

Using Statistical Models and Evolutionary algorithms in Algorithmic Music Composition

Main Author: Ritesh Ajoodha
South Africa

Co-author: Marija Jakovljevic
South Africa

Key-Words: Genetic Algorithm, Statistical Model, Probability of occurrence (POC), Context-free Grammar, Gaussian distribution, Genetic Phase, Statistical Phase, Music Representation, Fitness Function, Note counting.

INTRODUCTION

Algorithms can be used to implement a set of instructions or rules. The applications of algorithms are extensive, however, there are some areas where they can be constrained. In composition theory, there are many choices that are drawn from the artist's creativity. These choices are partly influenced by the rules of composition theory and partly influenced by the artist's personal perception or inspirations. It is for this reason that a computer would not be able to satisfy every artist's perception into one musical piece. The algorithmic components, however, can be achieved by satisfying the components of composition theory to develop a piece of music.

Composition theory is a branch from music theory which deals with the architecture and construction of music. This is achieved by a mixture of instruction from composition rules and the composer's continuous personal percept. The composer is commonly a human-agent in an environment which solely provides rules and agent intelligence. The rules in this environment are those directly given by composition theory, these rules include rules of harmony, structure, style, articulation, dynamics and sequence.

If one is to compose music artificially, one would need to reproduce these rules from composition theory as well as the emotional intelligence that the human-agent provides. Unfortunately, the human-intelligence that produces emotion cannot be achieved in a computer-agent (Johann Beukes. pages 58-59, 2011) which makes the latter impossible. Consequently, this leads us to believe that a computer-agent might be able to mimic the procedural aspects of composition theory but will fail when it is to include the emotional aspects. Let it be clear that this research only poses procedural traits to develop music and does not attempt to algorithmically mimic emotional traits which are found in human-agents.

If one is to ignore emotional trait constraints, then it is easy to develop music-strings through randomization. A way to achieve this is through a statistical model: A statistical model is a structure that produces a sample based on a probability of occurrence in each instance of production. The probability of occurrence is attached to each production in a context-free grammar (CFG). By doing this, any string can be derived from the CFG with a user defined occurrence. Obviously, this CFG will be able to produce a very extravagant music selection which is why it has become necessary to channel a workable sample from a selection through human intervention.

Now that a sample has been established, either through randomization or human intervention, the sample can proceed to be refined through genetic algorithms as per the title of this paper. A genetic algorithm is a search heuristic that mimic the process of natural evolution. The heuristic is routinely used to generate useful solutions to optimization and search problems (Melanie 1996). This research attempts to use genetic algorithms to refine the statistical sample. The genetic algorithm will consist of various phases which the statistical sample will be passed through. At the end of the genetic phase the statistical sample should be refined and can be finalized.

BACKGROUND

The Statistical Model

Conklin (2004) reviewed the process of music generation and equated it with the problem of sampling from a statistical model. One can represent a piece of music as a chain of events, which consist of music objects (notes, rests etc.) together with a duration and an onset time after instantiation into a piece. A statistical model of music assigns to every possible piece of music a probability and captures regularities in a class of music, be this a genre, a style, a composer's style, or otherwise. Conklin (2004) pointed out that, surprisingly, only a few of the proposed sampling methods have been explored in the music generation literature. He suggests that using statistical modelling in music composition can be beneficial. The main limitation of Conklin (2004) is that of Viterbi decoding, as its computation time increases exponentially with the context length of the underlying Markov model on states. Producing the highest probability pieces from complex statistical models is therefore a computationally expensive task (Conklin 2004), and heuristic search and control strategies must be applied.

The Genetic Algorithm

Matic (2009) reviewed that composing, as well as any other artistic activity, includes free choice by which a composer express their feelings, moods, intentions or inspiration. He maintains that these choices are seen as a series of instructions that can be relatively easy to interpret. Most composers apply certain rules and instructions when composition and thus any composing process in some way can be considered as an algorithm. On the other hand, the absence of human factors in the automatic composition will lead to the appearance of large amounts of objectively bad and useless music as a result of bad computational selected choices. The combination of genetic operators such as mutations, selections and crossovers in some way simulates the innovative process (as real composing is), enabling continuous "improvement" of their obtained results. The main limitation of Matic (2009) is that the user will not be able to specify a genre based on user perception as the class of the best individuals will be very similar every time. The following evaluation function was used by Matic (2009):

Equation 1 Total Fitness by Matic (2009)

$$Total\ fitness = f + g = \alpha \sum_{i=1}^m \vartheta_i (\mu_i - a_i) + \beta \sum_{i=1}^m \rho_i (\sigma_i^2 - b_i^2) + \gamma \frac{1}{bl}$$

Where $\sum_{i=1}^m \vartheta_i (\mu_i - a_i)$ and $\sum_{i=1}^m \rho_i (\sigma_i^2 - b_i^2)$ represent the similarity of the two individuals and $\frac{1}{bl}$ represent the collection of good tones. α , β and γ served as global weight factors. The experimental results that were produce by the algorithm meet some objective criteria of "beautiful" compositions: According to Matic (2009), they contain intervals that are pleasant to the human ear, the rhythm is meaningful, and, with a slight adjustment to the appropriate arrangement, the compositions sound unusual, but pleasant. George and Wiggins [1998] proposed a genetic algorithm which was used similarly in Matic (2009) like the use of evolutionary algorithms. Most of which include the work on modified genetic operations that can be traced back to Horowitz (1994); Burton (1996); Brown (2002) and Moroni et al. (2000).

George and Wiggins [1998] reviewed that genetic algorithms have proven to be very efficient search methods, especially when dealing with problems that have 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 choice for a search engine in a musical application. George and Wiggins [1998] new GA exhibits the following three significant characteristics which are uncommon in GA applications to music: An algorithmic fitness function for objective evaluation of the GA results; problem-dependent genetic operators; and symbolic representation of the structures and the data which helped them solve their problems. George and Wiggins (1998) state subjectively that their system often generated interesting music patterns. The main limitation of George and Wiggins (1998) is that the music pieces produced had no music structure as a result there was no room for time signatures which are important in music analysis. Their research was also biased to the study of jazz music. A very similar study was done by Biles et al. (1996).

