator.

Other use of ANNs can be their inherent ability to derive meaning from complex data. A typical machine learning application. Discussion on such advanced usage of these connectionist models can be found in (Nierhaus, 2009, p. 205-223), along with musical applications.

Swarm behavior are the phenomena seen in collective animal behavior. We are all familiar with, *flocking* birds, the *schooling* of fish or *herding* behavior of the wildebeest. The famous example of swarm behavior, simulated on a computer, is *Boids* by Reynolds (1987). It is a visual simulation of the emergent behavior that can be observed in flocks of birds. Each bird is seen as a autonomous *agent* or *particle* that can move in a 3D world. Each bird obeys to three simple rules in relation to the other birds in the swarm,

1. **Separation**: steer to avoid crowding local flockmates ('keep your distance')

2. **Alignment**: steer towards the average heading of local flockmates ('watch your neighbors')

3. **Cohesion**: steer or move toward the average position of local flockmates ('follow your neighbors')

The neighborhood around a bird is defined as a distance from its center, with a gap behind the bird (it can't see there). The result is complex emergent, even *life-like*, flocking behavior of the particles that are animated.

A musical application can be found in Jones where the author describes its relation to many terms already mentioned in this document, with:

> "AtomSwarm, like any complex dynamical system, is fundamentally a staging ground for a continuous parallel flux of interactions, between forces, agents and resources. Convoluted feedback loops arise between the multiple planes of interaction (human input, rules, hormones and genomes), with sufficient complexity to evoke the organic (in)stability of a natural ecosystem." (Jones, 2008, p. 431)

The system even includes a genomic representation for the properties of the agents. A subject to be discussed in the next chapter.

1.3 Complexity through Algorithms in Music

In the previous two sections concepts of evolution and complexity where discussed, sometimes in relation to musical results. Algorithmic composition is the technique through which these ideas can enter music.

An algorithm is an *effective method* described in a finite list of instructions, in order to perform a certain function. Or more loosely defined, it is a recipe, method or technique for doing something. (Wilson R., 1999, p. 11)

The use of algorithms in music can be traced back as far as the year 1025 with Guido of Arezzo's work "*Micrologus de disciplina artis musicae*". (Nierhaus, 2009, p. 1) He created a method for converting text into melodic phrases.

1.3.1 Chaos in Music

Iannis Xenakis is often referred to as a prime example of the usage of algorithmic techniques in music. Micro as well as macro level compositional decisions are extracted from *abstract automata* algorithms (Solomos, 2005) to *stochastic processes*. Concerning his work in *stochastic music* he describes in "*Formalized Music*",

> "For if, thanks to complexity, the strict, deterministic causality which the neo-serialists postulated was lost, then it was necessary to replace it by a more general causality, by probabilistic logic which would contain strict serial causality as a particular case. This is the function of stochastic science."

In response to the conclusions Xenakis has on the 'Crisis of Serial Music', he somewhat poetically continues to describes the potential of stochastic techniques, "But other paths also led to the same stochastic crossroads - first of all, natural events such as the collision of hail or rain with hard surfaces, or the song of cicadas in a summer field. These sonic events are made out of thousands of isolated sounds; this multitude of sounds, seen as a totality, is a new sonic event. This mass event is articulated and forms a plastic mold of time, which itself follows aleatory and stochastic laws. If one then wishes to form a large mass of point-notes, such as string pizzicati, one must know these mathematical laws, which, in any case, are no more than a tight and concise expression of chain of logical reasoning.

Everyone has observed the sonic phenomena of a political crowd of dozens or hundreds of thousands of people. The human river shouts a slogan in a uniform rhythm. Then another slogan springs from the head of the demonstration; it spreads towards the tail, replacing the first. A wave of transition thus passes from the head to the tail. The clamor fills the city, and the inhibiting force of voice and rhythm reaches a climax. It is an event of great power and beauty in its ferocity. Then the impact between the demonstrators and the enemy occurs. The perfect rhythm of the last slogan breaks up in a huge cluster of chaotic shouts, which also spreads to the tail. Imagine, in addition, the reports of dozens of machine guns and the whistle of bullets adding their punctuations to this total disorder. The crowd is then rapidly dispersed, and after sonic and visual hell follows a detonating calm, full of despair, dust, and death.

The statistical laws of these events, separated from their political or moral context, are the same as those of the cicadas or the rain. They are the laws of the passage from complete order to total disorder in a continuous or explosive manner. They are stochastic laws." (Xenakis, 1992, p. 8-9)

The motivation for using stochastic algorithms in composing or synthesizing new music, is a deep one. Fueled from intrinsic beauty, or fear to some, that might be observed in potentially large scale emergent phenomena. The similarity to expectations arising from evolutionary or complexity related ideas, seems striking.

On *self-organisation* in music Blackwell and Young write,

> "The development of higher level musical structure arises from interactions at lower levels, and we propose here that the self-organisation of social animals provides a very suggestive analogy." (Blackwell & Young, 2004, p. 137)

Taking somewhat of a leap to art theory and relating to the dynamics described by Xenakis, Galanter writes,

> "(...) in its purest form generative art using complex systems is about the dynamics of complex systems. Complexism not only rehabilitates formalism, it perhaps more importantly reintroduces the artistic notion of dynamism. As originally introduced by the Futurists, dynamism celebrated the aesthetic of the locomotive and the racecar, and called for the exploration of motion and process rather than portraying objects as being frozen in time.
>
> Dynamism in complex art is the visceral appreciation of the beauty of dynamics as more fully revealed in the context of complexity. In a sense, formalism is to nouns as dynamism is to verbs. With its focus on complex generative systems, complex art encourages artists to move from art objects to art processes, i.e., from nouns to verbs." (Galanter, 2008, p. 330)

Chapter 2

Evolutionary Computation

> One could say that a man can "inject" an idea into the machine, and that it will respond to a certain extent and then drop into quiescence, like a piano string struck by a hammer.
>
> Alan Turing
> *Computing Machinery and Intelligence*

As described in the previous chapter, biology can form a rich and often complex source of inspiration for computational models. Be it for artistic ends or not. Where the field of bioinformatics actually applies computational techniques to problems in biology, evolutionary computation concerns itself with models *of* a biological nature. These models in general have search, learning or optimization characteristics. A question on the use of such models within music composition might be: *What music composition tasks could benefit from search, learning or optimization techniques?*

Evolutionary algorithms are used in many real world applications in fields such as, electrical engineering, computer graphics or finance. As we will see later on, evolution also has a rich, but short history in the arts.

I will discuss genetic algorithms and genetic programming. As well as the workings of the fitness function, genetic representation and the concept of search space.

2.1 What is computed evolution?

In popular science (fiction) writing there are many references to actions a computer can't (or for that matter, can) perform. A computer is not *autonomous*—we tell it what to do. A computer doesn't *evolve*—it is restricted by it's programming. And even more striking is the fact that they do not *reproduce*. Evolutionary computation, or more in general the field of artificial life, concerns itself with these kind of problems.

