UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Artificial Intelligence and Music Cognition

Diploma Thesis

Athanasios Tzikas

Supervisor: **Aspasia Daskalopoulou**

Volos 2021

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Artificial Intelligence and Music Cognition

Diploma Thesis

Athanasios Tzikas

Supervisor: **Aspasia Daskalopoulou**

Volos 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνητή Νοημοσύνη και Μουσική Γνωστική Λειτουργία

Διπλωματική Εργασία

Αθανάσιος Τζήκας

Επιβλέπων: **Δασκαλοπούλου Ασπασία**

Βόλος 2021

Approved by the Examination Committee:

Supervisor	**Aspasia Daskalopoulou** Assistant Professor, Department of Electrical and Computer Engineering, University of Thessaly
Member	**Michael Vassilakopoulos** Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly
Member	**Panagiota Tsompanopoulou** Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Date of approval: 24-02-2021

Dedicated to my family

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

The Declarant

Athanasios Tzikas

24/02/2021

ABSTRACT

Over the past years, music compositions have involved a number of traditional activities, such as the creation of melody and rhythm, voice-leading, musical arrangement or orchestration, and notation. Nowadays, thanks to technology, all the above activities can be automated by a computer on various levels and specifically algorithmic composition has facilitated the automated procedure of generating music compositions. The tasks of producing music can be applied nowadays in two ways. The first one is to generate music which is similar either to a number of compositions or a specific style. The second way is to automate composition tasks and generate compositions without the need of human intervention. Modeling music cognition with artificial intelligence is commonly seen as a way of enhancing our knowledge in the field of human psychology and intellect. In fact, the procedure involves taking some problem solving steps. First, we need to figure out how to measure music in the form of input information. The next step is to find a way to present that information to the computer. Moreover, it is important to know how the information will be represented in the computer program so that the program can come to some understanding of its meaning. Finally, we should focus on what the computer will do with all this knowledge. This thesis is bibliographical and contains information from already existing research on algorithmic composition.

ΠΕΡΙΛΗΨΗ

Στο παρελθόν, η μουσική σύνθεση περιλάμβανε μια σειρά από παραδοσιακές δραστηριότητες, όπως τη δημιουργία της μελωδίας και του ρυθμού, την προσθήκη κύριας φωνής, τη διάταξη και ενορχήστρωση, και τη μουσική σημειογραφία. Σήμερα, χάρη στην τεχνολογία, όλες οι παραπάνω δραστηριότητες μπορούν να αυτοματοποιηθούν από ένα υπολογιστή σε διαφορετικά επίπεδα και συγκεκριμένα η αλγοριθμική σύνθεση έχει διευκολύνει την αυτοματοποιημένη διαδικασία δημιουργίας μουσικών συνθέσεων. Οι εργασίες παραγωγής μουσικής μπορούν να εφαρμοστούν πλέον σήμερα με δύο τρόπους. Ο πρώτος τρόπος είναι να δημιουργήσουμε μουσική που είναι όμοια είτε με έναν αριθμό συνθέσεων είτε με ένα συγκεκριμένο στυλ. Ο δεύτερος τρόπος είναι να αυτοματοποιήσουμε τις εργασίες σύνθεσης και να δημιουργήσουμε συνθέσεις χωρίς την ανάγκη ανθρώπινης παρέμβασης. Η μοντελοποίηση της μουσικής γνωστικής λειτουργίας με την τεχνητή νοημοσύνη προσεγγίζεται συνήθως ως τρόπος αύξησης των γνώσεών μας στον τομέα της ανθρώπινη ψυχολογίας και διάνοιας. Στην πραγματικότητα, η διαδικασία μοντελοποίησης περιλαμβάνει την ανάγκη να ακολουθήσουμε τέσσερα βήματα επίλυσης προβλήματος. Πρώτον, πρέπει να ανακαλύψουμε πώς να μετρήσουμε τη μουσική έτσι ώστε να έχει τη μορφή εισόδου πληροφοριών στον υπολογιστή. Το επόμενο βήμα είναι να βρούμε έναν τρόπο να παρουσιάσουμε αυτές τις πληροφορίες στον υπολογιστή. Στη συνέχεια, είναι σημαντικό να γνωρίζουμε πώς οι πληροφορίες θα αναπαρασταθούν στο πρόγραμμα υπολογιστή έτσι ώστε το πρόγραμμα να μπορεί να κατανοήσει τη σημασία του. Τέλος, πρέπει να επικεντρωθούμε σε αυτό που θα κάνει ο υπολογιστής με όλες αυτές τις γνώσεις. Αυτή η διπλωματική εργασία είναι βιβλιογραφική και περιέχει πληροφορίες από ήδη υπάρχουσες έρευνες σχετικές με την Αλγοριθμική σύνθεση.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Music plays an important role in today's society. People always find a way to express their inner world, through the combination of sounds, voices or instruments, to achieve beauty of form and expression. Music can also be defined as those temporally patterned human activities, which involve the production of sound and have no evident and immediate aim[1]. In other words, what music is for some at present is not what music is for others, was for our predecessors, or could be for the next generations in the future. We could also look at music as a multimedia activity of some patterned movements that has the ability and the capacity to coordinate the emotions of participants and thus lead to conjoinment*[1].*

Musical knowledge derives from the development of music through an individual's lifetime as well as its history through the years that human species exist. Music cognition is an advantageous area to study, because of its relation to intuitive and learned knowledge. Knowledge of intuition, is commonly shared among humans as a part of our evolutionary history, while learned Knowledge can be diverse across cultures as it is produced by cultural evolution.

What is interesting to Know about music cognition is the fact that our neural structure was shaped by natural selection throughout the evolution of the human species. According to researchers who have examined the evolution of music in our species, music has offered an advantage to the individual either in the social framework or via mechanisms of sexual selection [2]. Their studies reflect a more general trend in modern psychology to perceive cognitive abilities as specific adaptive solutions to evolutionary problems. Therefore, we can now see music as a product of cultural or mimetic transmission, which consists of concepts that are grounded in specific social interactions.

Many intuitive abilities that humans apply to music perception and music composing may not have evolved as musical processes by definition, but rather as processing mechanisms and knowledge selected for their utility within other domains such as conceptual representation, language expression, timing and emotion.

This idea is similar to the concept of exaptation, which is used to describe morphological forms that arise not because of direct selection pressures, but rather are the inevitable result of selection pressures for other qualities. The study of the human brain has always presented interesting results, which helped science to evolve and Artificial Intelligence is definitely a branch of it that has a lot to present in the not too distant future.

Nowadays, more and more researchers are trying to implement algorithms capable of imitating the way that the human brain works in terms of music perception and cognition. This thesis aims to present my research on the studies which have already been carried out in the specific field of computational creativity called Algorithmic Composition, as well as to display the results this composition has produced.

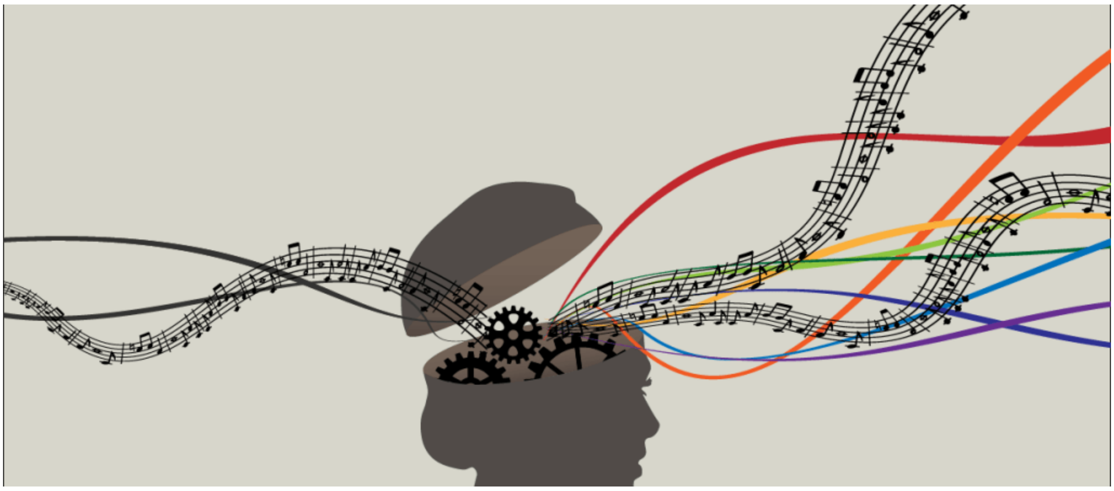
CHAPTER 2

COGNITIVE SCIENCE AND MUSICAL INFORMATION

2.1 Introduction

Having reviewed this issue from the perspective of cognitive psychology, it is time to discuss how the mental algorithms related to music cognition can be implemented in the human brain.

First of all, we shall define musical information and the difference it makes to us. Information in general is defined as one or more bits of awareness, which are combined by an information processor (e.g. a brain or a computer), to produce some meaning. A processor is required to make a difference. This difference takes place at the processing stage and is noted between a bit of awareness and the current state of the information processor. The human mind is a great example of this kind of a processor and the way it functions when it is aware of music information is worth mentioning.*(*Figure *2.1)*



2.1 Brain perception of musical information [3]

2.2 How do our brains receive musical information?

So what bits of awareness do we receive, when we listen to a piano playing? Using musical terminology, we hear a note with a certain pitch, duration, amplitude and timbre. Our brain already contains a lot of preconceived Knowledge about the specific space we are in, what usually happens in this space and that we are likely to hear piano sounds, because we know that there is a piano in the room. These new bits of awareness that come from a piano playing , are perceived as different because we hear a note which is different from the preceding silence. Since the piano is the only instrument in the room, we identify that the note has the timbre of a piano.

We could also identify if the note was short or long (based on our knowledge about the sustaining capabilities of a piano), the exact pitch of the note that is being played (if we have a perfect pitch). In addition to this information, the difference in time between the two notes has added another bit of awareness. Information in the other parameters has provided evidence of repetition. We need to mention that the role of repetition in music is different from redundancy in information theory. According to *Chris Dobrian (1993, p2),* repetitive events or information in music cannot be considered redundant since probability and expectation are reinforced by repetition. Regarding repetition, we need to bear in mind that past evidence is simply evidence, not an explanation for what happened in the past. As for musical applications of information theories, past events do not determine the future because we Know , for example, that a composer can produce something new unexpectedly due to his free will. In this case some evidence is missing. We only know what has happened, but we don't know how or why or which rules lead to this behavior. Therefore, the odds of repeating the pattern or not are still even after 99 repetitions, and we can't predict what will happen afterwards if we don't Know the reason why it will happen. *[4].*

As far as musical performance is concerned, it also involves how musicians, through the performance, communicate structure to the listener. The written notation of Western music represents pitch and duration much more explicitly than it does the structural and expressive principles, such as phrasing and tension-relaxation. However, the performer provides information about these unwritten aspects of the piece to the listener, often through systematic variances from the notated music. In many cases the qualities that affect the way music is perceived by the listener, are changes in tempo, dynamics, and synchrony, done in a systematic way as to bring the structure of the song across to the listener *[4].*

2.3 Aesthetic decisions

Two categories of computer music exist. One is about compositions that are produced with the help of computers and the other one is about compositions made by computers alone. We can only say that a computer composes music by itself, when it has the ability to make choose between random options that are based on some aesthetic knowledge that was provided to it by the programmer, or even on information that the program collected through a period of time. On the other hand, if the computer program just follows some commands or rules and has no ability of choosing between some options, it is just proceeding to the performance of specific pre-ordered by the programmer tasks. *[4]*.

What is an aesthetic decision?

By aesthetic decision we mean a choice which attempts to achieve an interesting and pleasant result. Artificial intelligence is mainly concerned with the idea of explaining decisions with algorithms because computers just follow some orders and they don't have the ability to understand the reason why. This means that it's the responsibility of the programmers to figure out how a computer is going to reach a pleasant result by making aesthetic decisions [*4] .*

How do humans make decisions when they compose music?

When it's time for the composer to decide about which the starting element of his/her song would be (e.g., tempo, pitch), he/she has to either be based on some aesthetic decision or on some arbitrary choice made among a corpus of ideas that he/she has collected. This last approach is a procedure that a computer can do way faster than a human-being. Therefore, we can see, that every decision that is based on some rules, needs to be based on some prior aesthetic or arbitrary decision in order to be made. By trying to trace these aesthetic or arbitrary criteria back to their prior choices, we are bound to reach a deadlock where we simply like the result or it doesn't matter. In this case, we meet a new attribute called taste. Until this moment, this kind of decisions that are made using the intellect, cannot be understood by the human mind and this is why intuition should be considered as a dimension that is included in making decisions. In the case we reach a point where we cannot decide which aesthetic decision is better, we choose to take the path of making an arbitrary choice between them. *[4].*

Computer programs and aesthetic decisions

To conclude, we can understand that a decision-making program cannot be independent of the programmer's taste and intuition and since the computer lacks any creative capability, some mechanism is needed as a substitute to human creativity for any AI algorithm applied to a creative problem *[4].*

2.3.1 Aesthetic decisions and creativity [3]

2.4 Definitions of important attributes of music

Among the temporal attributes of music are pitch, meter, rhythmic pattern, grouping, and tempo.