Alfonseca, Cebrian, & Ortega (2007) propose the use of interval distance as a fitness function which may be used by genetic algorithms to automatically generate music in a given pre-defined style. Alfonseca et al. (2007) focused on continuing the results of their previous work which helped them increase the efficiency of the procedures described by their previous paper on information theory. The main limitation of Alfonseca et al. (2007) is that their genetic algorithm still needed to be fine-tuned for the proposed application. Moreover, although the authors introduced the information about note

duration in the genetic process, it had been ignored so far. Alfonseca et al. (2007) still need to experiment with different strategies. Some of the pieces of music thus generated recall the style of well-known authors, in spite of the fact that the fitness function only takes into account the relative pitch envelope. Qualitative response by human audiences confirms that the results described in this paper are superior to those obtained previously with a different fitness function. Their use of the interval distance informed this research paper.

MAIN FOCUS OF THE ARTICLE

Problem

There has been several attempts to conduct algorithmic music composition by using techniques which resemble statistical models (Conklin, 2004) and genetic algorithms (Matic, 2009) (George & Wiggins, 1998) (Alfonseca, Cebrian, & Ortega, 2007). However, there has been no attempt to integrate these techniques into a suitable algorithm for future progress in music evolution and architecture. According to Conklin (2004), the simplest way to generate music from a history-based model is to sample, at each stage, a random event from the distribution of events at that stage. After an event is sampled, it is added to the piece, and the process continues until a specified piece duration is exceeded. Although this is a quick and easy way to create music, it is doubtful that the music produced by the Conklin (2004) random model would be meaningful as music-notes are just being randomly attached together. When creating a statistical model it is often convenient to view all components of the sample at once, especially when dealing with large samples, this can be achieved by using a context-free grammar (CFG). This brings us to the first problem statement (PS1):

PS1: Very few algorithms, in Composition theory, make use of Context free grammars for convenience in their statistical music production.

The music produced by a statistical model again lacks distinctive melody and meaningful structure. It is seen in Matic (2009); George & Wiggins, (1998); Alfonseca, Cebrian, & Ortega (2007) that even though the music produced by their research was developed using a genetic model, the music produced still failed to have a distinctive structure and a meaningful melody. Which brings us to the second and third problem statements (PS2 and PS3):

PS2: While there are many algorithms that create music using algorithmic composition, very few algorithms have been presented which produces tasteful structure and stylistic sense in its melody production.

PS3: Very few algorithms has been presented which make use of a hybrid of statistical and evolutionary methods.

Addressing these problems foregrounds music as a platform that inspires creative thinking thus motivating success.

Research Methodology

The research methodology is underpinned by a research model that presents five phases: The Statistical, Genetic, Variation, Structure, and Credibility phases. This paper only focuses on two of the five phases of algorithmic music production (Ajoodha, 2013): The Statistical phase, where a context-free grammar and statistical model is used to generate user-defined music-strings; and The Genetic phase, where a genetic algorithm is used to refine the statistical sample. The other phases included: the Variation phase, where a selection of melodies is viewed from the sample and an algorithm is applied to deduce musical variation from the output of the genetic phase; the Structure phase, where structure is automatically assigned to each melody with respect to its variation; and the Credibility phase, where the music is tested for credibility with respect to cultural acceptance. Figure 1 below shows the complete music development process through these phases, however, only the first two of the five phases will be explained in this paper.



Figure 1 The Phases of Music Production (Ajoodha, 2013)

A CFG has been identified which needs to be created along with a statistical model to generate user defined music-strings. A simple CFG is proposed to suit the representation of each note played, however, firstly the music representation that will be used throughout this paper is discussed.

Music Representation

The standard music representation will be used by focusing on music notes; octave; duration; and tempo representation. JFugue version 4.0.3 (Koell, 2013) and Sibelius version 6 were used for the creation of audio files and music manuscript representation respectively. The note structure will follow the standard manuscript note representation and accidentals will be used where necessary. Figure 2 below shows a list of note names ranging from middle C to C an octave higher by using a chromatic scale. The row of letters in Figure 2 represent the note name of the notes above each respective letter. A similar notation will be used throughout this paper.



Figure 2 Note Representation

Each note duration will follow the conventional manuscript notation. The note durations in this paper will be limited to the durations given by Table 1. Each row in Table 1 gives a duration representation that can apply to every note in Figure 2. Furthermore, every note duration can be defined by the note representation, where a note represents either a sound or a rest (i.e. where no sound is played). The rest representation is given by last column of Table 1.

Table 1 Note Duration Representation

Note Name	Note representation	Rest Representation
Whole Note		
Half Note		
Quarter Note		
Eighth Note		
Sixteenth Note		

Tempo is defined as the number of beats per minute. To define the tempo of a melody the tempo at the beginning of that melody is defined. The tempo can range from 0 to infinity, however only a selected range of tempos will be used in this paper. Figure 2 below shows the different tempo ranges and their respective classifications. Notice that different tempo classifications can overlap.

Table 2 Tempo Ranges and Classifications

No	Tempo Range	Classification
1	40 - 52	Largo
2	48 - 66	Adagio
3	60 - 88	Andante
4	84 - 108	Moderato
5	104 - 120	Allegret
6	120 - 144	Allegro
7	138 - 186	Vivace
8	160 - 200	Presto
9	200 - 208	Prestissimo

The Statistical Phase

In this phase a context-free grammar (CFG) is defined that is able to derive any of the 768 music notes (12 notes x 8 possible durations x 8 octaves) available by the representation. This CFG achieves this by a simple leftmost derivation. *The CFG is defined as follows:*

Figure 3 The Context-free Grammar

Let $G = (V, \Sigma, R, \hat{A})$ be a CFG where

$$V = \{\mathcal{L}, \hat{I}, \hat{A}, \hat{Y}, \hat{O}, \hat{E}, \hat{U}\}$$

$$\Sigma = \Sigma(1) \cup \Sigma(2) \cup \Sigma(3) \cup \Sigma(4) \cup \Sigma(5) \text{ where}$$

$$\Sigma(1) = \{A, B, C, D, E, F, G\}$$

$$\Sigma(2) = \{w, h, q, i, s, t, x, n\}$$

$$\Sigma(3) = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

$$\Sigma(4) = \{\#, b\}$$

$$\Sigma(5) = \{T, [,], 1, 2, 3, 4, 5, 6, \dots, u\} \text{ where } u \text{ is an upper-bound as tempo has none.}$$

