AI Methods for Algorithmic Composition: A Survey, a Critical View and Future Prospects

George Papadopoulos; Geraint Wiggins
School of Artificial Intelligence, Division of Informatics, University of Edinburgh
80 South Bridge, Edinburgh, EH1 1HN, Scotland
{georgep;geraint}@dai.ed.ac.uk

Abstract

In this paper we survey the use of different AI methods for algorithmic composition, present their advantages and disadvantages, discuss some important general issues and propose desirable future prospects

1 Introduction

Algorithmic composition could be described as "a sequence (set) of rules (instructions, operations) for solving (accomplishing) a [particular] problem (task) [in a finite number of steps] of combining musical parts (things, elements) into a whole (composition)", (Cope, 1993)[1]. From this definition we can see that it is not necessary to use computers for algorithmic composition as we often infer; Mozart did not when he described the Musical Dice Game.

The concept of algorithmic composition is not something new. Pythagoras (around 500 B.C.) believed that music and mathematics were not separate studies. Hiller and Isaacson (1959) were probably the first who used a computational model using random number generators and *Markov chains* for algorithmic composition. Since then many researchers have tried to address the problem of algorithmic composition from different points of view.

In this paper we give a short review of these attempts (section 2) followed by a general discussion (section 3), some desirable future prospects (section 4) and we finish with some conclusions (section 5).

2 The survey

A review such as this can never be exhaustive, firstly due to lack of space and secondly because there have been so many attempts. Here we briefly review some recent characteristic work. Most, if not all, of this work involves implementation and not just theory, and was carried out in the last decade.

In the following subsections we give some representative examples of systems which employ different AI methods which we categorise, based on their most prominent feature, as follows:

- Mathematical models
- Knowledge based systems
- Grammars
- Evolutionary methods
- Systems which learn, and
- Hybrid systems

The categorisation is not straightforward since many of the AI methods can be considered as equivalent: for example *Markov chains* are similar to type-3 grammars (Chomsky, 1957)[2]. Furthermore some of the systems have more than one prominent feature, for example *EMI* (see below) was categorised as a grammar, but it can also be seen as a knowledge based system or even a system which learns. In such cases we chose the method which was more responsible for the generation of the musical output.

The length of the subsections varies, reflecting the amount of research done in such field in the AI domain generally (evolutionary methods and systems which learn are the most popular methods).

The structure of each subsection is: we begin by stating the possible reasons for using such methods, then we present some representative examples and finally briefly discuss the disadvantages of each of the methods.

2.1 Mathematical models

Stochastic processes (reviewed by Jones, 1981) and especially Markov chains (Ames, 1989; Cambouropoulos, 1994) have been used extensively in the past for algorithmic composition (*e.g.,* Xenakis, 1971). Probably the most important reason for this is their low complexity which makes them good candidates for real-time applications. Many commercial programs use stochastic processes for this reason (*e.g.*, *M* and *Jam Factory*, see Zicarelli, 1987).

[1] Panel Discussion in the ICMC '93, concatenation of different definitions of the two words.

[2] See also Ames (1989, p. 185) for a different opinion.

Cybernetic Composer (Ames and Domino, 1992) is a representative example of such models. It composes pieces in different genres, such as jazz, rock or ragtime. One of the interesting features of this system is that it first deduces the rhythm of the melody using Markov chains, and then chooses the pitches at a later stage.

We also see computational models based on chaotic nonlinear systems (see, for example Pressing, 1988; Herman, 1993; Harley, 1994) or iterated functions (Gogins, 1991) but 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.

Conklin and Witten (1995) examine the prediction and generation of music using a multiple viewpoint system. They use *machine learning* techniques to extract the information from a number of examples in order to create their models. These are measured based on the notion of entropy or unpredictability (Shannon, 1951) and are used to create new pieces.

The main disadvantages of stochastic processes are:

- First, someone needs to find the probabilities by analysing many pieces. Something necessary if we want to simulate one style; and

- Second, the deviations from the norm and how they are incorporated in the music is an important aspect (it is difficult to capture higher or more abstract levels of music)[3].

