Chapter 3

Computer Automated Algorithmic Composition in Research and Practice

In the 1980s, Berg (1987) suggested that the fundamental contribution of the computer to music was that it "empowers the composer to hear that which could not be heard without the computer, to think that which could not be thought without the computer and to learn that which could not be learned without the computer. The computer can allow a composer to write music that goes beyond that which she is already capable of." (p.161). One of the earliest experiments in computer automated composition put its navigational powers to test in exactly this way. The Illiac suite (Hiller and Isaacson (1959)) consisted of four 'experiments' carried out on the ILLIAC I mainframe of Illinois University. The first two experiments were designed to demonstrate that standard musical techniques could be handled by computer programming, the third to show that computers might be used by contemporary composers to extend present compositional techniques, while the fourth experiment was intended to "show ... that computers might be used in highly unusual ways to produce radically different species of music." (ibid p.4). The article Hiller wrote at the time was pounced upon about by the popular press who whipped up a storm of controversy which lasted throughout much of the 1980s. The musical establishment of the time were so antagonised by this scientist operating his machines in the name of art that apparently neither Baker's Biographical Dictionary of Musicians nor the New Grove Dictionary of Music and Musicians recognised his existence until just before his death. Fortunately this hysteria has now subsided, and the computer, if not the algorithm, is now largely accepted as an integral part of compositional practice.

It was not only players and composers who took an interest in computational approaches: music theorists also looked to the quasi-scientific rigour of computational modelling as a means of raising their profile. In post-war universities, the hard sciences occupied pride of place, particularly in American academies. Disciplines like music theory tried to gain credit, making themselves look as hard as possible by adopting scientific language and symbol systems (Cook (1998)). By the late 1970s, attempts were made to implement the more objective of existing analysis techniques such as Schenkerian analysis in computer models (Cook (1998)). Others such as Sundberg and Lindblom (1976) attempted to objectify musicology by applying cognitive science methods. They developed new grammars specific to musical styles aiming "to describe facts in music theory by means of generative rule systems" (p.100).

Several researchers interested in the cognitive processes underlying musical composition also built algorithmic composition systems as cognitive models. Steedman (1984) declared that "generative rules are only really interesting when they can be used to drive

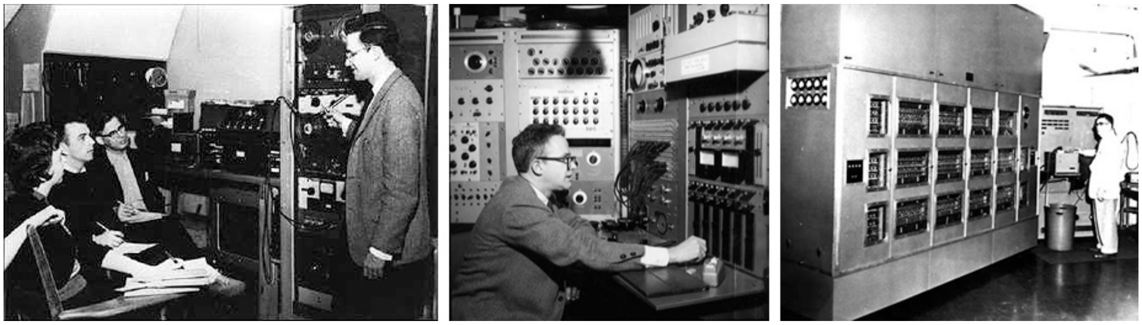


Figure 3.1: Hiller with colleagues in the Electronic Music Studios, University of Illinois (left), at the controls of the console (middle), and in front of the mainframe ILLIAC I (right).

a model of human performance," (p.75), and implemented a model of blues progressions. Johnsson-Laird (1991) set out to develop a theory of "what the mind has to compute in order to produce an acceptable improvisation" (p.291). Related research activities have gathered momentum in recent years with the burgeoning field of computational creativity which is beginning to tackle creativity in general as well as issues such as the evolution of musical behaviours (e.g. Miranda (2000b), Bown and Wiggins (2005)).

As this rapid historical jaunt suggests, the field is wide, and has a rather large number of different teams playing on it: those developing digital performance systems are only one of a large number of different groups exploring the use of algorithmic methods in music. Under the most inclusive use of term, the field draws ideas and motivations from a range of disciplines including music practice, musicology and music theory, AI, computer science, cognitive science, and more recently, evolutionary theory. Research, practice and investigation take place not only within AI departments, music departments, dedicated computer-music research centres, on new media courses, and in arts colleges, but increasingly outside institutes on the laptops of the world's more experimental electronic music producers and performers.

Testimony to the breadth of the field are the number of labels which have been used to describe overlapping, sometimes identical projects: algorithmic composition, automatic composition, composition pre-processing, generative modelling of music, generative music, active style synthesis, empirical style modelling, computer aided composition, computer composing, computer music, procedural composition, score synthesis, computer aided automated composition, machine musicianship, computational creativity and computational musicology. Attempts to distinguish these terms have been made in Spiegel (1989), Cope (1991) (p.220), Burns (1994) (p.195), Miranda (2000a) (pp.9–10), Taube (2004), Gerhard and Hepting (2004) (p.505), Ariza (2005) Roads (1985) (p.175), Ames and Domino (1992) (p.55) and Pearce et al. (2002).

This chapter focuses on compositional applications. Section 3.1 provides a rapid survey of the principle techniques that have been applied across the whole field. There are a great many of reviews and surveys of the field, so this overview is meant merely as a means of marking out the corners of the pitch and demonstrating the range of equipment available. Methods used to extract or imbue forms of musical 'knowledge' are considered separately from 'extramusical' approaches that are developed to explore new musics. Section 3.2 introduces some distinctions suggested by Pearce et al. (2002) which differentiate between scientific and creative applications. This section also raises some issues, perhaps specific to the academic community, which arise due to the cross-disciplinary nature of the field. Section 3.3 takes a more practical perspective and examines some of the motivations for using generative processes in composition or performance. The desirable characteristics of algorithms are discussed from this more practical perspective. Section 3.4 aims to gather together these different perspectives to mark out the framework for the current project.

3.1 Techniques

This section gives a brief overview of the some of the main techniques that have been applied to computer automated algorithmic composition. This whirlwind tour is meant only to provide a taste of the principle methods that have been applied to date. There exist many reviews positioned in different corners of the expansive field, adopting a number of different classification and evaluation systems. More detailed surveys of algorithm types are given in Loy (1989), Papadopolous and Wiggins (1998), Dodge and Jerse (1997) p.341, Miranda (2000a), and Todd and Werner (1999). Chronological overviews are given in Hiller (1981) and Burns (1994), whilst Ariza (2005) gives a taxonomy according to various descriptors[1]

At the start of the new millenium Miranda (2000b), asserted that the discussion as to whether or not computers could compose music was no longer relevant. Suggesting that one of the greatest achievements of AI to date was the demonstration that machines can compose music "of incredibly good quality", he cites Cope's EMI system (e.g. Cope (1991)), which is famously reported to have fooled the critics with its compositions in-the-style-of X, to demonstrate that computers *can* compose if programmed accordingly. What these AI systems are good at is mimicking established musical styles, either by being hard wired, deriving statistical patterns from exemplar of the extant cannon, or searching spaces of musical possibility using Evolutionary Computation (EC) techniques guided by a human ear or imbued more implicitly with musical knowledge. He goes on to suggest that the study of 'new' music is trickier, partly because "it is hard to judge what the computer creates in such circumstances because the results normally sound very strange to us. We are often unable to judge these computer-generated pieces because they tend to lack those cultural references that we normally hold on to when appreciating music." (ibid, p.1). Issues of 'judgement' will be discussed in Section 3.2 and 3.4, but this distinction between the replication of existing styles versus exploration of new ideas forms a natural division which is adopted here.

In the first half of this section, an outline is given of the various techniques that have been applied in an AI framework to imbue a system with some degree of musical knowledge. These are divided into *rule based*, *probabilistic*, *learning* and *evolutionary* systems. The boundaries are a little fuzzy as will become evident. These are often considered in terms of the extent to which a good balance of novelty and structure can be achieved. This may be a little over simplistic, but provides a good starting point from which to compare the characteristics of algorithms. In the second half a range of more experimental approaches is given. In current practice, there are a vast number of musicians using all sorts of personalised methods, many of which are never discussed, so the examples are primarily representative of those presented in academic research.

3.1.1 AI Methods of Embodying Musical Knowledge

Algorithmic composition systems are often compared according to the potential for achieving a suitable balance of structure and novelty (e.g. Todd and Werner (1999)). A random

---

[1] These descriptors are scale, process-time, idiomaffinity, extensibility, event production, sound source, and user environment.

number generator is an algorithm, but by itself has little structure[2]. The procedure:

```
IF note = A then play B, ELSE IF note = B THEN play A; note = A
```

is an algorithm but such an incessant trill, although structured, would get a little dull after a second or so. The attraction of many AI techniques then, is that they can not only be used to embody, or derive musical 'knowledge', but this 'knowledge' can then be used to generative a variety of different examples within the bounds defined by it.

Explicit Rules

The simplest form of algorithmic composition comes in the form of explicit sets of rules. These may be simple sets of transformations used to organise pre-existing material, or more complex sets of rules seen to embody aspects of a specific style (for a review see Loy (1991), Todd and Loy (1991)). The *Musikalisches Würfelspiel*, a popular parlour game of the 18^{th} century is often used to illustrate the former. A set of Minuet and Trios are composed, lets say twelve. Each bar is interchangeable across sets. The players then throws a pair of dice, once for each bar of the piece. The original material is composed such that any combination of individuals bars works harmonically, giving the players the joy of composing their own variations. Such an approach obviously guarantees a well formed result but leaves little room for novelty.

William Shottstaedt's automatic species counterpoint program writes music based on rules from Johann Joseph Fux' *Gradus ad Parnassum*, a counterpoint instruction book from the early 18th-century aimed at guiding young composers to recreate the strictly controlled polyphonic style of Palestrina (1525-1594) (Grout and Palisca (1996)). These 'guidelines' are then implemented in a formal language which allows some probabilistic deviation.

