Evolutionary Music Composition for Digital Games Using Regent-Dependent Creativity Metric

Herbert Alves Batista[1]* Luís Fabrício Wanderley Góes[1]† Celso França[1]‡
Wendel Cássio Alves Batista[2]§

[1]Pontifícia Universidade Católica de Minas Gerais, Instituto de Ciências Exatas e Informática, Brasil
[2]Universidade Federal de Minas Gerais, Instituto de Geociências, Brasil

Abstract

In recent years, the Computational Creativity field has been boosted significantly through researches in human abilities related themes, especially artistic abilities. One of the greatest challenges in working on this field is that the created artifacts are hard to evaluate. In this work is presented a creative music compositor for digital games that explores combinational and explorational creativity. The method is based on Genetic Algorithm and uses the Regent-Dependent Creativity metric to evaluate the generated artifacts. The results demonstrate that some melodies generated by the system are considered creative also by human evaluators.

Keywords: computational creativity, regent-dependent creativity metric, evolutionary music generation, digital games melody composition.

1 Introduction

Since its birth, the video game industry has grown rapidly. Along with it has grown the digital games music industry, starting from simple bleeps originated by transistors and wires to fully orchestrated compositions with long hours [1]. Such growth has made room for research into the computational generation of music for digital games.

The creation of computational music or Algorithmic Composition (AC) is a complex activity that involves aspects of other disciplines such as psychology and acoustics. Making music requires creativity, technical knowledge and some manual skills, making it a difficult task, even for humans [2]. In addition, music involves combinations of elements whose search space is quite extensive and whose relations are quite complex and heterogeneous, which makes it an excellent topic of research in the area of Computational Creativity, involving the study of human emotions, intelligence and creativity.

Among the most used approaches in the production of computational music are those that involve techniques of Evolutionary Computation. The use of Evolutionary Computing in Algorithmic Composition, in general, presents good results. Many studies in this area use Interactive Evolutionary Computation (IEC). This method uses human evaluation in part or in the whole evaluation process of the artifacts generated, however, the problem of using human evaluation in this process is that it creates a bottleneck. A user needs to hear and focus on the sound details, making the method expensive [3].

This work presents an automatic generator of melodies that uses a genetic algorithm in the construction of the artifacts. To avoid the bottleneck caused by human evaluation, the Regent-Dependent Creativity (RDC) metric was used in the evaluation step. The RDC consists of an independent domain metric for generating and evaluating artifact creativity. The musical style of the melodies was extracted from the dataset using a 4Th order Markovian Model [4]. The experimental project demonstrated that the population size and mutation probability are the most influential parameters in the performance of the genetic algorithm. The system proved to be able to generate melodies with quality, according to the research done through a questionnaire in which six short melodies generated by the system, each containing fifty sound or pause events, were presented to the interviewees for evaluation.

The work is organized in the following way: in sections 2 and 3 the theoretical reference necessary to the development of the work is presented; In section 4 the related works are presented; Section 5 describes the methods and tools used in the development of the work; Section 6 details the tests performed and the results obtained; Section 7 presents the conclusion of the paper and suggests some approaches that can be explored in the future.

2 Computational Creativity

Although the definition of creativity is still a subject under discussion, one widely accepted is that creativity is the ability to produce ideas or artifacts that are new and valuable. The term "valuable" can be understood by many ways, according to the context, like: interesting, beautiful, simple, richly complex, etc. The term "new" has two categories: psychological and historic [5] [6].

An idea or artifact is considered P-creative (psychological novelty) when it is new only for its creator, even if something identical has been generated before, on the other hand, it is said H-creative (historic novelty) when never before has been generated something identical [5].

New ideas or artifacts can be generated by means of combination, exploration or transformation. Combinational creativity is the technique of producing new ideas from associations of elements belonging to existent ideas. Exploratory creativity generate new ideas through the exploration of the search space. Several places are visited, including the non explored ones, trying to find the boundaries and potential of it. In the transformational creativity, the space itself is transformed through the changing or removal of one or more of its dimensions. In that approach there is the possibility to generate ideas that were not even possible before the change [5].

Therefore, the Computational Creativity is the Artificial Intelligence sub-field that studies the computational systems creation that, under the evaluation of an impartial observer, shows creative behavior to accomplish specific tasks [7].

3 Evolutionary Composition

3.1 Algorithmic Composition

The Algorithmic Composition (AC) is the field that studies the generation of fully or partially automated music by means of computational systems. Basically, it can be described as a set of rules that

*e-mail: herbert@pucminas.br
†e-mail: lfwgoes@pucminas.br
‡e-mail: celso@pucminas.br
§e-mail: wendel@ufmg.br

define how to accomplish the task of combining musical parts in a full composition [8].

The AC involves several activities such as melody, rhythm, harmony, counterpoint writing, arrangement and notation. Those activities not necessarily need to be applied in every kind of music, but they are a reasonable start point [9].

The range of methodological approaches to implements AC is remarkably wide, including different methods of Artificial Intelligence and also mathematics models of Complex Systems and Artificial Life, like: Grammars, Knowledge Based Systems, Markov Chains, Artificial Neural Networks, Evolutionary Computation and Cellular Automaton [9].

3.2 Evolutionary Computation

The Evolutionary Computation incorporates a wide set of algorithms inspired on the Charles Darwin's Theory that defines "Natural Selection or Survival of the Most Fitted as the preservation of the differences and individual variations that are favorable and the destruction of those that are harmful" [10].