The 'fathers' of computing already wrote and lectured about the philosophical problems relating to computing and biology. Philosophizing on the creation of an *artificial intelligence*, based on the mind of a 'child'—not an 'adult', Alan Turing wrote,

> "We cannot expect to find a good child machine at the first attempt. One must experiment with teaching one such machine and see how well it learns. One can then try another and see if it is better or worse. There is an obvious connection between this process and evolution, by the identifications
>
> Structure of the child machine = hereditary material,
>
> Changes of the child machine = mutation,
>
> Natural selection = judgment of the experimenter
>
> One may hope, however, that this process will be more expeditious than evolution." (Turing, 1950)

As we will see later, these are the basic components of the process of *interactive evolution*. Worth mentioning is also the work of John Von Neumann, specifically his book "*Theory of Self-Reproducing Automata*" (1966) where he lies the mathematical and logic fundamentals for 'reproducing machines'. Even earlier, in 1948, during his lecture at the Hixon Symposium, Von Neumann theorizes about complex systems, automatons and their relations to natural organisms.

> "Natural organisms are, as a rule, much more complicated and subtle, and therefore much less well understood in detail, than are artificial automata. Nevertheless, some regularities which we observe in the organization of the former may be quite instructive in our thinking and planning of the latter; and conversely, a good deal of our experiences and difficulties with our artificial automata can be to some extent projected on our interpretations of natural organisms." (Jeffress, 1951)

The need for extensive work on artificial evolutionary systems, was clearly in the air. In the 1960's and 70's the theoretical basis for evolutionary computation was being formulated. Three techniques where developed by the names of *genetic algorithms* (GA), *evolution strategies* (ES) and *evolutionary programming* (EP). The common feature of all these techniques being the evolution of a population of candidate solutions to a given problem. (Mitchell, 1996, p. 2) Ingo Rechenberg developed ES in his paper "*Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*" (1973). EP can be seen as a forerunner of what is nowadays called *genetic programming* (GP).

Evolutionary computation is in effect, a method of searching among an enormous number of possibilities for solutions. Out of the search property, overtime emerges a *design property*: novel solutions to problems.

2.1.1 Genetic Algorithm

The genetic algorithm was invented during the 1960's by John Holland. He set out to study the biological phenomenon of adaptation as it occurs in nature. In 1975 he presented his book "*Adaptation in Natural and Artificial Systems*" (Holland, 1975).

The basic genetic algorithm contains the following steps:

1. Generate a population of n individuals with random l-bit chromosomes
2. Calculate the fitness of each chromosome x in the population
3. Repeat the following until n members of a new population have been created;
   (a) Select a pair of parent chromosomes (according to fitness)
   (b) According to crossover probability perform crossover, which might result in offspring
   (c) Mutate the offspring according to mutation probability and place it in the new population
4. Replace the current population with the new population
5. Go to step 2

The concept of a population consisting of individuals is quite straightforward to implement in an appropriate data-structure. Each individual is genetically represented by it's chromosome, which in *classic GA* is just a bit string. The fitness (usually a real valued number) of each individual is calculated according to the *fitness function.* This fitness function encodes criteria for the evaluation of potential solutions to the problem posed. The actual core process of a GA is represented by selection, crossover (also called recombination) and mutation. These operations are called the *genetic operators.* Other operators like inversion are possible, if desired.

The selection of 2 parent chromosomes can occur by fitness value, where higher fitness gives a higher *chance* for reproduction. There are other possibilities for the selection method. With a interactive GA, the user can be the selector and choose, on the basis of the phenotype, which individual can reproduce. Similar to what biologists call *viability selection* is fitness-proportionate selection, the actual aging of an individual before it becomes viable for reproduction. The number of times an individual is expected to reproduce, while computing for one generation, is equal to it's fitness divided by the average fitness of the population. Reproduction probability is 'scaled' with overall fitness.

Other selection methods include, sigma scaling, elitism or rank selection. Which selection method to use according to the problem under investigation, is still an open question. (Mitchell, 1996, p. 124)

The recombination of 2 chromosomes occurs at a random *locus*, a point within the chromosome. According to a certain mutation probability, at a random locus, genes are mutated via bit-wise inversion.

Of course there are many variations or tweaks to the basic process described above. Many features and effects on performance have been debated.

2.1.2 Genetic Programming

Genetic programming (Koza, 1992) is a technique very similar, in its core mechanism, to the genetic algorithm. The main difference being that operations are performed on 'functional blocks of programming'. These functional blocks (the bit string, in GA) are represented as a parse tree. Programming languages like LISP form an excellent candidate for encoding these type of algorithms.

Each node in the tree can either be a *function* or a *terminal* (a variable). The set of functions or terminals used, should be representative of the problem that's being computed for.

Garcia (2001) has used GP and its search property in order to find 'Sound Synthesis Techniques' that are similar to those of acoustic instruments. The solution space (or search space) is a huge one. Given a certain sound synthesis paradigm as well as a 'target sound', GP is utilized to approach this target sound. The implementation of such a system requires a way of choosing syntactically correct solutions. But also, the right synthesis technique (FM, additive-synthesis). Garcia proposes to create a 'functional-form suggestion mechanism.'

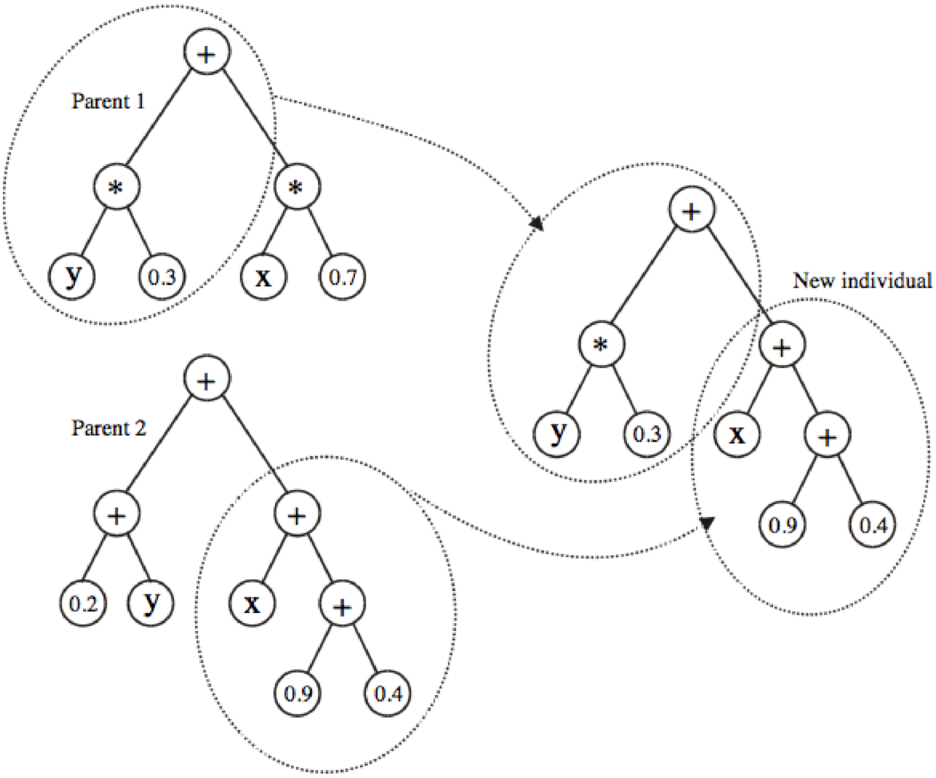

Figure 2.1: Parse tree (Garcia, 2001)

2.2 The fitness function

The fitness function deserves special attention. In popular belief *survival of the fittest* is often interpreted as survival of the *strongest*, when in fact it is survival of the *most adapted*. The fittest individual is the most adapted to his environment. This function computes the requirements for the sought after solution and implicitly encodes the adaptation and improvement that all solutions will undergo.

