Proceedings of the 2nd Conference on Nonlinear Science and Complexity
Porto, Portugal, July 28-31, 2008

ALGORITHMIC MUSIC COMPOSITION: A SURVEY

ISABEL M. FERREIRA
Faculty of Engineering of the University of Porto
Dept. Electrical and Computer Engineering, Porto, Portugal
imf@fe.up.pt

J. TENREIRO MACHADO
Institute of Engineering of Porto
Dept. Electrical Engineering, Porto, Portugal
jtm@isep.ipp.pt

This paper surveys some of the methods used for algorithmic composition and their evolution during the last decades. Algorithmic composition was motivated by the natural need to assist and to develop the process of music creation. Techniques and applications of algorithmic composition are broad spectrum, ranging from methods that produce entire works with no human intervention, up to methods were both composer and computer work closely together in real-time. Common algorithms used for music composition are based in stochastic, deterministic, chaotic and artificial intelligence methods.

1. Introduction

For many centuries, philosophers, music composers and mathematicians worked hard to find mathematical formulae that could explain the process of music creation. As a matter of fact, music and mathematics are intricately related: strings vibrate at certain frequencies and sound waves can be described by mathematical equations. Although, it is not possible to find an “equation” that will model all music works it is true that there are certain inherent mathematical structures in all works of music.

Through the history of Western music, we have been faced with the proposal of formal techniques for melody composition claiming that musical pieces could be created as a result of applying certain rules to some given initial material (Figure 1). More recently, the exponential growth of computing power made it possible to generate music automatically. We will now present an overview of the common techniques of algorithmic composition from Ancient Greece to nowadays, distinguishing the pre/non-computer methods from those which use modern computational systems.

Proceedings of the 2nd Conference on Nonlinear Science and Complexity
Porto, Portugal, July 28-31, 2008

Figure 1. Algorithmic composition.

Bearing these ideas in mind this paper is organized as follows. Section 2 describes the first approaches to algorithmic composition throughout centuries from ancient Greece to the beginning of the 20th century. Section 3 outlines the evolution of the properly called algorithmic composition methods, from Xenakis up to nowadays. Section 4 broadly classifies algorithmic composition methods attending to their structure and the techniques used to process musical data, pointing out their main characteristics. This classification could never be exhaustive given the so large number of attempts that have been proposed in the literature for the last decades. Section 5 briefly discusses the importance of algorithmic composition and its interaction with the traditional compositional methods based in the composer's creativity outlining the main conclusions.

2. Pre-computer Methods: From Pythagoras to Cage

The idea of using formal instructions and processes to create music dates back to the ancient Greece. Pythagoras believed in a direct relation between the laws of nature and the harmony of sounds as expressed in music. He was probably the first one to study the correlation between numerical ratios and consonant sounds and he constructed a musical scale using a method that was basically an iterative process. Starting with one particular note and using simple whole-number {1; 2; 3; 4}, ratios we could generate the Pythagorean scale. This process led the Pythagoreans to a general theory asserting that "if the ratio between any two given notes can be represented by a rational number p/q where p and q are small integers, then the two notes are consonant [1]. Applying this process repeatedly we obtain an infinite number of notes generating the so called spiral of Pythagorean fifths [2], as in Figure 2.

Ptolemy and Plato were two other Greek philosophers who wrote about this relationship among music, mathematics and nature, creating the myth of the "Music of the Spheres" (Figure 3).

In spite of all these ancient Greek formalisms rooted in nature observation and mathematical conjectures, the Greek music at the time can not be considered as a result of algorithmic composition in pure sense, since it was almost completely improvised.

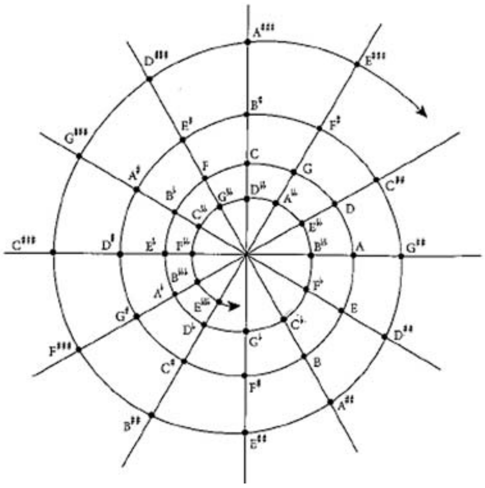

Figure 2. Spiral of Pythagorean fifths.