2.2 Knowledge based systems

In one sense, most AI systems are knowledge based systems (KBS). Here, we mean systems which are symbolic and use rules or constraints. The use of KBS in music seems to be a natural choice especially when we try to model well defined domains or we want to introduce explicit structures or rules. Their main advantage is that they have explicit reasoning; they can explain their choice of actions.

Ebcioglu (1988) implemented his own Backtracking Specification Language (BSL) and used it to implement CHORAL, a rule-based expert system for the harmonization of chorales in the style of J.S.Bach.

Harmonization is usually reviewed as a constraint satisfaction problem, dealing with constraints such as voice range, voice leading, *etc*. Tsang and Aitken (1991) and Pachet and Roy (1998) use constraint logic programming (CLP) and constraint satisfaction techniques (CSP) respectively for harmonization, with the former being much more efficient.

Ramalho and Ganascia (1994), Zimermann (1998) and Robertson et al. (1998) tried to simulate creativity by making intention-based music.

Ramalho and Ganascia used the notion of potential actions (Pachet, 1990; Pachet et al., 1996) in order to create real-time jazz improvisations. Instead of using randomness to emulate creativity they used a set of potential actions as an initial state (reference) for their musical problem (to create playable improvisations).

Zimermann used multimedia presentations as a reference for the intensional structure of the music. Robertson et al. adapted musical techniques used in films to generate atmospheric music suited to an educational virtual environment. The desired tension curve of the music was used as an input to the system.

Even though KBS seem to be the most suitable choice, as a stand alone method, for algorithmic composition they still exhibit some important problems:

- Knowledge elicitation is difficult and time consuming, especially in subjective domains such as music.

- Since they do what we program them to do they depend on the ability of the "expert", who in many cases is not the same as the programmer, to clarify concepts, or even find a flexible representation.

- They become too complicated if we try to add all the "exceptions to the rule" and their preconditions, something necessary in this domain.

For more knowledge based, and not only, approaches see Balaban et al. (1992) or Schwanauer and Levitt (1993).

2.3 Grammars

"The idea that there is a grammar of music is probably as old as the idea of grammar itself" (Steedman, 1996).

Experiments in Musical Intelligence (EMI) is a project focused on the understanding of musical style and stylistic replication of various composers (Cope, 1991, 1992). EMI needs as an input a minimum of two works from which extracts "signatures" using pattern matching. The meaningful arrangement of these signatures in replicated works is accomplished through the use of an augmented transition network (ATN).

Steedman (1984) devised a generative grammar for chord progressions in jazz twelve-bar blues and refined it (Steedman, 1996) using *categorial grammars* (Steedman, 1989) because "they allow left-branching analyses of what are traditionally viewed as right-branching constructions" making them "incrementally interpretable from left to right", which simulates more closely the listener's perception and interpretation of the chord progressions.

Johnson-Laird (1991) used also grammars for the generation of jazz chord progressions and bass line improvisations.

Some basic problems of the grammars are:

- They are hierarchical structures while much music is not (i.e improvisation). Therefore ambiguity

[3] See also section 4

might be necessary since it “can add to the representational power of a grammar” (Roads, 1985)

- Most, if not all, musical grammar implementations do not make any strong claims about the semantics of the pieces.
- Usually a grammar can generate a large number musical strings of questionable quality.
- Parsing is, in many cases, computationally expensive especially if we try to cope with ambiguity.

For a more complete review on grammars in music see Roads (1985).

2.4 Evolutionary methods

Genetic algorithms (GAs) have proven to be very efficient search methods (Holland, 1975; Goldberg, 1989), especially when dealing with problems with very large search spaces. This, coupled with their ability to provide multiple solutions, which is often what is needed in creative domains, makes them a good candidate for a search engine in a musical application.

We can divide the following attempts into two categories based on the implementation of the fitness function.

Use of an objective fitness function In this case the chromosomes are evaluated based on formally stated, computable functions. Horner and Goldberg (1991) used GAs for thematic bridging between very simple melodies. McIntyre (1994) generated a four part Baroque harmonization of an input melody, using only the C major scale in order to reduce the search space. Spector and Alpern (1994) used genetic programming (GP) (Koza, 1992) in order to generate programs that produce a four-measure melody as an output when given a four-measure melody as an input. They used five critical functions to evaluate the response.