In the nature, the individuals must be adapted to their environments to survive in a process called evolution. In that process the characteristics that give them advantage in the competition for survival are preserved when they reproduce, and those that turn them weak are eliminated. Those characteristics are carried by units called genes. After several generations of individuals, the genes also suffer selective pressure and those that carry the most favourable characteristics are preserved during the process of sexual recombination [11].

In an Evolutionary Algorithm (EA), the goal is to simulate the natural evolutionary process using a computer. Some basic components are necessary to implement an EA [11].

1. A represatation of a candidate solution.
2. A procedure to create a population of candidate solutions.
3. A fitness function that will perform the environment role, evaluating the candidates according to their fitness.
4. A selection procedure that will choose the candidate solutions that will give rise to the next generation of candidate solutions.
5. Evolutionary Operators that change the composition of the candidate solutions (usually mutation and crossover).
6. Parameter values like (population size, number of selected individuals, probability of application of the evolutionary operators, etc.).

The EA proceeds in a common mode: a set of candidate solutions that undergo repeated cycles of evaluation, selection and reproduction. Each candidate solution is evaluated by a fitness function, which is a metric in charge of quantify the quality of that solution. The best candidates are chosen in the selection phase and they will give rise to a new population of candidate solutions [9].

Evolutionary Computing applications include solving difficult optimization problems, designing digital circuits, creating new computer programs, and various other problems generally designated for humans [12].

3.3 Evolutionary Musical Composition

Evolutionary Musical Composition (EMC) is the task of Algorithmic Composition through the application of Evolutionary Computation.

In a EMC system, the chromosome is usually binary (Genetic Algorithm) or Tree-based (Genetic Programming) and can represent musical information in form of notes, frequencies or events. In the binary model, the events, notes or frequencies are represented by a sequence of bits, in which each bit or set of bits (gene) represent a music characteristic. With the same approach, although with more representation power, the bit sequence can be replaced by a vector of integer or real numbers. In the tree-based model, the leaf nodes represent the events, like pitch, duration, etc. and the internal nodes represent operations as concatenation, repetition, etc. [12].

The model that will represent the music in the system should denote at least the pitch and beginning and duration time of each note [4].

The tonal height can basically be represented in three ways: absolute, relative or with scale shift. Absolute mode is the simplest, a number is used to represent each note. In the relative mode the notes are represented in relation to a reference note, thus only the pitch difference between the note in question and the reference note is specified. In the model with scale shift, harmony can be used to determine a code for each note, so it is defined that the scales will always be respected, for example if we are using the Major Scale and the harmony in question is a C, each note of the C Major Scale will receive a number that will represent it [4].

The most common way to represent the beginning and duration time of each note is through the rhythm, that is, the time set in relation to the beats of the song. A set of pulses is also called a musical measure. Another, but less usual, approach is the absolute. In this mode the time set to actual measurements, such as milliseconds [4].

The creation of new individuals is done by the genetic operators that allow not only that the initial individuals are tested, but that, in fact, there is a process of evolution. The most common operators are the crossing over and mutation operators, but there may be others depending on the application area. These operators can be blind or guided, that is, produce changes completely randomly or not. The guided model is quite useful in the musical domain, where completely random changes can produce breaks in melodic or harmonic rules, which is not desired [4].

In the traditional crossover operator, a cutoff point is randomly chosen and the parent pieces are combined into new elements. Generally, in evolutionary music, the cutoff operates at the note level, since bit-wise cut can produce new notes that are not worth considering or do not generate significant change in the individual [4].

The mutation operator is used so that more options in the search space can be explored and is usually performed with a simple change of a gene. In evolutionary music, this operator rarely works, mainly because it generates very large horizontal displacements in the melody [4].

In the mutation phase there is the possibility of using other operators that make changes in the individuals. Because it is a complex coding problem, evolutionary music has some specific operators that can be introduced, for example: reversing the positions of the notes of a session randomly; Changing a place session or exchange a place session with another; Adding extra copies of genes into other parts of the composition; Erasing parts of the melody or turning them into pause; Changing the pitch of some or all notes; Moving the position of the notes in a measure; Sorting a note session by tonal height, among others [4].

3.4 Markov Models

Markov models use a data analysis to generate a sequence of states and transition probabilities. In a Markov chain, the probability of a future state X_{t+1} depends on the state X_t. Markov chains can be represented by graphs or transition matrices. When more than one past event is used to determine the probability of a future event, Markov chains of higher orders are given, where the order indicates the number of past events considered. Thus, in a model based on the notes of a dataset of melodies, the larger the order of the model, the more it approaches the original structure of the melodies in the base [4].

Typically, Markov Models are used for music style copying, but can also be used for genuine composite applications [4].

3.5 Related Works

There are several works using Evolutionary Computing to perform the Algorithmic Composition task, some of them are presented below.

In [13] a Genetic Algorithm to evolve Jazz solos from MIDI sequences was used. The representation model used involves two populations, one population of phrases and another of compass. Each individual in the compass population maps a sequence of MIDI events through a 32-bit string. Each 4 bits represent an event corresponding to an integer in the range 0-15, where 0 represents pause, 15 represents maintenance, and 1-14 represents notes. Each individual in the phrase population maps indexes in the compass population. The selection of the individuals is made from a modified process of selection by tournament. Four individuals are randomly chosen to form a family. Of the four individuals in a family, the two with the highest fitness are used as parents, and the other two are replaced by the children of those parents. The final evaluation of the individuals created is made by an element called mentor (person) who basically hears and responds to the system whether it is good or bad. The author concluded that his work demonstrated that genetic algorithms can be used for algorithmic composition and recognized that the human evaluator is a performance bottleneck that greatly limits population size.

