Synesthetic Musical Composition using Computational Intelligence

Alan Garcia-Zambrano¹, Yenny Villuendas-Rey¹, Oscar Camacho-Nieto²

¹ Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Mexico City, Mexico

² Instituto Politécnico Nacional, Secretaría de Extensión e Integración Social, Mexico City, Mexico

Abstract. Synesthesia is the combination of two senses in the same perceptive act; see the music, hear a red color or feel the texture of a green sound are some examples. The problem of a bad teaching in music theory turns music into a tedious and boring subject, causing in many cases a desertion. The musical composition with synesthesia will make it easier for the user to learn music through the association of sound with color and exploit a new way of learning with intelligent computing methods, where the advanced user creates visually pleasing musical compositions encouraging musical creativity. This article explains the use of evolutionary algorithms as a novel auxiliary model for musical innovation. In addition, it shows the existing tools of intelligent computation (IC) for musical composition and creation of computational art.

Keywords: computational intelligence, computational evolution, synesthesia, music composition, genetic algorithms.

1 Introduction

Musical language for centuries has been represented by musical signs embodied in manuscripts; systems of notation that determine the rules for the interpretation and together they integrate a melody [1]. Sometimes, the learning of *theoretical-musical* concepts is a rigorous process leading to ostracism. A study indicates that it takes about 10 thousand hours of practice in a musical instrument to be an expert; what represents practice for more than 10 years [2]. Recently, a large number of methods have been used of Computational Intelligence (CI) for compositions and expression of musical abilities. Genetic Algorithms (GA) [3], [4], Genetic Programming (GP) [5], Particle Swarm Optimization (PSO) [6]: metaheuristics for search and optimization. Its engine is the inspiration of nature as a biological metaphor, *selection, crossover* and *mutation*, proposed by Darwin and Mendel. Mitchell, Holland and Goldberg [7] who have managed to develop novel applications using CI for an endless number of scientific applications. Each individual in an evolutionary algorithm represents the solution to a problem, and those that offer better adaptability to the mechanisms of nature will

survive. The CI has managed to adapt models of artificial intelligence inspired by biological evolution; proving to be necessary in the study and development of new methodologies and applications as an excellent option to solve this type of problems.

We propose the intelligent modelling of the music composition problem as an auxiliary resource for learning, which allows the joint assimilation of perceptive sensations (hearing and sight) to improve techniques of improvisation and musical learning with *Synesthesia*. Its purposes are: (1) that musical learning with synesthesia avoids ostracism. (2). highlight the importance of the exercise of the senses in the acquisition of knowledge and musical ability and (3) adapts the evolutionary computation to a requirement of the company. In addition, it would work as a very useful tool for experienced musicians who are in the so-called "*Creative Stagnation*" due to lack of inspiration, stress or depression; creating visual art based on compositions and interpretations of musical language, supported by the principles of genetic evolution.

The paper is organized as follows: In Section one: a brief introduction on the importance of the project is detailed; in Section two: related works and bases for technological development are mentioned, in Section three: the methodology used, materials and methods for development are established. In Section four: the solution proposal is detailed; in Section five: the expected results are shown. Finally, the conclusions and future work are mentioned.

2 Previous Works

Composers like J. S. Bach, and W.A. Mozart used algorithms as a technique of improvisation and musical writing: the "Music Dice Game" used randomness, inspiration and a pair of dice creating complex musical pieces [8] or the "BACH Motive", which was used in cryptography [9]. Hiller, and Isaacson, designed the first computer-aided music generation (ILLIAC) [10]. The musical composition uses different methodologies for composition such as mathematics based on stochastic and probabilistic processes to predict musical sequences based on "Markov chains" [11] and fractal geometry as a form of musical expression [12]. In addition, grammar, which encode multidimensional arrays to create patterns of rhythm and also harmony. The authors of [13] built a system for electric bass. With an input line connected to the computer, it feeds the population and agents cooperate to generate complete musical pieces as a solution to an optimization problem. [14].

He uses musical compositions based on AG to create cryptograms. The encryption algorithm encodes a message into another intelligible with music. The authors of [15] generate images from L-Systems using AGI; because of visual interpretation, the system models growth and develops visual constructions with music, balancing technology and music with visual expressions. In the current market, there are a variety of software applications for musical composition and assistance, such as GarageBand [16] an auxiliary software for professional musicians which supports numerous synthetic instruments so that the user can compose a melody easily, Guitar Pro [17] is a sheet music editor that supports the MIDI format. Chordbot [18] is an application for

complex chords and accompaniment styles to produce tracks and mixes of piano and bass. TonePad [19] is an application with an interactive interface; the user creates melodies by touching a matrix on the device's screen. However, perhaps the most notable example of algorithmic musical composition is the Iamus Computer Cluster [20], it is the first computer that simulates and recognizes human language to perform musical improvisation.