Papadopoulos and Wiggins (1998) used a symbolic GA with problem dependent genetic operators, variable length chromosomes and a fitness function which evaluated eight different characteristics of the melody, such as consecutive intervals, note durations, contour, in order to evolve jazz melodies based on a given chord progression. This and another study on harmonization (Phon-Amnuaisuk et al., 1999) make clear that the efficacy of the GA approach depends heavily on the amount of knowledge the system possesses; even so they conclude that GAs are not ideal for the simulation of human musical thought because “their operation in no way simulates human behaviour” (Wiggins et al., 1999).

Use of a human as a fitness function Usually we refer to this type of GA as interactive-GA (IGA). In this case a human replaces the fitness function in order to evaluate the chromosomes. Jacob (1995) devised a composing system; Horowitz (1994) generated “tasteful” rhythmic patterns; Ralley (1995) developed melodies; and Biles (1994), evolved a “novice jazz musician learning to improvise”. All these attempts exhibit two main drawbacks associated with all IGAs:

- Subjectivity (Ralley, 1995), and
- Efficiency, the “fitness bottleneck” (see Biles 1994) – the user must hear all the potential solutions in order to evaluate a population.

Moreover, this approach tells us little about the mental processes involved in music composition since all the reasoning is encoded inaccessibly in the user’s mind.

Most of the approaches above exhibit very simple representations in an attempt to decrease the search space, which in some cases compromises their output quality.

For a more complete summary of GA/GP work in music see Burton and Vladimirova (1997b).

2.5 Systems which learn

In the category of learning systems are systems which, in general, do not have a priori knowledge (e.g. production rules, constraints) of the domain, but instead learn its features by examples. We can further classify these systems, based on the way they store the information, to subsymbolic/distributive (Artificial Neural Networks, ANN) and symbolic (Machine Learning, ML).

ANNs have been used extensively in the last years for musical applications (Todd and Loy, 1991; Leman, 1992; Griffith and Todd, 1997), and have been relatively successful, especially in domains such as perception and cognition.

Todd (1989) used a feed-forward ANN with feedback for melody generation. Mozer (1994) generated melodies using ANNs which, as he states, “are preferred over compositions generated by a third-order transition table” but still “suffer from a lack of global coherence”.

Bellgard and Tsang (1994) constructed an effective Boltzmann machine (EBM) for harmonization. The interesting feature of their system is that not only it generates harmonies non-deterministically, but it also provides a measure of their relative quality.

Toiviainen (1995) trained a ANN for jazz improvisation, while Hörnel and Degenhardt (1997) did the same for baroque-style melodic improvisation. Hörnel (1997), finally, created baroque-style chorale variations.

Melo (1998) used two cooperative ANNs operating on different levels in an attempt to capture harmonic tension in music. The ANNs were trained based on the mediant of the tension curve reported by 10 listeners who were asked to listen to the last movement of Prokovief’s 1st Symphony and to indicate their estimation of dynamic musical tension by pushing a sprung wheel. The ANNs could predict quite well the tension of an unseen part of the piece (80% was used as training data) and could also generate music, based on a given tension curve, but not as successfully.

ML implementations are not very common. Widmer (1992) used ML for the harmonization of melodies. Cope's work (see above) could also fit into this category. (Ponsford et al., forthcoming) derived a probabilistic grammar capturing the harmonic movement of a corpus of seventeenth-century dance music.

Schwanauer (1993) used five learning techniques, learning by rote, learning from instruction, learning from failure, learning from examples, learning by analogy and learning from discovery for the implementation of a system (MUSE) which could accomplish different harmonization tasks, from the simpler, completing the inner voices for a given soprano and bass, to the most general, harmonizing a chorale.

ANNs offer an alternative for algorithmic composition to the traditional symbolic AI methods, one which loosely resembles the activities in the human brain, but at the moment they do not seem to be as efficient or as practical, at least as a stand-alone approach. Some of their disadvantages are:

- Composition as compared with cognition is a much more highly intellectual process (more "symbolic"). The output from a ANN matches the probability distribution of the sequence set to which it is exposed (Bharucha and Todd, 1991), something which is desirable, but on the other hand shows us its limit: "While they (ANNs) are capable of successfully capturing the surface structure of a melodic passage and produce new melodies on the basis of the thus acquired knowledge, they mostly fail to pick up the higher-level features of music, such as those related to phrasing or tonal functions" (Toiviainen, 1999).