> ″The program is built around almost 75 rules, such as ′Parallel fifths are not allowed′ and ′Avoid tritones near the cadence in lydian mode.′ Schottstaedt assigned a series of ′penalties′ for breaking the rules. These penalties are weighted based on the fact that Fux indicated that there were some rules that could never be broken, but others did not have to be adhered to as vehemently. As penalties accumulate, the program abandons its current branch of rules and backtracks to find a new solution″ - Burns (1997)

Some see such programs as helpful in advancing a more generalised understanding of compositional practice, but in practical terms such programs leave little room for flexibility, and the adherence to such strict rules in actual practice is questionable (see Section 3.2.2).

Grammars

Various forms of grammars have been used in musical analysis as a means of representing hierarchical structure (e.g. Schenkerian analysis or Lerdahl and Jackendoff's (1983) Generative Theory of Tonal Music). When implemented as a computer program, grammars can be used to generate musical material within the specified structural form. The generation of traditional Jazz progressions has been tackled using a grammatical approach. Steedman (1984) devised a generative grammar for chord progressions in twelve-bar blues and refined it using categorial grammars (Steedman (1989)). Johnsson-Laird (1991) also applied grammars to the generation of Jazz chord progressions and bass line improvisations.

---

[2] Random number generators are probably the most widely used devices in Generative Music practice, but generally in the context of some other algorithm.

The grammars themselves can be derived by hand, or created automatically. For example David Cope's *Experiments in Musical Intelligence* project, which focuses on the understanding of musical style and stylistic replication of various composers (Cope (1991), Cope (1992)), uses pattern matching to extract the 'signatures' from two or more existing works. Augmented transition networks are then used to rearrange these patterns into meaningful structures.

Probabilities and Stochastics

Combining a rule based approach with some form of probabilistic reasoning gives greater scope for variation in output. Arguably some of the most successful examples of algorithmic composition are based on probabilistic processes which have designed intuitively by seriously musical minds. Although the inner workings of George Lewis' *Voyager* system are somewhat shrouded in mystery, in talks and discursive papers (e.g. Lewis (1999)) he outlines that the generative behaviour of the system is developed essentially from white noise, which is shaped and filtered with a series of stochastic rule sets. These stochastic processes have been designed and cunningly ordered by his intuitions as a Jazz improviser:

> "For the Voyager program, in addition to the idea of state, it is the sheer number of decisions, as well as their character and order of preference, that leads to a sense of directedness in the music that belies its humble origins in white noise." - Lewis (1999), p. 108

Similarly Nick Collins' Squarepusher patches, released as part of his BBcut SuperCollider library (Collins (2003a)) do an incredibly convincing job of emulating the intricately hand carved break-beats of Tom Jenkinsen, to the point where many cognoscenti of the breakcore universe adopt these procedures rather than sweat over sequencers with the GUI magnification turned up high.

	C3	D3	E3	F3	G3	A3	B3	C4	
C3	0.2	0.2	0.2	0.0	0.2	0.0	0.0	0.2	If C3, then C3, D3, E3, G3 or C4
D3	0.33	0.0	0.33	0.0	0.33	0.0	0.0	0.0	If D3, then either C3, E3, G3
E3	0.0	0.5	0.0	0.5	0.0	0.0	0.0	0.0	If E3, then either D3 or F3
F3	0.33	0.0	0.33	0.0	0.33	0.0	0.0	0.0	If F3, then either C3, E3 or G3
G3	0.25	0.0	0.0	0.25	0.25	0.25	0.0	0.0	If G3, then either C3, F3, G3 or A3
A3	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	If A3, then B3
B3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	If B3, then C4
C4	0.0	0.0	0.0	0.0	0.0	0.5	0.5	0.0	If C4, then either A3 or B3

Table 3.1: Example of a first-order Markov transition matrix (right) derived from probabilities specified in rules (right) for a hypothetical melody in C major.

Stochastic processes have also been widely employed. Stochastic processes evolve in time according to probabilistic law. One of the most widely explored are Markov chains. Markov chains have a discrete number of steps, with probabilities governing the transition from one to the other. These can be designed by hand. For example if we wanted to make a first-order Markov transition matrix to generate a melody in the key of C (Table 3.1, left), we could define which notes we could move to given the current note (Table 3.1, right) and then convert these into probability chains for each note. A second-order process would take the last two notes into account to define the subsequent note and so on. Simple applications include Brooks et al. (1993) who use matrices derived from an analysis of a series of traditional hymns to generate hymn-like melodies. A more sophisticated implementation was carried out by Ames and Domino (1992) who generated music in specific styles using Markov chains with transformation probabilities created from analyses of existing musical arrangements.

Rather than deriving probabilities by hand, learning systems have been employed to extract statistical information about the structure of a given set of inputs. Loy (1991) and Jones (1981) give early examples where analysis of overall pitch-transition probabilities is performed on a collection of set musical examples. Based on how often particular pitches follow each other in the examples, new compositions can be constructed with similar statistical structure. This can also be done in real time. *Jam Factory* (Zicarelli (1987)) for example, allows the user to train four 'artificial musicians' using real time MIDI input. Analyses are performed on notes and durations to create transition tables which are then used to generate music with the same transition probabilities. A similar technique is developed in Francois Pachet's *Continuator* (Pachet (2002)). This uses harmonic analysis of performance data and applies harmonic constraints on the material generated by the Markov engine to create a computer response which is impressively close to the performer's style with respect to phrasing as well as harmonic and rhythmic motifs.

Neural Networks

In theory, feed-forward and recurrent artificial neural networks (ANNs) are capable of learning more abstract, or longer term patterns than Markov-processes. This potentially allows higher level of musical structure to be preserved in the output (see Todd (1900), Todd (1989) and Todd and Loy (1991) for early examples of musical applications).

Very briefly, ANNs are connectionist models of information processing which are loosely inspired by the neural network structure of the brain. They consist of interconnected processing units which send signals to one another, changing levels of activation according to the sum of their incoming signals. They are commonly used as non-linear statistical data modelling tools, or for finding patterns in data as different learning techniques enable them to model complex relationships between inputs and outputs. (For a good introduction see e.g. Gurney (1997), and for a more detailed excursion, Haykin (1999)).

In musical applications, ANNs hold promise of being able to extract structural information which can then be used to re-synthesise music bearing a similar structure. Attempts have been made to model particular styles directly, as in Tovainen (1995) who worked on Jazz improvisation, and Hornel and Dagenhardt (1997) where a similar approach is adopted for the emulation of baroque melodic improvisation. Others have taken a more indirect approach, aiming to capture specific features of music. Melo (1988) for example used two cooperative ANNs operating on different levels in an attempt to capture harmonic tension in music. 10 listeners were asked to listen to the last movement of Prokofiev's 1st Symphony and to indicate their estimation of dynamic musical tension by pushing a sprung wheel. This data was split between use as training sets for the ANN and as means of testing their results. After training, the ANN's could predict quite well the tension of an unseen part of the piece. However music generated using the trained system was reported to be 'not as successful'.

Despite initial excitement, this application of ANNs does not seem to be living up to its promise: Mozer famously wrote of his own system that it produced "Music only its mother could love" Mozer (1994). One of the problems is that whilst formally ANNs are capable of extracting pertinent structure and generalising from it, it is often hard to predict exactly *what* structure will be learned: what is pertinent to the musical ear may not be statistically most pertinent.

Having said this, increasingly researchers are thinking more carefully about how

ANNs can be applied to musical situations. For example Ollie Glass' *Breakage*[3] project uses a feedforward ANN to generate breakbeats in-the-style-of the user. The nets are used to extract common patterns from a set of user-supplied drum patterns. The system then generalises, creating a variety of drum patterns bearing similar structure. This approach represents an exciting new take on automation in computer music composition which is further discussed in Section 3.3.

Evolutionary Approaches

Evolutionary computation (EC) techniques, such as genetic programming (GP) (Koza (1993)) and genetic algorithms (GAs) (Holland (1975)) model the Darwinian conception of evolution by random variation and natural selection and act as powerful search mechanisms. Various forms of EC have been extensively explored (see Burton and Vladimirova (1999) for a good overview), spawning a sub field of Evolutionary Art and Music (e.g. Bentley and Corne (2001)).

With respect to the structure/ novelty tradeoff, the approach is promising. The process of artificial evolution itself relies upon the accumulation of 'good solutions' which gives a degree of continuity, variation being provided by the recombination and mutation operators. The problem comes in ascertaining what a 'good' solution is. In natural evolution the criterion is ultimately the ability to survive and reproduce. In engineering applications, or optimisation problems, although non-trivial, it is possible to define a fitness function that can potentially be fulfilled by members of a population. In many music situation there is rarely a unique optimal solution, rather an ongoing process of innovation and refinement. This is arguably something that evolution, and artificial evolution systems excel at, however the problem of defining some measure of fitness criterion or selective pressure remains.

Several approaches have been developed to tackle this problem: some employ theory-derived rule sets to define operators, or desired patterns in output (e.g. Horner and Goldberg (1991)), or introduce a learning module to train an artificial critic (e.g. Spector and Alpern (1995)). One of the technical problems with simple GAs is premature convergence: the population can get stuck in a local minima. In addition, even if a solution is achieved that satisfies the fitness function no further evolution will occur. With no other pressure to do anything else, no further innovations will result. This can be a problem especially when the the population finds a loop hole in the fitness function, producing solutions that meet the criterion, and dominate the population but do not fulfil the goal (i.e. interesting music) that the function was intended to describe.

One solution that circumvents the difficulty of having to come up with a formally defined fitness function which has been popular in art and music applications is to use an Interactive Genetic Algorithm (IGA). In this case the fitness function is simply the subjective preference of a human listener. This is the technique used in the 'breeder art' described in the last chapter. However this approach is not without problems. GAs necessarily require a large population of solutions, each of which must be assessed. This is commonly referred to as the 'fitness bottleneck'. Some solutions to this include the development of hybrid systems, with multiple critics that include rule-based or trained critics operating on a lower level. This has proved successful in terms of reducing the number of examples which must be screened by a human listener (Biles (1994)).

Coevolutionary Approaches

An alternative technical means of circumventing premature population convergence, which does not require human assessment, is to coevolve critics and solutions. It has been shown that coevolution reduces the chance of solutions finding easy ways to 'trick' critics (Hillis (1992)). Coevolution can also produce diversity within a population. This

---

[3] `http://www.blackholeprojector.com/`

synchronic diversity can be generated through sexual selection leading to speciation - splitting the population into subpopulations of individuals with distinct traits and preferences (see e.g. Todd and Miller (1991)). Coevolution can also lead to diachronic diversity, producing traits in a population which change over time as in predator prey 'arms race' models such as described in Futayama and Slatkin (1983). This has been exploited by Werner and Todd (1997) in their model of the evolution of bird song.