3 Coding, Structure and Configuration

In the following section, we present the methodology of the modelling in four subsections: Structure, configuration and coding of the phenotype, primordial elements of the genetic algorithm, representation of the solutions and calculation of objective function.

3.1 Genetic Coding

The genetic structure is composed of the genotype and phenotype with the following *genes*: (1) *Clef*, (2) *Octave*, (3) *Scale*, (4) *Measure*, (5) *Note* and (6) *Accidental*.

- **1.** Clef: Determines the position in the musical staff with a binary coding [0,1].
- **2. Octave:** Each musical octave is composed of twelve semitones with their accidentals, that is, seven fundamental notes plus five accidentals defined by the pitch or height of the sound in a musical instrument. Using Octave 2 Octave 6 with the configuration [0 4] to delimit the range of sounds and colors is proposed.

Do Re Mi Fa Sol La Si Do Re Mi Fa Sol La Si	Do Re MI Fo Sol Lo SI Do Re MI Fo Sol Lo SI Do Re MI Fo Sol Lo SI	Do Re MI Fa Sol La SI Do Re MI Fa Sol La SI
1 Octava Dorre Mi Fa Sol La Si Dorre Mi Fa Sol La Si	3 Octava 4 Octava 5 Octava Do central	
	→ ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ►	DoRe Mi Fa Sol La Si DoRe Mi Fa Sol La Si

Fig. 1. Conventional keyboard of seven octaves differentiated by the clefs of G and F.

- **3. Scale:** It contains the coding for seven musical notes distributed progressively (Figure 2).
- **4. Measure**: Its phenotype distinguishes six figures; *Silence*, *Whole*, *Half*, *Quarter*, *eighth* and *sixteenth note* with coding: [0 6]. The spaces of the chromosome (5-36) determine time, duration, frequency and the number of elements.

A Whole note is equivalent to a structure of sixteen elements (*sixteenth note*) which is equivalent to two eighths of eight notes (*eighth note*). In addition, two figures of (*Half note*) are equivalent to four figures of one time (*Quarter*), that is: Half is worth two Quarters; the *eighth note* is worth two eighth notes. Therefore, the Whole note is equivalent to two Half notes, four Quarters, eight eighth notes (sixteenth notes). This



scheme (Figure 3) is proposed to represent the duration and distribution of the musical notes in the *measure* with a speed of 80 beats per minute (bpm).

Fig. 2. Genetic configuration of the notes by major scales with accidental and silence.



Fig. 3. The proposed compass is divided into four quarters of time with a total duration of 1920 ms; each quarter of time lasts 480 ms, in turn, it is divided into four segments of 120 ms.

- 5. Musical Notes: The notes are the interpretation of a frequency produced by the vibration of a sound, together they allow composing melodies. Its phenotypic representation is defined as (C, D, E, F, G, A, B and silence) with the coding [0-7].
- 6. Accidentals: They are the symbols that increase or decrease the intonation of the notes by semitone. Its phenotype is represented by sharp or flat (#, b), with coding [0,1] They are essential in the genetic mutation process.

3.2 Structure of the Chromosome

The genetic structure of the chromosome contains thirty-six spaces available to store the genes. The first four spaces belong to the configuration and the rest defines the structure of the musical system.

- Configuration: The genes in the chromosome (1 4) define the initial configuration and rules of the system.
- Structure: The genetic composition of a chromosome in the spaces (5-36) correspond to the musical notes and alteration. The odd spaces (5,7, ..., 35) determine specific I formation of the alleles corresponding to the musical notes (C, D, E, F, G, A and B), and the pairs (6,8, ..., 36) accidentals (#, b).

Research in Computing Science 147(12), 2018

236

- **Genotype**: Contains the genetic configuration of the genes in a chromosome, represents the information contained in the genes with the characteristics that define each individual, these are transmitted from generation to generation by inheritance.
- Phenotype: Represents the visible characteristics of each gene in the chromosome, during development, such as hair color, eyes, height, weight, etc.



Fig. 4. General outline of the configuration and block structure of a chromosome.