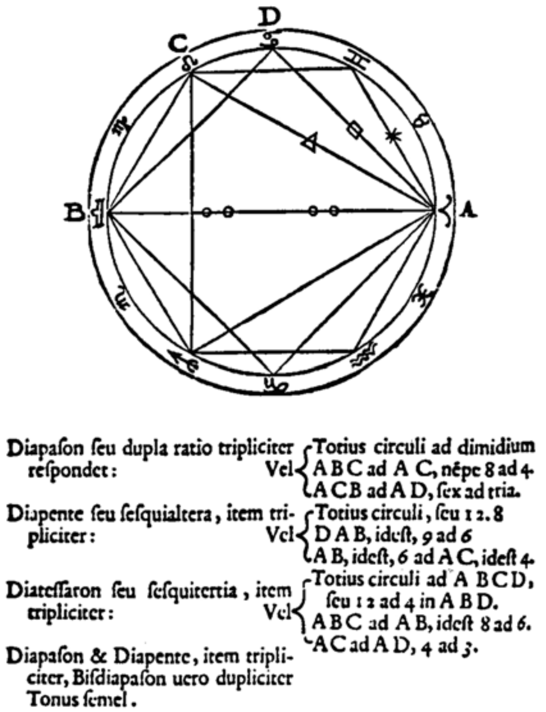

Figure 3. Musical consonances and astrological “aspects” from Ptolemy´s Harmonica [22].

However, these formalisms were a first attempt to establish formal extra human processes of music composition. The next important step towards the definition of an algorithm for music composition is given with the birth of the "canonic" composition in the late 15th century [3]. The term "canon" derives from the ancient Greek word of kanon which means rule or law. This method consists in defining a primary leader melody which is then accompanied by a number of imitations or variations on this part, called followers. In this case we find an effective removal of the composer from a large part of the compositional process. The composer's creativity is restricted to the definition of a single melody or section – leader part – from which an entire composition is automatically obtained by applying different transformations such as inversion or varying parameters such as the number of followers and the delays between them.

" The prevailing method was to write out a single voice part and to give instructions to the singers to derive the additional voices from it … the second voice might be instructed to sing the same melody starting a certain number of beats or measures after the original; the second voice might be an inversion of the first or it might be a retrograde…" [4].

Canonic compositions have a strong influence in contemporary music composition with the development of methods for systematic ordering of musical events such as serialism [5] and the twelve-note system devised by Arnold Schoenberg in the early twentieth century [6] (Figure 4).

These methods are rigorous systems of composition grounded in mathematical sources, such as set theory, that tried to completely control all music parameters and to abstract the compositional process as much as possible. Different elements such as notes, pitch, dynamic markings and other parameters are ordered strictly according to the pre-determined sets.

In the eighteenth century we were also faced with the popularization of the Musikalisches Würfelspiel or Musical Dice Games [7], probably de most well-known example of algorithmic composition prior to the 20th century. This technique usually credited to Mozart involved a collection of small music fragments whose sequence would be selected by throwing a number of dice (Figure 5).

The measures used were composed in such a way that they could be combined in any number of ways. This idea of leaving the creative decisions in the hands of chance forms the basis of modern probabilistic algorithmic composition methods.