Let G have the following productions:

Figure 4 The Productions of CFG G

$$\mathcal{L} \rightarrow I[\hat{I}]\lambda T[\tilde{N}]\lambda \hat{A}$$

$$\tilde{N} \rightarrow 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | \dots | u$$

where u is an upper-bound as tempo has none.

$$\hat{I} \rightarrow \text{Piano}$$

$$\hat{A} \rightarrow A\hat{O}\hat{Y} | B\hat{O}\hat{Y} | C\hat{O}\hat{Y} | D\hat{O}\hat{Y} | E\hat{O}\hat{Y} | F\hat{O}\hat{Y} | G\hat{O}\hat{Y} | \lambda$$

$$\hat{O} \rightarrow \#\hat{E} | b\hat{E} | \lambda \hat{E}$$

$$\hat{E} \rightarrow 1\hat{U} | 2\hat{U} | 3\hat{U} | 4\hat{U} | 5\hat{U} | 6\hat{U} | 7\hat{U} | 8\hat{U} | 9\hat{U} | 10\hat{U}$$

$$\hat{U} \rightarrow w | h | q | i | s | t | x | n$$

$$\hat{Y} \rightarrow \lambda \hat{A} | \lambda$$

From this CFG any derivation of the 768 notes can be produced at any time by a simple *leftmost derivation* starting from production £. A problem lies in the arrangement of these note values upon a specified tempo (i.e. the selection of u). So it has become necessary to intervene in the initial population selection as the sample can be infinitely large – infinite because of the infinite possible tempos. Using a statistical model for this intervention, by adding a probability of occurrence (POC) to each production in the CFG, a statistically channelled initial population is achieved for the evolutionary algorithm to act on in the Genetic phase. This can be done in four steps: Firstly, the note selection POC is specified, since there are only 12 different notes in every octave and 1 rest (possibility of no note occurring), each note's POC is set to 0.07. Secondly, an octave range is specified: by using the Gaussian function given by Equation 2. The Gaussian distribution is defined around a specific octave, tempo, and duration, with the mean given as μ and standard deviation given as σ^2 .

Equation 2 Gaussian distribution Function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Since most music is dominantly based around middle C, it would be fitting to take middle C as the mean with a selected standard deviation. Figure 5 below shows Gaussian distribution around the 5th octave by setting $\mu = 5$ and $\sigma^2 = 1$. The standard deviation is chosen as directly proportional to the increase in the probability of occurrence of the octave selection.

Normal Distribution for Octaves

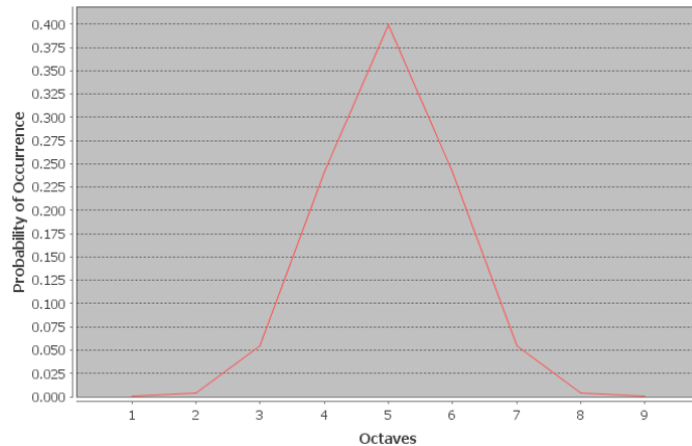


Figure 5 Gaussian distribution for Octaves with $\mu = 5, \sigma^2 = 1$

Thirdly, the duration range is specified. Again Gaussian distribution can be used to do this, setting $\mu = 2$ and $\sigma^2 = 1.5$, the distribution depicted by Figure 6 is achieved. These parameters were chosen such that the half notes get a better chance of occurrence as it is very common for music to dominantly have this trait. However, these parameters are subject to change.

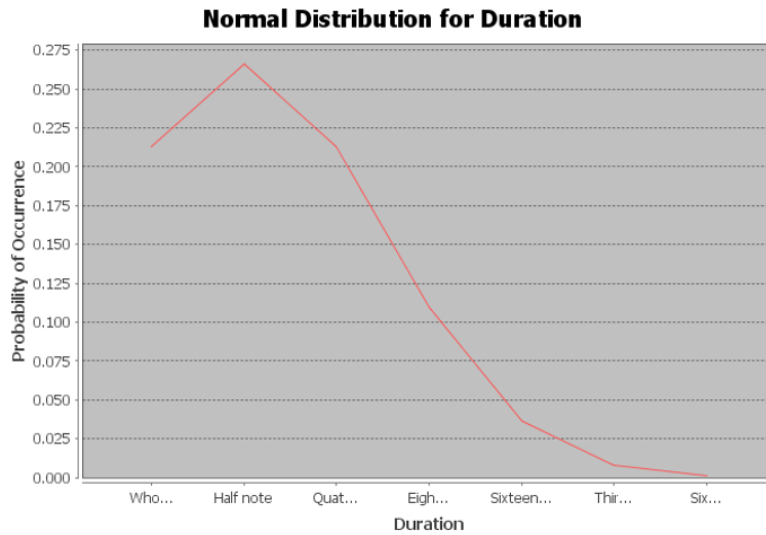


Figure 6 Gaussian distribution for Duration with $\mu = 2$, $\sigma^2 = 1.5$

Finally, a specified Tempo range is established. This again done using Gaussian distribution, only this time the median of any specified tempo and the standard deviation is chosen as directly proportional to the inclusion of more tempo ranges (i.e. the larger the tempo chosen, the more tempo classifications will be included). Figure 7 below shows a Gaussian distribution where $\mu = 133$ and $\sigma^2 = 4$ for tempo.