Pitch Scales

Most periodically vibrating objects, to which we attribute pitch, including the human vocal folds and the strings of musical instruments, vibrate at several frequencies simultaneously. These frequencies are approximately integer multiples (harmonics) of the fundamental frequency, and the complex is called a harmonic spectrum .Although each of these frequencies sounded alone would evoke a spectral pitch, when sounded simultaneously they make up a singular periodicity pitch*[5].*

Traditionally, pitch has been described as varying along a single dimension from low to high, called pitch height. Along this dimension, pitch is a logarithmic function of frequency. The Western equal-tempered tuning system divides each frequency doubling (octave) into twelve equally spaced steps (semitones) on a logarithmic scale, where one note is about 1.06 times the frequency of the preceding note *[5].*

EQUAL TEMPERED FREQUENCIES (Hz)

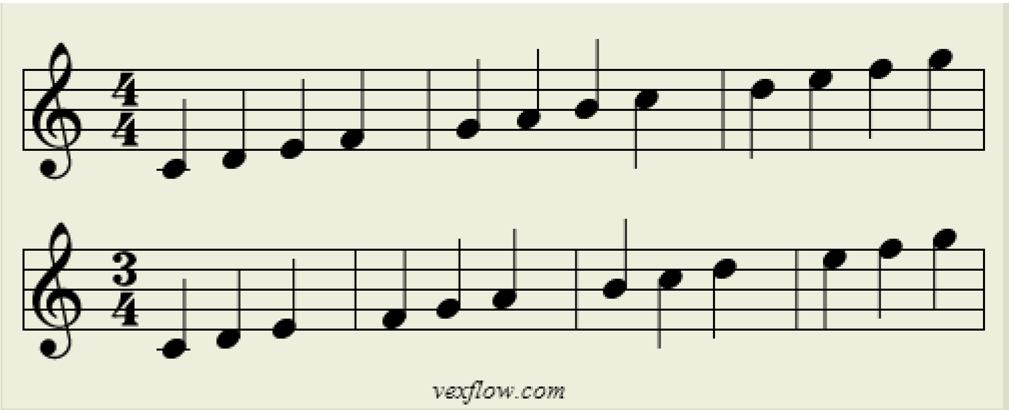
Note	Hz	Note	Hz	Note	Hz	Note	Hz
C_0	16.352	C_2	65.406	C_4	261.63	C_6	1046.5
	17.324		69.296		277.18		1108.7
D_0	18.354	D_2	73.416	D_4	293.66	D_6	1174.7
	19.445		77.782		311.13		1244.5
E_0	20.602	E_2	82.407	E_4	329.63	E_6	1318.5
F_0	21.827	F_2	87.307	F_4	349.23	F_6	1396.9
	23.125		92.499		369.99		1480.0
G_0	24.500	G_2	97.999	G_4	392.00	G_6	1568.0
	25.957		103.83		415.30		1661.2
A_0	27.500	A_2	110.00	A_4	440.00	A_6	1760.0
	29.135		116.54		466.16		1864.7
B_0	30.868	B_2	123.47	B_4	493.88	B_6	1975.5
C_1	32.703	C_3	130.81	C_5	523.25	C_7	2093.0
	34.648		138.59		554.37		2217.5
D_1	36.708	D_3	146.83	D_5	587.33	D_7	2349.3
	38.891		155.56		622.25		2489.0
E_1	41.203	E_3	164.81	E_5	659.26	E_7	2637.0
F_1	43.654	F_3	174.61	F_5	698.46	F_7	2793.8
	46.249		185.00		739.99		2960.0
G_1	48.999	G_3	196.00	G_5	783.99	G_7	3136.0
	51.913		207.65		830.61		3322.4
A_1	55.000	A_3	220.00	A_5	880.00	A_7	3520.0
	58.270		233.08		932.33		3729.3
B_1	61.735	B_3	246.94	B_5	987.77	B_7	3951.1
						C_8	4186.0

Lowest piano note
Middle C
Black keys
Top note on piano

2.4.1 Frequencies Table [6]

<u>*Meter*</u>

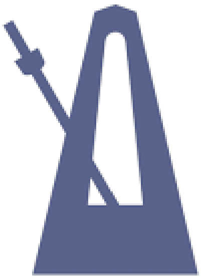
Meter is a hierarchical organization of beats .The first essential characteristic of meter is isochrony, which means that the beats are equally spaced in time, creating a pulse at a particular tempo. A beat has no duration and is used to divide the music into equal time spans, just as in geometry a point divides a line into segments[5].



2.4.2 Examples of Meter [7]

Tempo

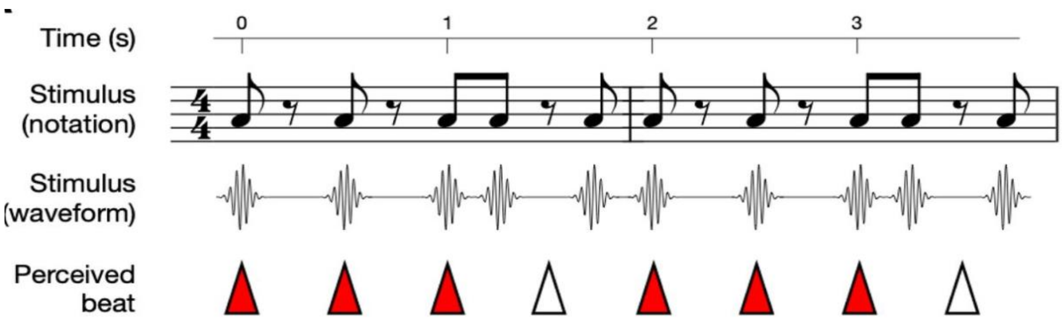
With the help of the meter we can identify the tempo, which is an attribute used to describe the rate at which the basic pulses of the music occur*[5].*



2.4.3 Metronome for Tempo Detection[8]

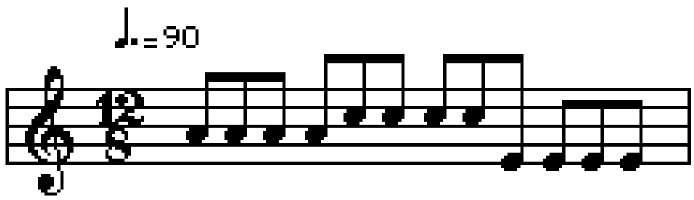
Rhythm Perception

Rhythm is perceived by detecting patterns of events in time. A listener has to detect a basic time interval, a beat or pulse, which remains stable for some period of time in order to detect a target rhythm. Any method of pattern detection can classify in the same group musical events which belong to the same pattern. As far as the perception of rhythm is concerned, we use the outline of time intervals within the patterns of identical events to find out the perceived rhythm in a piece of music. This rhythm is then analyzed for patterns which may indicate organizational concepts such as pulse, beat, and meter [5].



2.4.4 Perceived beats from music analysis [4]

Many factors can determine the perception of rhythm. In the example showing in the picture below (*Figure 2.4.*5), we notice that the dynamic accents and the pitch contour present two different additional rhythms, a dynamic accent every three eighth notes and a change of pitch every four eighth notes. Almost all Western music is characterized frequently by this type of interplay of different rhythms [5].



2.4.5 Western music is well known for interplaying different kind of rhythms[4]

CHAPTER 3

HOW DO COMPUTERS PERCEIVE MUSIC?

3.1 How we measure musical information?

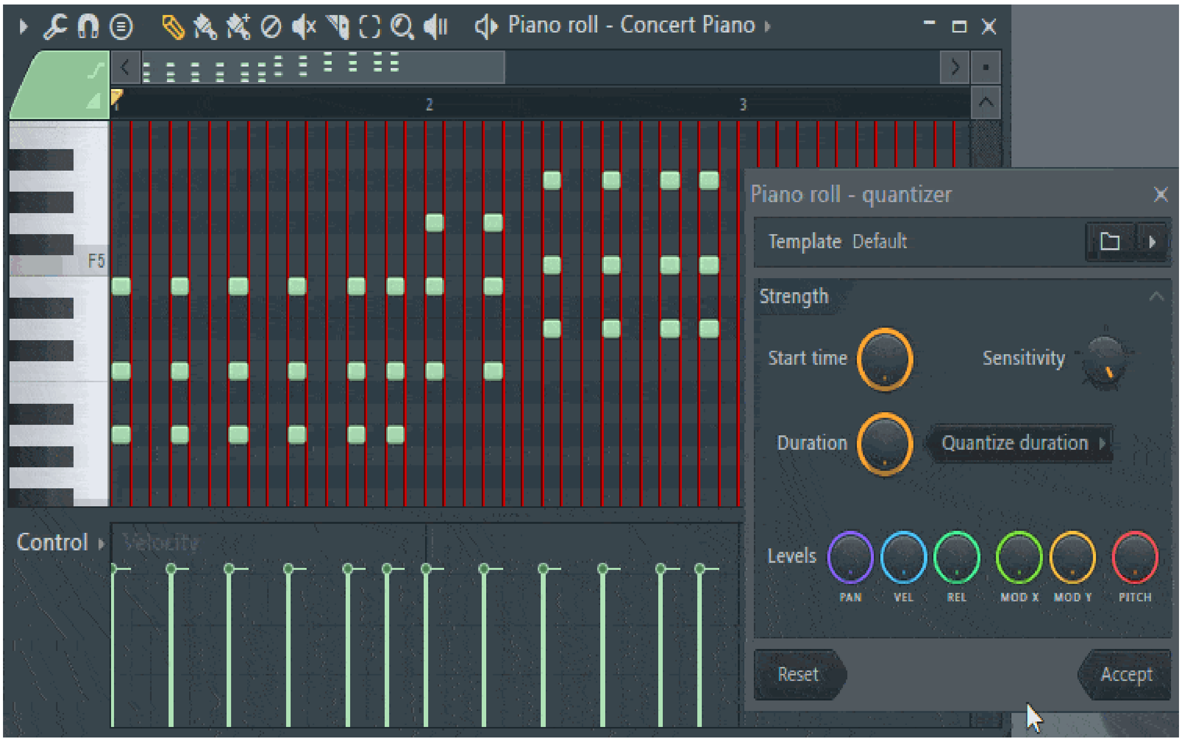
In order to produce computer input, we must decide on one or more parameters to measure because we cannot measure all aspects of a piece of music in any significant way. Yet, we must keep in mind that culture, musical style and individual preference affect our definition and choice of these parameters. Inevitably, a cognitive model will show the personal favor of the programmer. After we decide what to measure, we must tackle problems of how to measure.

We are now going to examine the representation of the input information in our program, and what we intend to do with it.

3.1.1 Graphic representations of three possible measurements, in order from minimum information to maxim [4]

3.2 MIDI Quantization

The unit of measure (such as milliseconds) in most computer implementations is by far smaller than the smallest interval to be considered a musical pulse. As a result, the input data must therefore be modified and quantized to correspond with a reasonable musical pulse unit. The best method to perform this is MIDI quantization. MIDI uses a simple rounding method of quantization, in which each event is rounded to the nearest multiple of a basic minimum quantum. Quantize resolution tells the computer how fine the grid should be. For example, if we pick eighth note resolution, all notes will be moved to the nearest eighth note position. If we happen to have played a rhythm that includes sixteenth notes, our phrase will get changed in a way we won't intend to. A good rule is to quantize to the shortest note we've played. This means that, if the phrase features eighth and quarter notes, we use eighth note resolution. This method does not permit changes in the tempo of the performance. Once the input data has been represented in the computer as a time-tagged set of MIDI bytes, the data is processed by the computer program in order to interpret its significance. *[4]*



3.2.1 MIDI Quantization as shown in DAW FL Studio [9]

3.3 A rule-based expert system for music perception

In *1988* Benjamin Miller, Jacqueline A. Jones and Don Scarborough, wrote an algorithm in Pascal, which aim was to take advantage of the theory of music perception and to illustrate a number of techniques. Their goal was to create an algorithm which perceives music in a similar way that human beings do. For example, a song can be divided into rhythmically similar half parts and each one of them will consist of two lines. Each half will also contain two notes per line and these notes will be equally spaced in time. This pattern of structure is the *metric structure*. What we call the *grouping structure*, is the process in which we divide these two half parts into lines, which is an essential part of music perception as well. A listener could also have the ability to perceive the key of the song (its tonality). When we listen to a part of a song we are familiar with, we already know what is the next chord of the part we perceive at this moment. These intuitions were formalized by Jackendoff and Lerdahl (1983) in a rule set of theories, which they called the "Generative Theory of Tonal Music" (GTTM). In this theory, an analysis is taking place, which is divided in four parts (stages). Each stage is embodied in a set of rules [10].

The use of a rule based system in this case, was due to the expression of the GTTM in a set of rules and due to the fact that rule based systems develop in rapid rhythms which made the researches browse through some of the available techniques. Therefore, it was the expression of theories as a set of rules that a rule based system supports, which made it appropriate for the implementation of this modular approach. The researchers wanted the simulation to implement the psychological processes which are related to the perception of music and this is the reason why their goal was to make this system function like a human which will perceive information in real time. As a result they preferred to use an architecture called Blackboard.