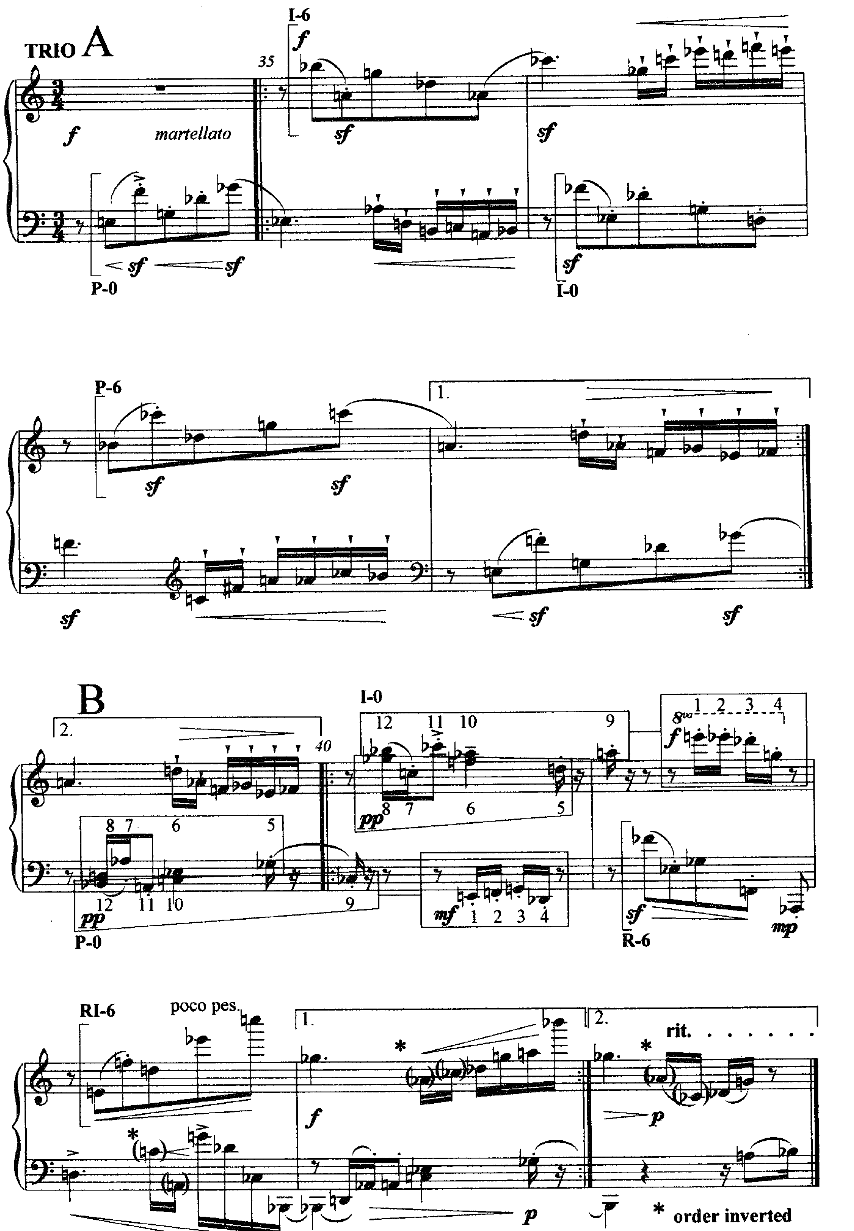

Figure 4. The opening of the Trio from the Minuet and Trio of Schoenberg's Piano suite, Op.25 [3].

Measure: / Dice:	1	2	3	4	5	6
2	70	34	68	18	32	58
3	10	24	50	46	14	26
4	42	6	60	2	52	66
5	62	8	36	12	16	38
6	44	56	40	79	48	54
7	72	30	4	28	22	64
8	114	112	126	87	89	88
9	123	116	137	110	91	98
10	131	147	143	113	101	[illegible]
11	138	151	118	124	141	127
12	144	153	146	28	150	154

Figure 5. Musical Dice Games

The work of John Cage and Pierre Boulez [8], in the twentieth century, was influenced by this principle. As a matter of fact, John Cage, an admirer of the wisdom of Buddhism, used randomness in many of his compositions, just like Mozart did. In his work Music of Changes, Cage created charts of pre-orchestrated combinations of sounds using the I Ching to select between them so attempting to rid his work of personal choice and ego [9]. Realizing that he had reduced the role of the performer to the of an automaton, John Cage developed his method and introduced randomness in later works such us Europera where the performers could choose between chance operations and their own free will [10]. One of his compositions named Reunion, was performed by playing chess on a board equipped with photo-receptors resulting in a different musical piece each time the game is performed [11].

3. Computer Composition

The incredibly fast development of technology in the 20th century, with the generalized use of computers more and more powerful, brought many innovations into algorithmic composition.

In the 19th century, Ada Lovelace, inventor of the “calculating machine”, had already predicted the importance of computers in the process of automated composition.

“Supposing, for instance, that the fundamental relations of pitched sound in the signs of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent" [12].

Among different composers that used algorithmic composition as a creative tool we find two important names: Lejaren Hiller and Iannis Xenakis.

Lejaren Hiller was the first one to use a computer, the Illiac, to compose music. With the help of Leonard Isaacson he wrote in 1956 the first computer algorithmic music composition, the Illiac Suite for String Quartet [13]. The result of the algorithm was then transposed into traditional musical notation, for performance by a string quartet.

The algorithm developed by Hiller and Isaacson relies on a generator-modifier-selector approach to simulate the composition process. First the algorithm generates some raw musical material (the composition core), which is then modified by the application of various techniques, and finally the best result is selected according to some predetermined rules [14]. Although the final composition is produced by an algorithm, it still operates within certain bounds and criteria defined by the composer.

The generator-modifier-selector approach was later applied to MUSICOMP (music simulator interpreter for compositional procedures), one of the first computer systems for automated composition, written in the late 1950s and early 1960s by Hiller and Robert Baker. They developed the Computer Cantata using MUSICOMP to demonstrate all its flexibility and generality. This algorithm was written as a library of subroutines giving the composer freedom in his creative work. This approach is well suited to automated composition and still used in many algorithmic composition systems nowadays.

Another important name in the use of computers for music composition is Iannis Xenakis. He used computers to aid in the composition of scores using statistical and probabilistic methods. Xenakis used stochastic algorithms to explore compositions as sequences of "clouds of sound" – individual sonic events made up of thousands of isolated point sounds [15]. It is worth saying that his book "Formalized Music" [16] stands as one of the most important books in the computer music bibliography used as a main reference even nowadays. However, with Xenakis "the computer has not actually produced the resultant sound; it has only aided the composer by virtue of its high-speed computations" [17]. In fact, the computer's output was not the composition itself but material with which Xenakis could compose. On the contrary, the work of Hiller and Isaacson attempted to simulate the entire compositional process.