- The representation of time can not be dealt efficiently even with ANNs which have feedback.

- Usually they solve toy problems, with many simplifications, when compared with the knowledge based approaches (Toiviainen, 1999).

- They can not even reproduce fully the training set and when they do this it might mean that they did not generalise.

- Even though it seems exciting that a system learns by examples this is not always the whole truth since the human in many cases needs to do the "filtering" in order not to have in the training set examples which conflict.

- Usually, the researchers using ANNs say that their advantage over knowledge based approaches is that they can learn from examples things which can't be represented symbolically using rules (i.e. the "exceptions"). However, we haven't seen such systems.

2.6 Hybrid systems

Hybrid systems are ones which use a combination of AI techniques. In this section we discuss systems which combine evolutionary and connectionist methods, or symbolic and subsymbolic ones.

The reason behind using hybrid systems, not only for musical applications, is very simple and logical. Since each AI method exhibits different strengths then we should adopt a "postmodern" attitude (Gutknecht, 1992) by combining them.

Gibson and Byrne (1991) created simple harmonizations, using only the tonic subdominant and dominant chords, of short melodies using a combination of a GA and cooperating ANNs. Spector and Alpern (1995) used GP to create a one measure response to a one measure 'call', with a ANN as a fitness evaluator of the response.

Biles et al. (1996), in an attempt to increase the efficiency of Biles' (1994) system, also used a ANN as a fitness function, without very successful results. Burton and Vladimirova (1997a) used an Adaptive Resonance Theory (ART) ANN to assign fitness measures to rhythms generated by a GA.

HARMONET (Hild et al., 1992; Feulner, 1993) is a hybrid system which harmonizes melodies using a combination of a ANN (which learns different harmonizations) with constraints satisfaction techniques (in order to fill in the inner voices). It uses distributive representation but unlike other ANN representations the input nodes do not represent notes rather harmonic functions (i.e tonic, dominant).

The main disadvantage of hybrid systems is that they are usually complicated, especially in the case of tightly-coupled or fully integrated models. The implementation, verification and validation is also time consuming.

3 Discussion

In this section we present a general discussion of some issues raised by our survey.

3.1 Evaluation of the systems

We can not help not to notice a twofold lack of experimental methodology in many research reports in this area. First there is usually no evaluation of the output by real experts (*e.g.,* professional musicians) in most of the systems and second, the evaluation of the system (algorithm) is given relatively small consideration with respect to the length of the report.

There are some musical questions about systems which only generate melodies (*e.g.,* Todd, 1989; Ralley, 1995; Spector and Alpern, 1995). How can we expect to evaluate the musical output if we do not have a harmonic context for it? Most melodies will sound acceptable in some context or other.

HARMONET (see above) is an example of successful task allocation. Even though the authors state that it's a ANN, we believe that it is its hybrid nature (ANN + CSP) which makes it effective. A ANN would be much less successful at filling in the inner voices. So it would be a big claim to say that the ANN is responsible for the success.

Finally, most of the systems deal with algorithmic composition as a problem solving task rather as a creative and meaningful process (see also sections 3.3 and 4, below).

3.2 Knowledge representation

Two almost ubiquitous issues in AI are representation of knowledge and search method. From one point of view, our categorisation above, reflects the search method, which however, constrains the possible representations of knowledge. For example structures which are easily represented symbolically are often difficult to represent with a ANN.

In many AI systems, especially symbolic, the choice of the knowledge representation is an important factor in reducing the search space. For example Biles (1994) and Papadopoulos and Wiggins (1998) (see section 2) used a more abstract representation, representing the degrees of the scale rather than the absolute pitches. This reduced immensely the search space since the representation did not allow the generation of non-scale notes (they are considered dissonant) and the inter-key equivalence was abstracted out.

Most of the systems reviewed exhibit a single, fixed representation of the musical structures. Some, on the other hand, use multiple viewpoints (*e.g.,* Ebcioglu, 1988; Conklin and Witten, 1995) which we believe simulate more closely human musical thinking.