These interactions between critic and composer in the coevolutionary approach is seen as a proto-social behaviour, and has been described as a 'cultural approach'. Miranda has produced a simulation in which a society of distributed, autonomous, cooperative agents evolve sound repertoires from scratch by copying each other (mimesis) (Miranda (2002)). The agents produce musical signals to which other agents react, influencing the signals they produce or their proclivity to mate. He argues that as the music created in the system influences the behaviour of the other agents living in the system, it gives it a social role and that the corresponding increase in the richness of the dynamics promises a musically more interesting output. The problem with this is that as a simulation, the system is cut-off from the real-world. Whilst the system may be argued to have cultural significance, culture references will accord with an agent, rather than human aesthetic. In this sense the approach is more amenable to scientific modelling rather than composition.

3.1.2 Techniques Employed in Exploration of New Musics

The application of AI methods described above frame composition as a 'problem solving' exercise aimed at capturing the essentials of existing musical idioms, (Cope (2005), Jacob (1996)). Other practitioners have adopted a more exploratory approach, using mathematical models or data drawn from other sources to explore new musical possibilities. The idiosynchratic nature of this approach makes it impossible to give a fully comprehensive review of all the techniques which are used, but this section aims to provide a representative overview. There may be many more innovative and effective algorithmic procedures in use which are never published or even described in words - primarily because discussion of compositional techniques is not considered relevant for many composers and producers (or maybe they are just so darn good they don't want anyone else to know).

Many of the techniques described above can be adapted to create new music. For example it has been proposed that you could train an ANN or derive Markov tables from two quite different musical styles (e.g. Tuvan throat singing and Black Sabbath) to create strange hybrids. In practice it is likely that the results will be a statistical rather than musical hybrid. In many instances composers have taken an established technique and applied it in unconventional ways. Many different algorithms have been explored, but by far the most common method of employing them is to directly sonifying the numerical outputs of the selected model. Where models are appropriated from existing mathematical or scientific theory (which is common), these tend to be models of growth, pattern formation or movement.

Evolutionary Approaches

Many of the techniques discussed above can be implemented in other ways which take advantage of the *structure* in the process of the algorithm, rather than its problem solving capabilities. In some instances the adaptations made to algorithms are such that its original function is no longer even preserved. For example Waschka II and Cristyn Magnus both used GAs, but rather than employing them as a search mechanism and 'listening' to the winning individual at the final generation, the process itself is sonified, conveying the changes that occur in the population across generations.

Waschka II used this technique to create one of the arias in his opera *Sappho's breath*.

The initial population was small and consisted of Greek and Medieval songs, and melodies that he had composed himself. There was no fitness function (and so no evolution in the traditional sense), individuals being randomly selected to form the next generation. The programme was run for only five generations. Even within this small population Waschka reports that there is a clear diminution of diversity in the population and a tendency towards replication of one or a few individuals over very few generations. These characteristics, i.e. the reappearance of individuals that survive unchanged across generations, and the population convergence, are used to create psychologically meaningful musical effects.

> "As the piece proceeds, the repetition of certain elements or whole individuals allows the listener to make connections with other parts of the work. Finally, the increased similarity of the musical material and the repetitions of motives or measures can provide for both the composer and the audience a sense of closure, and help to bring the piece or section to a natural and satisfying end."
> - Waschka II (2001), p.5

More traditional implementations of GAs have been applied as a means of setting synthesis parameters in complex DSP chains. For example *AudioServe* from Yee-King (2000) is an online interactive evolutionary synthesis tool. This uses an IGA to evolve frequency and amplitude modulation (FM/AM) circuits. A similar technique is employed by Palle Dahlestedt's *MutaSynth* (Dahlstedt (2001)) which can be applied to arbitrary synth engines.

Dahlestedt also experimented with simple coevolutionary programs to generate musical material. *Living Melodies* (Dahlstedt and Nordahl (2001)) uses a spatial multi-agent evolutionary model where populations of singing creatures wander about on a discrete virtual plain. Each agent's genome specifies their actions (WALK, TURN, REST, SING) and also includes 'IF' and 'LOOP' statements, allowing these actions to be combined (loops and ifs cannot be nested). Each agent moves around the discrete 2D space according to its specified actions. Reproduction occurs between agents which are spatially adjacent, have enough 'energy points', are of sufficient age, have walked about recently and have 'heard' some singing recently. A child is produced by cross-over, and genome lengths are dynamic. In the simplest mapping, each agent sings its specified note (voiced on a MIDI piano), which is determined by its species. This produces pulsing patterns which vary as species wax and wane. In order to thin out the cacophony of large populations, they also experimented with applying filters such as constraining note amplitude with 'listening pleasure' or energy points. What we hear in effect is a representation of various aspects of the population dynamics.

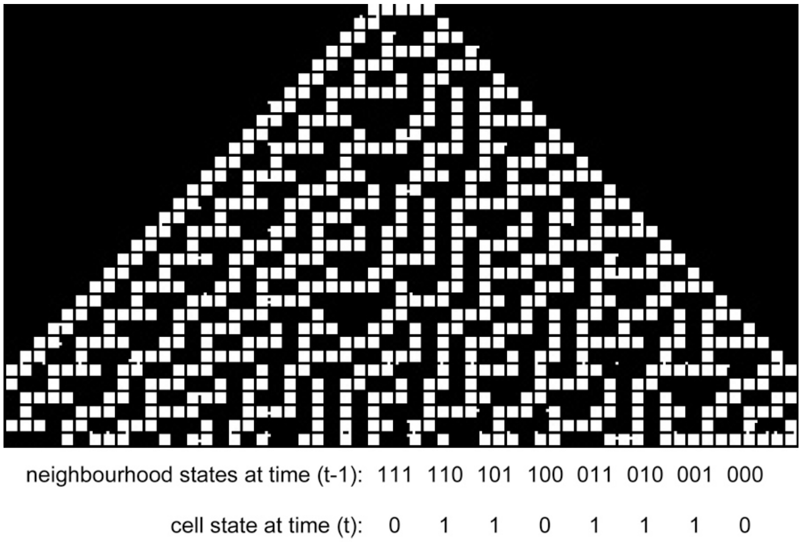
Cellular Automata

Evolutionary techniques are generally presented under the heading of Alife music alongside Cellular Automata[4] (CA). CAs are discrete dynamical systems operating on a regular lattice which change state according to rules based on their local interactions, creating self-organised patterns. In this respect they are comparable to the Swarm model used by Blackwell. The dimensions of the lattice and the number of possible states can vary. A one dimensional (1D) CA with binary states is depicted in Figure 3.2 (top) where each successive horizontal row represents the states of each cell at consecutive time steps. The state of each cell at time *t* is determined by the state of the cells in its defined neighbourhood

---

[4] In fact many prominent authors seem to conflate the two. Miranda for example repeatedly in papers and books refers to CAs as evolutionary, and Cope even describes CAs as "A form of GA". Here the term evolutionary and GA are reserved for models which deploy some form of fitness based selection and reproduction operators to specify the evolution of a population.

according to a sets of rules (Figure 3.2 (bottom)). In this example, the neighbourhood of each cell are the three cells immediately above it. Different rule sets produce different types of patterns; some very repetitive, some chaotic and some complex.



neighbourhood states at time (t-1):	111	110	101	100	011	010	001	000
cell state at time (t):	0	1	1	0	1	1	1	0

Figure 3.2: Graphical representation of a one-dimensional CA and its rule set (bottom).

Perhaps due to their inherent pattern propagation properties, CAs have been extensively explored by many researchers and composers as a basis for generating musical material (for a recent review of applications in MIDI based music see Burraston et al. (2004)). Bilotta and Pantano (2002), draw parallels between the development of variations on a theme, and the fact that CA rules can be run from different initial conditions, producing different, though correlated numerical sequences which can be transformed into musical passages. Others such as Miranda (1993) have applied CAs in the synthesis domain and suggest that they are more suited to specifying timbral rather than structural aspects of music (details of this are discussed in Chapter 4).

L-systems

Another model of structural growth that has been frequently implemented are L-systems. L-systems are generative grammars developed by biologist Prusinkiewicz and Lindenmayer (1990) to model the growth of plants. These have been used by many composers (e.g. Prusinkiewicz (1986)) as a method of generating or organising fragments of musical material. The attraction of such systems lies in their ability to model complex growth processes that, certainly graphically, develop in "aesthetically challenging opposition" (Supper (2001) p.50). In addition, non-predictable, often self-similar patterns are produced which would be hard to specify by hand. Examples of the sorts of branching structures which can be generated are given in Figure 3.3.

Statistical Models

Beside statistical models of musical structure, many other probability-based models are commonly used – most famously explored by Iannis Xenakis. Xenakis' background as an engineer and architect gave him a radically different perspective on music from his classically trained contemporaries. He used probabilistic models drawn from areas such as thermodynamics and game-theory to realise his fascination with the 'out of time' structures of the ancients. One of his best known works, *Pithoprakta* (1955-1956) is essentially a sonic incarnation of the Maxwell-Boltzmann kinetic theory of gases. The Boltzmann distribution can be used to simulate the speeds of gas particles. Xenakis mapped these onto

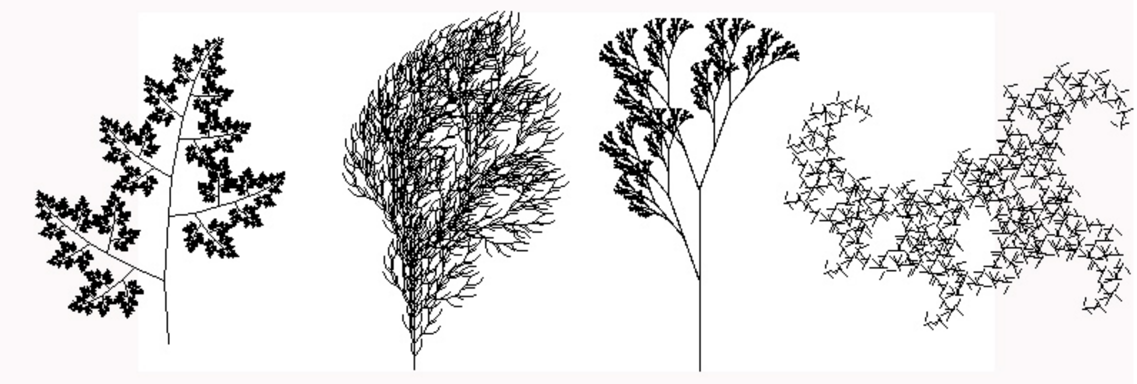


Figure 3.3: L-system generated plant forms.

tuning pitches for glissandi across different orchestral instruments. By slowly changing the temperature parameter in the gas simulation, he created a broad, slowly changing sonic cloud.