More recently many researchers have tried to address the problem of algorithm composition from different points of view. Cope developed an expert system that was able to generate a composition in the style of a given prominent classical composer making use of augmented transition networks (ATNs), a type of state machine commonly used in artificial intelligence processing [29]. Todd worked on a neural network which was trained to extract important features in a given music [30]. Mozer used a neural network system called CONCERT to predict note transitions based on learned style types [31]. Spector and Alpern proposed a genetic programming system which evolved responses to call phrases in jazz pieces [32]. Finally, a last reference to a jazz improvisation system called GenJam, developed by Biles [32]. The performance of this system is supervised by the user who behaves like a mentor to the algorithm, giving encouragement when the improvisation is good.

4. Algorithm Composition Methods

There is no universal rule to sort different compositional algorithms into categories. In fact, we can think about the way an algorithm takes part in the compositional process (i.e., in music composed by computer, or music composed with the aid of computer), examines the results of theirs compositional processes (sound-synthesis or event-generation algorithms) or, the most common, classifies compositional algorithms by their structure and the way of processing musical data. Using this last rule we can consider the following main methods: stochastic, deterministic, chaotic and artificial intelligence methods.

4.1. *Stochastic Methods*

A stochastic process depends on probability laws. In stochastic algorithm composition, a decision is taken according to the output of random generators so that the musical pieces are composed as a result of non-deterministic models. Stochastic processes work in parallel with probability distribution tables. One good example of stochastic algorithms is Markov chains. Hiller´s work uses Markov processes generating a matrix which maps out the probability that a certain note will be the next in the composition, given the previous note or sequence of notes that have occurred [14, 19] (Figure 6). In this figure the circles represent a particular note and the arcs indicate which notes can possibly occur next. The numerical values show us the probabilities of choosing a given arc. The matrices used are defined by the analysis of existing compositions.

It is obvious that when we use stochastic methods to generate musical pieces, a great importance has to be given in the weighting of the distribution table in order to get interesting results.

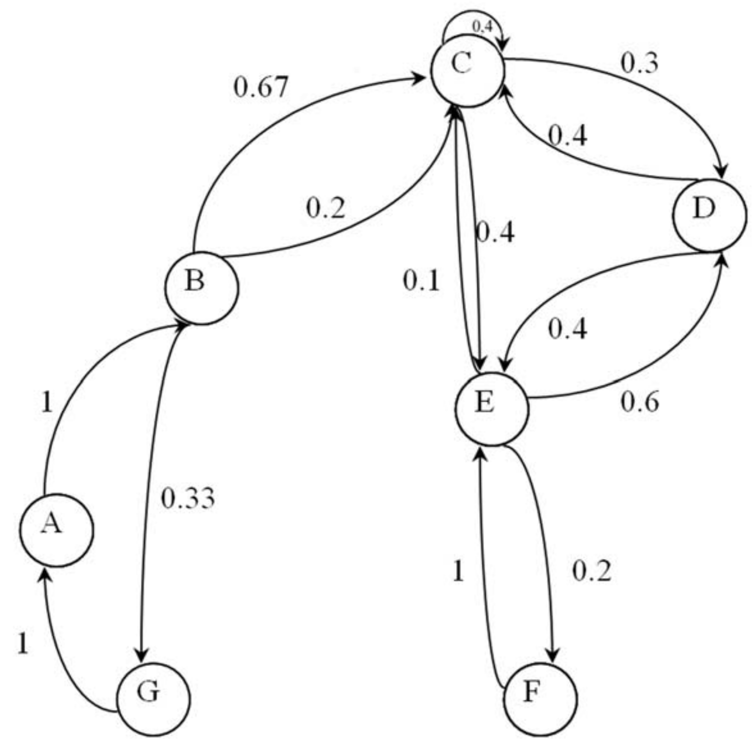

Figure 6. State diagram for a Markov process [19]

The earliest composition programs were almost exclusively driven by stochastic processes [14, 17], probably due to their low complexity which makes them suitable for real time applications. Since the early 1980s, developments in stochastic composition have been associated with Charles Anes, a student and follower of Lejaren Hiller. The Cybernetic Composer [23] is a good example of the use of stochastic models. It composes pieces in different genres, such as jazz, rock or ragtime, first deducing the rhythm of the melody using Markov chains. Different commercial programs such as M and Jam Factory [24] also use stochastic processes.

4.2. *Deterministic Methods*

Music composition using deterministic or rule-based techniques is not dependent of random generators but, instead, uses a system of rules to generate the score or other musical information. The Illiac Suite for a String Quartet is an example of this process. A deterministic algorithm defines a formal system of rules (a grammar) to be followed by the compositional process. Grammars often include rules for macro-level harmonies and rhythm, rather than single notes. These rules may be either collected from compositional techniques of the past or newly invented.