In [14] a Genetic Algorithm to generate melodies of a given musical style was used. As a representation of individuals was used a vector with N pairs of integers, each pair representing an event. The model requires a dataset with previously defined songs of a particular style. The fitness of each candidate solution was measured through the Normalized Compression Distance (LZ77 Algorithm) of the melody collection. The authors concluded that the LZ77 Algorithm is a promising tool to work with Genetic Algorithm when the intention is to generate elements similar to a desired target.

In [15] an Interactive Genetic Algorithm (IGA) was used to generate musical chord progression suited to the user's intention. It was used a vector of integers where each integer maps a chord (consisting of 3 notes) as the representation model . The evaluation of the individuals was made from two different experiments. In the first experiment, four users evaluate artifacts on a 5-point scale. Each user participates twice, each time with a goal: dark or light which, basically, are chord progression styles in Minor and Major scale, respectively. In the second experiment, the objective was to demonstrate the effectiveness of the system comparing the best individuals produced in the first experiment. The author concluded that the IGA proved to be suitable for the generation of chord progression, nevertheless recognized the need for improvements in the system to present a significant performance. In addition, he pointed out that assessing chord progressions by users is a difficult task having heard each, just once.

In [3] a Grammatical Evolution was used as a method for generating melodies on the piano. The system employs a context-free grammar as a model for representing individuals. The initial production of the grammar is one or more events, allowing recursive calls. Each event produces a concatenated sequence (style, octave, pitch, duration), where the style can produce a single note, a chord, an arpeggio, or a sequence of ascending or descending notes. An octave produces the region of the piano where the event occurs. A pitch produces a value between 0 and 11 that maps a note in the chromatic scale, and the duration produces a value that will correspond to the time that the event will last. To evaluate the melodies created, a tone-guided method was implemented, in which all instances of each note in the melody are counted. The fitness value is given by the difference between the number of instances of each note in the melody, the greater the difference between the note with more instances and the second note with more instances, and the more concentrated in a smaller number of notes are the notes of the melody, the better the fitness. The authors confess in their conclusion that the evaluation function used is a statistical tonal validation measure, rather than a measure of creativity or that it proves to be a good melody.

This work is based in the music representation techniques, as well in the models of reproduction, mutation and individuals selection widely used, however uses a fully automated approach in the assessment of the individuals fitness through the RDC metric, in order to avoid the bottleneck caused by a human assessment and trying to assure that the generated artifacts are creative.

4 Methodology

This chapter is divided into three parts. First, the definitions and delimitations referring to the scope of the study object of this work are established. Then, the genetic algorithm is detailed with each of its components, and finally, the experimental method used to determine the impact of each of the parameters of the genetic algorithm on the results is explained.

4.1 Specifications and Settings

The decisions that were taken to delimit and specify the object of study of this work are presented below.

1. It was decided that this work would deal with monophonic music (melody).

2. The interval of musical notes used comprises three octaves, which are the middle C or C4 (it is the central C in a piano), named in the system as the 'M' octave, one octave below, named as 'L', and one octave above, designated as 'H'. Figure 1 shows the beginning and end of each octave in a sheet music by disposing the notes C3 and B3, C4 and B4, and C5 and B5, respectively.

3. The translations of the genotypes into phenotypes and the phenotypes into music, used only to illustrate the created objects, are made in sheet music using treble clef and quadruple time.

4. The size of the songs has been standardized by the number of events, that is to say that in terms of real time (seconds), the produced songs may be shorter or longer, since there are events with different durations.

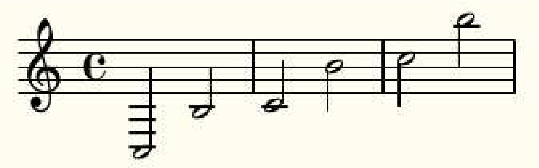


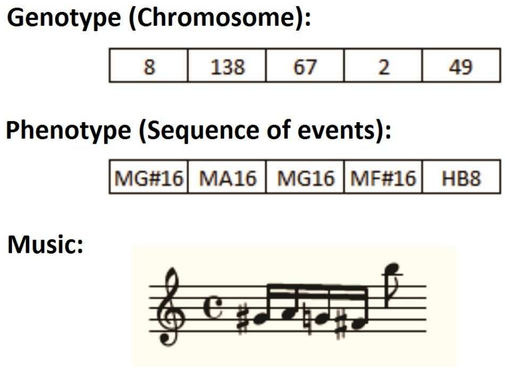
Figure 1: The interval of musical notes used comprises three octaves, which are the middle C, one octave below and one octave above.

The possible events to be generated correspond, then to the combinations of an octave, a note and a duration time. The octave, as explained above, may be L, M, or H, the note may be one of the notes of the chromatic scale: C, C#, D, D#, E, F, F#, G, G#, A, A# e B, and the duration time can be 1, which is equivalent to the time of 1 measure, 2, $\frac{1}{2}$ measure, 4, $\frac{1}{4}$ measure, 8, $\frac{1}{8}$ measure, 16, $\frac{1}{16}$ measure, 32, $\frac{1}{32}$ measure and 64, $\frac{1}{64}$ measure. Besides, it is possible to have pause events, represented by the symbol R, which can also be concatenated with all the duration times. Thus, 252 different types of sound events and 7 pause events are possible, totaling 259 different types of events. As an illustration, an example of an event can be "MF8", which corresponds to the note F played in the middle octave for $\frac{1}{8}$ of measure.

4.2 Genetic Algorithm

As an aid in the implementation of the Genetic Algorithm the Watchmaker Framework [16] was used. It is an extensible, high-performance, object-oriented framework for implementing evolutionary algorithms in the Java language.