3.3 Computational Creativity

Probably the most difficult task is to incorporate in our systems the concept of creativity. This is difficult since we do not have a clear idea of what creativity is (Boden, 1996).

Some characteristics of computational creativity, which were proposed by Rowe and Partridge (1993) are:

- Knowledge representation is organised in such a way that the number of possible associations is maximised. A flexible knowledge representation scheme. Similarly Boden (1996) says that representation should allow to explore and transform the *conceptual space*.

- Tolerate ambiguity in representations.

- Allow multiple representations in order to avoid the problem of "functional fixity".

- The usefulness of new combinations should be assessable.

- New combinations need to be elaboratable to find out their consequences.

One question that AI researchers should aim to answer is: do we want to simulate human creativity itself or the results of it? (Is DEEP BLUE creative, even if it does not simulate the human mind?) This is more or less similar to the, subtle in many cases, distinction between cognitive modeling and knowledge engineering.

Even after all these, will computers be able to emulate our musical thinking? Kugel (1990), in a very interesting article, expands on what Myhill (1952) seems to have first proposed, that there is more than computing in musical thinking. He proposes that we should add uncomputable processes "to our conceptual palette". These have also been called *trial-and-error* processes by Putnam (1965) and *limiting-computable processes* by Gold (1965).

4 Prospects for the future

In this section we discuss the possible future prospects in algorithmic composition.

The big disadvantage of most, if not all, the computational models (in varying degrees) is that the music that they produce is meaningless: the computers do not have feelings, moods or intentions, they do not try to describe something with their music as humans do. Most of human music is referential or descriptive. The reference can be something abstract like an emotion, or something more objective such as a picture or a landscape.

How can we incorporate concepts such as musical meaning in systems?

- We propose that future systems should explicitly refer to and evaluate factors such as *musical tension* (Lerdahl, 1988, 1996), *intension* (Ramalho and Ganascia, 1994; Zimermann, 1998), *expectation and melodic closure* (Narmour, 1990, 1992). Let us make this more clear with a simple example. If we have a statistical analysis (or train a ANN) on the dynamics of a piece and we find that 10% of the notes are relatively "loud" (or have something like this as "distributed knowledge" in the case of the ANN) we lose the important fact that it's the context which matters (these notes are probably related and not distributed randomly, creating for example a crescendo). It is this planned deviation from the norm and how it is accomplished that gives meaning to music (Meyer, 1956).

- The performance of a piece is another important factor and should not be regarded as irrelevant in the context of algorithmic composition. Especially in cases of improvised music the rhythmic deviations and the “gestures” are responsible for meaning (Keil, 1994). We have all felt “strange” the first time we heard a computerised reproduction of a known musical piece.

- We propose that geometrical models of pitch space and pitch perception (Longuet-Higgins, 1962a,b; Balzano, 1980; Shepard, 1982), even though they exhibit some disadvantages (Krumhansl, 1990), could prove useful if we incorporate them in such systems, something which was more implicit than explicit in some of the above implementations. ANNs also seem to be a valid choice for a cognitive model of pitch perception, especially tonality (see for example Rowe, 1993).
- We need multiple, flexible, dynamic, even expandable representations because this will more closely simulate human behaviour.

What is missing is a thorough-going account of musical cognition from the early stages of perception to the complex stage of creation and composition.

5 Conclusion

In this paper we have given a critical review of different AI approaches for algorithmic composition.

Systems based on only one method do not seem to be very effective. We could conclude that it will become more and more common to “blend” different methods and take advantage of the strengths of each one.

Finally, an exciting, but very difficult, prospect is that of an integrated system which evolves; a system which absorbs new knowledge (being able to combine different styles, showing creativity).

Acknowledgements

Special thanks to Ben Curry for his useful comments on drafts of this paper.

References

C. Ames. The Markov Process as a Compositional Model: A Survey and Tutorial. *Leonardo*, 22 (2):175–187, 1989.

C. Ames and M. Domino. Cybernetic composer: An overview. In M. Balaban, K. Ebcioglu, and O. Laske, editors, *Understanding Music with AI*, pages 186–205. AAAI Press, 1992.