Using GAs for optimization problems, the fitness function can become quite trivial. If for example we wanted to search for a function which outputs a certain value, we can test the genetic representation against this value. Encoded in the genetic representation would be simple mathematical operators and random values. Which, when executed, will produce the sought after value.

Applying a GA to the travelling salesman problem, the fitness function needs to test for the length of the path each phenotype will 'travel'. The GA will be searching—among an enormous search space, depending on the number of points to be visited—for the shortest path travelled where the solution will visit each point only once, and come back to the starting point.

As we will see, when entering the musical domain problems soon come to surface. Again, the evaluation of a static goal is trivial. Approaching a certain melodic phrase is easy to test for. Several general categories of fitness functions for musical utility are:

- **Deterministic**: there is a direct relation between the fitness function and the encoding in the *genotype*. Can also be seen as a pattern matching scheme.

- **Formalistic**: here the focus is on the *phenotype*. The fitness function will measure general 'style' rules and could be more concerned with musical form.

- **Computational-aesthetics**: controversial, philosophical approach. The difficulty being the formalization of aesthetics.

- **Interactive**: where the user determines the fitness of a certain phenotype.

Although computational aesthetic evaluation is a fuzzy and unsolved problem, attempts have been made. Agent based systems, where actors evaluate their peers and form judgments, have been constructed. I will discuss some examples in the next chapter. More dodgy solutions for aesthetic evaluation include the use of connectionist models such as neural networks or perceptual measures like, spectral analysis and complexity/order measurements.

An interactive fitness function asks the user of the system for an evaluation of the phenotype. The obvious problem that presents itself here, is our limited ability in evaluating large numbers of results. This is called the *fitness bottleneck*. Such a bottleneck would be the case if a user would need to evaluate every phenotype presented by the system. Such a system wouldn't be much better than purely random search, in the short run.

The *fitness landscape* is a means of visualizing the overall fitness of a population. Each genotype of the population is 'plotted' on the horizontal plane of the landscape, the x- and y-axis. Height, on the z-axis, is the individual's fitness. During the functioning of a GA, we can imagine the continual evaluation of the fitness of each individual, actually moving the individual around the landscape, until a largely stabilized landscape has been formed.

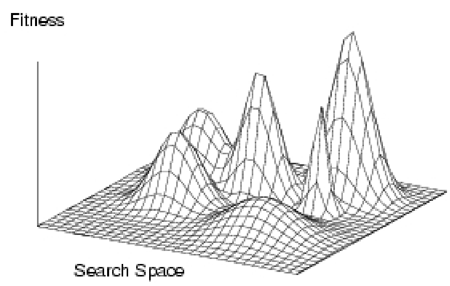

Figure 2.2: Fitness landscape

2.3 The representation problem

In the previous section the artistic choices involved in the fitness function became evident. Even more than the fitness function, the *genetic representation* relates directly to the complexity that can be displayed by a system. In effect the type and complexity of the genetic representation is what defines our 'movement' across the search space.

In nature even the genetic representation, or encoding mechanism, is subject to evolution. The genetic encoder/decoder is also encoded in our genome. As an analogy; think of a computer program that is able to print its own sourcecode.

There are at least four types of genetic representation, each with their own complexification potential.

Fixed Parametric
: Each phenotypic trait relates to a single gene, which can have binary or multiple state values; encoding a limited parametric space.

Extensible Parametric
: The variables of a phenotypic trait can be controlled in genes *separate* from the genes encoding for enablement of the trait.

Direct Mechanical
: The genome encodes 'algorithms' for the production of a variable trait. More variation is present.

Reproductive Mechanical
: Similar to DM, but with a reproductive capability. Genes can reproduce themselves, so to speak. Completely new traits can evolve.

The last option will obviously be the most complex, and as such exhibiting multiple levels of emergence where quite complex behavior could evolve. On a practical level chromosomes will need to be structured in some sort of data format. As we've seen this is in the most simple case just a bit string. A more extensive data structure is needed for the last 3 types of encodings. Data structures external to the actual genome could also be used, giving control over what information is evolved.

2.4 The multi-dimensional search space

The set of all possible solutions to a function being performed, is called the *search space*. This space, in the context of GAs, is typically a multi-dimensional space representing all possible combinations of phenotypic traits.

The fitness function used in a GA is actually our navigational guide, in navigating the search space. Imagine similar solutions being situated close to each other, and for example, the genetic mutation-operator causing 'jumps' to other locations in space. The complexity of navigating the space is also dependent on the genetic representation—complex representations can mean complex spaces. Natural biological diversity is an example of this.

The notion of search space implies that a GA is a searching method. It is important to note that this search property is not similar to for example, searching the internet or looking up a telephone number. The information to be found is not explicitly stored, but is created through mutation and recombination as the algorithm searches. In somewhat of a stretch, a GA can be likened to a so called *parallel terraced scan*. Each individual is seen as an agent exploring the space for solutions, the higher the fitness the more likely a certain direction (in the space) will be explored. The search is parallel because there are many agents exploring different directions. Not every direction is explored with an equal amount of resources (again dependent on the fitness). It is through the search space that the actual functioning of the fine grained architecture of a GA can be seen.

2.4.1 The metaphoric search space

In work by Dahlstedt (2001), he describes the exploration of so called parameter spaces. Any sound synthesis model resides within its own usually multi-dimensional parameter space. That of all possible unique combinations of parameter settings. GAs, and more specific the interactive variant, are excellent tools for exploring these spaces as they can search in a very direct way, for the unknown.

The step-wise nature of GAs can be a problem. For example in live performance. The generation and exploration of 'new sounds', on the spot, is an appealing thought to any electronic musician. In an attempt to make navigating the parameter space a more fluent one, a metaphoric approach to exploring is taken.

> “The user may like a specific variation of a mother sound, but he cannot access the continuum between the parent and the offspring. Also, the evaluation process may not be suitable for the audiences ears.” (Dahlstedt & Anders Nilsson, 2008, p. 479)

Dahlstedt and Nilsson propose a live performance system, where control inputs (in their case velocity sensitive pressure pads) actually control the exploration of parameter space. Each control input is mapped to a certain vector in n-dimensional geometric space. Depending on the control input this allows minute discovery of the space. Each vector follows the previous one and so, the point of origin *0* is your starting (‘parent’) sound, the end of the last vector, point *S*, is your sounding point (‘child’). *S* can be reset as the new *0* during performance.