3.3.1 The Algorithm

The Blackboard

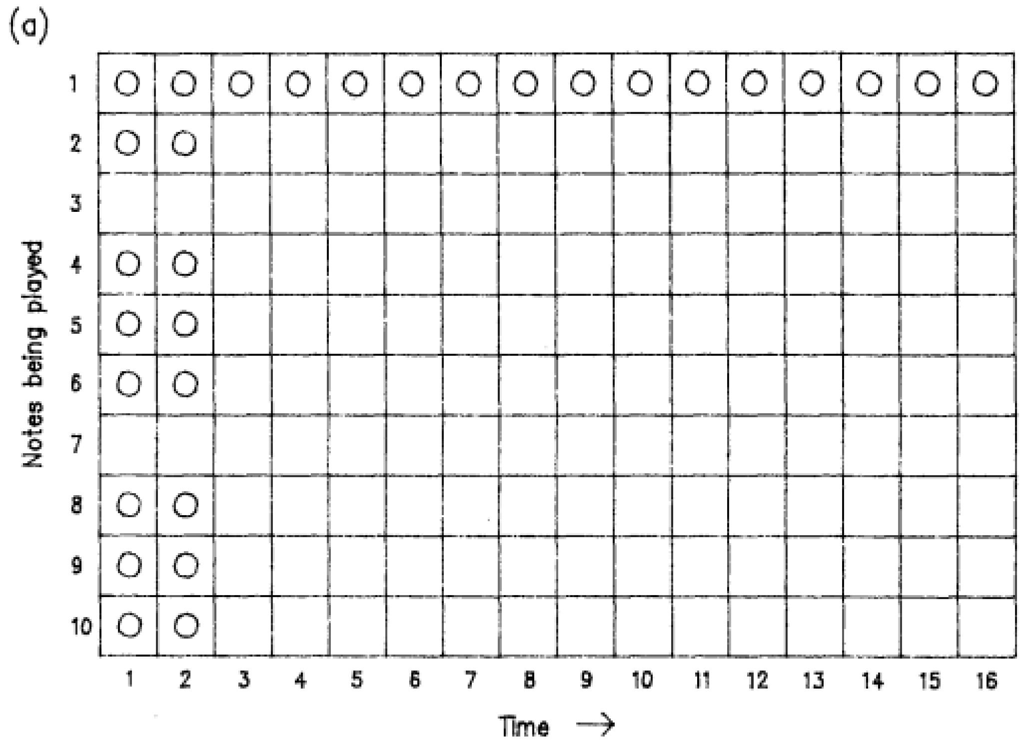
The model of a Blackboard system is divided in three parts:

1. A data structure which is global, the blackboard.
2. The so called "knowledge sources" (KSs). Each one of them encloses information considering a specific stage of the main problem.
3. A structure which has the role of monitoring and controlling which KSs will be active.

The input information as well as information gathered as the analysis proceeds is all stored in different levels. Each level, in which the blackboard is divided into, represents different kind of knowledge. In this model that researchers implemented, information considering the individual notes is stored at the base of the blackboard. [10].

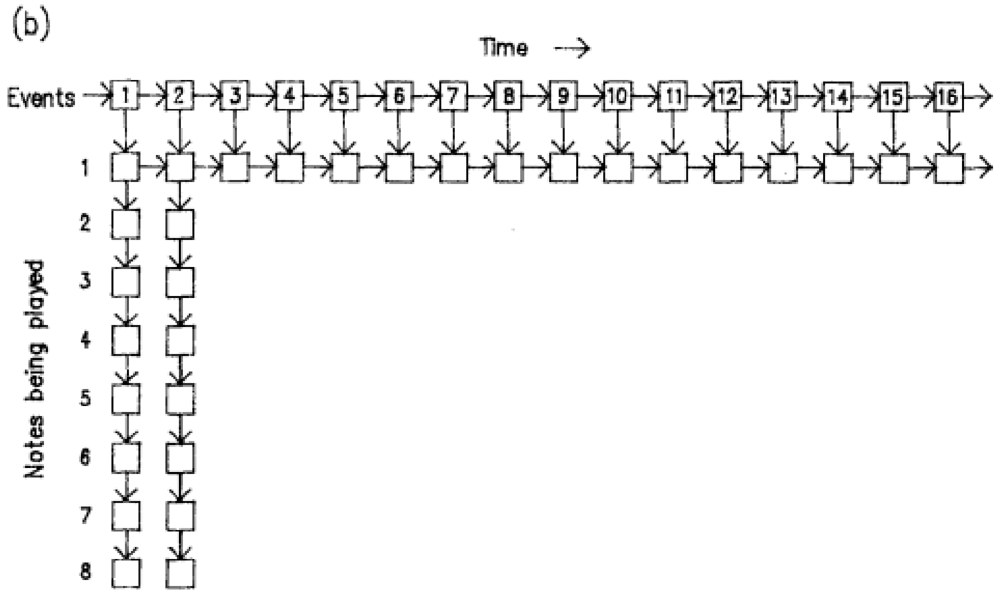
Other levels of the blackboard represent data related to the analysis of a song, such us metric information, grouping information, and tonality information. Other levels are also being represented, meaning the ones that are based on a mixture of this information, referring to a higher level. Researchers stored data related to the levels within in linked lists and used pointers which were essential, so that the different levels of the blackboard could be linked and remain synchronized at the same time. As for the representation of the musical input, it needed to be solid, because we know that notation in music can be very dense, supplying the listener with a lot of data for every note that it contains. For instance, what a listener first perceives in a listening experience is data related to the pitch, the duration of the notes etc. This kind of information, represented in a symbolic way, is stored in the base of the blackboard. Researches also used a matrix in order to capture data (notes) at the time they are being captured (played). Every row of the matrix represents a single instrument with the sequences of the notes it plays and every column includes all notes that instruments play at an exact time. [10].

Representing this kind of information with a classic array wouldn't be the best way to approach this problem, because most personal computers have limited memory and the result would have more cells being blank than including actual information (sparse matrix). Researchers tried to examine the results of representing piano music in this way. A 10 x 10 array was used so that it can represent the maximum amount of notes a piano player could play (rows) and their success in time (columns). We can see the results of such an example in *Figure 3.3.1* below. A chord which consists of 8 notes is being played for two intervals, and 14 single notes follow. Therefore, we can clearly notice that the use of a matrix in this case would lead to an array with 130 empty cells out of the 160 allocated. [10].



3.3.1 10x16 sparse matrix with 130 empty cells [10]

A dynamic list with nodes that are linked with pointers with each other is what researchers used as a solution to this problem. The allocation of a node happens only when a new element, containing information, needs to be added to the list. In *Figure 3.3.2*, arrows represent pointers, and each pointer is responsible for the connection between nodes, which is the main source of information. Pointers when followed can link to the next or the previous node, and this connection has the result of a linked-list being created. *Figure 3.3.2* below shows the same example as before, but this time with the approach of linked lists. In this figure, the first line consists of nodes that represent events in the order that were captured. Every node is responsible for storing data about a note. An array approach would lead to the allocation of 160 cells, compared to the linked-list approach that saves space requiring only 46 nodes instead. Therefore, it is quite understandable why the linked-list approach was the preferred one here, because it set the researchers free of the obligation to decide how many notes will be represented and the maximum amount of notes that will be played in every measure.



3.3.2 A linked-list representation of (3.3.1) [10]

As mentioned above, the blackboard is a global data structure which contains all 4 stages of this implementation (score, metric structure, grouping structure, tonality analysis). What researchers called the “backbone” of this structure is the so called “notochord”. The notochord is made of a doubly linked list , in order to capture the events that are happening in a specific time span in a specific time order. Every node of this list, only contains information about which is the time it took for an event to take place. This is simply the calculation of time between the previous event that started substituted from the time that the next event begins. This time, pointers also connect these nodes to other nodes with note data and to the representation of metric and grouping as well. No score data is included in these event boxes. On the other hand, each note node represents score data about a specific note that was played, and also contains a pointers’ pair, which role is to establish the connection of such a note to the “notochord” and to the rest of the notes that were played as well. [10]

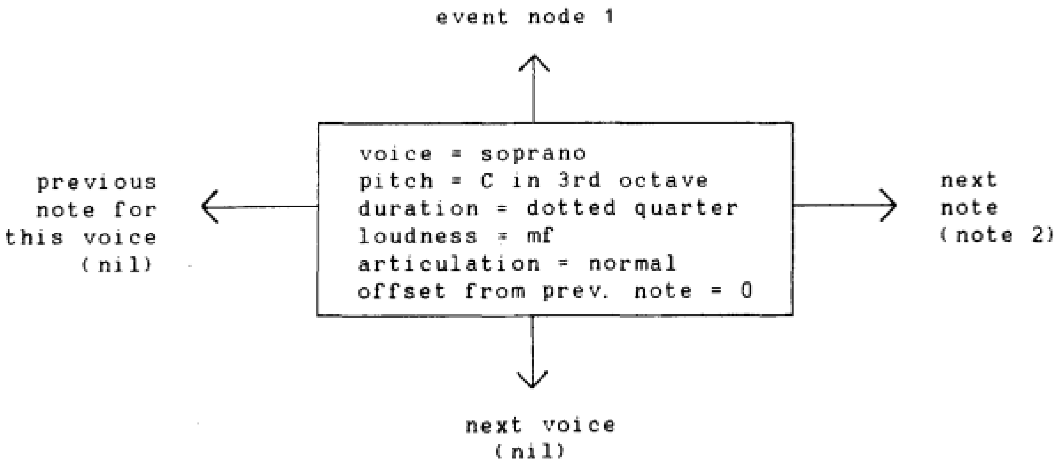


Figure 2. Schematic representation of the first note node for "Row, Row, Row Your Boat."

3.3.3 The first note node for "Row, Row, Row Your Boat" in a schema [10]

<u>Knowledge sources</u>

Researchers tried to create a single KS for the implementation of every GTTM rule. A KS follows the "if-then" concept. Therefore, KS orders an action when some specific conditions take place. The following instance of a simple GTTM rule is which is presented by the researchers, is the algorithm below:

"(IF n1n2n3n4 is a 4-note sequence,

and n2, and n3, differ in duration,

and n1, and n2 have equal durations,

and n3, and n4 have equal durations,

THEN there is evidence of a grouping boundary between n2 and n3)"

In this simulation, a KS structure contains two parts:

1. A function which role is to match patterns and to check the blackboard for a specified prior pattern.
2. A consistent procedure which role is to apply all the changes that are done in the blackboard.

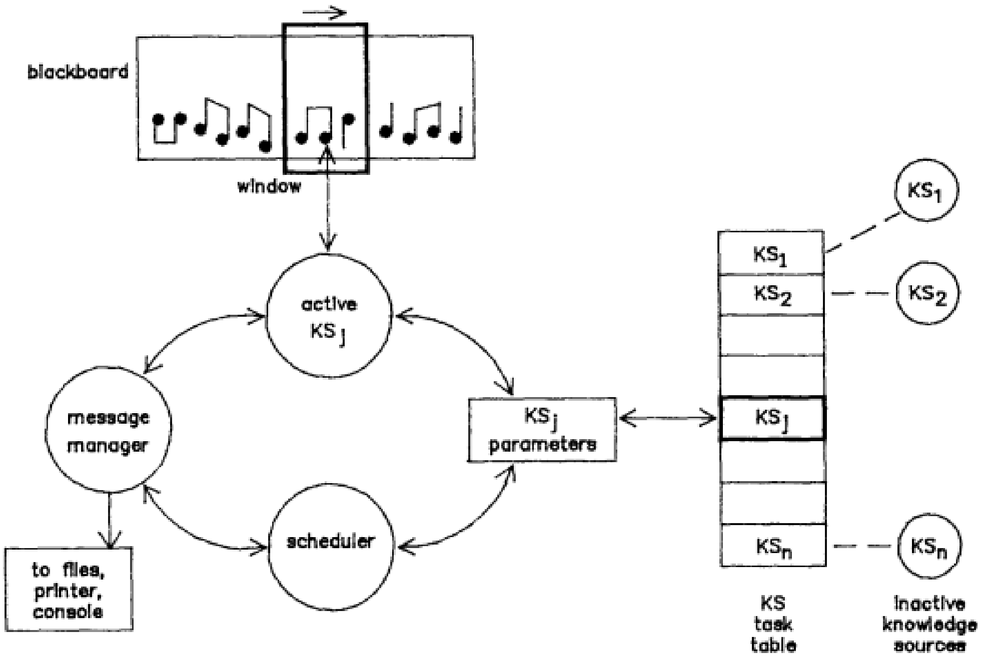
All these KSs needed to belong in a form-structure that will be able to make communication between KSs easier and this is why they chose to implement a structure based on “object oriented programming”. Therefore, every KS is considered an "object" which is responsible for the information in the structures, and encapsulates different methods that could be applied in each one of them. When the KS receives a specific message (initialize, execute, evaluate), it then applies the change that was asked to the data structure, a job that is simple due to the modular structure that this system is implemented (add replace nodes). [10]

Control structure

As mentioned above, a scheduling mechanism, is responsible to monitor what exactly KSs are about to execute and what is going to be the updated version of the blackboard. Although this procedure has the limitation that only a single KS can be running at a specific time, it is able to apply necessary changes as the analysis of song part is taking place. One of the most essential aspects of this implementation is the so called “window”, that defines which piece of the blackboard will be visible. This is the only piece of the blackboard that can be modified by a KS. Human beings have a short-term memory, when listening to music and this is the reason why researchers didn’t use unlimited memory for this implementation, in order to make it more familiar to the human listening experience.