M. Balaban, K. Ebcioglu, and O. Laske. *Understanding Music with AI*. AAAI Press, 1992.

G. J. Balzano. The Group-theoretic Description of 12-Fold and Microtonal Pitch Systems. *Computer Music Journal*, 4 (4): 66–84, 1980.

M. I. Bellgard and C. P. Tsang. Harmonizing Music the Boltzmann Way. *Connection Science*, 6 (2-3):281–297, 1994.

J. J. Bharucha and P. M. Todd. Modelling the Perception of Tonal Structure with Neural Nets. In P. M. Todd and G. Loy, editors, *Music and Connectionism*, pages 128–137. MIT Press, 1991.

J. A. Biles. Genjam: A Genetic Algorithm for Generating Jazz Solos. In *Proceedings of the International Computer Music Conference*, 1994.

J. A. Biles, P. G. Anderson, and L. W. Loggi. Neural Network Fitness Functions for a Musical IGA. Technical report, Rochester Institute of Technology, 1996.

M. Boden. What is Creativity. In M. Boden, editor, *Dimensions of Creativity*, pages 75–118. MIT Press, 1996.

A. R. Burton and T. Vladimirova. A Genetic Algorithm Utilising Neural Network Fitness Evaluation for Musical Composition. In *International Conference on Genetic Algorithms and Artificial Neural Networks*, 1997a.

A. R. Burton and T. Vladimirova. *Applications of Genetic Techniques to Musical Composition.* Submitted to the Computer Music Journal. Available by WWW at http://www.ee.surrey.ac.uk/Personal/A.Burton/work.html, 1997b.

E. Cambouropoulos. Markov Chains As an Aid to Computer Assisted Composition. *Musical Praxis*, 1 (1):41–52, 1994.

N. Chomsky. *Syntactic Structures*. The Hague : Mouton, 1957.

D. Conklin and I. H. Witten. Multiple Viewpoint Systems for Music Prediction. *Journal of New Music Research*, 24:51–73, 1995.

D. Cope. *Computers and Musical Style*. Oxford University Press, 1991.

D. Cope. Computer Modeling of Musical Intelligence in EMI. *Computer Music Journal*, 16 (2):69–83, 1992.

D. Cope. Panel discussion. In *Proceedings of the International Computer Music Conference*, 1993.

K. Ebcioglu. An Expert System for Harmonizing Four-part Chorales. *Computer Music Journal*, 12 (3):43–51, 1988.

J. Feulner. Neural Networks that Learn and Reproduce Various Styles of Harmonization. In *Proceedings of the International Computer Music Conference*, 1993.

P. M. Gibson and J. A. Byrne. Neurogen, musical composition using genetic algorithms and cooperating neural networks. In *Proceedings of the 2nd International Conference in Artificial Neural Networks*, 1991.

M. Gogins. Iterated Functions Systems Music. *Computer Music Journal*, 15 (1):40–48, 1991.

E. M. Gold. Limiting Recursion. *Journal of Symbolic Logic*, 30 (1):28–48, 1965.

D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

N. Griffith and P. M. Todd, editors. *Musical Networks.* MIT Press, 1997.

M. Gutknecht. The ‘postmodern mind’: hybrid models of cognition. *Connection Science*, 4 (3-4):339–364, 1992.

J. Harley. Algorithms Adapted From Chaos Theory. In *Proceedings of the International Computer Music Conference*, 1994.

M. Herman. Deterministic Chaos, Iterative Models, Dynamical Systems and Their Application in Algorithmic Composition. In *Proceedings of the International Computer Music Conference*, 1993.

H. Hild, J. Feulner, and W. Menzel. HARMONET: A Neural Net for Harmonizing Chorals in the Style of J.S. Bach. In R. P. Lippmann, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing 4 (NIPS 4)*, pages 267–274. Morgan Kaufmann, 1992.

L. Hiller and L. Isaacson. *Experimental Music.* New York: McGraw-Hill, 1959.

J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Mitchigan Press, 1975.

D. Hörnel. MELONET I: Neural Nets for Inventing Baroque-Style Chorale Variations. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing 10 (NIPS 10)*. MIT Press, 1997.