These methods, like Hiller´s MUSICOMP, usually assume the form of a computer program or a unified system of subroutines. One example of a deterministic method is that of William Shottstaedt´s automatic species counterpoint program that writes music based on rules stated in a counterpoint instruction book , Gradus and Parnassum, written by Johann Joseph Fux in the 18th century. This program follows almost 75 rules, such as “parallel fifths are not allowed” and “avoid tritons near the cadence in Lydian mode”. Based on Fux´s statement that some rules could never be broken but others were not so demanding, Schottstaedt defined a system of penalties for breaking the rules allowing the algorithm to leave a given path and go back to find a better solution [25]. Another example is that of Kemal´s Ebcioglu CHORAL system, which generates four-part chorales in the style of J.S. Bach following a set of over 350 different rules [25].

4.3. *Chaotic Methods*

As the name suggests this kind of algorithm is related to the theory of Chaos [19] which states that any system can be modeled by using very simple mathematical equations, no matter how complex it is.

Chaotic algorithm composition deals with a method fundamentally different from those already discussed. The main difference is that a chaotic algorithm exhibits non-linearity in oppose to the linear behavior of stochastic and deterministic methods. Different approaches have been proposed in the literature namely those of Pressing [26], Herman [27] and Harley [28]. Composers who have applied these mathematical principles of Chaos and Fractals to music composition are Rich Bidlack, Tommaso Bolognesi, Charles Dodge and Gary Lee Nelson among others. Most methods of relating fractal mathematics to musical compositions are based on selecting pitches according to fractal objects [20].

4.4. *Artificial Intelligent Methods*

These methods are similar to the deterministic approach in that they are programs or systems of programs, based on some pre-defined grammar, but they have the additional capacity of defining their own grammar, that is, they can learn from the composer/programmer inputs. David Cope´s system called Experiments in Musical Intelligence (EMI) , Figure 7, is an example of such a method [21]. EMI is based on a large database of style descriptions of different compositional approaches but it is able to update this database, creating its own grammar, based on the analysis of several scores of a specific composer [22].

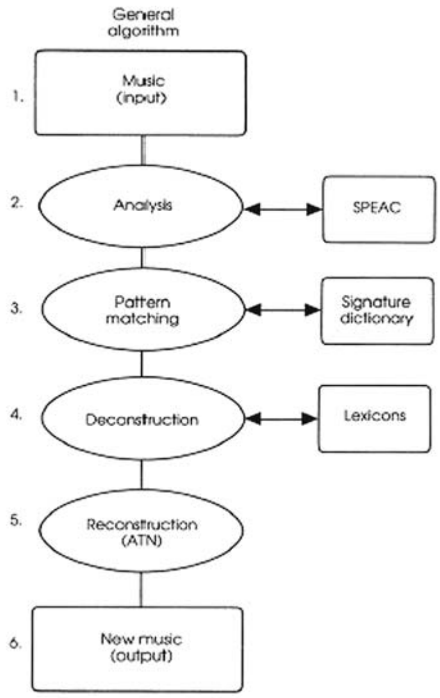

Figure 7. A general algorithm for EMI.

Genetic algorithms are known to show clear performance improvements when compared to enumerative, calculus-based and random searches of a given search space [34]. When we apply genetic algorithms to musical composition (Figure 8) three important problems arise that deserve particular attention, so that a meaningful and efficient algorithm might be developed.

The first subject to consider is the search domain. Attending to the large combinatorial possibilities of individual notes in time, rhythm, and harmony among other musical features, the search space basically has no limit. So, it is essential to impose constraints on the decision-making process so as to limit its size. Another important subject to study carefully is the input representation. Here we have to consider the musical information used by the algorithm to define pitch, rhythm, meter and other building blocks as well as the set of rules that formulate the evolution of a given composition during the successive iterations of the proposed algorithm. Finally, we arrive at the fitness evaluation, a measure of how well the rules are satisfied, which is the third critical subject to consider.

We can classify a given genetic compositional algorithm attending to the kind of fitness evaluation used. The most common evaluation methods proposed in literature are: deterministic, formalistic, user-determined and neural.

Deterministic evaluation methods use a mathematical function to give the fitness of each individual music event. Formalistic methods make use of stylistic features of existing music that are compiled into a set of rules. Algorithms employing user-determined fitness evaluation depend on the judgment of the user to assign fitness to individuals or to define the style of the output. Neural fitness evaluation methods implement a neural network that must identify a "good" individual according to its training data.