Chaos

Following the mathematics communities' enthusiasm for chaos theory, many researchers in the late 1980s and early 1990s explored the musical potential of fractal geometry and chaotic non-linear dynamical systems (e.g. Pressing (1988), Di Scipio (1990), Gogins (1991), Bidlack (1992), Leach and Fitch (1995)).

These are sometimes referred to as iterative functions due to the fact that they take the form $y_{(t+1)} = F(y_{(t)})$. Motivation for adopting these systems is often given in the form of a description of the existence of chaotic patterns in natural phenomenon such as the weather, population cycles of animals, the structure of coastlines, trees or leaves, bubble-fields or the dripping of water. Others point out similarities between the dynamics they exhibit and common musical structures:

> "Musical development or variation can be viewed as the transformation or distortion of a simple entity (a motif), often followed by some sort of return to the original motif. When certain values are chosen for the input parameters to these equations, very similar behaviour can be obtained from them. Thus a series of solutions can act like a repeated group of n notes for a number of steps in the iteration process, and then break away to more unpredictable (quasi-chaotic) behavior before eventually returning to the original n-note group, perhaps somewhat altered." - Pressing (1988)

Specific nonlinear dynamic equations that composers have experimented with include: the logistic map which is traditionally used to model a species' change in population; the Henon map which was originally introduced as a simple and efficient model of chaotic systems in general; and the three-dimensional Lorenz system which was developed from a simplified model of atmospheric turbulence.

Four pioneers of these methods are Jeff Pressing, Michael Gogins, Rick Bidlack, and Jeremy Leach. Gogins (1991) worked with what he called Iterated Functions System, a collection of functions which were selected from at each iteration giving an incredibly (perhaps excessively) broad selection of complex dyanmics. Bidlack (1992) explored higher dimensional dissipative chaotic systems as well as 'conservative' chaotic systems in which energy is conserved, producing a constant orbit rather than transience toward an attractor.

Pressing (1988) focused on the edge of the chaotic regions, where dynamics alternate between quasi-periodic and chaotic attractors: "The output shows unpredictability, but also traces of the nearby cyclic behavior . . . In musical terms, the overall effect is like a variation technique that inserts and removes material from a motive undergoing mildly erratic pitch transformations, in the style of an adventurous but development-oriented free jazz player, perhaps" (ibid. p.4). Maurer (1999) compares this to techniques of free improviser Ornette Coleman, who is stylistically notorious for improvising "motivic-chain-associations" (Jost (1994)).

In all these cases, the outputs of equations were generally used to define (MIDI) pitch information in a generally monophonic stream of notes. Pressing (1988) also experimented with two dimensional maps, mapping each dimension to two different characteristics of a musical event (e.g. pitch and length of a note), but reports that these results sound no better than the single dimension.

This approach was not without critics. Truax in particular was scathing about the use of non-linear functions at this level of musical form:

> "From a more philosophical or aesthetic point of view, it is not clear than an arbitrary mapping of a non-linear function [onto the pitch of successive notes] is inherently more musical than, for instance, a random or stochastic function. The musicality may reside in the musical knowledge of the mapper more than in the source function. The audience, if suitably primed with program notes, may be convinced there is more value or interest in the result because of the technique used, but the half-life of such interest seems to be short."
> - Truax (1990)

Despite his scepticism for using models at the level of pitch, Truax (1990), and later Di Scipio (1999) applied similar models to sound design tasks. They both experimented with models such as the logistic map to parameterise granular synthesis engines. The models were used to control a granulation process operating on a source sound and used to create chaotic textures. The degree of chaos could tuned, creating varying degrees of deviation from the original source sound: "Depending on the 'degree' of granulation on the equation system and on the region of the relative logistic map to be explored, the output sounds are clearly derivative from the original or completely extraneous to it" (Di Scipio (1990)). These equations were used to reproduce the dynamic progression of natural sounds such as increasingly heavy rain fall, or spreading fires.

There is a strong belief amongst both critics and practitioners that much of the creativity in using algorithms, the real compositional work, comes in defining the mapping that is used. This is true to an extent but the issue is more complex. In this case Truax expressed disdain at the use of 'arbitrary' mappings of non-linear systems to define musical pitches, but enthusiastically deployed very similar models in the synthesis domain. Whilst the details of the mappings used are of course vital, ensuring a good marriage between the algorithm and the level of musical organisation is an equally important consideration. This issue is discussed in more detail in Chapter 4, Section 4.4.

Summary

As this overview suggests a range of techniques, drawn largely from AI and other scientific endeavours, have been explored which could potentially assist in many different aspects of the compositional process. As Miranda noted, some are very good at generating music in their target style. Having said this surprisingly few techniques, either AI or extra-musical, seem to actually get used. In the case of AI techniques this may well be because until recently, the implementation of much more than a random number generator or probability model demanded sophisticated programming know-how which was perhaps beyond the average composer. Even when implemented, some require significant technical knowledge to set up and use: ANNs and GAs for example require a good understanding of their operation in order even to be able to pre-process and encode the input data in a suitable way. Such an understanding is also crucial to being able to tune them if their initial performance is less than perfect.

Information processing techniques have revolutionised many areas of engineering and data management, but musical composition is a very different task to the pattern recognition or optimisation problems in which these techniques excel. Within AI music literature many examples suggest that the compositional 'problem' has not been framed in quite the right way. Testimony to this are a number of reports of a mismatch between formal and aesthetic results. For example in their exploration of GAs in a harmonisation task, Papadopolous and Wiggins (1999) report that "The solo generator GA converged very quickly to high fitness" (p.11). However they also add that: "from an aesthetic viewpoint, the results are still far from ideal." (p.12).

Perhaps another reason that these techniques are less than ubiquitous in the musical community is that research tends to focus on the machine learning capabilities in the abstract, rather than their application in a specific musical domain. This is to be expected where algorithmic research is carried out as a computer scientific project, but as discussed below in Section 3.2, these divisions are not always made clear. Increasingly, those focused on music making are considering how these tools can be altered for more creative use, rather than applying them straight off the shelf. Al Biles for example focuses on EC techniques and contrasts the musician's creative perspective with the engineering perspective, which is more prevalent in EC circles and suggests how the musician's perspective alters how EC is applied in musical domains (Miranda and Biles (2007) , Chapter 2).

The dominant computer science/ AI research perspective may be one reason why the use of so called 'extra-musical' algorithms gets somewhat sidelined in academic writing. Beside Miranda's comment on the lack of 'cultural references', Papadopolous and Wiggins (1999) write that "It is difficult to judge the quality of their output, because, unlike all the other approaches, their knowledge about music is not derived from humans or human works" (p.12). In their context, judgement may refer to some form of empirical measure, which is indeed hard to define. But as Pearce et al. (2002) later suggests, the value of any algorithmic approach is best 'judged' in the same way that other music is, i.e. by a listening public.

It is difficult to know to what extent specific or similar techniques are adopted by performing musicians[5] (many of whom may not have access to the academic journals in which they are predominantly published). However, it is generally agreed that few formal models explored within the research community have been applied to anything more than generating small fragments of musical material which if used at all, are incorporated by hand into larger compositions. As a compositional aid, this is fine, indeed many composers view such processes as useful ways of throwing up new material to inspire new compositions. If we aim to develop generative systems for live performance however, we need something a little more powerful.

3.2 Aims and Inspirations: Ambiguity in Cross-disciplinary Enterprises

If the range of techniques, implementations and applications makes the field seem a little schizophrenic, it is because it is. Interdisciplinarity is trendy and increasingly encouraged in academic communities, and an operational interdisciplinary research pro-

---

[5] Allegedly Aphex Twin has employed some examples of generative music packages that are distributed with Miranda's 'Composing Music with Computers' (Miranda (2000a)) (Miranda, personal communication).

gramme would be welcome in algorithmic composition. But arguably, the field has not yet reached this level of maturity, existing more as a collection of cross-discipline projects situated in different fields. This section aims to highlight a few of the issues that arise in such situations.

In many academic papers describing algorithmic composition projects, authors, particularly those aiming to replicate existing musical idioms in some way, often present ambiguous and sometimes opposing representations of what they are aiming to achieve. Biles for example introduces *GenJam* – a GA which improvises jazz melodies over a supplied chord progression – as "a genetic algorithm based model of a novice jazz musician learning to improvise" (Biles (1994) p. 131). This makes it sound as if he has a cognitive scientific interest in understanding a process of musical development. In the final paragraph however, he writes that "GenJam shows that GAs can be a useful tool for searching a constrained melodic space"(ibid p.137), suggesting that GenJam is a compositional or performance tool.

Further confusion is added when authors write up the same systems in different contexts and submit them to journals in different fields. Todd and Werner for example, in one of the most cited reviews of algorithmic composition, present their coevolutionary model as a method of composition:

> "Thus, we propose a new algorithmic composition system based on a combination of all the approaches we have described here: rule-following, learning, and evolution." - Todd and Werner (1999) p.185

Exactly the same model is presented as a coevolutionary model of communications in an Alife context (Werner and Todd (1997)).

Similarly, Miranda's agent-based mimetic model appears in numerous computer music publications in the context of other sound generation/composition techniques, but it also appears as a model of the cultural evolution of music. For instance in his book *Composing Music with Computers* (Miranda (2000a)) he writes: "The adaptive musical games approach to generating musical materials for pieces of music seems a sensible way forward in order to break new ground in algorithmic composition practice." (ibid p.157)

The same system is presented in a paper entitled *Towards an Artificial Life Approach to the Origins of Music* (Miranda (1999)), in which he describes the contribution of Alife simulation to theoretical biology and proposes that "the Alife paradigm also has great potential for Musicology." (p.1), and presents his model as an investigation of "how musical forms may originate and evolve in artificially created worlds inhabited by virtual communities of musicians and listeners. " (ibid p.2). This is not perhaps a problem, if each is suitably assessed in each respective domain, but as Pearce et al. (2002) have pointed out this is not always the case.