D. Hörnel and P. Degenhardt. A Neural Organist improvising baroque-style melodic variations. In *Proceedings of the International Computer Music Conference*, Thessaloniki. Greece, 1997.

A. Horner and D. E. Goldberg. Genetic Algorithms and Computer-Assisted Composition. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, 1991.

D. Horowitz. Generating Rhythms with Genetic Algorithms. In *Proceedings of the International Computer Music Conference*, 1994.

Xenakis I. *Formalized Music.* Indiana University Press, 1971.

B. L. Jacob. Composing with Genetic Algorithms. In *Proceedings of the International Computer Music Conference*, 1995.

P. N. Johnson-Laird. Jazz Improvisation: A Theory at the Computational Level. In P. Howell, R. West, and I. Cross, editors, *Representing Musical Structure*. Academic Press, 1991.

K. Jones. Compositional applications of stochastic processes. *Computer Music Journal*, 5 (2), 1981.

C. Keil. Motion and Feeling Through Music. In C. Keil and S. Feld, editors, *Music grooves : essays and dialogues*, pages 53–76. University of Chicago Press, 1994.

J. R. Koza. *Genetic Programming.* MIT Press, 1992.

C. L. Krumhansl. *Cognitive foundations of musical pitch.* Oxford University Press, 1990.

P. Kugel. Myhill’s Thesis: There’s More than Computing in Musical Thinking. *Computer Music Journal*, 14 (3):12–25, 1990.

M. Leman. Artificial neural networks in music research. In A. Marsden and A. Pople, editors, *Computer Representations and Models in Music*, pages 265–301. Academic Press Ltd, 1992.

F. Lerdahl. Tonal Pitch Space. *Music Perception*, 5 (3):315–349, 1988.

F. Lerdahl. Calculating Tonal Tension. *Music Perception*, 13 (3):319–363, 1996.

H. C. Longuet-Higgins. Letter to a Musical Friend. *The Music Review*, 23:244–248, 1962a.

H. C. Longuet-Higgins. Second Letter to a Musical Friend. *The Music Review*, 23:271–280, 1962b.

R. A. McIntyre. Bach in a Box: The Evolution of Four-Part Baroque Harmony Using the Genetic Algorithm. In *Proceedings of the IEEE Conference on Evolutionary Computation*, 1994.

A. F. Melo. A Connectionist Model of Tension in Chord Progressions. MSc Thesis, School of Artificial Intelligence, University of Edinburgh, 1998.

L. B. Meyer. *Emotion and Meaning in Music.* University of Chicago Press, 1956.

M. C. Mozer. Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing. *Connection Science*, 6 (2-3): 247–280, 1994.

J. Myhill. Some Philisophical Implications of Mathematical Logic: Three Classes of Ideas. *Review of Metaphysics*, 6 (2): 165–198, 1952.

E. Narmour. *The Analysis and Cognition of Basic Melodic Structures. The Implication-Realization Model.* University of Chicago Press, 1990.

E. Narmour. *The Analysis and Cognition of Melodic Complexity. The Implication-Realization Model.* University of Chicago Press, 1992.

F. Pachet. Representing Knowledge Used by Jazz Musicians. In *Proceedings of the International Computer Music Conference*, 1990.

F. Pachet, G. Ramalho, and J. Carrive. Representing Temporal Musical Objects and Reasoning in the MusES System. *Journal of New Music Research*, 25 (3):252–275, 1996.

F. Pachet and P. Roy. Formulating Constraint Satisfaction Problems on Part-Whole relations: the Case of Automatic Harmonization. In *Workshop at ECAI’98. Constraint Techniques for Artistic Applications*, Brighton, UK, 1998.

G. Papadopoulos and G. Wiggins. A Genetic Algorithm for the Generation of Jazz Melodies. In *STeP’98*, Jyväskylä,Finland, 1998.

S. Phon-Amnuaisuk, A. Tuson, and G. Wiggins. Evolving Musical Harmonization. In *ICANNGA'99*, Slovenia, 1999.

D. Ponsford, G. Wiggins, and C. Mellish. Statistical Learning of Harmonic Movement. *Computer Music Journal*, forthcoming.