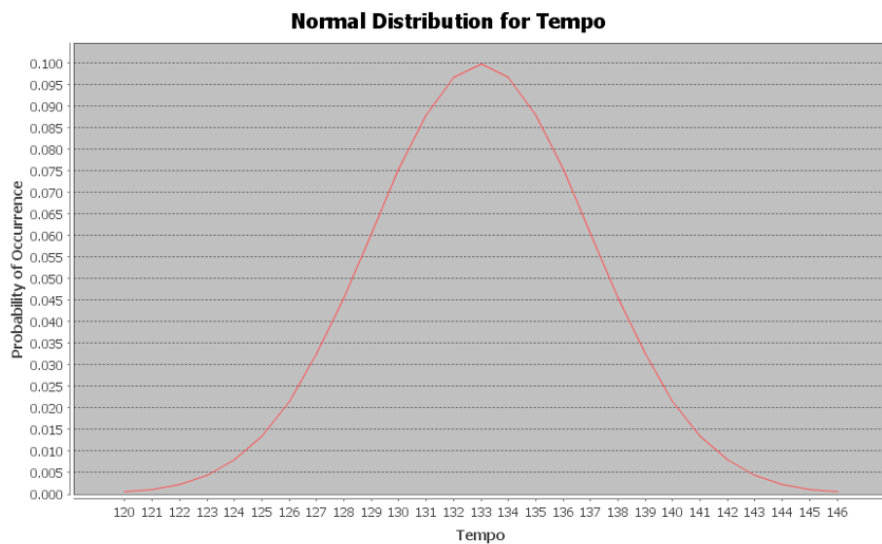


Figure 7 Gaussian distribution for Duration with $\mu = 133$, $\sigma^2 = 4$

The Genetic Phase

This phase presents a genetic algorithm that refines the sample created in the statistical phase. Figure 10 outlines the refinement process of the genetic algorithm performed in the Genetic phase.

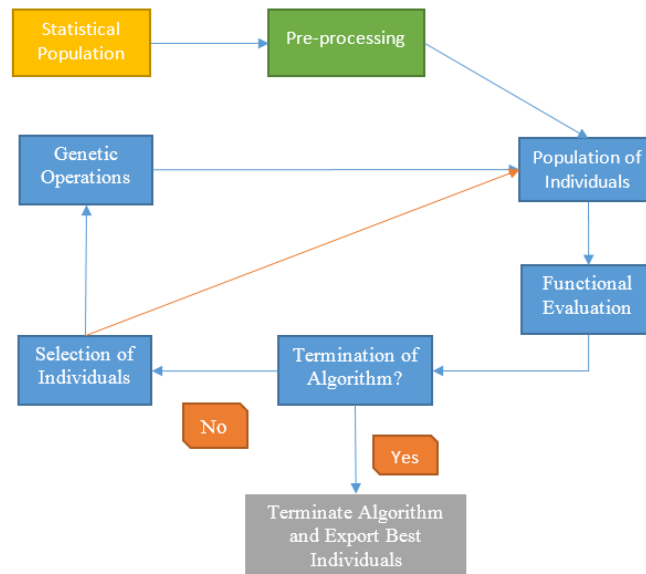


Figure 8 States of the Genetic Algorithm

In Figure 10 the statistical population is firstly pre-processed. The pre-processing stage is essential to setup the evaluation function. As part of the pre-processing step a key signature is established and the initialization of genetic operational parameters are setup, this includes the inversion and translocation thresholds. The Initialized population is then passed to the genetic algorithm where the functional evaluation takes place. In the *functional evaluation state* the music-strings are scanned and assigned an evaluation. The evaluation is derived from the evaluation function. Thereafter, the algorithm is prompted to *terminate*. The Algorithm will *terminate* if and only if the iteration counter for the genetic algorithm exceeds the specified generation threshold. Upon termination the individuals are released as best individuals. If the algorithm fails to terminate then there is a selection of music-strings. Music-strings are selected based on specified selection parameters. The selection categorises the individuals as either candidates for mutation or cloning. Only mutation candidates are allowed to pass through to the *genetic operation state*, whereas clone candidates are immediately passed through to the *Population of individuals state*. In the *genetic operations state* a selection of mutations occur and the mutated individuals as passed through to the *Population of individuals state*. The cycle continues until the algorithm terminates (i.e. iteration counter > generation threshold number).

Pre-processing

Before the genetic algorithm is initiated it is important to establish some constraints and rules. An example of such a rule is a key signature for the populations to strive towards. A key signature can be selected in various ways, however, for this sake of this paper the key signature will be chosen optimally by a NoteCount() procedure. A scale counter for a corresponding scale will count the occurrences of each note in the sample that belong to the corresponding scale. The scale counter with the largest value will be chosen and the corresponding scale will be used. Other constants that should be established in this step include the constants for the influence of each instance of the evaluation function; the number of generations; the mutation and clone decision constant; the mutation translocation threshold; and the mutation inversion threshold.

Functional Evaluation

In each iteration of the genetic algorithm (GA) an objective function is established that the music-strings can be evaluated by and a decision can be immediately reached to whether or not the music string, or part of it, will be accepted in the next generation of the GA. Recall that the algorithm initially scans through the sample with the NoteCount() procedure, the output of this NoteCount() will be an appropriate key signature which will be embedded to the evaluation criteria. Each individual's note is evaluated in order to determine if the consecutive notes are abiding the established key. Let the evaluation function for determining a key signature be defined as:

Equation 3 The Key Signature Fitness Function

$$f_1 := \theta \sum_{i=1}^n \frac{1}{\alpha^2}$$

Where θ is the influence of the evaluation, n is the number of notes, and α is a value that describes whether or not the note is accepted by the established key signature. α can be one of two values. If the note is in the established key signature then α assigned a very high value as music in key is highly encouraged. Similarly if a note is not in the established key then α assign a much lower value. See Table 3 for proposed weights for α .

Table 3 Key Signature Evaluation Weights

Evaluation table for the Key Signature	
Decision In key	Weight Proposal
Yes	0.8
No	0.2

A crucial part of determining the quality of music is examining what each note interval is. That is determining what relationship the note played had with the previous note played as the relationship of these two notes greatly affect the quality of the entire piece. It for this reason that every note played is evaluated by its previous note's in its respective key. The interval is the difference between two pitches. To get the interval of a pitch with respect to another simply compute:

$$\text{Interval Dist} = |\text{Position of pitch A} - \text{Position of pitch B}|$$

One way of examining the intervals is to make sure all the intervals are accepted by the key signature. The following evaluation is proposed:

Equation 4 In Key Interval Fitness Function

$$f_2 = \vartheta \sum_{i=1}^m \beta$$

Where ϑ represents the influence of the evaluation, β represents the interval cost and m is the number of intervals in the individual. The following weights in Table 4 have been proposed for β .