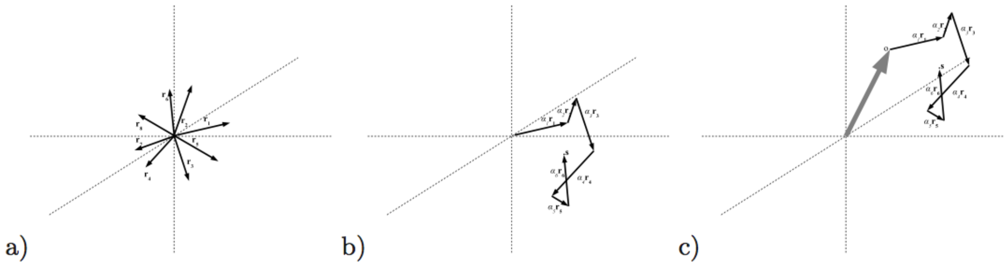

Figure 2.3: Parameter space (Dahlstedt & Ander Nilsson, 2008)

This is a technique clearly focussing on sound and gestural expression, not ‘parameter fiddling’-on-stage.

Chapter 3

Applications and Problems in Music

> Few of us have bred pets or livestock, let alone experimented with genetic engineering; many more of us have watched fish in a pond, insects in the grass, birds in the trees.
>
> Mitchell Whitelaw
> *Metacreation: Art and Artificial Life*

In the previous two chapters I have discussed the technical and theoretical overview of evolutionary computation, with some references to musical applications. Here I will discuss concrete music composition examples that utilize evolutionary computation techniques.

3.1 Evolutionary techniques in Music

The first know documented work of the use of a evolutionary technique in music is that of Horner and Goldberg (1991). They set out to solve a clearly defined problem, that of 'thematic bridging'. The transformation of a initial musical note pattern to a final pattern over a specified amount of time. They implemented their work on a Kyma Workstation, where the GA was implemented in Smalltalk. Around the time techniques like neural networks, just started to be used in algorithmic composition. (Todd, 1989)

Horner and Goldberg discovered the basics; the importance of the operator set, genetic encoding and the fitness function—in musical applications. But also hinted at the possibility of future use of GA techniques in sound synthesis. The musical parameters upon which the GA would operate are pitch, amplitude and duration. They used a real valued, direct mechanical genetic representation. Binary tournament selection and single-point crossover (recombination) are used. Binary tournament selection selects the 2 fittest individuals for reproduction, from a randomly selected sub-population.

In order to reach the goal of bridging two note sequences, they encoded *not* the notes in the genome, but operations on those notes. In a 'parallel auxiliary structure' the parameters to these operators are kept. The operators for performing transformations on the notes are: add, delete, rotate, exchange, mutate and no-operation. Each operator was encoded in one gene. The phenotype of an individual (to be executed from left to right) could look like:

```
(Delete 5)(Rotate 1 forward)(Delete 4)(Mutate 1 3)(Rotate 1 forward)
```

A deterministic, double fitness measurement function is used. The first part of the function calculates the degree to which the generated note pattern matches the final, user defined pattern. The number of notes desired, their ordering and the actual pitches are taken into account. The second part of the function concerns itself with desired and the produced duration's of the notes.

3.1.1 Musical Organisms

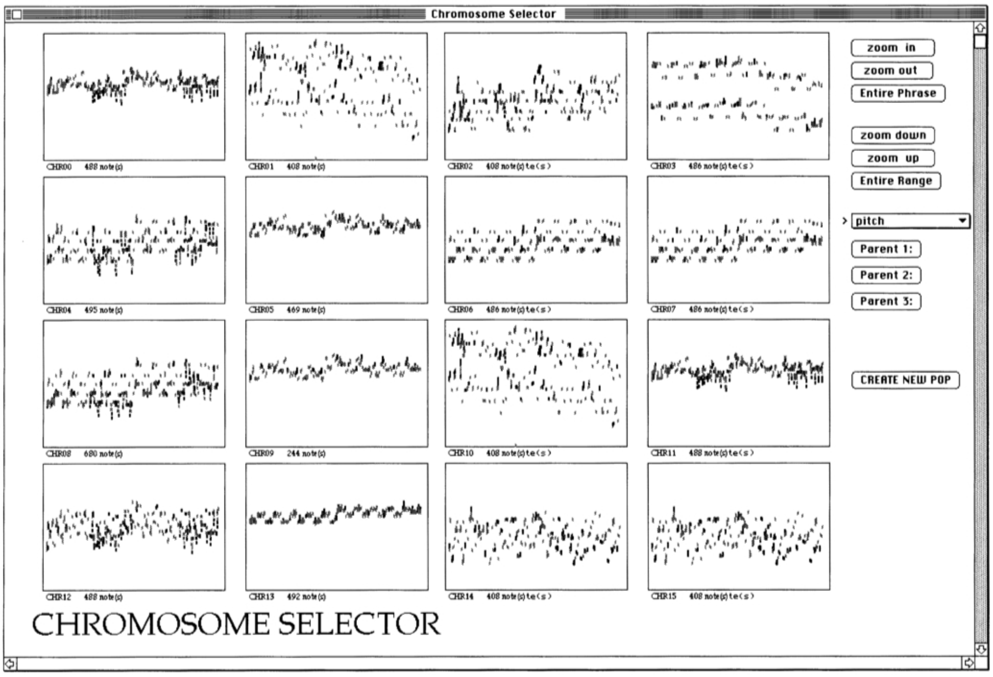

Figure 3.1: *MOE*

In "*The Evolution of Musical Organisms*" (Degazio, 1997), Degazio explains his work on a interactive evolution system for algorithmic music composition. An analogy can be drawn to the *Biomorph* system by Richard Dawkins, where a user can evolve 'stick figures' through an interactive GA. The genetic representation contains simple encodings like, the angle of branching, symmetry rules and depth of recursion.

As noted before, a system utilizing interactive fitness evaluation contains the fitness bottleneck, the user has limited ability for phenotypic evaluation. Depending on the goals of the user, this can be a problem. Careful slow evaluation of results in a compositional context has probably never hurt anyone. Next to that, the *Musical Organism Evolver* also presents the user with a visual aid, making it more convenient in practical use.