Regarding the representation model, each candidate is represented in the genetic algorithm by a vector of integers with N positions (chromosome). Each position of the vector (gene) receives a value in the interval [0-258] that maps a complete event, with indication of the octave, the note and the duration of the note. Figure 2 shows an example of an individual with N = 5 and their representations in events and music.



Genotype (Chromosome):

8	138	67	2	49

Phenotype (Sequence of events):

MG#16	MA16	MG16	MF#16	HB8

Figure 2: Genotype, phenotype and music note representation that can be translated into a sequence of events and interpreted in a song.

The generation of the initial population is done in a random manner. The vector size is pre-defined, so each position is populated with a random integer in the configured range. Then a draw is made to define which event each integer will map to. This map is stored so that it is possible to translate the vector into a sequence of events.

The evaluation of the individuals is performed by the RDC metric. RDC is a metric presented by [17] that evaluates artifact creativity. Because it is an independent domain metric, it is necessary to make adjustments to make the model work. In order to be used, it requires that the artifacts be described in a Regent-Dependent Model, in which the artifact characteristics are represented as pairs of dependencies, in addition, it requires that the relations between different characteristics be informed. To measure creativity, RDC considers two aspects in the artifact: novelty and value. The novelty is calculated through the Bayesian Surprise metric, which allows us to calculate how new is an artifact compared to those already existing in the dataset. The value is calculated from the synergy of the artifact. The artifact is modeled as a graph, in which each characteristic is a node and the edges are inserted between two or more characteristics that work well together. The synergy is then calculated through metrics that evaluate the relationship between vertices in a graph, especially connectivity and density.

In order to meet the RDC need for the representation mode, it was defined that the regent element is equivalent to the order in which the events occur, and that the dependent element is equivalent to the event itself, that is, each dependency pair is represented by a sequence number and an event, for instance: P(regent, dependent) = $P_1(1; HB2)$, the first event in the song is a B played in the High octave for $\frac{1}{2}$ compass.

To calculate the novelty, the music is converted into a vector of real numbers of length equal to its number of events. In the initial position all artifacts are given a value of 0, and in the following positions, the value is equivalent to the pitch difference between the note corresponding to that position and the initial note normalized in the interval [0, 1]. The normalization was done from the division of the value referring to the tonal difference found by the maximum possible value, in the case of this work the maximum value corresponds to the tonal difference between the lowest note (Low C) and the highest note (High B). This strategy was adopted to avoid that a song transposed from another already existing song (written with another reference note, but with the same intervals between the notes) be considered new. The RDC calculates the novelty through the Bayesian Surprise metric. This metric measures the degree of surprise generated by a new artifact from the modification it causes in the dataset variance.

With regard to the relationships between events, it is very difficult to precisely define which events work well together, especially in relation to rhythm. There are many different styles, many cultures and musical tastes. What looks good to some may not seem so good to others. Therefore it was decided to extract this information from the used dataset, in this way, it is expected that the system will produce songs of similar quality to those in the dataset.

The strategy used to extract information about the relationships between music elements from the dataset is supported on the Markov Models and considers that the next event depends on the current event, for example: since the current event is an LF2, the most likely events to occur in the sequence are considered synergistic to LF2. To arrive at these probability values, the dataset is scanned and for each possible event *a*, it is counted and stored how many times each possible event *b* occurred next. The probability of event *a* being succeeded by event *b* is given by the number of times that this occurred, divided by the total transitions between events in the base.

In a song the order in which the events happen is very important, so an adaptation was implemented in the RDC to try to improve the quality of the generated artifacts. Based on the way to generate creativity through the association of elements belonging to existing artifacts, extra edges were inserted in the synergy graph whenever the artifact builds sequences that exist in the dataset. For simplicity, sequences of up to 5 events were considered. With this, it is expected that individuals who find sequences of up to five events considered to be good are best evaluated by the RDC.

Figure 3 shows an example of synergy graph for an artifact of ten events. For ilustration, lets consider that the sequence comprising the first 5 events of the artifact in question is found in the dataset, that there is synergy between all subsequent pairs of events and that there is also synergy between the pairs of events: 1st and 8Th, 4Th and 6Th, 7Th and 10Th, 2Th and 6Th.

The synergy calculation for the above example is given by the connectivity, which in this case is equal to 1 (connected graph) and by the density, which is obtained by dividing the number of edges by the maximum number of edges for that graph, in case 19/90 = 0.21. Synergy is the simple mean of the two values, that is, 0.605. If the red edges were not inserted, the value of the synergy would be 0.544. However, the intention to implement adaptation in the calculation of synergy is not to allow greater values of synergy. The standard model is able to generate graphs with many edges, but suppose that the events in P1 and P2 have synergy. Without adaptation, the system would be induced to generate an artifact that would repeat these two events throughout the melody. This would have maximum value of synergy, but without any musical meaning.