Table 4 In Key Interval Evaluation Weights

Evaluation table for the interval in an <u>Established</u> Key		
Interval:	Description	Weight Proposal
1	Perfect unison	0.8
2a	Minor 2 nd	0.4
2b	Major 2 nd	0.4
3a	Minor 3 rd	0.6
3b	Major 3 rd	0.6
4	Perfect 4 th	0.8
5a	Diminished 5 th	0.6
5b	Perfect 5 th	0.8
6a	Minor 6 th	0.5
6b	Major 6 th	0.5
7a	Minor 7 th	0.4
7b	Major 7 th	0.4
8>	interval greater than 8	0.1

Further work need to be done to make sure that the current interval is accepted by the previous note's respective key, however this is not a crucial factor when determining quality so the influence of this will

be relatively low. The Following evaluation function is proposed with proposal weights given by Table 5:

Equation 5 Relative Key Fitness Function

$$f_2 = \rho \sum_{i=1}^m \gamma^2$$

Table 5 Relative Key Interval Evaluation Weights

Evaluation table for the interval in a Referenced Key: γ		
Interval:	Description	Proposed Weight
1	Perfect unison	0.8
2a	Minor 2 nd	0.4
2b	Major 2 nd	0.4
3a	Minor 3 rd	0.6
3b	Major 3 rd	0.6
4	Perfect 4 th	0.8
5a	Diminished 5 th	0.6
5b	Perfect 5 th	0.8
6a	Minor 6 th	0.5
6b	Major 6 th	0.5
7a	Minor 7 th	0.4
7b	Major 7 th	0.4

Finally, the entire evaluation of the individual is considered and its mean, μ , and standard deviation, σ^2 , are computed. The standard deviation (in this context) is inversely proportional to the quality of the piece. These values are added to the evaluation as well. Therefore, the total fitness for an individual is given by the following equation:

Equation 6 Total Fitness Function

$$Tf := \theta \sum_{i=1}^n \frac{1}{\alpha^2} + \vartheta \sum_{i=1}^m \beta + \rho \sum_{i=1}^m \gamma^2 + \mu \frac{1}{\sigma^2}$$

Where θ is the influence factor of the key signature; n is the number of notes in the individual; α is the evaluation of the key signature with respect to the individual; ϑ is the influence factor of the interval with respect to the key signature; β is the weight of the interval with respect to the key signature and the individual; m is the number of intervals; ρ is the influence factor of the interval with respect to the previous note's key signature; γ is the weight of the interval with respect to the previous note's key signature and the individual; μ is the mean of the individual's evaluation; and σ^2 is the standard deviation with respect to the mean. It is from these individual evaluations that the quality of every individual is calculated. The evaluation is directly proportional to the music-string's quality.

Selection of Individuals

After the evaluation of a generation by the evaluation function the max and min evaluations in the generation are extracted and used to create a new evaluation relative to the min and max evaluations. The following equation is used to calculate the relative evaluation:

Equation 7 Calculating Relative Evaluation

$$Relative\ Evaluation(i) = \frac{Evaluation(i) - Min}{Max - Min}$$

The relative evaluation is measured as a percentage with respect to all the other individuals. Based on the *mutation and clone decision constant*, chosen in the pre-processing step, the generation is split into two different categories: clone candidates or mutation candidates. Figure 9 shows the clone and

mutation decision constant set as 0.8. The cloned individuals will be automatically added to the next generation whereas the mutation candidates will await the mutation process that occurs in the genetic operation state.



Figure 9 mutation and clone decision constant

Genetic Operations

Recall that when an individual is cloned the individual is taken unchanged and posted into the next generation of the genetic algorithm. On the other hand, when an individual is mutated, the individual's content is changed and then put into the next generation. There are five different types of mutations that could randomly occur in a mutation candidate: An inversion; deletion; duplication; correction; or a translocation.

An inversion occurs when a segment of the music-string is reversed end- to -end given a specified threshold. The threshold serves as an upper bound for the inversion segment length. The inversion threshold is specified in the pre-processing step. Figure 10 shows an example of a music string being inverted. Given an inversion threshold, say 5, an inversion length of 4 is randomly selected and the segment EFGA is selected from the complete string CDEFGABC. In the rightmost image the sub-string is inverted and placed back into the same position. The goal of this operation is to rectify mistakes by structurally rearranging interval groups which could render the individual better for survival.

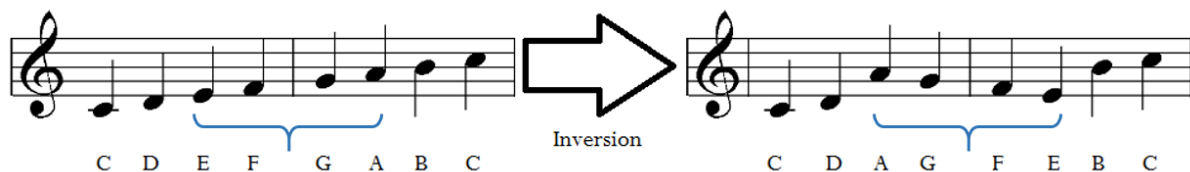


Figure 10 Inversion Operation of a Music-String

A deletion occurs when a random note is removed from a music-string. Figure 11 shows an example of a deletion. The music-string CDEFGABC is mutated by the complete removal of a random note in the music-string. The resulting string is CDEFGABC. The goal of this operation is twofold. Firstly this operation can remove an unwanted note from a music piece, and secondly the removal of the note could give rise to a better sounding interval that could exist between the previous and next notes of the note removed.