Applications that use deterministic fitness functions include melodic development and thematic bridging. One particular problem of algorithmic composition is that of generating variations on a preexisting music by artificial means. This approach involves taking a given musical phrase and applying specific rules to the phrase that modify it in some way. Ralley [35] suggested a GA system that greatly reduced the number of possible melodies of interest. By defining a compromise between user-assigned fitness, which is time consuming and context dependent, and deterministic fitness assignment, that may prevent any innovation due to the tight constraints on the search space. The problem of thematic bridging using GA was studied by Horner and Goldberg [36]. Thematic bridging is the process of transforming an initial note-set into a final note-set over a specified interval of time, using a sequence of simple operations, such as note insertion, deletion and rotation.

Some examples of formalistic rule-based fitness functions include:

- the composition of Baroque music using GAs with a fitness measure that points out how well genetically created chords follow the traditional rules of the style [37];

- a system, called GeNotator, which evolves the melody, rhythm and dynamics of a musical piece using rules based upon user-defined requirements [38];
- a system proposed by Werner and Todd [39] that co-evolves composers and evaluation methods using note-transitions probability tables to determine aesthetic properties of a given composition.

Applications of user-determined fitness functions include namely the GenJam system, developed by Biles [32], and the approach proposed by Jacob [40]. The GenJam system was able to create jazz-like solo improvisations from information relating to the chord structure of a piece of music, with good results. Initially the user was required to enter a fitness evaluation for each individual from a computer keyboard, but this process was later improved by averaging the evaluations from an audience. The approach of Jacob suggests building a melody from previously constructed musical phrases, which had already been considered as possessing aesthetic value rather than using individual notes. This approach uses three main modules, the Composer, the Ear and the Arranger which respectively generates the raw music material, evaluates its aesthetic properties and defines the compositions.

Finally, applications using neural fitness functions include the work of Gibson and Byrne [41] who attempted to design a system that generates rhythmic patterns using a neural network previously trained to distinguish “good” from “bad” rhythms. Another neural fitness evaluator was proposed by Biles et al [42] to reduce the fitness bottleneck problem associated with genetic systems using user-evaluated fitness functions. This neural fitness evaluator acts as a filter to prevent individuals with low musical structure to be presented to the mentor for evaluation, speeding up the generation of fitter individuals by the evolutionary process.

A recent approach to algorithmic composition using AI methods is that of Genetic Programming. These algorithms differ from EMI, because they generate their own musical materials as well as they form their own database and grammar [23]. The composer/programmer defines a set of functions for the algorithm and defines what is desirable in the output (critic function). The algorithm will then choose the more suitable functions to get a solution that fits the composer’s pre-defined desirable output as close as possible.

Genetic programming techniques [43] allow some degree of relaxation of the constraints upon the search space imposed by genetic algorithms, by genetically generating the functions that will better solve a given problem. Different compositional approaches, adopting genetic programming techniques, have been proposed in the literature, using various types of fitness evaluation. The algorithm developed by Laine and Kuuskankare [44] aimed to generate simple single-voice melodic patterns using mathematical representations of how the melody changes with time. In this case the fitness is evaluated according to the error between the output of each individual function and some predetermined target result. Another approach was proposed by Spector and Alpern [45].