The Implementation

At first the algorithm receives an input data file regarding the part of a song and creates the “notochord” of this input. Then, the window parameter is specified, and a specific part of the structure that was created is removed from the window. What is contained in the window is the exact data that a listener will first receive (researchers used windows that had the size of six – ten notes). The scheduling procedure maintains a task array, with each element of this array, being a different KS. The scheduler receives information from this array, such us information about the runnability and the priority of a KS and when it is time for a specific KS to run, a global variable is used in order to copy the task table there. After that, a notification is sent to the KS, informing it to execute a specific action (e.g., modification of the blackboard). A KS is responsible to inform the scheduler in case of a modification of its status or in case of an action being completed. An overview of this implementation can be shown in *Figure 3.3.4* below, which also shows the “Message Manager” which has the ability to control communication between the scheduler and all KSs [10].

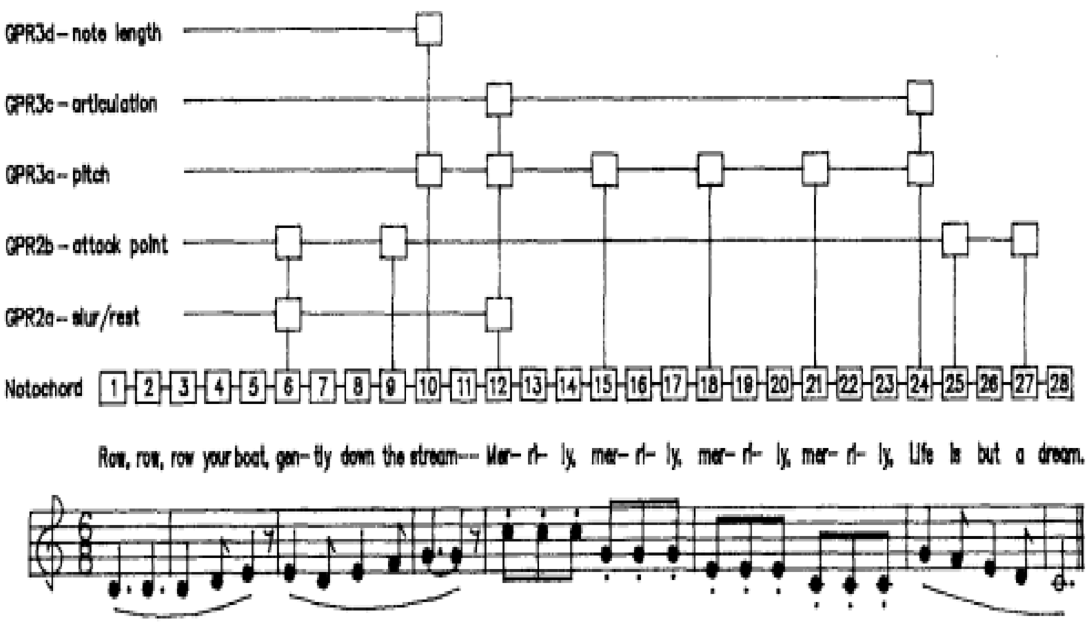


3.3.4 The “Control Structure” of the simulation [10]

In this implementation, the scheduler maintains a list of all the KSs sorted, based on their priority and calls the one which has the highest one, allocating some specific time that this KS will run. After this time has passed, another KS is selected to be executed. In this experiment, it is not the scheduler, but the size of the window which informs the KS about how much data it has to work with. For instance, a small window would trick us into thinking that all KSs are being executed at the same time [10].

3.3.2 Grouping Analysis

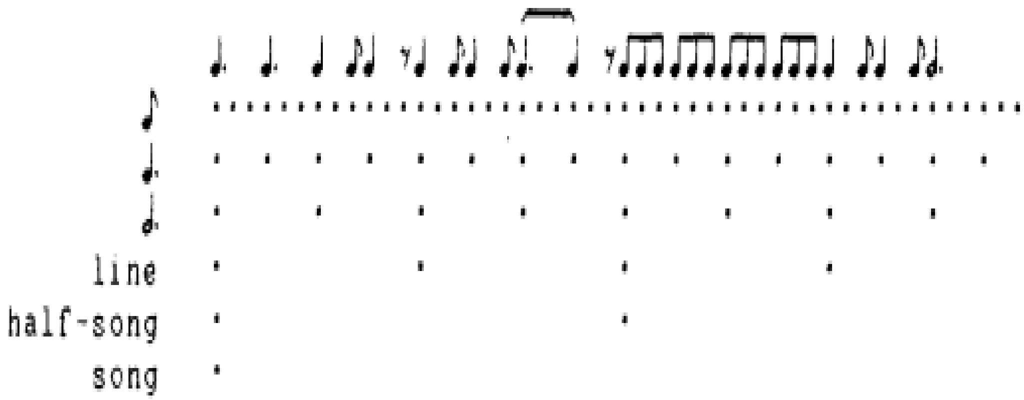
The rules that are responsible for the grouping of qualities such us the distance in the scale a song is in (e.g. C minor) among two notes, the difference of time that the notes are being played and the difference in the length of two notes are called grouping rules (sub-rules). If a boundary between some notes is found by a sub-rule, then the creation of a grouping node is completed, while also creating and adding this node to the list which contains the boundaries that this specific rule produced. At the end, all these boundaries form a new structure, which is being examined by rules of a different (superior) level: A single linked list is created for every grouping that was made, and each of its nodes represents a candidate boundary it has found. Every grouping node is also connected to its related event in the notochord, and its nearest grouping node that the same KS created. Last but not least, if some other grouping rules have noticed a boundary at the exact event, then they already produced a node, which also needs to be connected with the nodes which have the same boundary. This connection of all nodes that have the exact same boundary makes it easier for a higher rule-set to examine the evidence at this given spot. *Figure 3.3.*5 below, illustrates the grouping analysis of song "Row, Row" that researchers used for this example. The most important notes, according to the rules that were applied by the researchers were the combination of boundaries between the eleventh and the twelfth note. Therefore, the piece is divided in two parts due to this boundary. The boundaries between the fifth and the sixth note (after "boat") and between the twenty-third and the twenty-fourth note (after last "merrily"), will define the second most important part of the piece [10].



3.3.5 "Grouping analysis" for song "Row, Row" [10]

3.3.3 Metric Analysis

The direct translation of GTTM metric rules into algorithms is not possible. On the one hand, grouping rules include analysis algorithms, while on the other hand, metric rules' role is to only point out what the form of the metric hierarchy is going to be when the parsing is over. Therefore, the analysis was implemented in one pass, using routines which are based on "the grid theory".[11] According to Povel's grid theory, we can try to fit grids of different size and of equally spaced marks to the musical events as well as to determine how much stress, or how good it fits, among these grids and events. Researchers proceeded in the modification of grid theory in different ways. For example, only the grids that the intervals proposed are taken into consideration and all grids that fit, correlate to a specific metric level in the GTTM hierarchy. In this experiment, grid stress is used in order to delete the parts (levels) of the hierarchy that a listener cannot hear. As for the way that researchers represented the metric structure, data considering the metric hierarchy at each beat is encapsulated in metric nodes, in addition to pointers which points to the notochord and to other nodes as well. The results of metric analysis of song "Row, Row" that researchers used for this example, is shown in the *Figure 3.3.6. [10]*



3.3.6 Metric analysis of "Row, Row". The relative rhythmic importance of the notes can be indicated by the number of dots [10]

3.3.4 Tonality Analysis

The key of a music piece can be defined by human beings, and such thing can influence meter and the grouping as well. In the example we are examining, as for the metric structure, strong beats fall on notes C, E, and G which are the essential pitches which reflect the key that this song is in. The key of the song "Row, Row," is C Major and the piece contains twenty-five notes. What we notice in this example is that only five of the potential twelve chromatic notes occur (C (8), E (5) ,F ,G (5), and D). C, G, and E, when added, occur eighteen times out of the twenty-five notes and knowing that these 3 notes are the fundamentals of C Major, helps us extract the tonality of the song. Therefore, the most frequent notes of the chromatic scale, supply us with information about the key of the song part. A pitch node is used to represent every note and nodes are linked to chord nodes. Each one of these chord nodes gets input data from only three pitch nodes. The tonic, the subdominant, and dominant chords of a specific key, in groups of three chord nodes are linked to the key nodes. During the analysis, if notes are met in the input data, they activate the pitch nodes. Chord nodes are also activated by the pitch nodes and this finally leads to the activation of key nodes. The key node that is activated more times at some part of the piece sets the key that someone will perceive at any that exact point. In order to find the key of the piece, a single KS is designed from the researchers, and this mechanism is activated when new input data (notes) appear in the window. Therefore, in case of a new input, a new tonality calculation is done and this creates a new (temporary) tonality node, which is posted by the KS [10].

CHAPTER 4

APPLICATIONS OF ARTIFICIAL INTELLIGENCE IN MUSIC COMPOSITION

4.1 How are we going to use musical information?

So know that we know what music information consists of and how to get information from it, it’s time to decide what we want to do with it .This questions will influence what we are about to measure in musical information, the ways that these measurements are going to be represented, and how we are going to proceed with the data that are being represented. What is going to help us solve this problem is an insight into our own mental processes. A human-being active in music making, is always influenced by already existing songs he/she likes. Different kind of musical data can be used as initial inspiration (information) in a generative process like this. This kind of process may also be implemented as a compositional algorithm.

4.2 Grammars

In broad terms, a formal grammar may be defined as a set of rules to expand high-level symbols into more detailed sequences of symbols (words), representing elements of formal languages. Words are generated by repeatedly applying rewriting rules, in a sequence of so-called derivation steps. In this way, grammars are suited to represent systems with hierarchical structure, which is reflected in the recursive application of the rules [12].

To compose music using formal grammars, an important step is to define the set of rules of the grammar, which will drive the generative process. The rules are traditionally multi-layered, defining several subsets of rules for different phases of the composition process, from the general themes of the composition, to the arrangement of individual notes. There are methods like examining a corpus of pre-existing musical compositions to distill a grammar able to generate compositions in the general style of the corpus, or using evolutionary algorithms. What is also important is the mapping between the formal grammar and the musical objects that it generates, which usually relates the symbols of the derived sequences, with elements of the music composition like notes, chords or melodies. The use of the derivation tree to determine the various aspects of the musical composition is another possible mapping. The problem with a grammatical approach to algorithmic composition is the difficulty to manually define a set of grammatical rules to produce good compositions. This problem can be solved by generating the rules of the grammar (and the way they are applied) automatically. Most styles of music are hierarchically structured, so the fact that formal grammar theory can be applied to music composition is not a surprise.

4.2.1 L-Systems

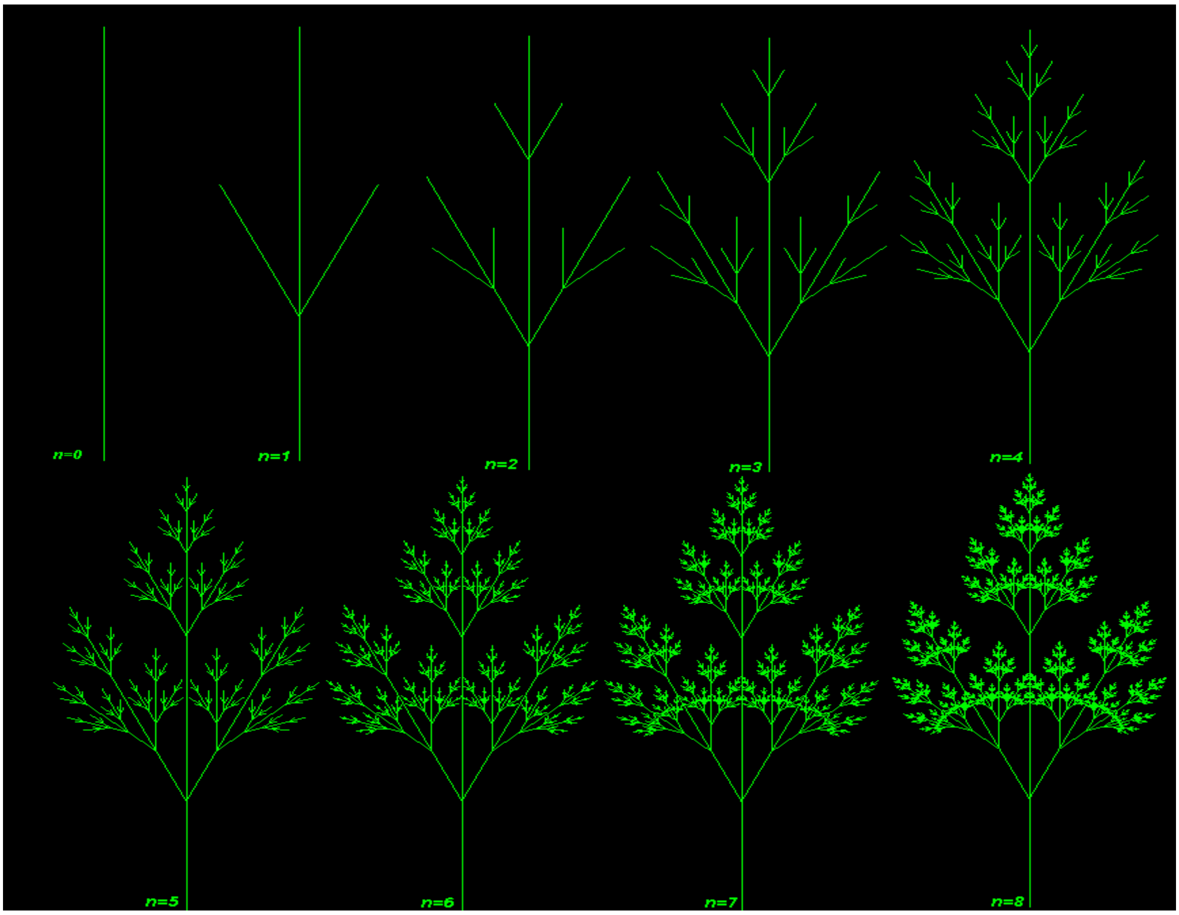
Lindenmayer Systems, also known as L-systems, is a form of grammar, which is most known for its feature called parallel rewriting. This means that in every step of the derivation, the application of all the possible rules happens at once instead of applying one rule per step. L-Systems have been applied successfully in experiments who examine the growth and shape of plants, because their structure is similar to the hierarchical one that these organisms have. This is the main reason why they have become so popular in algorithmic composition as well. Their basic function is string rewriting, which is mainly used for the transformation of an input based on a set of rewriting rules. This can work recursively with a feedback loop. If we successively replace parts of a simple initial object, complex objects can be defined and this makes rewriting a very powerful technique [12].