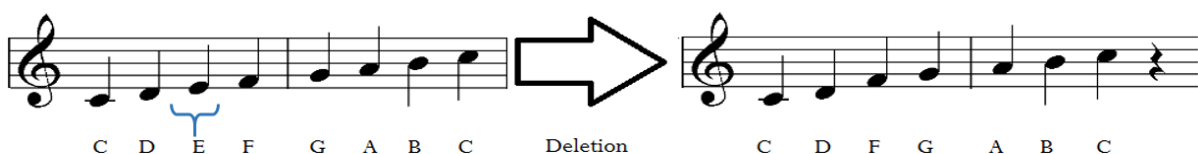


Figure 11 Deletion Operation of a Music-String

A duplication occurs when a random note in the music-string is repeated again consecutively. Figure 12 shows an example of a duplication. The String CDEFGABC to mutate to CDEFFGABC. The goal of this operation is to periodize the music-string. This gives the piece a rest from interval leaps and opportunely gives the listener a more wholesome sound.

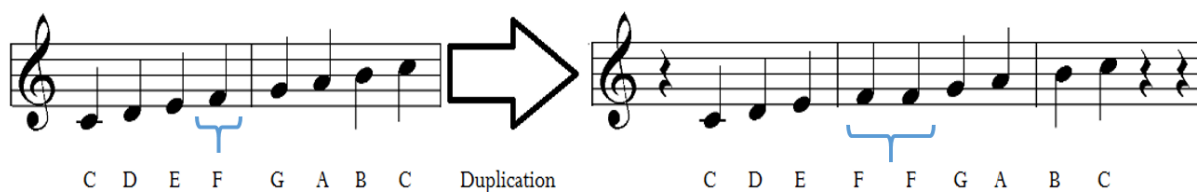


Figure 12 Duplication Operation of a Music-String

A translocation is a rearrangement of parts between two music-strings given a translocation threshold. The threshold serves as an upper bound for the translocation segment length. The translocation threshold is specified in the pre-processing step. Figure 13 shows an example of a translocation being performed by Piece A and Piece B. Given translocation threshold, say 5, a translocation length of 3 is randomly selected and the segment ABC is selected from the complete string CDEFGABC. In the rightmost image the segment ABC is translocated to Piece B. In this case the operation attaches the segment ABC to the end of piece B. The goal of this operation is to salvage a segment of good music and transplant it to another mutation candidate which will be reevaluated and perhaps could evolve into a better individual.

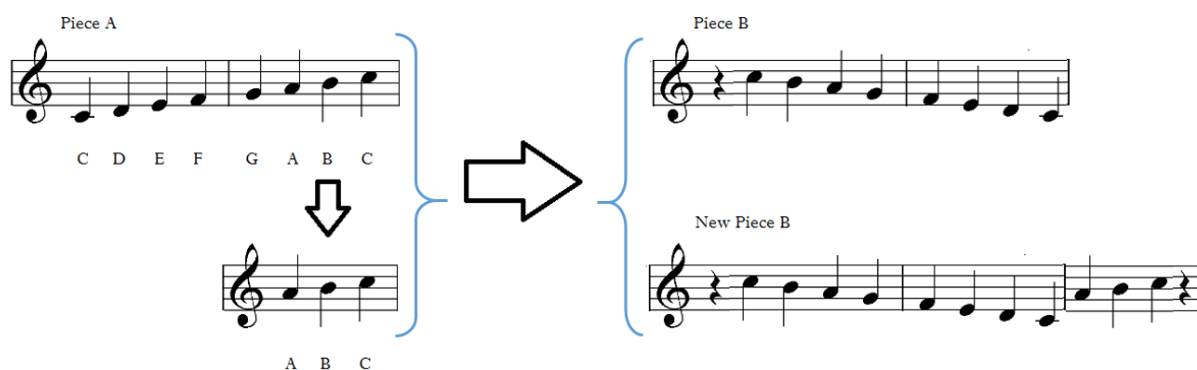


Figure 13 Translocation Operation of a Music-String from Piece A to Piece B

Phase Results and Recommendations

Statistical Phase Experimental Results

Using the parameters given above; a specified length of notes of each piece taken as 25; and the number of individuals per population taken as 100, we achieve 100 statistically modelled pieces. Only two examples are provided in this paper: Individual 1 and 87 given by Figure 14 and 15 respectively.

Individual 1

Alahya

Allegro ♩ = 126

Figure 14 Individual 1 from the Statistical Population

Individual 87

Alahya

Vivace ♩ = 144

Figure 15 Individual 87 from the Statistical Population

Although some segments of these examples make musical sense namely the first few bars of Individual 1 and the last few bars of Individual 87, both pieces have a lot to be desired. This however is a good place to start and can serve as seeds for the genetic algorithm.

Genetic Phase Experimental Results

The following experimental results were obtained by using the following static parameters on the statistical sample: 500 generations; a mutation and clone decision constant of 0.8; a mutation translocation threshold of 4; and a mutation inversion threshold of 11. The influence factors: θ , ϑ and ρ took the values 8, 5 and 1 respectively and the proposed weight factors: α , β and γ took the values of Tables 3, 4 and 5 respectively. After the NoteCount() procedure the key B flat major was established and embedded to the evaluation function.

Figure 16 shows the 5th individual of generation 500. It is clear that this piece contains reasonable interval leaps and a consistent voice range. There are only two large leaps in the passage, those being the first and last intervals. The melody is distinctive and reasonable and there seems to be a sense of key except in bar two, however, bar two's key conflict is resolved in bar 3. The melody is quite short (12 notes) which suggests that it must have undergone several translocations and or deletions from a length of 25 to 13. The piece however shows character and style.

Individual 5 Generation 500

Alahya

Allegro ♩ = 131

Figure 16 Best Individual 5 Generation 500

Figure 17 shows the 99th individual of generation 500. It piece contains leaps no greater than 2 octaves. There is a chromatic build up in bar 2 and 3 which was achieved as a result of the relative key interval criteria. The voice range is much broader and has 3 notes out of key. There are only two large leaps in the passage, those being the first and last intervals. The melody is distinctive and reasonable. The melody is also quite short (17 notes) which suggests that it must have undergone several translocations and/or deletions from a length of 25 to 19. The piece however shows character and style.

Individual 99 Generation 500

Alahya

Allegro ♩ = 130

Figure 17 Best Individual 99 Generation 500

Figure 18 shows the 84th individual of generation 500. It is clear that this piece contains reasonable interval leaps and a consistent voice range. The melody is played relatively quickly (Vivace) and shows clear patterns and stylistic sense. The melody is distinctive and reasonable and there is a sense of key throughout the piece. The melody is relatively longer than the other pieces and demonstrates character and clarity.