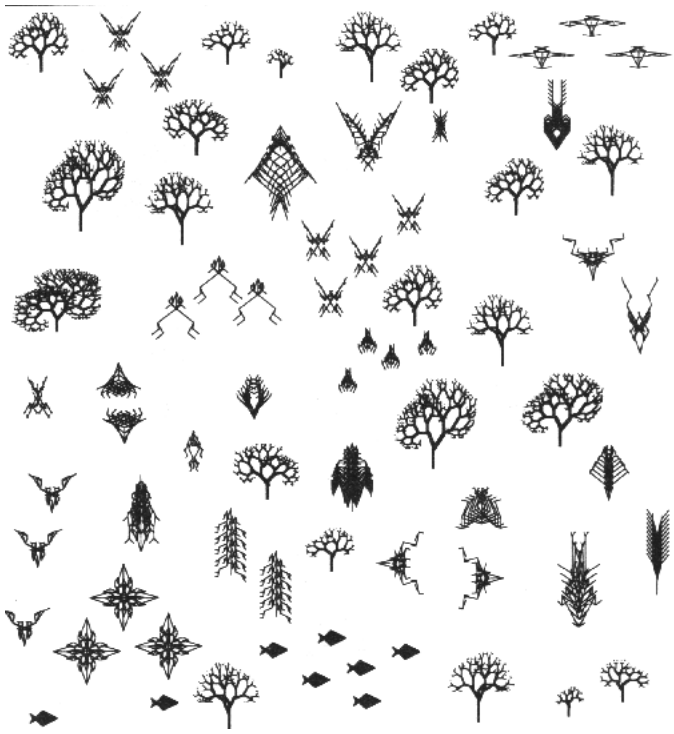

Figure 3.2: Several Biomorphs

The implementation is based around the existing algorithmic composition software *MIDIFORTH*, consisting of functions able to operate on MIDI information. He explains his *Arbitrary Pattern Generator*, where a pattern can be generated to operate on various MIDI data types. All the parameters of the generator are mapped in a '1-to-1'-fashion to the genome, which is a fixed parametric representation.

He extends this functionality greatly by adopting somewhat of a GP approach where genes aren't simple bit sequences, but represent higher level data structures. These structures are comprised of 128 bytes that encode a specific function from his *MIDIFORTH* system. The 128 bytes are structured as follows,

```
1 byte: operation_code
1 byte: grouping_structure
126 bytes: operation-dependent parameter fields
```

Each algorithmic composition function has a `operation_code` and the parameter fields will be populated with parameters specific to that function. Depending on the algorithmic function, not all the bytes for parameter fields are used. On the use of this extended system, on top of GA for note patterns he writes,

> "I believe that the imposition of this structure, above and beyond the structure evolved by the genetic algorithm process, is essential to the musical success of this project." (Degazio, 1997, p. 31)

He also theorizes on the use of 'application-specific' fitness functions that would enable automated fitness evaluation. For example, formalistic evaluation based on 'contrapuntally correct' percentage of notes compared to a cantus firmus.

3.1.2 Unconventional techniques

So far we have seen some elementary examples of the application of GA to sound objects, notes or even phrases. An unconventional approach might be taken on a lower level, with the application of GA to time-domain waveforms. (Magnus, 2004)

Magnus defines a gene as a segment of a waveform between two zero crossings. This directly shows the metaphorical approach to GA. The 'lifespan' of an individual is determined by the playback duration. A choice which already will push the audible results in a certain direction. The fitness of a waveform is determined by calculating its similarity towards a target waveform. An error rate is calculated by summing the difference between the desired and actual amplitude, of each instantaneous sound pressure value.

Her system selects the two highest ranking individuals from a population and performs single-point crossover. There are several mutation operations possible on the waveform segments (genes): reverse, remove, repeat and swap. Amplitude based mutation is also possible: addition with a random value, exponentiation and normalization.

There is also a 'spatial environment' defined, that she describes as:

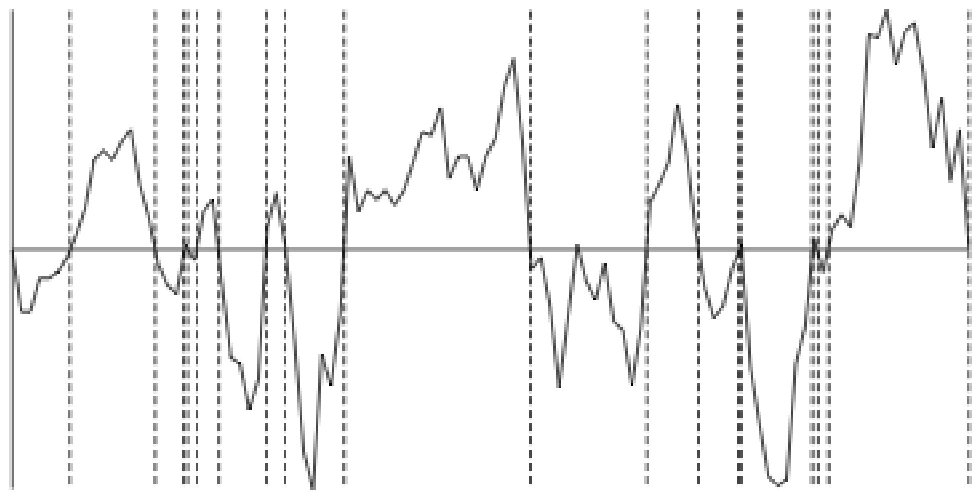

Figure 3.3: Waveform genome (Magnus, 2004)

> "For a given piece, the world will be characterized by some number of locations. These locations may be mapped spatially onto speakers. The environment at these locations will initially be defined by some target waveform and some set of mutation probabilities. Individuals within the population will have some probability of migration. In this way, sounds with new characteristics will enter each location, enhancing biodiversity." (Magnus, 2004, p. 3)

This approach can be seen in a similar category as many of the sample-based techniques used in contemporary computer music.

3.1.3 From musical interaction to audible traces

A more metaphorical approach of evolutionary and cybernetic mechanisms to the construction of a site-specific sound installation can be found in Di Scipio (2003). He theorizes on the subject of interaction, specifically that of interactive signal processing. He takes a *bio-cybernetic*, or eco-systemic perspective on the process of interaction for musical utility. Live electronic music performance can be seen as a simple feedback loop involving; performer, instrument and sound. He comments,

> "In a broader perspective, in this standard approach, the sound-generating system is not itself able to directly cause any change or adjustment in the 'external conditions set to its own process, i.e. it has no active part in determining the control data needed for its changes of internal state to take place. The only source of dynamical behaviour lies in the perfomers ears and mind."

Proposing a solution, through the use of the ecosystem concept (to be discussed later on) he writes,

> "(...) *a shift from creating wanted sounds via interactive means, towards creating wanted interactions having audible traces.* In the latter case, one designs, implements and maintains a network of connected components whose emergent behaviour in sound one calls music."

In essence he proposes a sound installation with interactive properties, emphasizing the interaction with the *local* ecosystem; the acoustic environment. Noise, or ambience, is a crucial element—it is the medium through which the dynamical behavior of the system can exist.

The *Audible Eco-Systemic Interface* (AESI) is a system with the following control flow, in a local acoustic environment:

1. Sound is played back from a computer, through loudspeakers.
2. Sound is fed back to the computer via microphones (crucial placement in relation to the loudspeakers).
3. Analysis is performed on the microphone signals, relevant sonic features are extracted.
4. Control signals are generated from the extracted analysis data. Signal processing modules are controlled by these signals.
5. The processing modules operate on the original played back sound.

6. A difference-signal is calculated from the microphone and original signals.

7. The difference signal (in effect, the acoustic room modes) is used to adapt other signal processing parameters, over a variable time-span.