The calculation of creativity by RDC, used in this work as a measure of fitness of the candidate solutions, is given by a real number

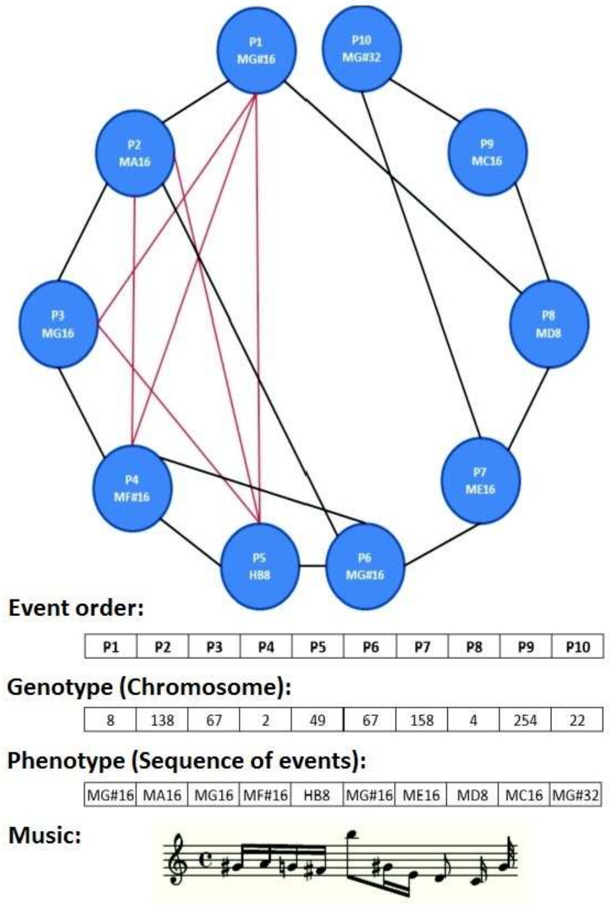


Figure 3: Synergy graph model. The black edges were inserted using standard RDC while the red ones were inserted from the adaptation implemented in the RDC.

in the interval [0,2] obtained from the sum of the synergy and novelty values, subtracting a penalty factor given in virtue of the difference between the values of synergy and novelty, the greater the difference, the greater the penalty.

The selection of candidate solutions that will give rise to a new population was made using the technique of selection by tournament. At each iteration, after the evaluation, the algorithm chooses the individuals in pairs, being the fittest among them selected and the other discarded. This strategy was adopted because it is a technique widely used and it allows a good genetic variability in the population without losing possible good solutions, as it happens in the selection by Russian roulette.

The evolutionary operators used were the crossover and mutation. The crossover operator was used to generate new individuals. Two parents are chosen randomly and two children are raised from the combination of parts of these parents. The mutation operator was used to expand the Genetic Algorithm search space in an attempt to increase fitness faster and avoid stagnation. Five types of mutation were defined: random, session reversal, ascending order, descending order, and guided mutation. In the random mutation a gene is chosen randomly to have its value also changed randomly; In session reversal, an interval of events is randomly chosen to have its events positioned in reverse order; In ascending and descending ordering of sessions an interval of events is also chosen at random, but in this case the events are ordered relative to their tonal height in an ascending or descending manner, respectively, and in the guided mutation a *x* gene is randomly chosen, then it is checked whether *x* and its subsequent *y* occur in the dataset in the same order. If not, *y* is replaced by any event occurring next *x* in the dataset, if so, this check proceeds successively until the 5Th subsequent event.

Table 1 shows the result of each mutation type when applied in the sequence used in Figure 3, having the 3rd gene drawn for mutation and the size of the interval set at 3. The event inserted in the random mutation can be any event belonging to the search space. The event inserted by the guided mutation will be any event in the dataset that occurs after the sequence composed by the events at positions P3, P4 and P5, if there is any.

Table 1: Different implementations used by the mutation operator.

SEQ.	MUTATION TYPE					
	Original	Random	Reversion	Ascend. Sort	Descend. Sort	Guided
P1	MG#16	MG#16	MG#16	MG#16	MG#16	MG#16
P2	MA16	MA16	MA16	MA16	MA16	MA16
P3	MG16	*	HB8	MF#16	HB8	MG16
P4	MF#16	MF#16	MF#16	MG16	MG16	MF#16
P5	HB8	HB8	MG16	HB8	MF#16	HB8
P6	MG#16	MG#16	MG#16	MG#16	MG#16	LA16
P7	ME16	ME16	ME16	ME16	ME16	ME16
P8	MD8	MD8	MD8	MD8	MD8	MD8
P9	MC16	MC16	MC16	MC16	MC16	MC16
P10	MG#32	MG#32	MG#32	MG#32	MG#32	MG#32

As an example of a guided mutation, suppose that there is an artifact in the dataset containing the following sequence of events: MG16, MF#16, HB8, LA16. As the 3rd gene is drawn for mutation and the configuration used previously in which the sequence composed by the first five events of the artifact occurs in some instance of the dataset, the system will verify that the sequence composed by the genes P3 and P4 occurs in the Base, that the sequence composed of the genes P4 and P5 also occurs in the base and that the sequence composed by the genes P5 and P6 does not occur in the dataset. Then, the P6 gene will have its event replaced by something else that occurs in the dataset after the sequence composed of the genes P3, P4 and P5, if any. In the case of this example, event LA16 will be inserted into the P6 gene.

After the generation of the new individuals by the crossover process, for each of them there is a possibility of one mutation occur. The type of mutation to be performed in each individual is randomly defined, with only one of the five types of mutations being able to occur in an individual.

4.2.1 The Dataset

To create the dataset used in the system, fifteen pieces of classical and baroque music were extracted from randomly collected documents in the Mutopia Project repository. These styles were chosen because they are considered as high quality musical segments, in addition, several medieval themed games use soundtrack in these styles, for example Lord of the Rings, World of Warcraft, among others. These fifteen melodies were used as seeds of style to introduce into the system an initial notion of quality in music. In order to allow a better evaluation of the system in the aspect of attributing novelty to the artifacts, the dataset was expanded in two hundred melodies generated by the system itself with the knowledge acquired initially. The melodies generated for the dataset had a value between 0.4 and 0.6. This interval was defined considering that during the preliminary tests, it was observed that melodies generated with a value below 0.4 remained with many random stretches, which could harm the quality notion of the system, and the melodies generated with value above 0.6 replicated very large stretches of the original melodies, in addition to spending relatively high time to produce them.