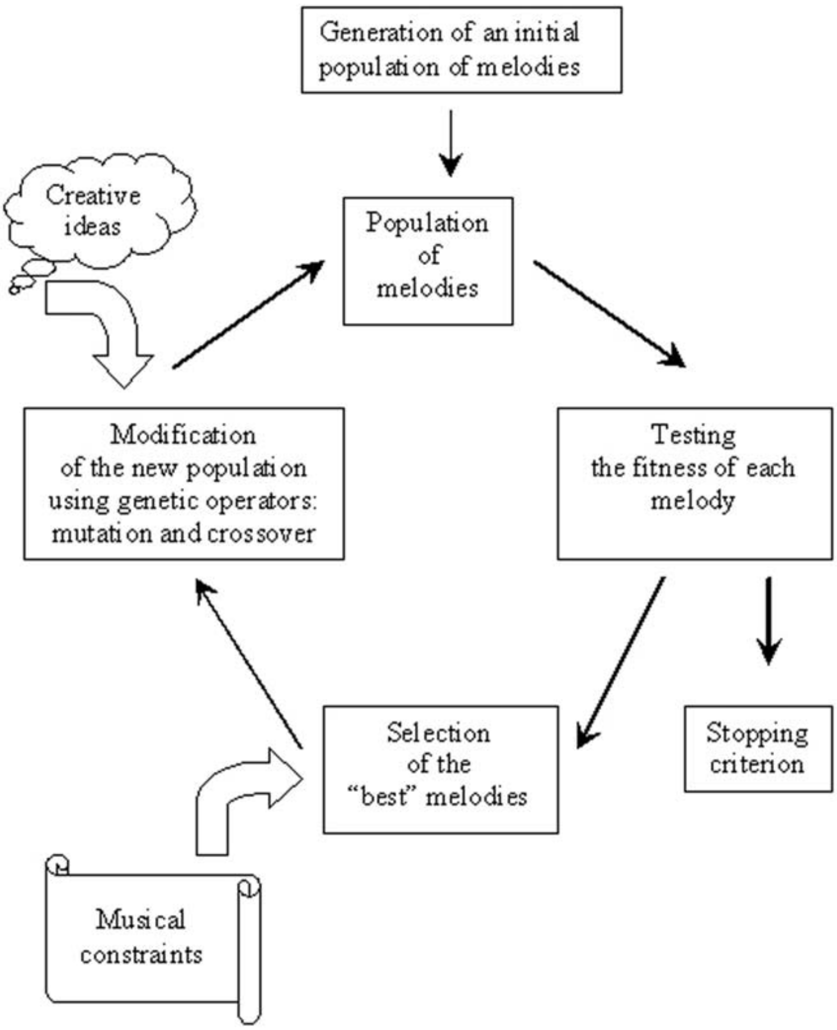

Figure 8. A genetic compositional algorithm

The initial population was defined as a number of randomly generated functions that operate on an input, which is a numerical representation of a sequence of notes. The fitness of each individual is obtained as a result of a set of comparisons (critical criteria) between tonal and rhythmic properties of the generated material and a database of stylistically similar works. This approach was later improved [32], using additional information of the music as a whole and adopting a fitness evaluation routine based upon a neural network that deduces musically relevant information in the generated musical phrases.

A final comment to point out that, since each AI method possesses different strengths, it seems logical to use a combination of AI techniques to benefit from their particular features to solve the various problems we are faced with in the development of compositional algorithms. We come then to the so called Hybrid systems [46, 47] that seem to deserve further investigation.

5. Conclusion

Musical algorithm composition is an interesting and vast subject of research. Algorithmic composition refers to a methodology of using some formal processes to compose music minimizing the human intervention.

This attempt is as old as music composition. We find it between the ancient Greeks in a rudimental form which achieves an extra level of abstraction in the Medieval period with the arising of Canonic composition. The term “algorithm” has been adopted from the fields of computer and information science by the half of the 20th century. Computers have given composers new possibilities of automating the compositional process, allowing them to work more quickly. It also simplifies the musical composition task enabling more creative people without a sound musical knowledge to bring their original ideas to this field of music creation.

Algorithm composition also raises some new and important questions still to be answered like: What about creativity? Who is responsible for the creativity, as far as it exists, the algorithm itself or the programmer?

Perhaps the success of algorithmic composition relies on achieving a close match between the composer’s creative inspiration and the implemented algorithm.

We will finish claiming that the algorithmic composition concept is no longer a mere curiosity, and it offers a great help in the understanding of the internal mechanisms of composition itself.

References

1. D. J. Grout. and C. Palisca. *A History of Western Music*. W.W. Norton & Company, 5th edition (1996).
2. B. Reitman. History of Mathematical Approaches to Western Music. http://www.mlahanas.de/Greeks/PDF/math_music_hist.pdf (2003).
3. J. F., R.F. and R.J. Wilson, Music and Mathematics: From Pythagoras to Fractals. *Oxford University Press* (2003).
4. J. Kepler, *Mysterium cosmographicum*. Tübingen (1956).
5. Wikipedia – The Free Encyclopedia. Wikipedia Foundation, http://www.wikipedia.com (2004).
6. R. Kelley, The Relationship Between Contrapuntual and Serial Composition Techniques As Seen in Works of Webern and Stravinsky. Furman University, Greenville, SC, April (1999).
7. A. Schoenberg, Composition with Twelve Tones. Style and Idea (ed. L. Stein, tr. L. Black), Faber & Faber, London (revised 1984).
8. http://sunsite.univie.ac.at/Mozart/dice/- Mozart dice game.
9. S. Richards. John Cage As … Amber Lane Press (1996).
10. P Griffiths. Modern Music and After – Directions since 1945. Oxford University Press (1995).
11. J Corbett. Extended Play – Sounding off from John Cage to Dr. Funkenstein. Duke University Press (1994).
12. J. Maurer. A brief history of Algorithm Composition Stanford University Center for Computer Research in Music and Acoustics (1999).
13. E. Bowles. Musicke´s Handmaiden: Or Technology in the Service of the Arts. Cornell University Press (1970).
14. C. Roads. The Computer Music Tutorial. The MIT Press (1996).
15. G. Assayag. Computer Assisted Composition Today.
16. I. Xenakis. Formalized Music. Thought and Mathematics in Composition. New York: Pendragon Press, (revised edition 1992).
17. D. Cope, New Directions in Music, 4th ed W.C. Brown: Dubuqe Iowa (1984).
18. S. H. Kellert In the Wake of Chaos.
19. F. R. Moore. Elements of Computer Music Prentice Hall (1998).
20. D. Cope Experiments in Musical Intelligence, A-R Editions (1996).
21. A. Horner and D. Goldberg. Machine Tongues XVI: Genetic Algorithms and Their Application to FM Matching Synthesis. *Computer Music Journal*, **17(4)**:17-29.
22. A. Barker, Greek Musical Writings, 2 vols (I: The musician and his art; II: Harmonic and acoustic theory), Cambridge (1984, 1989).
23. C. Ames and M. Domino., Cybernetic Composer: An Overview, in M. Balakan, K. Ebcioglu and O. Laske editors, Understanding Music with AI, 186-205, AAAI Press (1992).