3.2.1 Implications for Research Methodologies

Pearce et al pick up on the motivational ambiguity within the field and suggests that it is symptomatic of a methodological malaise. To remedy what they fear may hamper research progress, they propose that the field should be subdivided and propose distinct labels and methodological and evaluation procedures for each subdivision.

A principal distinction is proposed between creative and scientific projects. These are further bisected. The term 'algorithmic composition' is used to describe projects aimed at expanding the compositional repertoire, as distinct from the 'design of compositional tools'. Scientific projects they suggest should also be divided according to whether systems pursue the musicological aim of evaluating stylistic theories or attempt to make contributions to cognitive science by modelling the cognitive processes that underlie composition.

Not unreasonably, they suggest that each requires quite distinct approaches to evaluation. Creative applications, where the ultimate output is intended for public consumption they suggest can only be evaluated "in the same way that composers and compositions are usually appraised: through audience reactions at performances, record sales, critical reviews and so on" (ibid p.5). For many years this was an all-too-rare event. Any form of objective appraisal of musical effectiveness in papers tend to offer rather vague subjective judgements such as "[the program] seems to be capable of producing musical results" (Ebcioglu (1988) p. 49).

They also note that there is a lack of evaluation from those offering their systems as cognitive science models. A cognitive theory is evaluated by implementing it in a model and then making empirical comparisons between the behaviour of the model and the human behaviour that the theory is intended to explain. Granted little is know about the cognitive mechanisms underlying composition, but those who have attempted to model aspects of this process in cognitive terms pay little attention to the basic scientific modus operandii. For example, the only evaluation that Johnsson-Laird (1991) offers for his model of Jazz improvisation is that "The program performs on the level of a competent beginner." (p. 317).

Pearce's suggestions are relevant to individual disciplines using algorithmic composition methods for specific goals and they promote more rigourous application of these established methodologies. It would undoubtedly be easier for us as readers and researchers to judge the value of individual contributions if authors were a little clearer in stating their aims, intentions and evaluations. But there is also an argument that the field needs to look outside of its constituent disciplines to develop new methodologies. Computer assisted algorithmic composition is a relatively young endeavour which straddles speculative and empirical disciplines as well as arts practice. This is what makes it exciting, as well as slightly confused. Too close an allegiance to its parent methodologies could also be restrictive. It seems reasonable to encourage purposeful cross-fertilisation of ideas which may be a crucial ingredient for success. For example, in yet another paper describing his mimetic model of music evolution, Miranda proffers:

> "Should such experiments with adaptive games corroborate our hypothesis that we can improve algorithmic composition systems considerably by including mechanisms that take into account the dynamics of cultural evolution and social interaction, then we believe that a new generation of much improved intelligent composing systems will soon begin to appear over the next few years." - Miranda (2002)

Clearly defined speculative enquiries are central to the development of new methodologies and activities are arguably vital for the maturation of a nascent interdisciplinary project such as algorithmic composition.

3.2.2 Implications for Software Design

I would like to suggest that there is a more worrying side effect of cross-disciplinary projects which impacts on the use of algorithmic composition as both a practical creative pursuit and as a method of scientific modelling. These problems stem from the adoption of ideas and theories from one discipline into another practice without careful consideration of their true nature. The adoption of AI techniques, invariably, although not inevitably, is accompanied by the framing of composition as a 'problem' that needs to be solved. This is implicit in much work, and utterly explicit in others (Cope (2005), Jacob (1996)). This is fine in itself: we could agree to confer that some aspects of composition could be usefully framed as a problem solving task. In the context of AI, music becomes an engineering problem, to be solved by a search mechanism, constraint satisfaction procedure or pattern matching algorithm. In both engineering and scientific research applications, such techniques are developed into computational models. Models implement theories which are generally derived from experimental observations of the phenomenon under study. In a scientific context, the theory is then evaluated by empirical comparison of the output of the model and the natural phenomenon. In an engineering context the hope is that this theory is sufficiently powerful that the model simulates the natural world phenomenon.

As noted in the last chapter, this approach can be very effective in related areas of machine listening, where theories of human auditory perception and cognition provide the starting point for beat-induction or pitch discrimination algorithms (e.g. Collins and Cross (2003)). In algorithmic composition, for either scientific or compositional ends, Western Musicology offers an attractive body of theory which can be embodied as a rule-base or set of constraints in a similar fashion. As well as explicit encoding in Knowledge Based systems, theoretic principles can be embodied more implicitly in evolutionary or learning algorithms.

At first sight, this offers a means of imbuing the necessary domain knowledge to ensure musical structure in the output. The rules of harmony, for example, offer an apparently neat solution to the headache of designing a fitness function for a GA; spatial representations of tonal distance (e.g. Schoenberg (1954), Shepard (1982)) could provide a basis of error measurement for an ANN. And indeed this is common practice: the "Fitness function judges the fitness of each chromosome according to criterion taken directly from music theory" (Phon-Amnuaisuk and Wiggins (1999)). The problem, which does not seem to be discussed, is that music is not a natural phenomenon, it does not conform to natural law, and music theories do not serve the same function as theories of biology or physics or even cognition.

Music theory and analysis are post hoc endeavours, which aim neither to describe what the composer did, nor necessarily what the listener hears. For researchers aiming to model the compositional process or develop systems with which to compose music aimed for public consumption, it seems important to keep this in mind. This section serves as a gentle reminder that care should be taken when inspiration is drawn from music theoretic tenants either for creating models of composition or in designing systems for generating music for people to listen to.

The Analyst's Music

> "Writing about music is like dancing about architecture: it's a really stupid thing to do" - Elvis Costello, in an interview by Timothy White (1983) p. 52

Music theories in their many guises, represent attempts to *understand* music in an analytical sense; the analyses from which rules are derived aim to achieve possible coherent sets of principles and ideas with which to rationalise, analyse and investigate the structurally functional aspects of music. This is neither exhaustive, nor aimed primarily at describing music in terms of the listener's perception.

> "Each musical culture rationalises only a few selected aspects of its musical production . . . [so] any cultural representation of music (i.e. music theory) must constitute a thoroughly incomplete specification of the intended musical experience... A formal analysis is a kind of mechanism whose input is the score, and whose output is a determination of coherence...In other words, it purports to establish or explain what is significant in music while circumventing the human experience through which such significance is constituted; to borrow a phrase from Coulter, it aims at 'deleting the subject'." - Cook (1990), p.241

Cook's basic argument is that there is an important and inevitable discrepancy between the experience of music aurally, and the ways in which it is imagined or thought about. He draws a useful distinction between 'musical listening' which is concerned with the aesthetic gratification in being absorbed in a non-dualistic sense, and 'musicological listening' for the purpose of establishing facts or the formulation of theories about music as a 'perceptual object'. This is not to say that the thoughts and ideas of theorists and analysts are inadequate or misplaced, but simply that they are not aimed primarily at giving an account of the listener's phenomenological experience of music.

An extreme example of this discrepancy between analytic and experiential reality is provided by experiments in which two versions of short classical piano pieces were played to music students: their original form, which began and ended in the same, or closely related key, and an altered form which had been modified so as to modulate to, and end in, a different and unrelated key (Cook (1987a)). In standard music theory, tonal closure - or more generally the influence, or organising function of the overall tonic - is the very core of the traditional forms of eighteenth and nineteenth century music. However in these trials, there were no statistically significant differences in preference for the original over the altered forms. In fact, there was a general preference for a Brahms Intermezzo in which the final section had been transposed up a minor second.

In another set of tests, Cook (1987b)) played the first movement of Beethoven's G major sonata (Op 49 No 2) to music students, stopping the recording just before the final two chords. He then asked them how much longer they thought the piece continued for and reports that they frequently predicted that the music would carry on for another minute or more. Theoretically, the recapitulation and coda are key informative aspects of form, signifying the close of a piece. These studies suggest that in aural presentations they seem to be slightly less than effective in fulfilling this function. In both these studies, there seems to be a discrepancy between what is seen as functional theoretically, and what actually affects us perceptually.

Perhaps more elegantly designed studies would be needed to make any strong claims, but it is common for musicologists to differentiate between the aural and analytic aspects of a piece of music. Bailey (1983) writes of Webern's symphony that it consists of "two quite different pieces - a visual, intellectual piece and an aural, immediate piece, one for the analyst and another for the listener." (p.195) Clifton (1970) expresses this more incisively: "For the listener, musical grammar and syntax amount to no more than wax in his ears". (p.71)

Within the music community, the differences between analytic treatments and compositional practices are well recognised. The fugue, for example is often qualified with a prefix 'composer's' or 'classroom' to differentiate between the actual fugues written by composers, and the idealised abstractions which appear in textbooks.

> "[The classroom fugue] insists on certain exigencies which place the texture upon the highest level of skill, the actual musical result being, seemingly, of secondary importance. Many noted theorists have set down all the things which should happen in every fugure, but composers have always followed their own dictates. It is impossible to align No.1 Book One of Bach's Forty-Eight, for example, and many others, with the dictates of the theorists. At one time it was not regarded as musical expression but was used solely as a mental exercise." - Demuth (2003)

In some discussion of algorithmic composition, these distinctions seem to get papered over. In his list of historical examplars of algorithms, Cope (2000) writes: "Fugues and other contrapuntal formalisms represent constraints and often require severely limiting algorithms" (p.3). He uses these and many other examples of pre-digital algorithms to support his claim that 'all composition is algorithmic'. But as Demuth notes - in a textbook of musical forms - although a textbook description of a fugue might look like an algorithm, and so be suitable for implementation and realisation by a procedural device like a computer, this is not what composers did, or do. The clean, algorithmic-looking version came *after* the composer's intuitive one.

> "Schemes of musical arrangement, even if they exist a priori, should only be discovered after they have been used." - Schoenberg (1954)

The dangers of too close a marriage between academic theory and the development of compositional algorithmic programs has been expounded by Gartland-Jones and Copley (2005). They focus on the level of form, clearly illustrating their arguments with insightful and deep consideration of sonata form which shows just how artificial the text book abstractions are that researchers such as Cope use as a point of departure to justify algorithmic research. This is perhaps especially important for those aiming to model compositional processes, as these theoretical imprints depart considerably from what composers actually do. For those interested in algorithmic composition for making music, there may be another problem with the blind adoption of music theoretic principles.