These functional properties will be familiar to the person interested in site-specific sound installations or the mechanisms and properties of acoustic space. Art installations concerning themselves with sound and space have long history in the usage of feedback mechanisms.

3.2 Towards virtual ecologies

A distinctive direction, utilizing evolutionary computation techniques, is that of the ecosystem. The concept of a virtual ecosystem tries to provide a solution for a problem such as the fitness bottleneck. Next to that there is the potential for emergence on many levels.

Todd and Werner (1998) drawn an analogy between algorithmic music systems and the creation of the monster by Victor Frankenstein. This could be further expanded on and seen as an analogy between artificial life and algorithmic music systems. One which especially holds merit in the context of a virtual or artificial ecosystem.

They formulate a problem present in probably any algorithmic music composition system as, the structure/novelty trade-off. In their words:

> "Thus the trade-off: More structure and knowledge built into the system means more reasonably structured musical output, but also more predictable, unsurprising output; less structure and knowledge in the system means more novel, unexpected output, but also more unstructured musical chaff." (Todd & Werner, 1998, p. 2)

In their work they divide algorithmic composition techniques into three categories:

- Formal rule-based
- Example learning
- Evolutionary (descent with modification)

Descent with modification of course refers to the reproductive capabilities a evolutionary system can posses. In the discussion on evolutionary techniques they propose a system of *coevolution* where artificial composers as well as critics are evolved, in order to produce musical material. According to them; evolution might lead to beauty or, monstrous creations.

On the question of what type of fitness function or, testing criteria to use, they return to the three techniques listed above, adding interactive testing. Coevolution can metaphorically be viewed as the exchange of information between a music creator and music critic;

> "In the same way, the two halves of the creative loop, creator and critic, or performer and audience, should dance around each other, shaping and being shaped by the others behavior. This is a very general principle–essentially that of a feedback loop–that lies at the heart of a wide range of dynamic systems, whether within the psychology of a single creative mind, or between a pair of interacting individuals, or among the groups generating and responding to artifacts in a particular culture, or even among species interacting in an ecosystem. If only one side of any of these systems can change, then it will only change until it is in line with the other fixed component, and then creativity and innovation will stop. Both sides must be free to adapt to the other for continuing novelty to be generated." (Todd & Werner, 1998, p. 14)

Through coevolution, selective pressures are exchanged between the creator and the critic. Coevolution can,

- **reduce trickery** on the part of the creators, in order to achieve high fitness (new fitness criteria will continuously evolve)
- **produce synchronic diversity**, through speciation (sexual selection will generate rich variation)
- **generate diachronic diversity**, through arms races between species (traits change in response to adaptation)

Summarized to:

> "Thus, to generate musical diversity both across time and at any given instant–both diachronically and synchronically–we must build a system that can create a multitude of distinctly defined 'species' within one population, and that can further induce those species to move around in musical space from one generation to the next. Sexual selection through mate choice allows the former, leading a population to cluster into sub-populations with unique (musical) traits and preferences." (Todd & Werner, 1998, p. 15)

No doubt, this will be a highly complex system to construct. An explanation of their actual implementation is given in the paper referred to above.

Their coevolutionary model was inspired by evolution of birdsong, where critical females choose which singing males to mate with. In the end their goal of musical diversity and novelty was reached, but at the cost of 'musical structure'. They finally conclude with,

> "Musicians intent on creating algorithmic composition systems with the spark of human creativity would do well to adopt this combination of coevolution, learning, and rule-following, and thereby with luck avoid the horrors that were visited upon Victor Frankenstein and his creation." (Todd & Werner, 1998, p. 21)

And so it seems the task of constructing, a eco-systemic evolutionary algorithmic music composition system, is not a easy one.

3.2.1 Artificial Life for Music

A similar agent-based system to the one described above, was created by Dahlstedt and Nordahl (2001). The departure point is their statement that interactive evolution, although being a optimization technique, inherently emphasizes the creative nature of evolution.

A similar point is made by Goldberg (1994), on the relation of innovation and creativity to the functioning of GAs.

> "What is it we are doing when we are being innovative or creative? Often we take a set of solution features that worked well in one context and a set of solution of features that worked well in a different context and bring them together—possibly for the first time— to try to solve the problem at hand.
>
> This emphasis and juxtaposition of human creativity is similar to the selection and recombination of genetic algorithms. Of course, the willy-nilly juxtaposition of GAs seems much less directed than that of our own creative efforts, or so we would like to believe; nonetheless, the alignment of the fundamental processes in the two situations is appealing (...)" (Goldberg, 1994, p. 4)

The system constructed for *Living Melodies*, is an artificial world inhabited by coevolving creatures that produce sounds, and judge each others sounds. The vocalization of the creatures is inspired by the mechanisms of animal mating calls. A few important properties of the system follow.

- Two part genome: sound and procedural, both variable in length. The sound genome contains notes, ordered by decreasing listening pleasure for the individual. The procedural genome contains movement or sing/listen actions.
- The world is a square lattice, with or without boundaries. Also the 'air' (for sound propagation) is modelled. The current sound (note), amplitude and direction for each discrete cell is modelled.

- There can be food for the creatures to consume. If consumed, their life-points (energy) increase.
- The creatures have life-points, decreased by time, mating or singing.
- Creatures can reproduce if they are near each other, their life-points are high enough and their listening pleasure has reached a certain value. A child is spawned within a few cells of the parents, and one of the parents is displaced a few cells, to prevent successive mating attempts.
- A creature has 3 sensorial inputs that can be used in the procedural genome. Notes of neighboring creatures in several directions, sound volume at the current coordinates or age of the creature, are some of the sensor values that can be detected.

Deduced from these properties can be the fact that there are numerous parameters that control the dynamics of this system. One important parameter concerns the filtering on the vocalization of the creatures. Filtering can be done according to listening pleasure or life-points values, each resulting in different musical results. Filtering according to listening pleasure only vocalizes the 'happiest' individuals, whereas life-points filtering is a more steady filtering method.

Their work is concluded in an relativistic and optimistic fashion,

> "In this world, the main feature we emphasize is not so much that the individual creatures develop interesting sounds, but that of listening to the process of evolution of simple creatures working together, triggering each other and playing a small part each. The process is more like creating a singing ant colony than trying to create a Pavarotti ant (that will be our next project)." (Dahlstedt & Nordahl, 2001, p. 247)