4 Genetic Algorithms for Intelligent Musical Composition

Genetic Algorithms (GA) are metaheuristics of search and optimization inspired by nature's mechanics. They are used to solve problems or optimize solutions and are essential in the construction of Computational Intelligence. This technology is perfectly applicable in musical composition and computational creativity [21]. They result in the adaptability of individuals to better search regions for generations. The genetic selection, crossing and mutation operators carry out the evolution of the population and repeats until a stop condition is fulfilled [4].

Selection: The best parent individuals (P1, P2) are chosen from the initial population by a selection method such as roulette or tournament [22] that will determine their qualification. They seek to improve the quality of descendants of the population [23].



Fig. 5. Example of genetic structures to represent two quarters of best two individuals in the population.

Crossover: We propose to exchange information by crossing under the following conditions: (1) It is required to implement an elitist operator that does not touch the

configuration of the chromosome, because it would alter the structure producing amorphous individuals. (2) Place the cut points only in even spaces of the chromosome to preserve the note with its alteration. (3) The minimum cutting range must be made after the fourth space and (4) it is advisable to use the cut in two points; in this case the operator would represent improvement [24]. (5) The probability of crossing (Pc) must be high.



Fig. 6. Example of genetic structures (P1 and P2) to represent the product of crossing at two points and its musical staff.

Mutation: Occurs in the genes of both descendants after crossing, randomly changes one of the genes of the individual child. The mutability considers the following restrictions: (1) Probability of mutation (*Pm*) in each gene must be very small. (2) Applies only to: *Cliff, Octave, Music Note* or *Accidental*. (3) The compass and scale are omitted. As in the crossing, it would decompose the genetic structure.



Fig. 7. Example of mutation (X1) The Quarter Note C becomes an eighth note (X1').

Research in Computing Science 147(12), 2018 238

4.1 Generated Individual and Objective Function

A population represents the group of individuals as possible solutions, and with the passage of generations they will manage to evolve until they reach an objective. Each generation must carefully perform the selection of the best individuals to cross them and in very few cases to mutate their elements so as not to stagnate the evolution.

Generated Individual: looks like an objective function, and has to calculate how it looks and what its distance function is.



Fig. 8. Example of individual generated.

Objective function: seeks to reward the adaptability of their individuals towards a desired solution. The initial fragment of two-time melody "Imagine" by composer and musician John Lennon. Figure 9 represents the configuration of the phenotype with the individual coding for a compass.



Fig. 9. Example of objective individual.

4.2 Objective Function Calculation

We propose to apply the distance calculation in: *Octave*, *Scale* and *Musical Note*. The distance calculation procedure for the values of each chromosome is made as follows:

Calculation of Objective Function in the Octave: The fundamental values of the notes in each musical octave are taken; these are measured in a fundamental frequency range (Hz) and their corresponding average is calculated by set of tones (Table 1).

The generated individual indicates that his Octave contains a coding [2] (Octave 4) and the target individual contains a value [4] (Pitch 6).

$$d_1(Pitch4, Pitch2) = 360 - 90,d1 (360 - 90) = 270.$$

Calculation of Objective Function in Scale: We propose adding the coding of each major scale to obtain a real number (Table 2). The generated individual has a coding [3] (Scale 4) and the objective individual contains a value [2] (Scale 3).

 $d_1(Escala4, Escala3) = 27 - 22,$ $d_1(27 - 22) = 2$

ISSN 1870-4069

239 Research in Computing Science 147(12), 2018

Calculation of Objective Function in Musical Note: Each musical notes according to the protocol of communications MIDI (Musical Instrument Digital Interface), represents a numerical value (36 - 95), depending on the octave in which it is located. Each octave is a group of twelve notes (Table 3). We proposed that the silent note is assigned a high value, otherwise it can be confused with another MIDI note and the same procedure is performed for distance calculation.

Coding	0	1	2	3	4
Octaves	Pitch 2	Pitch 3	Pitch 4	Pitch 5	Pitch 6
	65.4064	130.813	261.626	523.26	1046.52
	69.2957	138.591	277.183	554.37	1108.73
	73.4162	146.832	293.665	587.33	1174.66
	77.7817	155.563	311.127	622.26	1244.51
cies	82.4069	164.814	330	659.26	1318.51
nenc	87	174.614	349	698.46	1396.91
requ	92.4986	184.997	370	739.99	1479.98
Ē	97.9989	195.998	391.995	783.99	1567.98
	103.826	207.652	415.305	830.61	1661.22
	110	220	440	880.00	1760.00
	116.541	233.082	466.161	932.33	1864.66
	123.471	246.942	493.883	987.80	1975.60
Average	90	180	360	720	1440

Table 1. Shows the corresponding average of frequencies by Octave.