L-systems generate strings of symbols by repetitively substituting predecessors of given productions by their successors. However, there is an essential difference between L-systems and the other known grammars. In these grammars productions are applied , one at a time, while in the case of L-systems productions are applied concurrently to all symbols in a given string. L-systems were conceived to formally describe the growth process of living organisms and in this context they were extensively studied by biologists and theoretical computer scientist. Another area which L-Systems were used was for the generation of realistic images of plants and trees for computer imagery purposes *(Figure 4.2.1) [12].*

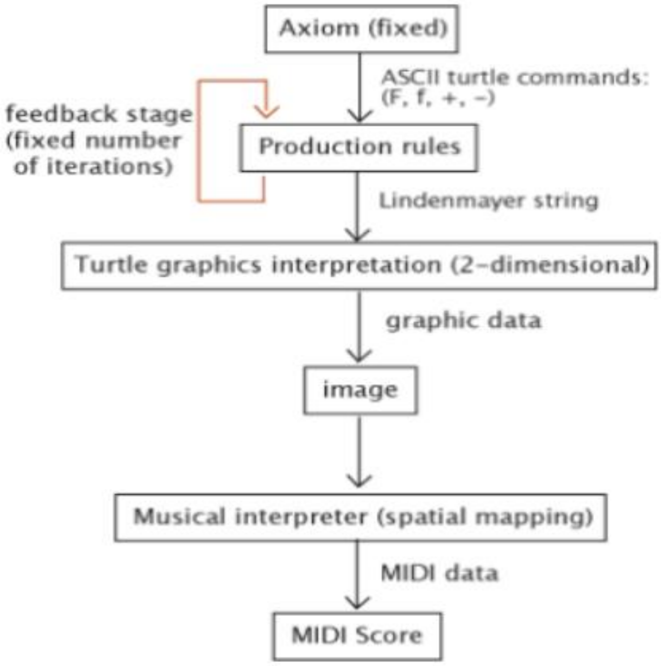
Graphical interpretation of L-Systems

Przemyslaw Prusinkiewicz from Department of Computer Science University of Regina, in a paper written in 1986, presents a technique for generating musical scores with L-

Systems, which consists of 3 steps. “At first, a string of symbols is generated by an L-system. Then, this string is interpreted graphically as a sequence of commands controlling a turtle. A state of the turtle is a triplet (x, y, IX), where the Cartesian coordinates (x, y) represent the turtle's position, and angle IX, called the turtle's heading, is interpreted as the direction in which the turtle is facing.” The system Prusinkiewicz proposes would look the diagram in *Figure 4.2.2*.



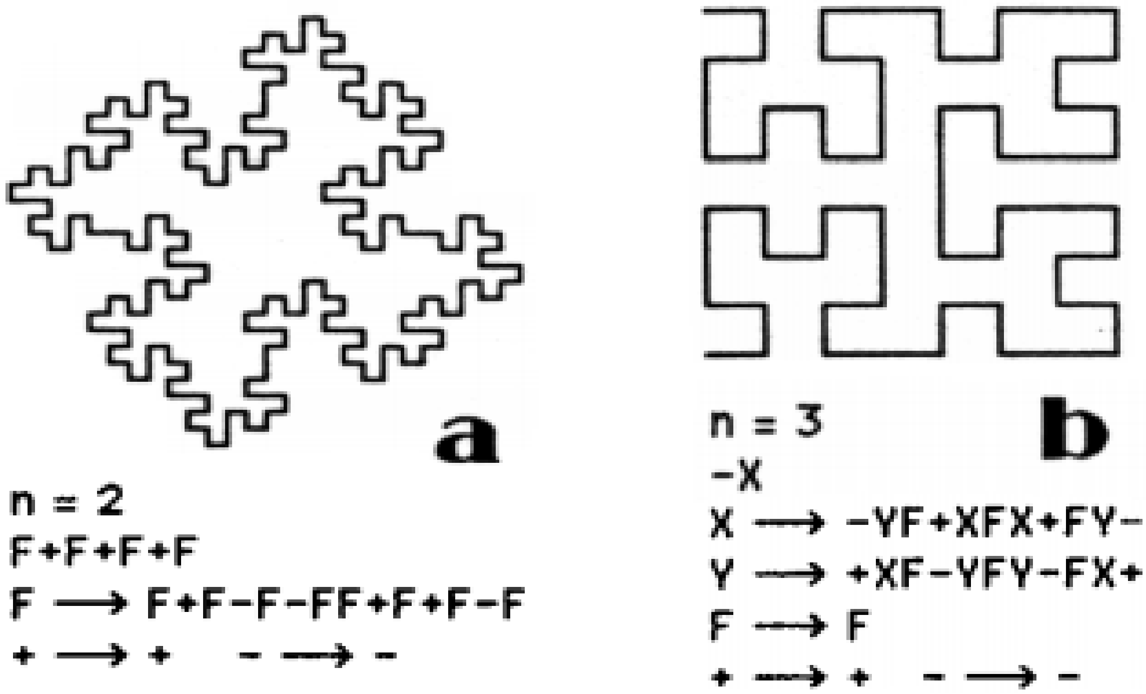
4.2.1 Simulation of plants using L-Systems [14]



4.2.2 Graphical representation of Prusinkiewicz's algorithm [12]

We assume that the initial state of the turtle is a string G, (κ0, λ0, α0), and b, ε are fixed parameters. The orders that the turtle follows (if we name the size of the step as b and the increase of the angle as ε) are the following:

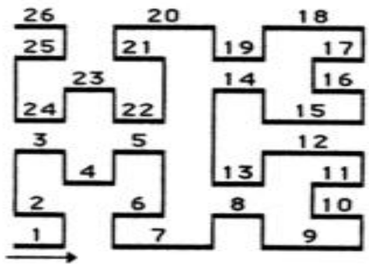
- F Move a step of size b, where the turtle moves to (k2, l2, α), where k2 = k1 + b* cos(ε) and l2 = l1 + b* sin(α) and sketches a line between (k1, l1) and (k2, l2).
- f The same as the step above, but the turtle doesn't sketch a line
- + A right turn of angle ε. The turtle moves to (k1, l1, α+ε). (clockwise rotation is used as positive)
- - A left turn of angle ε. The turtle moves to (k1, y1, α-ε). [13]



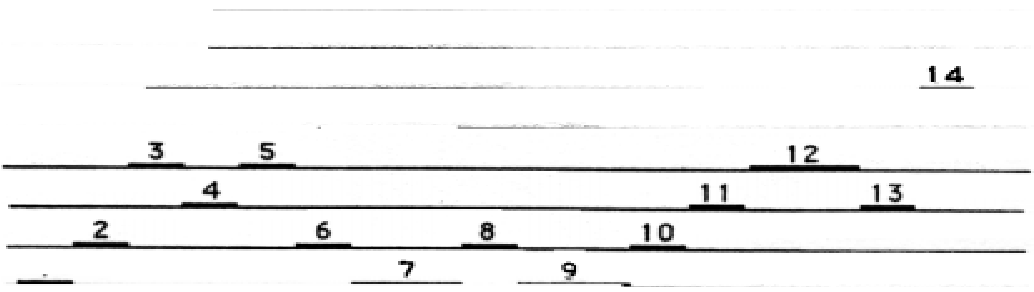
4.2.3 The turtle draws lines following the commands of the programmer [13]

After the turtle is finished, the resulting geometric shape is traced in the order in which it was drawn. Then this figure is interpreted musically, as a sequence of notes with pitch and duration determined by the coordinates of the figure segments.

Suppose that the Hilbert curve is traversed in the direction indicated by the arrow (as shown in Fig 4.2.4) and the consecutive horizontal line segments are interpreted as notes. The pitch of each note corresponds to the y-coordinate of the segment, and the note duration is proportional to the segment length. The resulting sequence of notes forms a simple score shown in *Figures 4.2.5* and *4.2.6.* Any curve consisting of horizontal and vertical segments can be interpreted in a similar way. In Przemyslaw Prusinkiewicz's example, it is assumed that the notes belong to the C major scale and the first note is C. The use of a lookup table is always convenient, which allows the specification of an arbitrary mapping of y coordinates into note pitches. Prusinkiewicz suggests that other parameters, like velocity and tempo, could be controlled as well.



4.2.4 Traversing the Hilbert curve [13]



4.2.5 The score associated with the Hilbert curve in the piano-roll notation [13]

4.2.6 The score associated with the Hilbert curve in common musical notation [13]

The scores generated using the above method, were actually interesting. Despite the simplicity of the underlying productions, they were relatively complex, but they also had a legible internal structure so that they do not make the impression of sounds accidentally put together .[13]

<u>4.2.2 Symbolic, Knowledge-Based Systems</u>

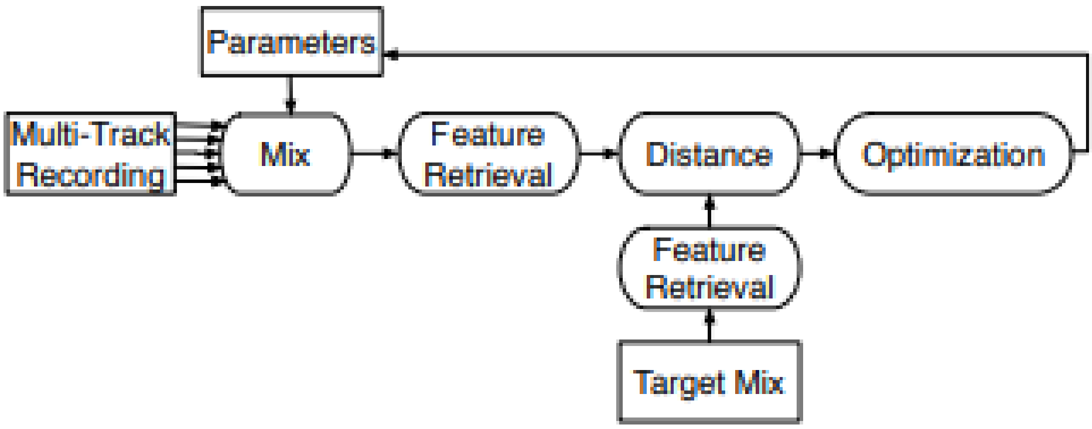
The implementation of algorithmic composition has been achieved with knowledge-based systems, because music theory as we know it is mainly based on rules that have the ability to control musical symbols. While the knowledge implemented in rule-based systems is usually static, some of the knowledge may be dynamically changed. The natural term for this concept is machine learning. As for the advantages of knowledge-based systems they have the ability to consider multiple objectives and find an optimal solution, which fulfills the largest number of targets. As a result, this system is able to consider when and how to break certain sets of rules or allocated targets, and interpret different rules with different levels of priority in parallel. There are also some computational approaches, which may never give a single optimal solution, but are able to propose a series of comparable scored different answers. Similarly, in music production, a single generally accepted optimal mix of a music track cannot exist, but a series of different options that are all considered appropriate of a good mix from some points of view. This is due to the subjective nature of human beings and music production. [15]

Automatic target mixing

In 2009, researchers from University of London developed an algorithm which has the ability to mix songs without any human intervention. Songs that were used in this experiment consisted of many tracks and this is the reason why we refer to them as multi-track recordings. For example if the producers used a guitar, a piano, some drum hats and a flute for the construction of a song, each instrument is considered a distinct track and if all tracks are played simultaneously, we get the result of the whole song being played. For this experiment, the purpose that automatic target mixing had, was to derive the parameters in the mixing of a multi-track recording based on a target mix. The user was able to choose any audio signal as the target, due to some specific qualities like an equalization curve for a specific instrument or even a balance of the amplitude among different tracks [16].