The Listener's Music

Music as we listen to it is indefatigably temporal. Music as it is studied on the score lies out of time. The contradiction between the ever-present 'inner' time in which music is experienced, and the retrospective 'outer' time which is imposed in the act of reflection and measured by musical notation is a fundamental dilemma for many theorists (Cook (1990)). Schutz (1976) goes as far as to suggest that attempts to describe musical experience in 'outer' time poses a variant of the Eleatic paradox – i.e. that the flight of Zeno's arrow cannot be described because it is impossible to represent the ongoing quality of its' motion. As he puts it "you may designate the spot occupied by the arrow at any chosen constant during the flight. But then you have dropped entirely the idea of an ongoing motion." (ibid p.30). At the heart of this lies the discrepancy between the static, symbolic nature of music in notated form - which is the principle object of music theoretic concern – and the dynamic immediate nature of music in sonic form - which is the subject of concern for the listener. Many more progressive critics and theorists both within and beyond traditional 'note-based' music discuss the distortive nature of the musical notation, and how it gets in the way of an understanding of our actual listening experience.

> "The principle point I am going to develop is that the priorities of notation do not merely reflect musical priorities - they actually create them. It is fundamentally important to grasp the point if we are going to understand an approach to music based on our listening experience .... A preoccupation with conventional notation can lead us into formalism, a situation where there is no longer any experiential verification of our theories about how to compose music." - Wishart (1985), p.11

Wishart of course works primarily in electronic music and is keen to ensure that new possibilities and ideas aren't constrained by conceptions carried over from archaic idioms. But this is not just an issue relevant to the acoustic-electronic divide. Very similar ideas were expressed years before, by Ernst Toch.

> "I never expected so much fascination to come from investigations of the nature of musical theory and composition. Aspects unfolding to me show why the rules of established musical theories could not be applied to 'modern' music, why there seemed to be a break all along the line, either discrediting our

> contemporary work or everything that has been derived from the past. To my amazement I find that those theories are only false with reference to contemporary music because they are false with reference to the old music, from which they have been deduced; and that in correcting them to precision you get the whole immense structure of music in your focus." - Toch (1948), p.xii

Ernst Toch was a masterful and original classical, and later modernist, composer who was also renowned for his Paramount Studio film scores. Later in life he became preoccupied with the reconciliation of theories of classical music with contemporary modernist trends. *The Shaping Forces in Music* (Toch (1948)) is his account of how all musical writing must respond to the psychological wants of the listener, and how similar goals may be achieved in different styles. If harmonic structure is the cornerstone of traditional music theory, Toch sees the movement of melodic 'impulses' (not dissimilar to Wishart's dynamic morphologies) as the central force of music from the listener's perspective. He describes harmony as 'arrested motion' by which he means to stress the fundamental Heraclitean flux in music.

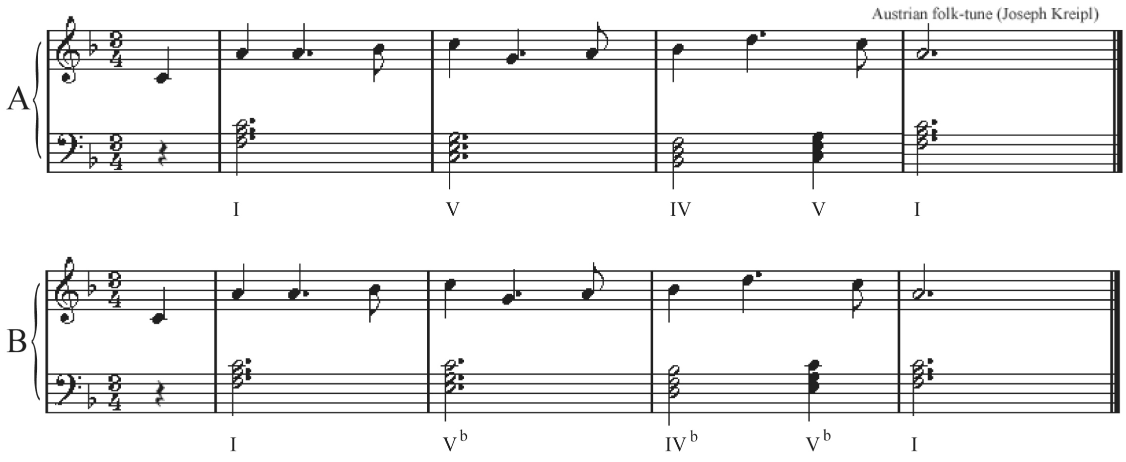


Figure 3.4: A Natural harmonisation of a phrase from a simple folk tune using I (tonic), V (dominant) and IV (sub-dominant) (top) and appropriate chordal inversions (IV^b etc) (bottom).

Harmony as Arrested Motion

In an example that is typical of the tasks used in algorithmic approaches to harmonisation, Toch presents a phrase from a folk tune, that invites a simple I, IV V harmonisation (Figure 3.4 A). This is something that a GA could perhaps achieve. We could even potentially incorporate theoretic axioms for finding appropriate chord inversions into the fitness function: by minimising the number of steps that each note must take into membership of adjacent harmonies we could feasibly find the first inversions needed to create a smoother chordal structure in bars two and three as shown in Figure 3.4 B. The apparent simplicity and efficacy of this kind of 'rule' is precisely what is attractive to the algorithmic composer, but as Toch warns: "While this axiom seems a simple expedient for the beginner, it implants in him a dangerous misconception, namely the view point of rigidly preconceived harmony as a fixed unit, within the frame of which each voice seeks to take up its appropriate place." (p.5).

This point is illustrated by considering a common or garden Chorale harmonisation shown in Figure 3.5 A., which concedes to all the traditional rules of harmony. Toch then

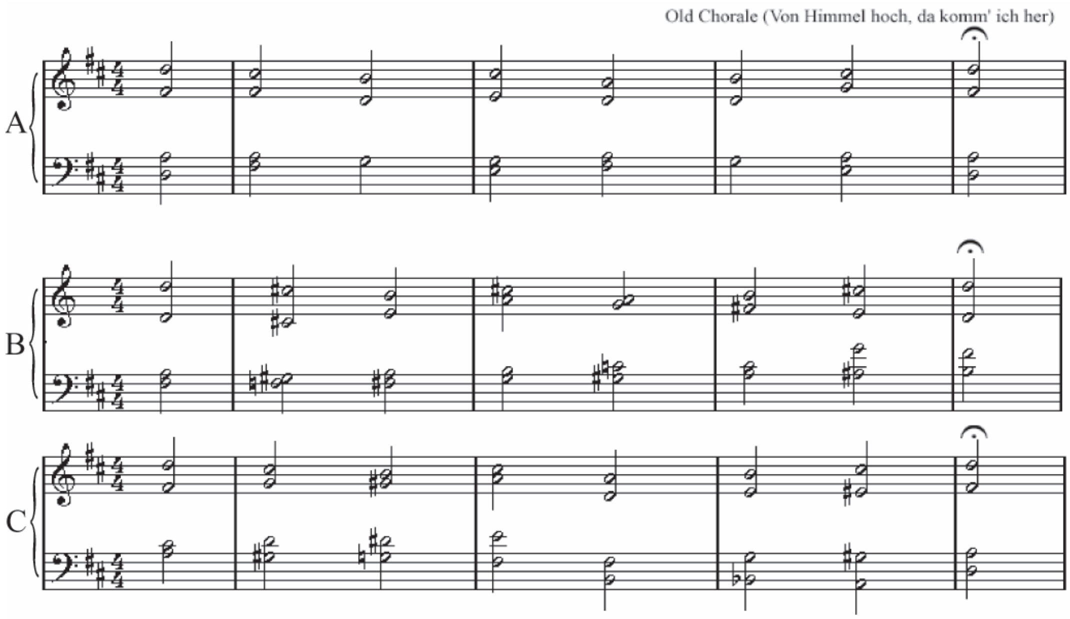



Figure 3.5: Standard harmonisation of a phrase from a Chorale (A) and examples from Toch's alternative harmonisations (B) and (C).

offers twelve other possibilities, examples of which are given in Figure 3.5 B and C. These are written using a more general principle which he calls 'linear voice leading' - a term he uses to describe the dynamic impulse of each voice. In contrast to the 'appropriate' harmonisations of Figure 3.5 A, some harmonies in Figure Figure 3.5 B and C go against every rule in the book: consecutive fifths, cross relations, arbitrary dissonances etc.

> "And yet we hope that the reader, even though these harmonisations may appear unusual and strange, will feel their logic and organic life.[6] That they are arrived at by the movement of melodically independent voices is obvious. The truth is that the melodic impulse is primary, and always preponderates over the harmonic; that the melodic, or linear impulse is the force out of which germinates not only harmony but also counterpoint and form. For the linear impulse is activated by *motion* and motion means life, creation, propagation and formation." - Toch (1948), p.10

Toch's point here, is that harmonies are not dictatorial pillars which define the pitch of the constituent notes, but snapshots of coincidences that emerge from the interplay of separate melodic lines as they develop in time. This stands in stark contrast to the way in which the 'harmonisation problem' is sometimes conceived and approached in algorithmic composition: "We apply the following criteria: we avoid parallel fifths, we avoid hidden unison, we forbid progression from diminished 5th to perfect 5th; we forbid crossing voices ... " Toch himself might have enjoyed the authors' own description of the output of the system, as it reinforces one of his central propositions. "The harmonisation produced by the GA has neither clear plan or intention" (Phon-Amnuaisuk and Wiggins (1999) p.5).

---

[6] Toch invites the reader to play each line, separately at first and then with the soprano, before playing the full harmonies, listening to each separately to appreciate their movements.