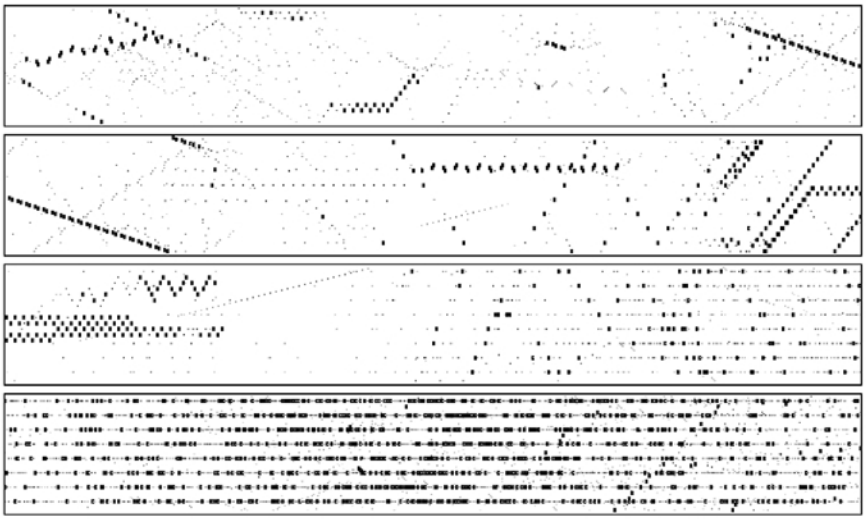

Figure 3.4: Sonogram of Living Melodies (Dahlstedt & Nordahl, 2001)

3.3 Principles of Complex Adaptive Systems

In light of the virtual ecosystem concepts presented above, I will take one final sidestep into complex adaptive systems theory. In "*Hidden Order*", Holland (1995) explores general principles of CAS. The principles are divided into properties and mechanisms, and serve as description of phenomena that can be encountered when studying CAS. Taking this as a departure point, these phenomena could serve simulational realism—enabling many levels of emergence—when constructing a virtual CAS.

Aggregation (property) of agents.

In the first sense, to simplify and understand the system. *How* an agent is defined within the system, and what their common properties are. In the second sense we are concerned with *what* the agents do, and what this aggregate behavior leads to.

Tagging (mechanism) facilitates the formation of aggregates.

The ability to tag, or mark, objects (agents) in order to track their behavior. This enables other agents to 'select', or simply distinguish. Think of it as rudimentary information exchange.

Non-linearity (property) of the aggregate behavior.

The interactions among agents cause aggregate behavior, that is not explained by simply summing or averaging its constituent parts. Think of the mathematics of chaotic systems, for example the *logistic map*.

Flows (property) of resources or information exchange.

A possible activity in a network, between nodes (agents). Think of signal flow between signal processing and generating elements in a sound synthesis network. Two important properties of flows are the *multiplier effect* and the *recycling effect*. Multiplication is the scaling of flows in each consecutive node. As the name implies, often manifesting in an increase of resources. The recycling effect considers 'feedback loops of flow', where the resource in the flow is 'under constant re-use' at several nodes before it ends at its natural end.

Diversity (property) of agents in a system as a cause for novelty.

Closely related are ecological concepts like, *niche construction*, *mimicry* or *symbiosis*. All constructs which enable diversification of the environment. The continuous adaptation of agents leads to diversity; enabling the adaptation of other agents (reacting to others, etc.). Diversity is a dynamic pattern.

Internal models (mechanism) for anticipation.

Where anticipation within agents, affects behavior between agents. Evolution rewards successful actions resulting from anticipation of consequences. Possible consequences can be modelled. There are *tacit* and *overt* internal models. The tacit model is taking a simple action, under a implicit prediction ('there is more energy in direct sunlight'), in order to reach a desired future state. The overt model, explicitly explores several possible outcomes of an action. Think of contemplating chess moves.

Building blocks (mechanism) of experiences in internal models.

Internal models can only be useful if the resulting anticipations conform with the actual experienced result of an action. The experiences gained can in turn be building blocks for other internal models, allowing adaption to novel situations. Think of learning. Tacit building blocks are usually combined and discovered on a evolutionary timescale. Overt building blocks could change on a much shorter timescale.

These theoretic principles are one way to look at CAS. They are not *the* way, as is often the case in theoretical science. I do believe that distilled versions of these ideas can be powerful tools, enabling emergence, novelty and in general interesting behavior for compositional practices.

Chapter 4

Ecosystem approach and conclusions

The previous chapter has described the possible implementation complexities involved in constructing a virtual ecosystem. The following chapter will describe an experiment into creating a virtual ecosystem, in order to gain a understanding of the software complexities involved. The software is written in the SuperCollider scripting language, contained in three classes. There are numerous parameters of the ecosystem that can be controlled. The functionality of the current system will be discussed. Some final remarks in relation to the previous chapters will be made.

4.1 Creatures

Creatures is a simple ecosystem where individuals roam a virtual world. The individuals are able to make sound. The world contains Food for the individuals to consume and Speakers enable them to make sound. Each individual contains its own unique genome. The genome can be translated into an audio synthesis network. When individuals bump into each other, they are able to mate and produce offspring.

When a *World* is started, a number of *Individuals* is spawned. The world consists of a 2 dimensional lattice where at random locations a certain amount of food, and a certain amount of speakers are materialized. The world has soft borders, individuals that bump into them are transported to the other side.

The World is represented in the class `World`, the Individuals in the class `Individual`. The genome to phenome conversion resides in the class `Gen2Graph`, which is a modified version of `GAGraphCreator2` (Stowell, 2006). Genetic single point crossover functionality is provided by the RedGA library (Olofsson, 2011). All software is distributed under the GNU General Public License.

All sourcecode for the system and sound examples with an explanation can be found at (Jöbses, 2011).

4.1.1 Simulation Functionality

The global functionality of the individuals is as follows.

- A Individual is spawned in a World. It is able to move one step, either left/right or up/down. Moving costs life points.
- The Individual will have a certain amount of life points to begin with. Life points can be increased by finding food. Mating will decrease the life points.
- Each Individual has a variable length, real valued genome. This genome contains a extensible parametric representation of audio synthesis units.
- When an Individual finds a Speaker, it is able to translate the genome into an audible synthesis network. The volume of the Individual is scaled to the individuals total life points.
- When two Individuals with sufficient life points mate, a single point genetic crossover is performed on their genome. With a certain probability the parents will either die or be displaced. Before mating; parents that are making sound, will be silenced.

The functionality of the World contains the following properties.

- The World is created according to specified x and y values, representing the dimensions. The specified number of Individuals is spawned at random locations, avoiding food or speakers.
- Food and Speakers are distributed randomly trough out the world. Each Speaker contains a certain rank; newbie, advanced or elite. Allowing for different types of synthesis networks, when a Speaker is used.
- The World is run in cycles, of a pre-determined timestep, in which each Individual's standard check routine is performed.
- This routine is performed during the life cycle of an Individual.
  - A new location, based on the current one, is generated.
  - Boundary conditions for this new location are checked.
  - The collision check is performed; the possibility of two individuals bumping into each other.
  - When the life points of the colliding individuals are high enough, mating is allowed. Otherwise, life points are decreased because of the collision—and the main routine is reset to the beginning.
  - The two final checks are for Food or a Speaker on the new location. If the Individual hasn't already encountered a Speaker, it is allowed to use it for making sound.
- At the end of each World cycle, dead Individuals (less than 0 life points) are silenced and cleaned up.

4.1.2 Sound Functionality

The `Gen2Graph` class constructs a SuperCollider function that can be interpreted as a *SynthDef.* This synthesis definition describes the audio synthesis network that originates from the Individuals genome. According to the rank contained with the speaker, this class constructs a different type synthesis network. Increasing in complexity, the ranks of newbie, advanced or elite allow the usage of more advanced audio- and control-rate Unit Generators

(UGens). At the elite level there are 59 different audio- and control-rate UGens and mathematical or sequencing operations available for the Individual.