Individual 84 Generation 500

Alahya

Vivace ♩ = 145

Figure 18 Best Individual 84 Generation 500

Future Research Directions

Future Research Directions can attempt to improve current processes in automated music production by emphasize on improving methods in the statistical; genetic; variation; and structure phases. Additional research can also attempt to explain why it is important to include a living sample in automated music composition, by using a credibility phase, as producing music samples which are culturally rejected by the current society are meaningless.

The algorithm presented can be changed based of the user defined commands with respect to the parameters. If the user wanted to produce a statistical population that would encourage the production of Jazz music than he/she would simple define the statistical phase with jazz elements, for example, decreasing the probability of a fifth occurring and decrease the tempo to adagio or largo classifications. Consequently, when the NoteCount() is performed a jazz scale will be selected and embedded to the evaluation criteria.

Conclusion and Discussion

An evolutionary approach to algorithmic music composition is presented in this paper through statistical and genetic models. Some melodies produced by the algorithm displayed significant interval changes and stylistic sense. The best individuals showed character and obeyed aspects of composition theory to certain extents, for example staying in key and avoiding incongrual intervals.

The results can be further refined by optimizing the weight and influence factors: α , β , γ , θ , ϑ and ρ . By doing this the genetic algorithm will be able to produce an optimal set of best individuals for a desired application. The algorithm has been designed to run in $O(mn)$ time and space, where n is the length of the individual and m is the number of generations. Increasing the number of generations and the length of the individuals consequently, increases the time the algorithm will take. However, the greater the length of the individual and the number of generations it undergoes, the better the quality of the output sample.

The research can be extended in several ways. Since the static parameters can be optimised so that the algorithm outputs a set of optimally best individuals, it would be interesting to see whether it is

possible to classify these parameters into parametric ranges that code for different genres. The best individuals of this algorithm can also be varied and given structure where necessary.

The five phased model strengthens the algorithmic music composition methods, particularly in the genetic and statistic phases where the key melody is defined. The potential for this research can be applied to a wider context of research in education and human-agent composition.

References

- Ajoodha, R. (2013). Algorithmic Composition: Using Statistical Models and Evolutionary Algorithms in Music Evolution and Structure. (pp. 20 - 21). Johannesburg: Unpublished manuscript.
- Alfonseca, M., Cebrian, M., & Ortega, A. (2007). A simple genetic algorithm for music generation by means of algorithmic information theory. In *PROCEEDINGS OF STEP 98* (pp. 3035-3042). Congress of Evolutionary Computation.
- Biles, J., Anderson, P., & Loggi, L. (1996). *Neural Network fitness functions for musical iga*. Rochester Institute of Technology.
- Brown, A. R. (2002). Opportunities for evolutionary music composition. *ACMA* (pp. 27-34). Melbourne: Australasian Computer Music Conference.
- Burton, A. R. (1996). A Hybrid neuro-genetic pattern evolution system applied to music composition. PHD thesis, University of Surry, School of Electronic Engineering.
- Conklin, D. (2004). Music generation from statistical models. In *Proceedings of the Second international conference on Computer Music Modeling and Retrieval* (pp. 295 - 312). Berlin, Heidelberg: Springer-Verlag.
- George, P., & Wiggins, G. (1998). A genetic algorithm for the generation of jazz melodies. In *Proceedings of Step 98*, (pp. 7 - 9).
- Horowitz, D. (1994). Generating rhythms with genetic algorithms. *Proceedings of the 1994 International Computer Music Conference*. San Francisc: ICMA.
- Johann Beukes. page s 58-59, 2. a. (2011, July 12). *Is Artificial Intelligence Truly Possible?* Retrieved from <http://johannbeukes.com/is-artificial-intelligence-truly-possible/2011/10/25/>
- Koell, D. (2013, July 14). *jFugue Users Guide*. Retrieved from JFugue Users Guide: <http://www.jfugue.org>
- Matic, D. (2009). A genetic algorithm for composition music ., (pp. 157 - 177: 295 295 - 312).
- Melanie, M. (1996). *An Introduction to Genetic Algorithms*. A Bradford Book The MIST Press, first mit press paperback edition.

ADDITIONAL READINGS

- Balkema, W., & van der Heijden, F. (2010). Music playlist generation by assimilating GMMs into SOMs. *Pattern Recogn. Lett.*, 31(11), 1396–1402. doi:10.1016/j.patrec.2010.02.001

- Blostein, D., & Haken, L. (1999). Using Diagram Generation Software to Improve Diagram Recognition: A Case Study of Music Notation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(11), 1121–1136. doi:10.1109/34.809106
- Bottoni, P., Labella, A., Faralli, S., Pierro, M., & Scozzafava, C. (2006). Interactive composition, performance and music generation through iterative structures. In *Proceedings of the 14th annual ACM international conference on Multimedia* (pp. 189–192). New York, NY, USA: ACM. doi:10.1145/1180639.1180693
- Feng, J., Ni, B., & Yan, S. (2010). Auto-generation of professional background music for home-made videos. In *Proceedings of the Second International Conference on Internet Multimedia Computing and Service* (pp. 15–18). New York, NY, USA: ACM. doi:10.1145/1937728.1937732
- HUA, X.-S., LU, L., & ZHANG, H.-J. (2004). Automatic music video generation based on temporal pattern analysis. In *Proceedings of the 12th annual ACM international conference on Multimedia* (pp. 472–475). New York, NY, USA: ACM. doi:10.1145/1027527.1027641
- Ishizuka, K., & Onisawa, T. (2008). Generation of variations on theme music based on impressions of story scenes considering human's feeling of music and stories. *Int. J. Comput. Games Technol.*, 2008, 3:1–3:9. doi:10.1155/2008/281959
- Knees, P., Pohle, T., Schedl, M., & Widmer, G. (2006). Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval* (pp. 147–154). New York, NY, USA: ACM. doi:10.1145/1178677.1178699
- Liang, R.-H., & Ouhyoung, M. (1994). Impromptu Conductor: a virtual system for music generation based on supervised learning. In *Proceedings of the second Pacific conference on Fundamentals of computer graphics* (pp. 257–266). River Edge, NJ, USA: World Scientific Publishing Co., Inc. Retrieved from <http://dl.acm.org/citation.cfm?id=212531.212556>
- Marinos, D., & Geiger, C. (2009). An immersive multiuser music generation interface. In *Proceedings of the International Conference on Advances in Computer Entertainment Technology* (pp. 433–434). New York, NY, USA: ACM. doi:10.1145/1690388.1690486
- Marques, V. M. (2010). Plenary lecture 4: on fitness function and evolutionary computation techniques for music generation. In *Proceedings of the 10th WSEAS international conference on applied informatics and communications, and 3rd WSEAS international conference on Biomedical electronics and biomedical informatics* (p. 18). Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS). Retrieved from <http://dl.acm.org/citation.cfm?id=2170353.2170357>
- Moerchen, F., Mierswa, I., & Ultsch, A. (2006). Understandable models Of music collections based on exhaustive feature generation with temporal statistics. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 882–891). New York, NY, USA: ACM. doi:10.1145/1150402.1150523
- Morris, H., & Wainer, G. A. (2012). Music generation using cellular models. In *Proceedings of the 2012 Symposium on Theory of Modeling and Simulation - DEVS Integrative M&S Symposium* (pp. 37:1–37:8). San Diego, CA, USA: Society for Computer Simulation International. Retrieved from <http://dl.acm.org/citation.cfm?id=2346616.2346653>