This is an extreme, although not atypical application of music theory to the design of algorithmic systems. Considered in the light of Cook's comments on the nature of musicological listening, and Toch's comments on the nature of musical listening some potential problems with this approach come to light. Music theory, working with a static representation of music, forms abstractions and generalisations. As Cook suggests, some of the key functional structures may be aurally imperceptable. It seems far from inevitable that in using theoretic principles as guiding principles to design systems capable of creating and playing music that we will recreate the temporal phenomenon from which the theory was derived. And without this, it may be hard to produce a sense of plan or intention that underlies the structures from which theory generalises. This could be one reason why we frequently see comments such as: "while conforming to classical triadic harmony, the music seems lifeless" (Cope (1999) p.21) or "The music often wanders with unbalanced and uncharacteristic phrase length. No musical logic is present beyond the chord-to-chord syntax" (ibid, p.22)

This is not to say that music theories are wrong, or worthless. The problem comes only in interdisciplinary settings where words have different meanings and theories do different jobs. Music theory is not aimed at providing a model of the phenomenon of music in the same was that biology does of living organisms. It offers ways of understanding music, ways of imagining music, and can undeniably alter the way that we appreciate it. But it does not aim to explicate the key phenomenon of relevance to us as listeners, or the things that composers do. It seems sensible to bear this in mind when applying it to the design of algorithmic composition systems aimed at generating material for people to listen to.

This section leaves behind scholarly academic discussions, and focuses more closely on the practical application of algorithmic processes in composition and live performance. The touch of ivory tower syndrome that dogged what we might call 'institutional algorithmic composition' for some time, whereby the standard method of evaluation was the we-think-it-sounded-quite-nice comment in a paper's conclusion is drawing to an end. Within academic circles, there are increasing numbers of conferences and dedicated workshops springing up which accompany the standard paper/ poster presentation with demos and concerts. ICMC, Ars Electronica and Generative Art have been joined by the Iteration series in Melbourne, the EvoMusArt workshop at EuroGP as well as countless other local groups. More dramatically, the increased accessibility of music programming languages mean that the days of computer music being the reserve of specialist institutes is long over. Music programming languages are not only more accessible and widely available than ever before, but open source communities such as PD and SuperCollider mean that young electronic music enthusiasts can work alongside pioneers such as Miller Puckette.

3.3 Performance Issues

In addition to those persuing electro-acoustic interactive performance, there is now a significant and rapidly expanding community of ardent laptopists, dedicated to performing live with little but their shiny machines and perhaps the odd cross-over cable uniting the smaller-ego'd in a mini laptop orchestra. This new breed of programmer-composer-performer typifies the dissolved hierarchies of contemporary culture and answers the call of many academic researchers for increased collaboration between computer scientists and composers by combining them in the same skin. The first two sections of this chapter focused on academic research in algorithmic composition, where performance exists as some kind of evaluation method. This section focuses on the practicalities of music making where algorithmic processes exist as a compositional or performance tool.

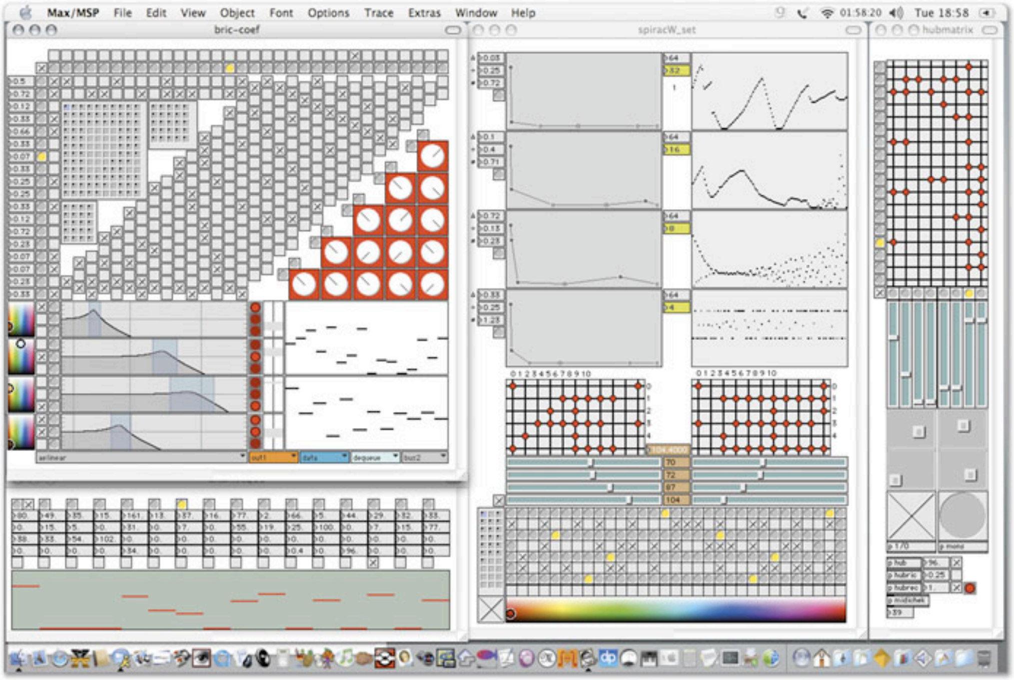


Figure 3.6: Screen shot of an Autechre Max patch.

3.3.1 Aims: Automation – Augmentation – Autonomy

Beside basic curiosity and nerdy experimentalism, there are several distinct reasons why composers in a number of fields might adopt algorithmic methods. Most pragmatically, they are very efficient, both in terms of time and storage space. Conceptually, and arguably aesthetically, the use of live processes brings back a sense of significance-to-the-now and uncertainty to electronic performance, a loss which was most famously bemoaned by Bartok with the invention of mechanical recording techniques. Creatively, algorithmic processes open up all sorts of possibilities which would be either impossible or incredibly laborious to create by hand. On the one hand of course generative processes are central to the various forms of interactive performance discussed in Chapter two in which the computer must respond to an instrumentalist or some other time-specific environmental cue. On the other hand, even in off-line composition the minutiae of the sonic/rhythmic/textural complexities afforded by the machine are invariably incredibly laborious, if not impossible, to specify by hand. Algorithms offer a crucial method of automation which is required to meet certain compositional aims. Increasingly generative processes are developed to augment these aims, pushing the user further along their chosen path, and even encouraged to wander off in new directions, suggesting new ideas.

Pragmatically, algorithmic processes can be incredibly efficient in terms of both composition time and (storage) space. This underlies the attraction of generative music for internet and games applications: indefinate amounts of never-repeating material can be specified in a definite and small number of bytes. In purely practical terms for installations which are required to run for long periods of time, algorithmic processes come into their own with respect to a composition time to runtime payoff.

In a paper discussing generative music in laptop performance, Nick Collins describes the use of live generative processes as a “tenuous hidden conceptual thrill” Collins (2003b). Tenuous in laptop performance, because, as he is the first to point out, what the audience sees is a glowing apple and a dimly lit look of concentration. It is often hard to differentiate the activites of the inventive algo-mentalist juggling hot-off-the-press SuperCollider procedures from the guy pressing ‘go’ on his pre-programmed Ableton Live set.

But for many the never-to-be-heard-again aspect of generative music represents a turning point in electronic music. Brian Eno has perhaps voiced this enthusiasm most loudly, demonstrating the joy of incommensurate loops in his *Music for airports*, but also raving about his experiences with other people’s generative software. Describing his first encounter with Sseyo’s KOAN he rejoiced:

> “I’m so thrilled by it that it is very difficult for me to listen to records anymore. Putting on a record and knowing I’m going to hear the same thing I did last time has actually become a little bit irksome. It feels quite Victorian to do that (laughter). I think this has really moved up into a new phase of music.”
> - Eno (1996))

Whilst Eno may be fairly unique in sitting and listening to generative music at home, live generative music arguably brings with it a sense of uncertainty, the loss of which was feared in the age of mechanical reproduction, and again as MIDI sequencing became de rigueur in the pop music of the 1980s. Scored music is sometimes referred to with disdain by extreme proponents of the generative approach, but of course the nuances of expression and interpretation are significant enough for us to need both Pablo Casal’s and Paul Tortelier’s recordings of the Bach ’cello suites. And it is the freshness and uncertainties of live performance which draw us to go and see our favourite bands again and again despite having listened to their albums upteen times at home. In 1990, critic Jon Pareles disdainfully commented on the burgeoning use of sequenced MIDI material as a substitute for human players in pop music :

> “If I wanted flawlessness, I’d stay at home with the album. The spontaneity, uncertainty and ensemble coordination that automation eliminates are exactly what I go to concerts to see; the risk brings the suspense, and the sense of triumph to live pop perhaps the best we can hope for is that someone will come up with a way to program some rough edges too.” - Pareles (1990)

As anyone working with live algorithmic processes will attest, uncertainty, suspense and spontaneity are algorithmic composition’s middle names. The use of live algorithms certainly provides the possibility of some rough edges (though not perhaps of the sort Pareles was after) but arguably offers one way of regaining “A music that says this time is special, now is privileged ” (Collins (2003b), p.71).

In live performance, but also in offline studio situations, algorithms are a welcome means of removing the laboriousness of dense or intricate composition. As Xenakis’ work is often used to illustrate, computational procedures take the tedium out of calculating the millions of sonic events which the super-human capabilities of the computer can deliver. Both on and off-line, even fairly basic randomising techniques are warmly welcomed by any producers experimenting with slightly broken beats. The tedium of sequencing break-beats is removed by simple plug-ins such as Marcus Clements’ ColdCutter[7] which randomly splices and/or reverses samples, creating complex dense rhythmic textures. Even the break-beat cognoscenti adopt more powerful implementations of stochastic break-beat generators such as Nick Collins’ BBCut SuperCollider libraries which includes the squarepusher emulator mentioned above. In this situation, algorithms act as a method of automation, sometimes merely for saving time. In live situations, automation is the only option for achieving “. . . desirable but physically preposterous [effects such as] machine-gun-fire buffer stutters or micro-polyphonic granular swarming or Gaussian distribution counterpoint” (Collins (2003b) p.74)

---

[7] `http://www.brightonart.co.uk/coldcutter.shtml`

Others offer similar devices which are capable of learning the rhythmic styles of individuals. Once such project is Ollie Glass' *Breakage* system which generalises from user-supplied drum patterns. We might describe these as augmentations rather than automations, doing a bit of stylistic development on behalf of the user. Such devices were prophetically yearned for back in the 1980s by Laurie Spiegel. Bemoaning the over-use of randomness in algorithmic composition she expressed a desire for precisely this kind of algorithm:

> "A good algorithm should be a composer's amanuensis, an invisible assistant who reaches for a knob or a note for you, knowing just what you want to do and when. It should be an extra pair of hands who know their job well, and are able to execute or elaborate for you because they embody knowledge of what music is and how musicians really work. Ultimately, an algorithmic music program should be able to learn each individual user's unique personality, procedures, habits, and preferences, and to use this knowledge to take initiative and make musical suggestions when asked, to add to the individual's power by "automating" what s/he would do anyway, to extend the individual's music much further in directions it already takes. "
> - Spiegel (1989)

There is undoubtedly a calling and a place for automation and augmentation, but random number generators still feature large in the composition tools and performance patches of many using algorithmic methods. Whilst performing musicians may not pursue the world of out-of-control emergence which obsess Alife artists, there is increasing interest in semi-autonomous processes which demand answers and push the individual in entirely new directions. In the manifesto for a current EPSRC focus group 'Live algorithms for Music' (LAM), the need for algorithms which are 'not merely automatic, mechanical and predictable, but comprehensibly interactive and capable of novelty' is expressed.