How it works

The recording which consists of many tracks is mixed according to some parameters, and then the extraction of some features from this mix as well as the target mix takes place. After calculating the distance between these features, this distance becomes input to an optimization algorithm. This algorithm then calculates the set of parameters which minimize the distance from the target mix, in order to use this same set to build the estimated mix. The algorithm's purpose is to find what gains to apply to every track which minimizes the Euclidean distance between the spectrum histograms of estimated and target mix [16].

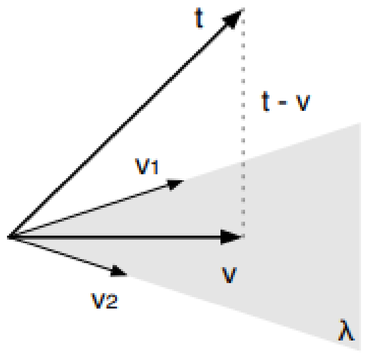


4.2.7 Automatic Target Mixing Framework [16]

The Algorithm

The Algorithm that Daniele Barchiesi and Josh Reiss implemented for this experiment is described below:

Let's assume that the gains applied to each track mi, are the components of the mix that we want to estimate. We are also extracting and comparing a linear feature from the mix and the target, that is a feature U such that: $a1*U(m1) + a2*U(m2) = U(a1*m1 + a2*m2)$ $\forall a \in R$. Assuming that t = U(target mix) is what we have extracted from the target mix and that vi = U(mi) is the feature extracted from the i-th track of the multi-track, t can be designed as a vector in an M-dimensional space (M = size of the feature), as is shown in *Figure 4.2.8.* Any linear combination of the vectors vi with positive coefficients, generates the subspace λ. The number of tracks N, is equal to the dimensionality of λ (in general N << M). The projection of t on the subspace λ is the mix which minimizes the distance from the target.



.

4.2.8 Geometric Representation of Target Mix and Multi-track Recording [16]

The projection vector v can be written as a linear combination of vi with the gains aˆ that we want to retrieve. *Figure 4.2.9 a),* shows this linear combination:

$$v = \sum_{i=0}^{N-1} \hat{\alpha}_i v_i$$

$$A = \left(\begin{bmatrix} \\ v_1 \\ \\ \end{bmatrix} \begin{bmatrix} \\ v_2 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ v_N \\ \\ \end{bmatrix} \right)$$

4.2.9 a) v as a linear combination of the tracks vectors vi with the gains aˆb) A, a matrix whose columns are the vectors vi and v [16]

A is a matrix which has the vectors vi and v as columns and it can be written as: v = Aaˆ, as shown in Figure 4.2.9. Every vector vi is orthogonal to the vector (t- v) and this is why the inner products between (t- v) and every vi is zero: A^T*(t- v) = 0 .If we substitute, we get that A^T*t –A ^T*Aαˆ = 0, which leads to: αˆ = (A ^T*A)^ (−1) * A^T *t (least squares method*)*. We assume that all vectors vi are be linearly independent, which is the only condition for the matrix A^T*A to be invertible.

Target Equalization

Researchers applied different gains to different frequency bands of the tracks. This extended the least squares approach to the equalization problem and had the result of increasing the dimension of λ, choosing pairs of orthogonal vectors wi,j ⊥ wi,k. This is better shown in *Figure 4.2.10* below:

$$\sum_{j=1}^{P} w_{i,j} = v_i$$

4.2.10 Linear combination of pairs of orthogonal vectors wi,j ⊥ wi,k. P is the number of frequency brands [16]

The number of frequency bands is symbolized with P. This algorithm's aim is the estimation of the equalization curve which is applied to the various tracks in the target mix using a constant function defined by multiple sub-functions for every track. Researchers refer to this function as the sub-band estimator. Estimation becomes more accurate and more able to approximate any transfer function as P increases. FIR filters technique is used in this algorithm, which finds the constant values of filters which minimize the Euclidean distance between the estimated mix and the target mix, when applied to the tracks of the multi-track recording. This technique is clearly inspired by linear prediction.

If v is an audio signal, its value v(n) in time n can be calculated as a linear combination of its S previous samples as shown in *Figure 4.2.11* below:

$$\tilde{v}(n) = \sum_{j=0}^{S-1} \alpha_j v(n-j)$$

4.2.11 Audio signal written as a linear combination [16]

What is considered as the goal of this linear predictive model, is the finding of the coefficients a'j that minimizes the squared Euclidean distance between signal and predicted signal. A function J(α) = ||v˜ − v||^ 2 can represent this squared distance. By setting the gradient ∇J(a) to zero , researchers also computed the coefficients a'. This has the result of solving this linear equations' system that is shown in *Figure 4.2.12* below:

$$R_l = \sum_{j=0}^{S-1} \hat{\alpha}_j R_{j-l}$$

,

4.2.12 System of linear equations that needs to be solved [16]

With Rl we refer to the autocorrelation of the signal v and with Rj−l to the j-shifted autocorrelation. An FIR filter is defined by the coefficients α' and is applied to the signal v, leading the researchers to the definition of a new squared distance function so that the target equalization problem can be solved. [16].

Estimation of the FIR coefficients

Researchers continue with the estimation of the FIR coefficients: If target mix is symbolized with t, the length of the target is symbolized with M, N is the number of tracks in the multi-channel track and P is the order of the filter we want to calculate, a squared distance function is defined as shown in *Figure 4.2.13* below:

$$J(\alpha) = ||t - \sum_{i=0}^{N-1} v_i * \alpha_i||^2$$
$$= \sum_{n=0}^{M+P-1} \left[t(n) - \sum_{i=0}^{N-1} \sum_{j=0}^{P-1} \alpha_{ij} v_i(n-j) \right]^2$$

4.2.13 Squared distance function [16]

After that, by computing the partial derivative and then by setting the gradient ∇J(α) to 0, researchers were lead to solve the system of linear equations which is shown in F*igure 4.2.14*.

$$\frac{\partial J(\alpha)}{\partial \alpha_{kl}} = 2 \sum_{n=0}^{M+P-1} \left[t(n) - \sum_{i=0}^{N-1} \sum_{j=0}^{P-1} \alpha_{ij} v_i(n-j) \right] v_k(n-l)$$

$$\sum_{n=0}^{M+P-1} t(n) v_k(n-l) = \sum_{i=0}^{N-1} \sum_{j=0}^{P-1} \hat{\alpha}_{ij} \sum_{n=0}^{M+P-1} v_i(n-j) v_k(n-l)$$

4.2.14 Partial derivative calculation [16]

Correlation Cl(t, vk) between target t and the k-th track vk is represented in the first sum on the left side of the last equation and the shifted correlation Cj−l(vi , vk) between the i-th track and the k-th track is represented in the sum over n on the right side. The above equation can be summarized as shown in *Figure 4.2.15* below:

$$C_l(t, v_k) = \sum_{i=0}^{N-1} \sum_{j=0}^{P-1} \hat{\alpha}_{ij} C_{j-l}(v_i, v_k)$$

$$\forall\, k = 0, \ldots, N-1 \quad l = 0, \ldots, P-1$$

4.2.15 Correlation between (t,vk) and (vi,vk) written as an equation [16]

The correlation Cl(v, w) between two vectors v and w is the inner product between v and the l-shifted version of the vector w. Therefore, researchers built A, a matrix whose columns contained the tracks v as well as the l-shifted tracks as shown in *Figure 4.2.16*:

$$A = \left(\begin{bmatrix} v_1 \\ \\ \\ \\ \end{bmatrix} \begin{bmatrix} \\ v_1 \\ \\ \\ \end{bmatrix} \cdots \begin{bmatrix} \\ \\ \\ v_1 \end{bmatrix} \cdots \begin{bmatrix} v_N \\ \\ \\ \\ \end{bmatrix} \cdots \begin{bmatrix} \\ \\ \\ v_N \end{bmatrix} \right)$$

4.2.16 Matrix A columns contain the tracks v as well as the l-shifted tracks [16]

This equation can be written in a form of a matrix: (A^T*A)*a’ = A^T*t , and the least square method can be used again in order to solve it: α’ = (A^T*A)^ −1*A^T*t .

Daniele Barchiesi and Josh Reiss conclude that the estimation of FIR can be viewed as a generalization of the geometric approach in order to retrieve the gain settings. The norm of the error function J (α) in the time domain is minimized by this least squares estimation. As a result, the distance in the Fourier domain between target and estimated mix will also be minimized. P is the only parameter that we have to choose. Finally, as far as the sub-band estimator is concerned, we understand that the more parameters we have, the more accurate our algorithm is going to become, but its computational cost is also going to increase as well.

4.2.3 Evolutionary and Other Population-Based Methods

Most evolutionary algorithms follow some specific steps. At first, the generation of candidate solutions for the initial set takes place, which can happen from user examples or in more random ways. A fitness function is then used in order to measure the quality of each candidate. The second step is the selection, where by copying candidate solutions from the old set of candidate solutions, a new one is generated. After that, every candidate solution is copied multiple times in proportion to its fitness. The diversity of the population is decreased and restored in this step, with the application of specific operators (e.g. mutation, recombination), made to increase the variation, to a part of the candidate solutions. Due to the fact that the application of the steps happens repeatedly, best and worst fitness have the tendency to gradually increase. A common implementation of a fitness function is the calculation of a weighted sum of the features of composition. What could also be a fitness implementation, is calculating the fitness as the distance to a corpus of compositions or even a target composition, where the fitness of each one of the compositions that exist in a population. For example, if we choose to use pitch for our fitness function, the fitness will be calculated by adding the variation in pitches between a target composition and the pitches of each individual [15].

Can a Genetic Algorithm (GA) mimic the way a composer thinks?

In 2002 Gartland-Jones, implemented an experiment using GAs in order to compose music. Assessing fitness is one of the main problems that needs solving in the GAs area of composition. If we want to determine what we are going to consider as good in a specific genotype, we have to decide what we consider as good generally in music.

The two most known approaches for the assessing of fitness are:

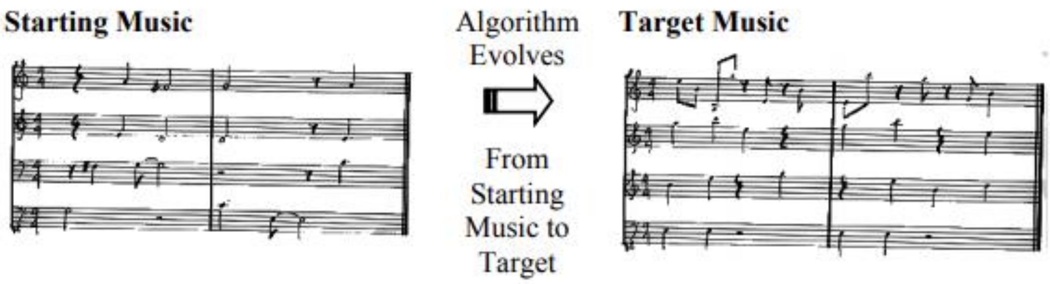
1. The interactive way, by using a human-being to decide which members of the population should take the promotion to the next generation, based on his knowledge. (IGA)
2. The automatic fitness assessment way, by providing the system with enough encoded information, so that it will be able to make the fitness assessment by itself, after every crossover and mutation process. (AFA)

These are the two goals that implementations of composing music with GA try to reach:

1. The creation of original music, which is closer to the composition of music from a human-being, a goal which is mainly creative and subjective.
2. The use of a specific rules-set to compose an output, as well as the examination of what the results of such an output would be. This goal is referring to music theory and music analysis, and this is why is more objective. [17]

The Algorithm

Gartland-Jones decided to create an algorithm, which aim was to evolve from a starting musical piece towards a target musical piece (Figure 4.2.17). This, according to the author, makes the directing of the search as well as the creation of fitness assessment quite simpler. Therefore, the fitness function can be defined as the similarity of an exact genotype to the target piece provided.