4.3 Experimental Project

As a way to determine the effects that each parameter has on the artifacts generated, was performed the factorial 2^k experiment. In this experiment two values are defined (minimum and maximum) for *k* variable factors that influence the result of the proposed model. The model is executed with all combinations of possible values and the result is accompanied by metrics that identify characteristics of the generated objects and the execution of the model [18].

In the case of this work, five factors were defined: (a) crossover probability, with values set at 50% and 100%, (b) mutation probability, with values set at 1% and 100%, (c) cut-off points (how many cut-off points in the crossover process), with values set at 1 and 5, d) population size, with values set at 50 and 1000, and (e) session length (part of the melody that will mutate) with values set at 2 and 25. The factors and their values are schematized in Table 2.

Table 2: Factors with the minimum and maximum values set.

	FACTOR	VALUES	
		Min	Max
a	crossover prob.	50	100
b	mutation prob.	1	100
c	cut-off points	1	5
d	population size	50	1000
e	session length	2	25

The metric followed in the execution of each experiment was the creativity, whose measurement is obtained from the fitness function (RDC). Each experiment was performed three times and the final value considered for the calculation was the simple average of the values obtained in the three runs.

The stopping criteria of the algorithm during the experiments were defined as stagnation by thirty consecutive generations or artifacts with fitness greater than or equal to 1.6, which is equivalent to 80% of the maximum possible value of fitness, or elapsed time of 20 minutes, any of the three situations stops execution.

Table 3 shows the results of the experiments, the (- or +) signals indicate that the values used were minimal or maximum, respectively, the variation source column shows the factor, or combination of factors, whose impact is being observed in the experiment.

Table 3: Experiment results

	FACTOR					RESULT		
Experiment	*a*	*b*	*c*	*d*	*e*	Creativity	Variation Source	Impact
1	-	-	-	-	-	0,477	none	
2	-	-	-	-	+	0,474	*e*	0,03%
3	-	-	-	+	-	0,481	*d*	17,22%
4	-	-	-	+	+	0,618	*de*	0,20%
5	-	-	+	-	-	0,470	*c*	0,07%
6	-	-	+	-	+	0,429	*ce*	0%
7	-	-	+	+	-	0,509	*cd*	0,13%
8	-	-	+	+	+	0,551	*cde*	0,03%
9	-	+	-	-	-	1,364	*b*	46,75%
10	-	+	-	-	+	1,363	*be*	0,04%
11	-	+	-	+	-	0,742	*bd*	33,29%
12	-	+	-	+	+	0,756	*bde*	0,06%
13	-	+	+	-	-	1,327	*bc*	0,25%
14	-	+	+	-	+	1,359	*bce*	0,12%
15	-	+	+	+	-	0,530	*bcd*	0,31%
16	-	+	+	+	+	0,605	*bcde*	0%
17	+	-	-	-	-	0,429	*a*	0,31%
18	+	-	-	-	+	0,406	*ae*	0,07%
19	+	-	-	+	-	0,506	*ad*	0,03%
20	+	-	-	+	+	0,618	*ade*	0%
21	+	-	+	-	-	0,467	*ac*	0,28%
22	+	-	+	-	+	0,427	*ace*	0%
23	+	-	+	+	-	0,634	*acd*	0,02%
24	+	-	+	+	+	0,665	*acde*	0,02%
25	+	+	-	-	-	1,528	*ab*	0,11%
26	+	+	-	-	+	1,418	*abe*	0,03%
27	+	+	-	+	-	0,699	*abd*	0,61%
28	+	+	-	+	+	0,687	*abde*	0%
29	+	+	+	-	-	1,512	*abc*	0%
30	+	+	+	-	+	1,520	*abce*	0%
31	+	+	+	+	-	0,626	*abcd*	0%
32	+	+	+	+	+	0,601	*abcde*	0,02%

The impact of each factor or combination of factors on the metric was obtained from a non-linear regression model of the form y = $q_0 + q_a x_a + q_b x_b + ... + q_{abcde} x_a x_b x_c x_d x_e$. Replacing the x_a, x_b, etc., by 1 or -1, and y by the metric, according to the experiment, is obtained 32 equations. As a result of these equations, is obtained the coefficients of the regression equation, from which is calculated the portion of variation that corresponds to each factor. It is recommended to read [18] for a more detailed description of the model used.

As Table 3 demonstrates, the factors mutation probability 46.75%, population size 17.22% and the combination of these 33.29% are the most impacting on the generated artifacts. The experiments with larger populations presented low computational performance, arriving at the predetermined time limit of 20 minutes with 42 generations, on average, whereas the experiments with smaller populations showed good results when the mutation probability was high allowing better exploration of the search space.

For the generation of the artifacts used in the tests, the parameters were fixed, as shown in Table 4.

Table 4: Genetic Algorithm parameters used during the generation of artifacts for tests.

PARAMETER	VALUE
crossover prob.	100%
mutation prob.	25%
cut-off points	1
population size	250
session length	3

5 Tests and Results

5.1 Questionnaire Interview

In order to verify the effectiveness of the RDC metric in the evaluation of the artifacts generated by the system, a research was carried out by means of an electronic questionnaire in which the interviewees evaluated some previously selected artifacts. The questionnaire had eight pre-selected melodies, and for each one the interviewee should evaluate it in two aspects: value and novelty. To assess the value, the interviewee should score on a 10-point scale how much he thought the melody sounded good in relation to the harmony between the notes. To evaluate the novelty, he should score on a 10-point scale how much he considered the melody rhythm regular and familiar.