24. D. Zicarelli, M and Jam Factory, *Computer Music Journal*, **11(4)**:13 (1987).
25. H. Burns Algorithm composition, a definition, http://music.dartmouth.edu/~wowen/hardware/algorithmdefinition.html, Florida International University.
26. J. Pressing Nonlinear Maps as Generators of Musical Pitch., *Computer Musical Journal*, **2(2)**:35-46 (1988).
27. M. Herman, Deterministic Chaos, Iterative Models, Dynamical Systems and Their Application in Algorithmic Composition. In Proceedings of the International Computer Music Conference (1993).
28. J. Harley Algorithms Adapted From Chaos Theory. In Proceedings of the International Computer Music Conference (1994).
29. D. Cope, An Expert System for Computer-assisted Composition, *Computer Music Journal*, **11(4)**:30-46 (1987).
30. P. M. Todd, A Connectionist Approach to Algorithmic Composition, *Computer Music Journal*, **13(4)**: 27-43 (1989).
31. M.C. Mozer et al., Neural Network Music Composition by Prediction: Exploring the Benefits of PsychoAcoustic Constraints and Multi-scale Processing, Connection Science, **6(2-3)**: 247-80 (1994).
32. L. Spector and A. Alpern, Induction and Recapitulation of Deep Musical Structures, Proceedings of the IJCAI-95 Workshop on Music and AI.
33. J. A. Biles, GenJam: A Genetic Algorithm for Generating Jazz Solos, Proceedings of the 1994 International Computer Music Conference, ICMA, San Francisco (1994).
34. D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Reading, Massachusetts, Addison-Wesley (1989).
35. D. Ralley, Genetic Algorithms as a Tool for Melodic Development, Proceedings of the 1995 International Computer Music Conference, ICMA, San Francisco (1995).
36. A. Horner and D.E. Goldberg, Genetic Algorithms and Computer-Assisted Music Composition, Proceedings of the 1991 International Computer Music Conference, ICMA, San Francisco (1991).
37. R.A. McIntyre, Bach in a Box: The Evolution of Four-Part Baroque Harmony Using the genetic Algorithm, Proceedings of the First IEEE Conference on Evolutionary Computation, Washington, DC (1994).
38. K. Thywissen, GeNotator: An Environment for Investigating the Application of Genetic Algorithms in Computer-Assisted Composition, , Proceedings of the 1996 International Computer Music Conference, ICMA, San Francisco (1996).
39. G.M. Werner and P.M. Todd , Too many Love Songs: Sexual Selection and the Evolution of Communication, Proceedings of the Fourth European Conference on Artificial Life, Cambridge, MA: MIT Press.

40. B.L. Jacob, Composing with Genetic Algorithms, Proceedings of the 1995 International Computer Music Conference, ICMA, San Francisco (1995).
41. P.M. Gibson and J.A. Byrne, Neurogen, Musical Composition Using Genetic Algorithms and Cooperating Neural Networks , Proceedings of the Second International Conference on Artificial Neural Networks, Stevenege, England: Institute of Electrical Engineers.
42. J.A. Biles, P.G. Anderson and L.W. Loggi Neural Network Fitness Functions for a Musical IGA, http://www.it.rit.edu/~jab/GenJamPop.html
43. J.R. Koza, Genetic Programming, Cambridge, Massachusetts: MIT Press (1992).
44. P. Laine and M Kuuskankare, Genetic Algorithms in Musical Style Oriented Generation, Proceedings of the First IEEE Conference on Evolutionary Computation, Washington, DC (1994).
45. L. Spector and A. Alpern, Criticism, Culture and the Automatic Generation of Artworks, Proceedings of the Twelfth national Conference on Artificial Intelligence, AAI-94, Cambridge (1994).
46. H. Hild, J. Feulner and W. Menzel, Harmonet: A Neural Net for Harmonizing Chorals in the Style of J.S. Bach, In R. P. Lippman, J.E. Moody and D.S. Touretzky, editors, Advances in Neural Information Processing 4 (NIPS 4), Morgan Kaufmann (1992).
47. J. Feulner, Neural Networks that Learn and Reproduce Various Styles of Harmonization, In Proceedings of the International Computer Music Conference (1993).