4.2.17 Evolving from supplied starting musical selection, to supplied target musical selection [17]

The individual steps that Andrew's Gartland-Jones algorithm follows to reach a specific target music fragment are detailed below:

1. At first a two-bar, four-part MIDI file is used by the algorithm, for the creation of the initial population of identical genotypes.
2. The second step is the performance of mutation and crossover operators to a selected population member, which is selected in turn. Mutation operations have taken the form of methods used in musical composition, like adding and deleting notes, to transposing and reversing. Furthermore, as for selecting new pitches, the operators that are required in such selection are based on a harmonic assessment of the target.
3. Evaluation of fitness of the population member that is being mutated, by comparing the similarity of the genotype to a musical target that is provided to the Algorithm.
4. The fourth step, checks the fitness value of the member that is being mutated. If this value is higher than the lowest fitness that exists in the population, then the low fitness member is replaced and stored as a musical point on the evolutionary path to the target. In any different scenario, it is discarded.
5. If the target isn't reached yet, all steps are repeated. When the Algorithm is done, the musical fragments that it produced are converted to MIDI files that become available for further use. [17]

Fitness Assessment

As mentioned above, the evaluation of fitness of a genotype is made considering the similarity it has with a specific target. Each object that represents a note in the genotype encapsulates its fitness value. This note has a fitness of value 1 if it has the same pitch as the note in the target musical piece which has the same position. Otherwise, the note has a fitness of value 0 . This is also described below:

Musical Section fitness = numSection / numTarget, where:

numSection = num of notes in Musical Section with a Fitness of 1

numTarget = num of notes in the Target

Therefore, we are referring to the notes in the target piece, that have a matching pitches with the notes in the genotype. [17]

The Mutation Operators

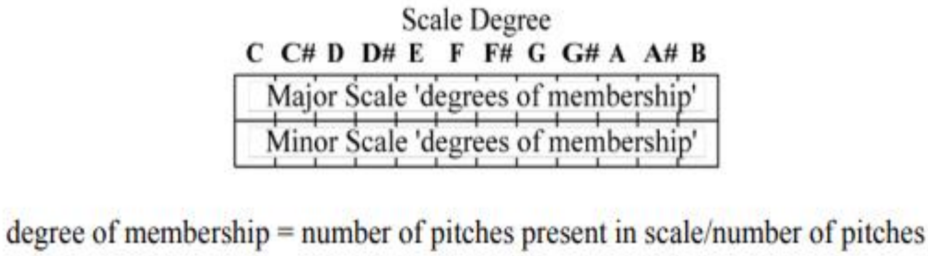
The goal of this Algorithm is not only to reach a target, but also produce interesting results. In order for this to happen, the author used some mutation operators that make it more possible to produce a musically pleasing output. These mutation operators include:

- The addition of a note
- A swap between 2 neighbor notes
- Changing the pitch of a note by an arbitrary interval
- Changing the pitch of a note by an octave
- The mutation of the velocity of a specific note
- The change of the position of a note
- The reversion of some notes between a start point and an end point that are selected in an arbitrary way
- The inversion of some notes between a start point and an end point
- The mutation of the duration of a specific note.
- The deletion of a selected note

These mutations are applied to single notes or to groups of notes, when a user specifies a probability and only when their application will not modify the notes that already have a target fitness value equal to 1, because these notes are not being mutated any more. After every mutation, the algorithm calculates the fitness of every genotype again until it has the value of 1 [17].

<u>*Selecting New Pitches*</u>

At first, the system makes an analysis of the musical target we are trying to reach. This is followed by the construction of a table (*Figure 4.2.18*) where every table location encloses what is the target's degree of membership of that scale type (Minor or Major), for that scale degree (from C to B).



4.2.18 Table representing the degree of membership of the target analyzed [17]

These analysis values, combined with weightings that users supplied, help find the probability as to which scale a new pitch should be taken from when creating a new note or when transposing a pitch within an octave. After the selection of a scale is done, a similar process is used to choose the final pitch within that scale. As a result, this table consists of an analysis of the other notes in the musical section being mutated, instead of the target music being analyzed. This is done for the part that the note is present in, when tending to create melodies from the same scale type, and the chord that this pitch will become part of, when tending to create harmonies from the same scale type. Because of the fact that estimating harmonic analysis is rough, new pitches are allowed to be selected by the user, which are more likely to be met in the target music. At the end, when all mutations are done, the application of a random crossover to the population takes place and this is where population members exchange note patters between each other [17].

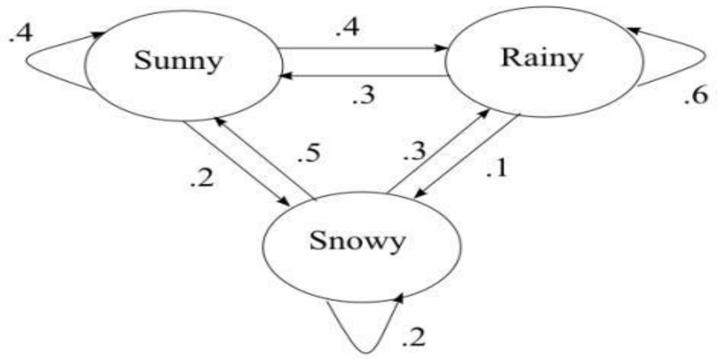
4.3 The Magma Algorithm

MAGMA (Multi-AlGorithmic Music Arranger) is an experimental AI system to compose music, designed by Richard Fox and Adil Khan. In order to generate songs, MAGMA is based on Markov chains, a routine planning algorithm, and a genetic algorithm. The input of the system is user specifications, while the output is a MIDI file. [18]

4.3.1 AI algorithms used in MAGMA

MARKOV CHAINS

A Markov chain is a static model and is defined as a diagram with probabilities in its edges, which represents the transitions from one state to another. *Figure 4.3.1* below, shows a daily weather Markov chain. By looking at this diagram, we can get information about the probabilities of the transition from one state to another.

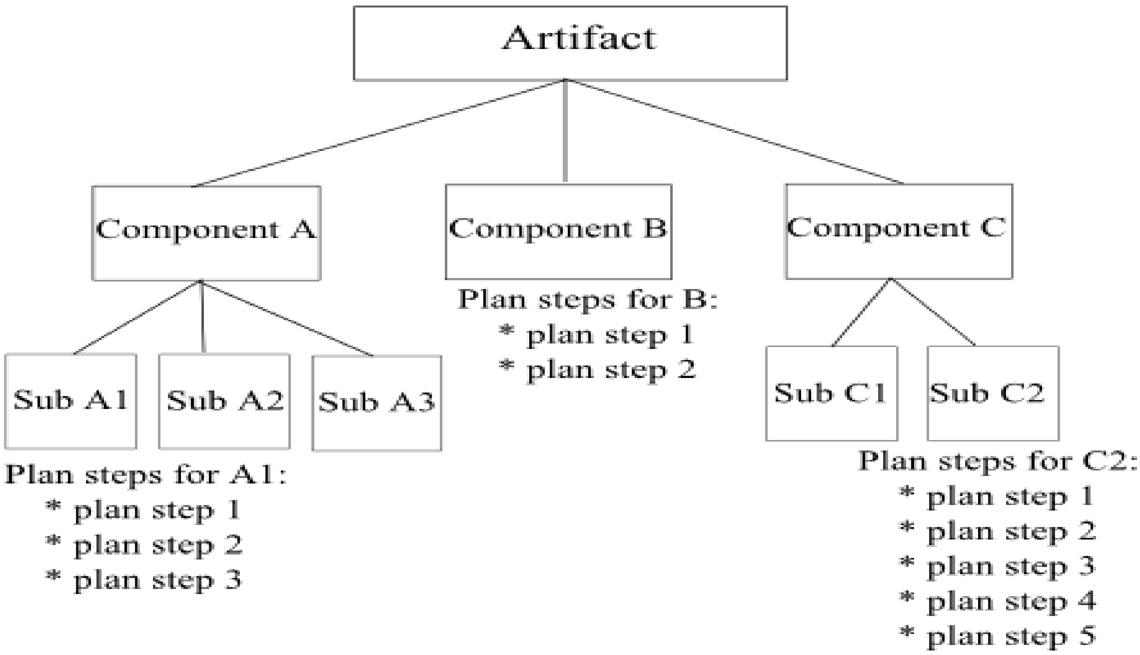


4.3.1 A simple Markov chain of daily weather pattern [4]

Markov chains can be used to analyze the chord changes from a corpus of songs (MIDI input), in order generate a new structure, by finding the frequency in which each sequence of chords occurs. After this, they are able to produce/compose a progression of chords, that is going to contain these sequences in the same relative proportions of occurrence, a procedure which doesn't ensure that the result is going to be aesthetically pleasing. As a result, although composition with the use of Markov chains includes the recognizable elements that were extracted from the analysis of the model, it doesn't have the same aesthetic result with a model composed by a human being *[4].*

Routine planning

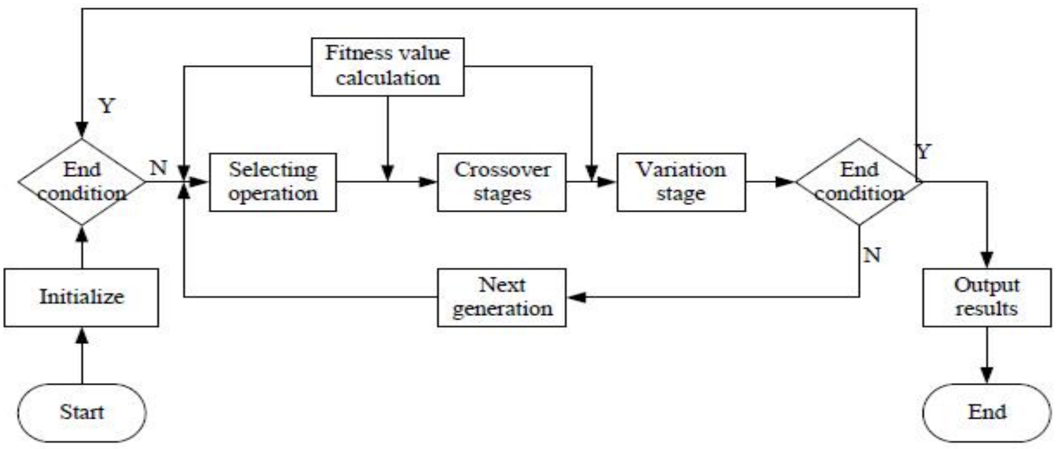
According to Chris Dobrian, routine planning is a knowledge-based approach which mimics the way that an expert with routine knowledge would solve a problem designing an artifact. This routine knowledge is split in categories. At first, the representation of the plan decomposition takes place, which shows that by designing every component and sub-component we can construct a given artifact. Furthermore, for every component that is going to be designed, some plan steps need to be made by the expert. Each component can have many different steps. In order to select the appropriate plan, the expert needs to take into consideration the pattern-matching information/knowledge which identifies the plan step, that has the greatest chance to provide success to the designed object, meeting user specifications, as well as the decisions that were already made on the design of other components.



4.3.2 Routine planning captures the prototypical sequence of problem solving activities that a domain expert might undertake in planning or designing an artifact [4]

Genetic Algorithm

In the genetic algorithm approach, the use of a natural selection is preferred in order to reach a better solution. The way that the GA model works is better described in section 4.2.3 of this thesis.



4.3.3 The genetic algorithm uses a form of natural selection to evolve a better solution [4]

Having discussed the basics of these three algorithms, it is time to proceed as to how the MAGMA algorithm is implemented.

To begin with, the user specifications define the type of song the users will select. The following five preferences are rated on a 5 point scale each.

1. Transition: It determines the size of a transition from one chord or note to the other. Thus, a higher transition would lead to a song with a more inharmonious sound while a small transition could make a song dull.

2. Repetition: It dictates the likelihood of chords/notes to repeat before a transition happends. A more creative a song would include less repetition.

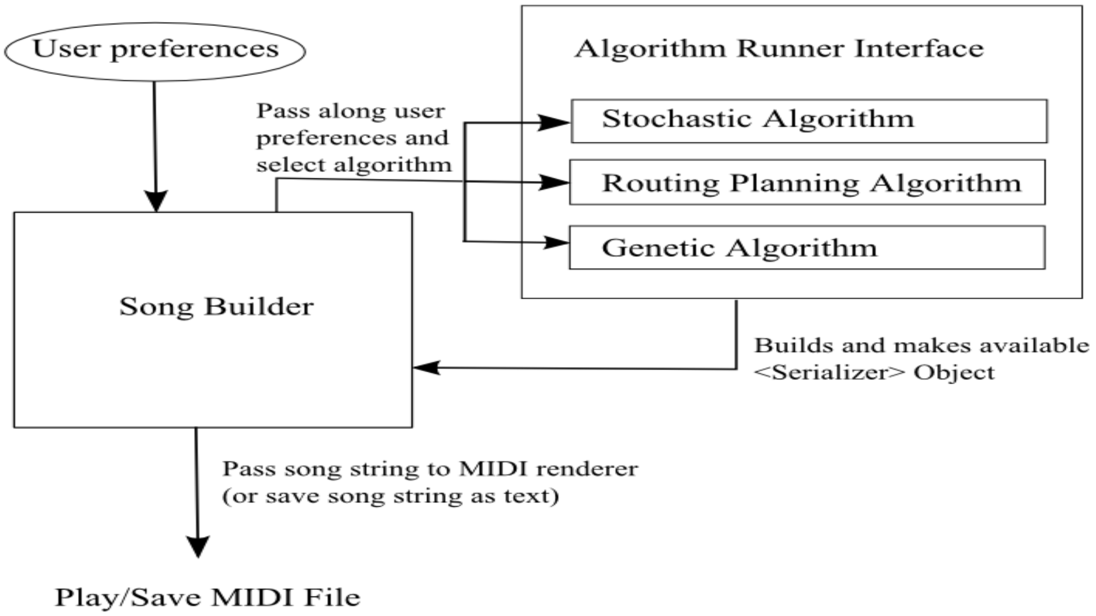
3. Variety: It has an impact on the amount of chords or notes or song components which are generated. A higher variety results to a song with more parts and it can add diversity within its parts. A simple song will usually be the result of lower variety.