J. Pressing. Nonlinear Maps as Generators of Musical Pitch. *Computer Music Journal*, 2 (2):35–46, 1988.

H. Putnam. Trial-and-Error Predicates and the Solution to a problem of Mostowski. *Journal of Symbolic Logic*, 30 (1): 49–57, 1965.

D. Ralley. Genetic algorithms as a tool for melodic development. In *Proceedings of the International Computer Music Conference*, 1995.

G. Ramalho and J. G. Ganascia. Simulating Creativity in Jazz Performance. In B. Hayes-Roth and R. Korf, editors, *National Conference on Artificial Intelligence 1994*, volume 1, pages 108–113. The AAAI/The MIT Press, 1994.

C. Roads. Grammars as Representations for Music. In C. Roads and J. Strawn, editors, *Foundations of Computer Music*, pages 403–442. MIT Press, 1985.

J. Robertson, A. de Quincey, T. Stapleford, and G. Wiggins. Real-Time Music Generation for a Virtual Environment. In *ECAI98 workshop on AI/Alife and Entertainment*, Brighton, England, 1998.

J. Rowe and D. Partridge. Creativity: A survey of AI approaches. *Artificial Intelligence Review*, 7:43–70, 1993.

R. Rowe. *Interactive music systems : machine listening and composing.* MIT Press, 1993.

S. M. Schwanauer. A Learning Machine for Tonal Composition. In S.M. Schwanauer and D.A. Levitt, editors, *Machine Models of Music*, pages 511–532. MIT Press, 1993.

S. M. Schwanauer and D. A. Levitt. *Machine Models of Music.* MIT Press, 1993.

C. E. Shannon. Prediction and entropy of printed English. *Bell System Tchnical Journal*, pages 50–64, 1951.

R. N. Shepard. Geometrical Approximations to the Structure of Musical Pitch. *Psychological Review*, 89:305–333, 1982.

L. Spector and A. Alpern. Critisism, Culture, and the Automatic Generation of Artworks. In B. Hayes-Roth and R. Korf, editors, *National Conference on Artificial Intelligence 1994*, volume 1, pages 3–8. The AAAI/The MIT Press, 1994.

L. Spector and A. Alpern. Induction and recapitulation of deep musical structure. In *Proceedings of the IJCAI-95 Workshop on Artificial Intelligence and Music*, 1995.

M. Steedman. A Generative Grammar for Jazz Chord Sequences. *Music Perception*, 2:52–77, 1984.

M. Steedman. Grammar, Interpretation and Processing from the Lexicon. In W. Marslen-Wilson, editor, *Lexical Representation and Process.* MIT Press, 1989.

M. Steedman. The Blues and the Abstract Truth: Music and Mental Models. In A. Garnham and J. Oakhill, editors, *Mental Models in Cognitive Science.* Erlbaum, Mahwah, NJ, 1996.

P. M. Todd. A Connectionist Approach to Algorithmic Composition. *Computer Music Journal*, 13 (4):27–43, 1989.

P. M. Todd and G. Loy, editors. *Music and Connectionism.* MIT Press, 1991.

P. Toiviainen. Modeling the target-note technique of bebop-style jazz improvisation: An artificial neural network approach. *Music Perception*, 12 (4):399–413, 1995.

P. Toiviainen. Symbolic AI Versus Connectionism in Music Research. In E. Miranda, editor, *Readings in Music and Artificial Intelligence.* Gordon and Breach, 1999.

C. P. Tsang and M. Aitken. Harmonizing music as a discipline of constraint logic programming. In *Proceedings of the International Computer Music Conference*, Montréal, 1991.

G. Widmer. Qualitative Perception Modeling and Intelligent Musical Learning. *Computer Music Journal*, 16 (2):51–68, 1992.

G. Wiggins, G. Papadopoulos, S. Phon-Amnuaisuk, and A. Tuson. Evolutionary Methods for Musical Composition. *International Journal of Computing Anticipatory Systems (in press)*, 1999.

D. Zicarelli. M and Jam Factory. *Computer Music Journal*, 11 (4):13, 1987.

D. Zimermann. Modelling Musical Structures. Aims, Limitations and the Artist's Involvement. In *Workshop at ECAI'98. Constraint Techniques for Artistic Applications*, Brighton, UK, 1998.