Among the eight artifacts selected for evaluation, six were actually generated by the system, one was generated randomly and another, a control artifact, was extracted from a real song. In order to guarantee the impartiality of the interviewees, according to the requirement indicated by [7], they were informed that all the melodies of the questionnaire had been generated by the system. The control point was used to disregard the interviews with people who mistakenly evaluated the control artifact. Table 5 shows the evaluation by the RDC of the eight melodies selected for the questionnaire.

Table 5: RDC of the eight melodies selected for the questionnaire.

MELODY	VALUE	NOVELTY	CREATIVITY
Random	0,003	0,9	0,368
Real	1,00	0,808	1,492
S1	0,20	0,93	0,544
S2	0,40	0,93	0,782
S3	0,677	0,698	1,346
S4	0,692	0,797	1,340
S5	0,805	0,813	1,605
S6	0,821	0,875	1,606

Fifteen interviews were collected, of which the evaluations average for each melody normalized in the interval [0,1] are shown in the Table 6. Normalization was made dividing the simple average of the interviewees evaluations to the melody in question by the maximum score possible, in this case, 10.

Table 6: Interviewees evaluations for the melodies used in the questionnaire

MELODY	VALUE	NOVELTY	CREATIVITY
Random	0,38	0,74	0,781
Real	0,80	0,31	0,680
S1	0,43	0,69	0,863
S2	0,48	0,65	0,953
S3	0,74	0,45	0,890
S4	0,64	0,53	1,048
S5	0,71	0,46	0,911
S6	0,71	0,42	0,845

Although the values attributed to each melody by the interviewees were different from those attributed by the system, the order of quality between the melodies was practically the same, as can be seen in Figure 4. The exception was the melody S3, which was considered by the interviewees better than the melodies S5 and S6, which did not occur in the system's evaluation. Another important aspect observed was the fact that the melody generated in a random way was evaluated by the interviewees as the one that sounds less well, which shows that the system fulfills its objective of attributing quality to the artifacts.

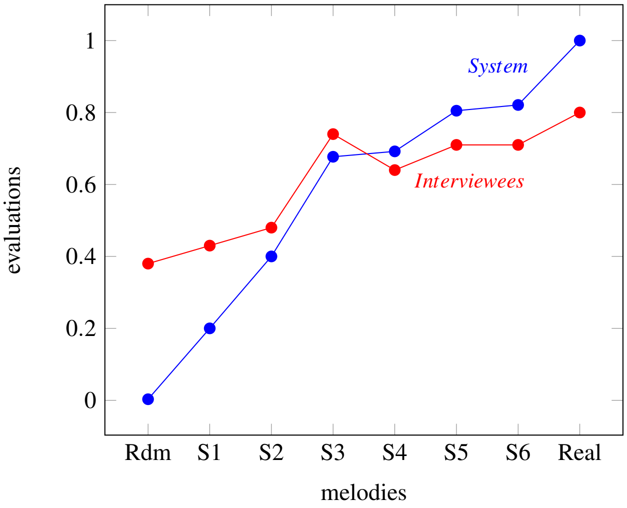


Figure 4: Values assigned to each melody.

In relation to the novelty there was not the same correspondence in the evaluations, as can be seen in Figure 5, however, it should be emphasized that the metric used by the system to evaluate the novelty is given by the degree of modification of the variance that the new artifact causes in the current state of the dataset. Two weightings can then be made: 1 - although the dataset used has been able to provide the system with an adequate parameter of quality, it does not accurately reflect the musical collection known to the interviewees; 2 - the system is more detailed in the calculation of the novelty, computing the gain of novelty in each attribute, while the interviewees only analyze the familiarity of the sounds.

The Figure 6 shows that the evaluations of creativity on the part of the system and the interviewees were quite different. Creativity is calculated from the sum of the value and novelty measures and of the result is subtracted a given penalty depending on the difference between these two measures. Such penalty is important, because through it the system is induced to produce artifacts that contain both characteristics.

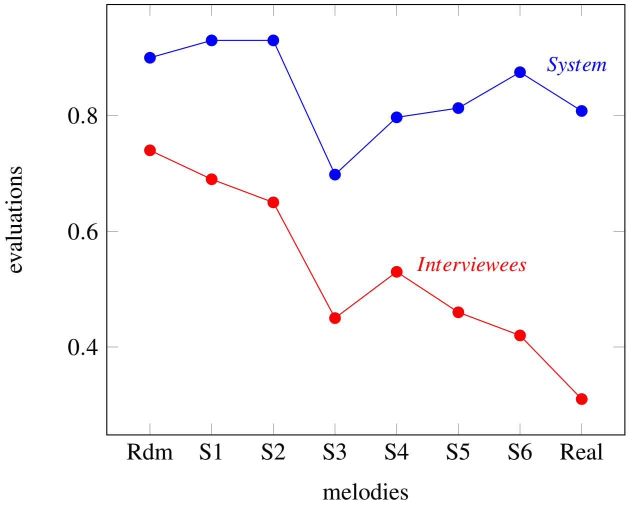


Figure 5: Novelties assigned to each melody.