4. Range: Range controls the number of octaves in which the chrods/notes belong and the instruments that the algorithm is going to choose for the MIDI file.

5. Mood: If we make the choice of a sad mood the tempo is usually going to be slow. The opposite effect (a faster tempo and a major key) would be the result of an upbeat mood

while an intermediate mood can generate both types of keys for the different components of a song, like the verse or the chorus. Another thing that mood impacts are the instruments being selected.

Eventually, it depends on the user which of the three algorithms to utilize (stochastic approach, planning approach, genetic algorithm approach) *[4].*



4.3.4 The MAGMA Algorithm in a nutshell. User selects a preferred algorithm to proceed. MAGMA then produces an output MIDI file [4]

4.3.2 The MAGMA Algorithm

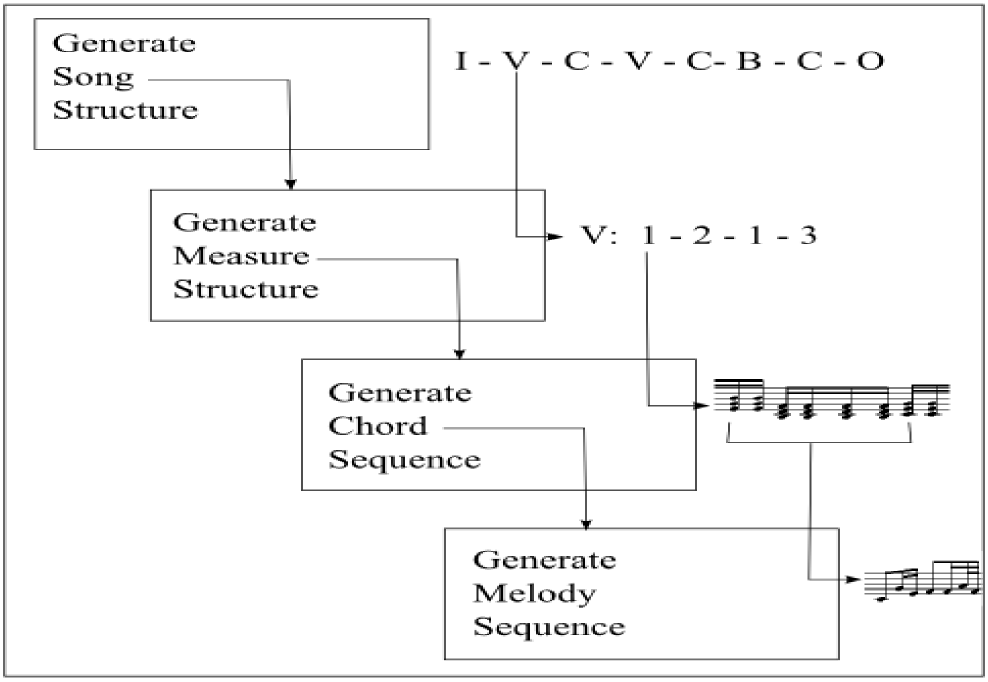
After receiving the user's specifications, the program is ready to apply the algorithms to generate the song by performing the following steps.

Initially, the process involves generating the song's general structure which can be simple or elaborate depending on the order or repetition of its basic components, namely, the introductions symbolized as I, verses symbolized as V, choruses symbolized as C, bridges symbolized as B, solo sections symbolized as S and outros symbolized as O. A plain structure would be I-C-V-C-V-C while a more complicated one would be I-C-V-B-C-V-B-S-C-S-O.

After that, it is time to generate the structure of the separate song components which is associated with the number of measures and the fact that these measures may be the same, or they may switch between various chords/notes. The generation of the chords of every measure follows next. Each measure is made of a number of beats. In a measure which consists of 4 quarter notes or 8 eighth notes or 1 whole note (4/4 timing) , the existence of chords with the same duration or longer and shorter duration will be determined by the variety.

When the chords chapter is completed, it is time for a melody sequence to be generated over the chords, independently of their sequence but in accordance to the chord sequence's duration and the song's key.

Although each of the three algorithms (stochastic, planning, genetic algorithm) performs the above four steps in different ways, they all follow the same order of performance, as it can be shown in *Figure 4.3.5* below.



4.3.5 The steps that all three algorithms perform [4]

The *stochastic algorithm* uses Markov chains for every of its four steps, each of which was generated using a plain algorithm from a corpus of songs. The parser provides the Markov chains as transition probabilities matrixes (e.g., chord transition probabilities), as shown in *Figure 4.3.6*. MAGMA uses four transition probabilities matrixes in total, one for the song's structure, one for the measure structure, one for the generation of the chords and one for the creation of the melody. For every song that is produced, the probabilities of the matrixes can change depending on the specifications of the user in order to suit the user’s preferences *[4].*

	-7	-6	-5	. . .	12	13	14
-7	.11	.21	.31	. . . .	.33	.18	.17
-6	.1	.32	.39	. . . .	.43	.28	.67
-5	.7	.2	.9	. . . .	.25	.46	.72
.	.	.	.		.	.	.
.	.	.	.		.	.	.
.	.	.	.		.	.	.
12	.11	.13	.2	. . . .	.6	.23	.17
13	.4	.14	.82	. . . .	.1	.2	.11
14	.19	.11	.5	. . . .	.9	.21	.16

4.3.6 Markov chains as transition probability matrixes, provided by the parser [4]

The *planning approach* involves plan steps in order to define specific sequences of the song’s structure: measures, chord sequences and note sequences all of which are selected in a way that matches the user specifications as closely as possible.

The *Genetic Algorithm* approach includes four types of chromosomes for the structure of the song and its contents, as well as the sequences of the chords/notes. The approach makes use of five fitness functions, one for each type of user preference, which are combined using a weighted average. At this stage, each member of the final chromosome represents a distinct measure for the song component. The generation of a chord sequence for a given measure is the next stage and it is also influenced by the user in terms of repetition, variety and transition preference. a chord sequence repetition and variety will also affect the size of the chromosome. The same factors influence the chromosome composing the melody in a chord sequence on the next stage. The melody that is generated might be trimmed in order to fit the duration of the chords in the measure, for every measure. Chromosomes take values between 0 and 1 in every phase and in order for this number to be produced, the algorithm takes into consideration the user’s preferences. The use of Nashville notation helps with the required calculations, where negative numbers are used to represent lower octaves while numbers greater than seven represent higher octaves.

The initial population for each section of song generation is created in a random order by the genetic algorithm and it consists of 10 chromosomes. 4 parents are chosen from the algorithm's starting population, and splits them in 2 parts, with the one having a higher fitness level than the other. This leads to the generation of 6 children by crossover, and the generation of 4 more children by mutation. This is usually repeated 12-15 times as far as the song structure and components are concerned, and 100 times for the production of the chords/melody. [*4].*

KEY	1	2	3	4	5	6	7
A	A	Bm	C#m	D	E	F#m	G#dim
Bb	Bb	Cm	Dm	Eb	F	Gm	Adim
B	B	C#m	D#m	E	F#	G#m	A#dim
C	C	Dm	Em	F	G	Am	Bdim
Db	Db	Ebm	Fm	Gb	Ab	Bbm	Cdim
D	D	Em	F#m	G	A	Bm	C#dim
Eb	Eb	Fm	Gm	Ab	Bb	Cm	Ddim
E	E	F#m	G#m	A	B	C#m	D#dim
F	F	Gm	Am	Bb	C	Dm	Edim
Gb	Gb	Abm	Bbm	Cb	Db	Ebm	Fdim
G	G	Am	Bm	C	D	Em	F#dim
Ab	Ab	Bbm	Cm	Db	Eb	Fm	Gdim

4.3.7 The Nashville Number System [19]

4.3.3 MAGMA Examples

These are instances of what the MAGMA algorithm generated for *song structure*, according to some specific user preferences. R symbolizes repetition, and V is used to symbolize variety.

- {R=2,V=3}=I|V|C|V|C|O
- {R=2,V=4}=I|V|C|V|C|B|C|O
- {R=1,V=5}=I|V|C|B|O

As for the generation of the *chords and the melody*, the algorithm deals with the pitch and the rhythm. In the examples that follow, numbers are used to symbolize which chords are being played. For example, if our key is D a 3 is an F chord and 6 is an B chord, while 0 is a rest. The duration of the chord/note that is being played is symbolized with a letter following the (w = whole note, h = half note, q = quarter note, i = eighth note, s = sixteenth note). The three lists below are examples of chord sequences and melody sequences as well.

- {R=1,T=1,V=2,H=3}=4q|4q|5q|5q
- {R=1,T=1,V=2,H=2}=5h|6h
- {R=1,T=1,V=1,H=1}=5w

The algorithm uses 3 more procedures, so that the match of the melody with the chords is guaranteed.

1. The first one is responsible for trimming the melody according to the duration of the measure in the chord sequence phase.
2. The second one makes sure that the rules of music theory are applied and this is why its mechanism matches the melody that is generated, to the key of every song component. For instance, a melody that is being played in a verse with a key of B must agree to this key.

3. The third procedure is responsible for transitioning from one song component to another, by modifying the sequence of chords that are involved in this transition. An example would be a transition from a verse to chorus.

CHAPTER 5

CONCLUSIONS

To conclude, it seems that Algorithmic composition can imitate, to some degree, the way that the human brain perceives and processes music. Researchers all over the world have accomplished to automate various music composition tasks mainly by applying two approaches. According to the first approach, music can be generated by imitating a corpus of compositions of a specific style. This approach has been tackled with many different methods, which have frequently turned out to be reasonably successful. The second approach refers to the automation of composition tasks to varying degrees, from designing a base for human composers, to generating compositions without human intervention. The latter approach needs further improvement since the results produced by computers and algorithms have sparked controversy. This is due to the fact that the concept of artistic creativity eludes a formal and effective definition that can be widely acknowledged, which means that the evaluation of these systems can scarcely be achieved in a precise way. Although it is still debatable whether the point at which a computational system may become truly creative , it is easy to see that the amount of systems capable of independent creativity will increase in the future. This should not be seen as another case of computers replacing humans in a sophisticated activity, but as an opportunity for human artists, because if computers reach a point of composing human-like music, this will enrich their catalogue with new songs and therefore it will enhance their own creative music compositions and innovative ideas.

BIBLIOGRAPHY

[1] Iann Cross, "Music, Cognition, Culture, and Evolution, 2001", Faculty of Music, University of Cambridge, Cambridge CB3 9DP, United Kingdom, pp. 28-42.

[2] Charles Darwin, "Sexual Selection and the Descent of Man: The Darwinian Pivot, 1971", p. 73.

[3] "Music psychology", 2-2-2021, https://en.wikipedia.org/wiki/Music_psychology

[4] Chris Dobrian, "Music and Artificial Intelligence, 2002", University of California, Irvine, pp. 2-10.

[5] Timothy C. Justus and Jamshed J. Bharucha, "Music Perception and Cognition", CHAPTER 11: pp. 453-480.

[6] "Equal tempered frequencies", 5-12-2020, http://hyperphysics.phy-astr.gsu.edu/

[7] "Examples of Meter", 2-1-2021, http://www.Vexflow.com/

[8]"Tempo and the beat",3-1-2021,https://www.musical-u.com/modules/rhythm/tempo-and-the-beat/

[9] "How to Quantize in FL STUDIO", 7-1-2021, ***https://www.synapticsound.com/quantize-fl-studio/***

[10] Jacqueline A. Jones, Benjamin O. Miller, and Don L. Scarborough," Behavior Research Methods, Instruments, & Computers , 1988",Brooklyn College of the City University of New York, Brooklyn, New York, pp. 255-262.

[11] Povel, D.-J., & Essens, P, "Perception of temporal pattern, 1985", https://online.ucpress.edu/, pp. 411-440.

[12] Stelios Manousakis,"Musical L-Systems, 2006", Master's Thesis - Sonology The Royal Conservatory, The Hague, pp. 53-55.

[13] Przemyslaw Prusinkiewicz,"Score generation with L−systems",Proceedings of the 1986 International Computer Music Conference, pp. 455−457.

[14] "L - SYSTEMS" ,8-12-2020,http://progsystem.free.fr/

[15] Jose David Fernández, Francisco Vico, "AI Methods in Algorithmic Composition: A Comprehensive Survey 2013", Universidad de Málaga, Calle Severo Ochoa, pp. 513-582.

[16] Daniele Barchiesi, Josh Reiss, "Automatic target mixing using least-squares optimization of gains and equalization settings", Queen Mary, University of London United Kingdom, pp. 513-582.

[17] Andrew Gartland-Jones, "Can a Genetic Algorithm Think Like a Composer?", Computational Creativity Research Group, The University of Sussex, Brighton, UK, pp. 141 – 151.

[18] Richard Fox, Adil Khan, "Artificial Intelligence Approaches to Music Composition", Department of Computer Science Northern Kentucky University, Highland Heights, pp. 1-7

[19] “The Nashville number system” , 8-12-2020, http://www.elim.org.uk/