The conversion is completely deterministic; based on the genome and rank, the same UGen graph is constructed. Due to the nature of the complete process there is the possibility that a UGen graph will not make any sound. Based on the number of genes in the genome, the number of UGens or other operations used, will vary. All interconnections are mono with a range of -1.0/+1.0, with the possibility of branching the output of one UGen or, other operation, to several others.

According to the skeleton definition of the UGens (which input requires, what type of, possibly scaled, input signal) the conversion constructs a complex UGen graph. This final graph is preceded by a default number of 'input UGens', that provide input to the genetically constructed synthesis network.

This complete conversion procedure allows for a large number of different synthesis networks and so, audible results. As there is no 'intelligence' in the procedure for what types of interconnections have proven audible results or, a certain audible 'paradigm'—this intelligence is needed elsewhere, and as such provided in a limited way with the possibility of Individuals mating.

4.2 Conclusions

Provided is a broad overview on the usage of evolutionary biological and complex systems concepts in music. The question whether music composition could benefit from search, learning or optimization methods—has been answered by some of the examples in the third chapter. As music is strongly related to discovery, search techniques could definitely benefit the composer. Machine learning is a complex subject, maybe unnecessary complexifying the already intricate concept of a virtual ecosystem with coevolving entities. As we have seen, optimization techniques merely enable style imitation.

I am convinced that there is still a lot to be discovered in the application of evolutionary and complex system concepts to musical composition.

The most appropriate warning being that of too much complexity. Drowning in extra-musical systems during implementation serves no goal, except that of personal experience. Extensive research and *design* is needed before attempting to construct such systems. There probably is a whole body of scientific work that could aid in a less painful way of constructing virtual ecosystems. Even if this is true, one of the fundamental problems of, at least computer, music remains. The problem of mapping extra-musical constructs to musical events, objects or gestures—or, even harder, the mapping to abstract sonic properties and events.

We have seen that analogies and a metaphoric approach to the concepts provided in the first two chapters, can mitigate or solve, among others, the problem of mapping. On the matter of *analogies and metaphors*, I want to leave you with one final lengthy quote summarizing many of the concepts already touched upon.

> "When we appraise great art, music, architecture, literature, or other human outputs that are commonly associated with the term "creativity," some features that commonly distinguish great from not-so-great work are the number, quality, and type of interconnections that the work makes with other works and other aspects of life. For example, we look at a piece of art or literature for what it *represents* (the direct connections it makes with the object or situation it portrays), for its *allusions* (the direct interconnections it makes to other works of art or literature), and for its *symbolism* (the meaning we infer by analogy through conceptually constructing interrelations between the work of art and some situation not directly represented in the work).
>
> In things creative, analogy *transfers* a set of fit features to the subject work, thereby infusing it with a well-adapted richness, a complexity that would have been difficult to obtain by other means. In genetic algorithms a method of *analogizing* or *metaphoric transfer* would enable *deep, difficult building blocks* to be transferred from a well-understood situation to a poorly understood situation without the cost of explicit search." (Goldberg, 1994, p. 4-5)

References

Blackwell, T., & Young, M. (2004). Self-organised music. *Organised Sound*, *9*(2), 137-150.

Bowler, P. J. (2003). *Evolution: The history of an idea* (3rd ed.). University of California Press.

Dahlstedt, P. (2001). Creating and exploring huge parameter spaces: Interactive evolution as a tool for sound generation. In *Proceedings of the 2001 international computer music conference, habana, cuba* (p. 235-242).

Dahlstedt, P., & Anders Nilsson, P. (2008). Free flight in parameter space: A dynamic mapping strategy for expressive free impro. *LNCS 4974: EvoWorkshop 2008*, 479-484.

Dahlstedt, P., & Nordahl, M. (2001). Living melodies: Coevolution of sonic communication. *Leonardo*, *34*(3), 243-248.

Degazio, B. (1997). The evolution of musical organisms. *Leonardo Music Journal*, *7*, 27-33.

Di Scipio, A. (2003). 'sound is the interface': from interactive to ecosystemic signal processing. *Organised Sound*, *8*(3), 269-277.

Galanter, P. (2008). Complexism and the role of evolutionary art. In *The art of artificial evolution: A handbook on evolutionary art and music* (p. 311-332). Springer, Berlin.

Garcia, R. (2001). *Automatic generation of sound synthesis techniques.* Unpublished master's thesis, MIT.

Goldberg, D. (1994). The existential pleasures of genetic algorithms. (IlliGAL report no. 94010)

Goldstein, J. (1999). Emergence as a construct: History and issues. *Emergence*(11), 49-72.

Holland, J. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. University Michigan Press.

Holland, J. (1995). *Hidden Order: how adaptation build complexity.* Basic Books.

Horner, A., & Goldberg, D. (1991). Genetic algorithms and computer-assisted music composition.
(IlliGAL report no. 61801)

Jeffress, L. (1951). *Cerebral mechanisms in behavior: The hixon symposium.* John Wiley & Sons.

Jöbses, S. (2011). *Creatures.* (`http://rxj.nl/creatures/` (Retrieved August 2011))

Jones, D. (2008). Atomswarm: A framework for swarm improvisation. In *Evoworkshops 2008* (p. 423-432). (LNCS 4974)

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection.* MIT Press.

Magnus, C. (2004). *Evolving electroacoustic music: the application of genetic algorithms to time-domain waveforms.* (in Proceedings of the 2004 International Computer Music Conference)

Mitchell, M. (1996). *An introduction to genetic algorithms.* MIT press.

Mitchell, M. (2009). *Complexity: A guided tour.* Oxford University Press.

Nierhaus, G. (2009). *Algorithmic composition.* SpringerWienNewYork.

Olofsson, F. (2011). Redga supercollider extension [Computer software]. (`http://fredrikolofsson.com/` (Retrieved August 2011))

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. In *Computer graphics* (p. 25-34). (SIGGRAPH '87 Conference Proceedings)

Solomos, M. (2005). Cellular automata in xenakis' music. theory and practice. In *Definitive proceedings of the international symposium iannis xenakis.*

Stowell, D. (2006). Ga library for supercollider [Computer software]. (`http://www.mcld.co.uk/supercollider/` (Retrieved August 2011))

Todd, P. (1989). A connectionist approach to algorithmic composition.

Computer Music Journal, *13*(4), 27-43.

Todd, P., & Werner, G. (1998). Frankensteinian methods for evolutionary music composition. *Musical networks: Parallel distributed perception and performance*.

Tudor, D. (1995). *Neural synthesis nos. 6-9.* (2CD Album, Lovely Music Ltd., in Liner notes)

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *49*, 433–460.

Wilson R., K. F. (1999). *The mit encyclopedia of the cognitive sciences.* Bradford Book.

Xenakis, I. (1992). *Formalized music.* Stuyvesant, NY. Pendragon Press.