- Nierhaus, G. (2010). *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer Publishing Company, Incorporated.
- Pauws, S., Verhaegh, W., & Vossen, M. (2008). Music playlist generation by adapted simulated annealing. *Inf. Sci.*, 178(3), 647–662. doi:10.1016/j.ins.2007.08.019
- Ragno, R., Burges, C. J. C., & Herley, C. (2005). Inferring similarity between music objects with application to playlist generation. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval* (pp. 73–80). New York, NY, USA: ACM. doi:10.1145/1101826.1101840
- Reese, K. (2012). A framework for interactive generation of music for games. In *Proceedings of the 2012 17th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games (CGAMES)* (pp. 131–137). Washington, DC, USA: IEEE Computer Society. doi:10.1109/CGames.2012.6314564
- Shen, H.-C., & Lee, C.-N. (2011). An interactive Whistle-to-Music composing system based on transcription, variation and chords generation. *Multimedia Tools Appl.*, 53(1), 253–269. doi:10.1007/s11042-010-0510-6
- Thalmann, F., & Gaelli, M. (2006). Jam Tomorrow: Collaborative Music Generation in Croquet Using OpenAL. In *Proceedings of the Fourth International Conference on Creating, Connecting and Collaborating through Computing* (pp. 73–78). Washington, DC, USA: IEEE Computer Society. doi:10.1109/C5.2006.22
- Van Der Merwe, A., & Schulze, W. (2011). Music Generation with Markov Models. *IEEE MultiMedia*, 18(3), 78–85. doi:10.1109/MMUL.2010.44
- Wang, J., Xu, C., Chng, E., Duan, L., Wan, K., & Tian, Q. (2005). Automatic generation of personalized music sports video. In *Proceedings of the 13th annual ACM international conference on Multimedia* (pp. 735–744). New York, NY, USA: ACM. doi:10.1145/1101149.1101309
- Wu, X., Xu, B., Qiao, Y., & Tang, X. (2012). Automatic music video generation: cross matching of music and image. In *Proceedings of the 20th ACM international conference on Multimedia* (pp. 1381–1382). New York, NY, USA: ACM. doi:10.1145/2393347.2396495
- Yi, L. (2009). *Decision-theoretic planning in social welfare and music generation*. University of Kentucky, Lexington, KY, USA.
- Yoon, J.-C., & Lee, I.-K. (2007). Synchronized background music generation for video. In *Proceedings of the international conference on Advances in computer entertainment technology* (pp. 270–271). New York, NY, USA: ACM. doi:10.1145/1255047.1255119
- Yoon, J.-C., Lee, I.-K., & Byun, S. (2009). Automated music video generation using multi-level feature-based segmentation. *Multimedia Tools Appl.*, 41(2), 197–214. doi:10.1007/s11042-008-0225-0
- Yu, Y., Shen, Z., & Zimmermann, R. (2012). Automatic music soundtrack generation for outdoor videos from contextual sensor information. In *Proceedings of the 20th ACM international conference on Multimedia* (pp. 1377–1378). New York, NY, USA: ACM. doi:10.1145/2393347.2396493

Zhang, T., Fong, C. K., Xiao, L., & Zhou, J. (2009). Automatic and instant ring tone generation based on music structure analysis. In *Proceedings of the 17th ACM international conference on Multimedia* (pp. 593–596). New York, NY, USA: ACM. doi:10.1145/1631272.1631364

KEY TERMS AND DEFINITIONS

Key Words:

Genetic Algorithm: - A genetic algorithm is a heuristical model of machine learning that is based on the process of natural selection.

Statistical Model: - A statistical model is an interpretation that uses variables and equations to show mathematical relationships.

Probability of occurrence (POC):- The probability of occurrence is a static constant assigned to every music object in the Statistical Phase to produce a sample.

Context-free Grammar: - A Context-free grammar is a formal grammar in which every production rule is in the form $V \rightarrow u$, where V is a single non-terminal symbol and u is a string of terminal and/or non-terminal symbols, u can also be empty.

Gaussian distribution: - Gaussian distribution, sometimes referred to as normal distribution, is a mathematical function that defines the probability of a number in some context falling between any two real constants.

Genetic Phase: - The Genetic Phase is the second phase of a five phased model. The Genetic Phase presents a genetic algorithm that refines a statistical sample through a fitness function and genetic operators.

Statistical Phase: - The Statistical Phase is the first phase of a five phased model. The Statistical Phase presents a Context free grammar and statistical model that produces an initial population.

Music Representation: - a notational portrayal of acoustic music.

Fitness Function: - A fitness function is an objective function that is used to evaluate how close a given construction is to achieving the pre-determined criteria.

Note counting: - Is a procedure where a scale counter for a corresponding scale counts the occurrences of each note in the sample that belong to the corresponding scale. The corresponding scale with the largest counter value is returned.