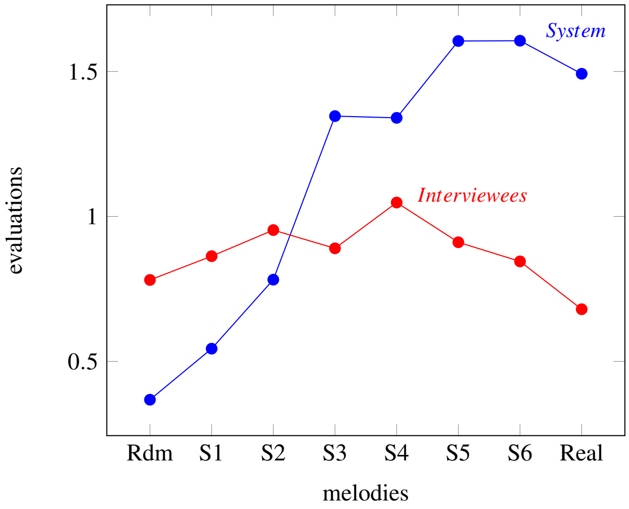


Figure 6: Creativities assigned to each melody.

6 Conclusion and Future Works

Regarding the genetic algorithm, it was noted that the evaluation of the artifacts spent a relatively high computational time, which made the use of very large populations unfeasible, something that would be quite interesting, since the search space in question is extensive.

The tests performed with the questionnaire allowed to identify that the system is able to attribute quality to the generated artifacts and also to measure with good precision the degree of quality in them. Regarding the novelty, the tests with the questionnaire were not so cohesive, which generated a certain contradiction in the calculation of creativity, as for example, the fact that the melody considered by the interviewees with less quality was more creative than the melody best evaluated. However, it is important to point out that the human evaluation of novelty is rather superficial in comparison to the evaluation performed by the system, especially in the aspect of combinational creativity. Through the Bayesian Surprise metric the system is capable of generating novelty by combining, in new ways, elements of existing artifacts, whereas the human evaluator can only detect familiarity in sounds. The Bayesian Surprise metric ensures mathematically that artifacts generated with high novelty value produce change in the dataset variance. In this way it can be said that the system is capable of generating creative music considering aspects of novelty and value.

In the future can be explored: the expansion of the model to generate polyphonic music using a more robust representation model, such as the one proposed by [3]; The use of other types of novelty metrics that compare the created artifact with each item in the dataset individually; The use of a model of synergy proposed by a music expert, rather than extracting it from the dataset; The modification of the event model representation to one that takes into account only the interval between notes, which would considerably reduce the search space and allow larger populations to be used in the Genetic Algorithm. All are interesting approaches and their results could help to consolidate the RDC metric as an efficient method for generating and evaluating creativity in artifacts.

References

[1] S. Frishert, "Implementing algorithmic composition for games," *Utrecht School of the Arts, Department of Art, Media and Technology*, 2013.

[2] M. Delgado, W. Fajardo, and M. Molina-Solana, "A state of the art on computational music performance," *Expert systems with applications*, vol. 38, no. 1, pp. 155–160, 2011.

[3] R. Loughran, J. McDermott, and M. O'Neill, "Tonality driven piano compositions with grammatical evolution," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 2168–2175.

[4] A. R. Freitas, "Métodos de Composição Algorítmica," http://www.alandefreitas.com/assets/material/ca/livroca.pdf, 2015, accessed: 2017-08-06.

[5] M. A. Boden, "Computer models of creativity," *AI Magazine*, vol. 30, no. 3, p. 23, 2009.

[6] ——, *The creative mind: Myths and mechanisms*. Psychology Press, 2004.

[7] S. Colton and G. A. Wiggins, "Computational creativity: The final frontier?" in *Proceedings of the 20th European conference on artificial intelligence*. IOS Press, 2012, pp. 21–26.

[8] G. Papadopoulos and G. Wiggins, "AI methods for algorithmic composition: A survey, a critical view and future prospects," in *AISB Symposium on Musical Creativity*, 1999, pp. 110–117.

[9] J. D. Fernández and F. Vico, "AI methods in algorithmic composition: A comprehensive survey," *Journal of Artificial Intelligence Research*, vol. 48, pp. 513–582, 2013.

[10] C. Darwin and J. W. Burrow, *The origin of species by means of natural selection: or the preservation of favoured races in the struggle for life*. Penguin Books, 1985.

[11] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer methods in applied mechanics and engineering*, vol. 191, no. 11, pp. 1245–1287, 2002.

[12] J. H. Jensen, "Evolutionary music composition: A quantitative approach," Master's thesis, Institutt for datateknikk og informasjonsvitenskap, 2011.

[13] J. Biles, "GenJam: A genetic algorithm for generating jazz solos," in *Proceedings of the International Computer Music Conference*. INTERNATIONAL COMPUTER MUSIC ACCOCIATION, 1994, pp. 131–131.

[14] M. Alfonseca, M. Cebrián Ramos, and A. Ortega, "Evolving computer-generated music by means of the normalized compression distance," *WSEAS Transactions on Information Science and Applications*, vol. 2, pp. 1367–1372, 2005.

[15] M. Fukumoto, "Creation of Music Chord Progression Suited for User's Feelings Based on Interactive Genetic Algorithm," in *Advanced Applied Informatics (IIAIAAI), 2014 IIAI 3rd International Conference on*, 2014, pp. 757–762.

[16] D. W. Dyer, "Watchmaker Framework for Evolutionary Computation," http://watchmaker.uncommons.org/, 2006, accessed: 2017-08-06.

[17] C. França, L. F. W. Góes, A. Amorim, R. Rocha, and A. R. da Silva, "Regent-Dependent Creativity: A Domain Independent Metric for the Assessment of Creative Artifacts," in *Proceedings of the Seventh International Conference on Computational Creativity*, 2016, pp. 68–75.

[18] R. Jain, *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons, 1990.