3.3.2 Adaptability vs Dependability: Inspiration vs Freedom

Regardless of framework, the thrill of the unpredictable brings with it the danger of dull or irritating as well as the genius. In live performance in particular, the trade-off between adaptability and dependability is vital: if we are to make the most of the potential of live generative processes, they should be able to show variation in response to the moment; conversely if we are going to invite them onto the stage, we need confidence that they are going to behave well and not embarrass us. This is as true of tailor-made patches as it is of ready made algorithmic composition tools. Similarly any particular system should ideally provide enough structure to inspire us and prove its worth, whilst at the same time leave sufficient flexibility to allow us the creative freedom to work with it toward our personal compositional/ performance ends.

If generative processes are being unleashed live, the possibility exists that they may wander off into some dull or ear-bashing corner of its universe. Processes must be somehow constrained. But if constraints are so tight as to narrow the field to the point where differences are imperceptible or irrelevant musically, why bother writing a procedure? Why not just re-write the entire set and press play? When improvising with the digital other, there is only so long you can cover a system stuck in a stubborn silence, and there are only so many moves you can make in response to a sustained high frequency noise convulsion.

In situations when there is someone at the helm of the laptop, there are several tried-and-tested practical solutions to this problem. Assume that we adopt the 'generate and test approach', used by Hiller and Issacson in the composition of the Illiac suite, and employed in some form by many algorithmic users. The simplest solution, which works well in some situations, is to simply preview the current output - much like a DJ beat-matching before crossfading. Tasks such as tempo-matching can take place out of the audience's ear shot: algorithmic vagaries can be screened before being unleashed on the audience. Search mechanisms such as the Interactive GA technique described above could be applied to actively search large spaces for suitable candidates, a technique Palle Dahlstedt deploys in performances with *Mutasynth* (Dahlstedt (2001)).

Presets could also be made on the fly during the performances (with maybe a couple of last-resort settings made in advance in case of emergency). Max/MSP for example includes the *pattr* object and its family, which allow snapshots of parameter settings to be captured and saved. This makes it easy to capture any particularly classy behaviours which you might wish to return too. Collins suggests designing code which allows operation in 'autopilot' mode. Here, algorithmic processes are given free reign, but the procedures and interface are designed so that if a process wanders off into undesirable territory, the user can step in and take manual control.

Others not wishing to have to perform any screening or nannying adopt a more conservative approach, combining some generative mechanisms as the icing on the cake of predetermined material. For example John Eacott, working under the pseudonym *jnrtv*, performs dance sets where the rhythmic patterns and high-level structure is fixed, and surface effects are applied algorithmically to provide what he calls 'fluidity' (Eacott (2000)).

All these solutions are fine if generative process are used in solo or group laptop extemporisations, but in situations with a live instrumentalist, or when the the laptop is unmanned, we may need the algorithm itself to be a little more reliable. It needs to offer unlimited variation, but to operate within an acceptable arena: it needs to be able to do its own thing but respond to our calls.

When used live or for off-line composition, an algorithm should ideally offer a balance of structure yet freedom both in terms of control and implementation. A random number generator gives complete freedom of application, and the degrees of freedom allowed can be controlled by the composer, but it does not provide much structure. At the other extreme, many 'ready made' generative music applications leave little room for creative application on the behalf of the user. Sseyo's KOAN, which so pleased Eno was one of the most widely marketed of early systems aimed at allowing broad application, but other users don't find such joy:

> "in the case of a system like KOAN, although the documentation that accompanies it encourages you to interact with the parameters to create your own unique settings and hence, create a new piece of music, what is mostly achieved (in my experience) could be described as a remix of pieces preprogrammed by the development team." - Eacott (2000), p. 5

If mainstream commercial software is seen to be limited by the representational assumptions of the design team, aesthetic predilections of the programmer of ready-made algorithmic systems often shackle the user just as strongly. This is a common problem across much creative software, but is of particular issue for generative tools. Systems like KOAN include not only algorithmic specifications, but determine how these are mapped onto sound. As Eacott says, we are offered parameters to control certain aspects, and may be able to select between different MIDI instruments etc. but essentially we are constrained within a certain field delineated by the designers. Within generative arts more broadly, this is seen as a problem for applications offering themselves as 'creative tools', as although the process itself may offer many possibilities, decisions made by the programmer at the time of implementation introduce constraints. These issues are discussed in Chapter 7.

3.4 Discussion

In the last chapter, it was suggested that the Alife methodology was suitable for implementing software capable of supporting a rich interactive experience. Having looked at a range of approaches, potential pitfalls and pragmatic requirements, this section considers the potential for similar adaptive dynamical systems as methods of generative composition.

Section 3.2 provided a taste of the range of techniques that have been applied to the generation of musical material. Distinctions were made between scientific projects, which aim to model aspects of particular styles or the cognitive processes involved in composition, and those aimed at making practical contributions to composition methods. A further delineation was made between those aiming to develop systems with some form of representation of musical know-how derived from existing styles of music, and those aiming to explore new musical ideas employing what is often referred to as 'extra-musical' systems.

Section 3.2.2 aimed to illustrate some of the problems in the way that algorithmic composition is both framed and implemented: imbuing a system with representations of musicological thought may enable you to maximise your fitness function and help you achieve 'correct harmonisations' but it does not necessarily guarantee a decent musical output. The problems endemic in some areas of research are perhaps summed up by presentations of systems which are set the task of 'harmonising a Bach Chorale'. From a composer's perspective, this itself is a misnomer as it suggests a vertical rather than a linear approach that would have been entirely foreign to Bach's own aesthetic.

If we are not necessarily tied to the emulation of particular existing styles, we have more freedom in algorithm design; at the same time, without the trappings of familiar harmonies or forms we may have to work harder to engage the listener. The fact that the most popular mathematical models to date have been models of growth, pattern formation and population evolution suggest that one alternative is to focus on achieving a strong linear impulse. Several of these seem to be useful in sound design situations, emulating the temporal evolution of complex sound progressions. These successes suggest that at least in some circumstances, formal properties of these dynamical systems can be used to create comparably dynamic sonic effects.

In contrast, applications of Alife and dynamical models at higher levels of musical organisation do not seem to have been so successful. Whilst systems such as CAs, L-systems and chaos models are capable of generating some interesting fragments, they have not been so successfully applied to the generation of any significant musical works. There are at least three possibilities which can be investigated toward developing upon existing approaches.

Part of the reason perhaps is that they do not present immediately obvious hierarchical structures, generally being mapped onto a stream of monophonic notes. Agent-based ecosystem models as used by Dahlstedt in Living Melodies, do perhaps offer richer structural possibilities, but their system was sonified using mappings that stuck to low level pitch determinations. One obvious move then, is to explore other approaches to mapping.

Another alternative is to examine some other models. In Alife music in particular, 'Alife' has become synonymous with GAs, CAs and multi-agent models. There are many, many other possibilities: the existing artillery employed within Alife research is itself extensive, but also provides illustrations from which bespoke models can be developed. Using off-the shelf models is a starting point, but as suggested above in the application of AI learning and search algorithms, real headway may only be made once these techniques are tailored to meet our specific purposes. Of the existing techniques used in Alife research, there are many which offer an attractive middle-ground between the single dimensional chaos model and the complex co-evolving ecosystems examined in the last chapter. In particular, the sub-field of autonomous systems research develops many models for investigating adaptive control of single agents.

A third alternative is to combine different models with different characteristics. As noted above, new offerings in this area are, perhaps inevitably, given in the form of single models. Generative artists talk about composing processes, but there has been little exploration of the potential for composing *sets* of processes. These possibilities will be explored in Chapter 6.

Selection and development of these models can most sensibly proceed by taking into consideration some of the desirable properties discussed in Section 3.3. Most crucially, it was suggested that if we are going to risk relinquishing control to a live algorithm on stage, it needs to be adaptable, but dependable. Further that if it is going to have more than a one-off idiosyncratic appeal, it is desirable that aspects of its global behaviour can be tuned, giving both a vast range of possibilities, and some degree of control over where in that space it travels. CAs and the chaos models that have been used begin to address these balances. Both are sensitive to initial conditions, meaning that there is a pragmatically indefinite variation in their output. On the other hand, as deterministic systems, this initial seed can be saved, and the same pattern recreated. In the case of CAs it is also possible to define particular classes of behaviour - i.e. chaotic, ordered or complex - according to defined rules or experimental experience. Chapter 6 presents explorations of models with at once greater adaptive potential, and richer possibilities of control.

Adaptive dynamical systems open a potentially interesting pathway for computer music composers. Carving a line between the relentless unfolding of uninterruptible mathematical equations and the offline search for new musical forms illustrated by GAs. Collins suggested that algorithms could be released live on stage given the implementation of GUI controls which allow the human to take over. The use of adaptive systems makes for a more collaborative form of man-machine performance in allowing the possibility to influence or coerce behaviours, rather than grab the reigns back.

Discussions in the last two chapters suggested that the adaptive, dynamical framework adopted by Alife research offers the potential to meet the latent wants of some in the interactive and generative music communities. The initial aesthetic motivation for this whole project stemmed from observing the artificial agency apparent in visualisations of Alife style systems and wanting to bring it onto the stage. But it is perhaps easy to get wooed by this artificial agency and assume that this can be readily transported into the auditory domain. Before exploring the musical potential of adaptive systems, the next chapter takes a step back and considers the source of this apparent agency: is it something inherent in the system architecture ? or an illusion created by their physical presence or the visual trappings which allude to biological creatures?