Table 2. Shows the proposal of distance corresponding to the calculation of scales.

																			0	1	2	3	4	5	6	
_				ш	1	E	2	E	3	Е	4	Е	5	Ш	6	Е	7	Σ	21	23	25	27	22	24	26	
	0	E1	Do	0	0	1	0	2	0	3	0	4	0	5	0	6	0	21	0	2	4	6	1	3	5	0
	1	E2	Re	1	0	2	0	3	1	4	0	5	0	6	0	0	1	23	2	0	2	4	1	1	3	1
M2 =	2	E3	Mi	2	0	3	1	4	1	5	0	6	0	0	1	1	1	25	4	2	0	2	3	1	1	2
	3	E4	Fa	3	0	4	0	5	0	6	6	0	0	1	0	2	0	27	6	4	2	0	5	3	1	3
	4	E5	Sol	4	0	5	0	6	0	0	0	1	0	2	0	3	1	22	1	1	3	5	0	2	4	4
	5	E6	La	5	0	6	0	0	1	1	0	2	0	3	1	4	1	24	3	1	1	3	2	0	2	5
	6	E7	Si	6	0	0	1	1	1	2	0	3	1	4	1	5	1	26	5	3	1	1	4	2	0	6

Table 3. Contains the MIDI values for this proposal is delimited from octave 2 to 6.

Octave	С	C#	D	D#	Е	F	F#	G	G#	Α	A#	В
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127	128	129	130	131

The difference between the objective function (Fo, Figure 9) and the individual function (*Fi*, Figure 8) is expressed by the following formula:

Research in Computing Science 147(12), 2018

240

Synesthetic Musical Composition using Computational Intelligence



Fig. 10. Visual comparison between objective (a) and the generated individual (b).

The difference between both individuals is remarkable in sound as in color scale; although the calculation focuses only on the first two frames, the objective function shows harmony and the ordered combination of colors. While the generated individual is the opposite. However, with the passing of generations it may be able to adapt to achieve the objective.

5 Conclusions and Future Work

Music is a universal language that should not be difficult to understand or interpret, and with the appropriate software, it can be made even easier. The evolutionary algorithms prove to be ideal for the creation of better visual proposals that contribute to the creative process of musical learning. The user is the one who conducts the feedback process of the system, evaluating how much the individual generated is similar to the solution. The best solution generated could sound (or look) slightly different from the objective, but it is part of the purpose to stimulate the creativity process. As future work, we will implement its use with other configurations for the Genetic Algorithm.

Acknowledgments. The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, CIDETEC and SEIS), the Consejo Nacional de Ciencia y Tecnología, and Sistema Nacional de Investigadores for their economical support to develop this work.

ISSN 1870-4069

241 Research in Computing Science 147(12), 2018

References

- 1. Herrera, E.: Teoría musical y armonía moderna, 1st ed. Antoni Bosch editor (1995)
- 2. Ericsson, K. A., Krampe, R. T., Tesch-Römer, C.: The role of deliberate practice in the acquisition of expert performance. Psychol. Rev., vol. 100, no. 3, pp. 363–406 (1993)
- Goldberg, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning. MA, USA: Addison-Wesley (1989)
- 4. Holland, J. H.: Adaptation in Natural and Artificial Systems: An introductory Analysis with Applications to Biology, Control and Artificial Intelligence (1975)
- 5. Koza, J. R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: MIT Press (1992)
- Kennedy, J.: Particle swarm optimization. In: Encyclopedia of Machine Learning, New York, USA: Springer, pp. 760–766 (2010)
- 7. Mitchell, M.: An introduction to genetic algorithms. MIT Press, vol. 32, no. 6, p. 162 (1996)
- Acevedo, A. G.: Fugue composition with counterpoint melody generation using genetic algorithms. In International Symposium on Computer Music. Berlin, Heidelberg.: Springer, pp. 96–106 (2004)
- 9. Boyd, M., Butt, J.: J.S. Bach. (1999)
- Hiller, L., Isaacson, L.: Experimental music: composition with an electronic computer. New York, USA: McGraw-Hill (1959)
- 11. Diaz-Jerez, G.: Algorithmic music: Using mathematical models in music composition. New York, USA: Manhattan School of Music. (2000)
- Hsu, K. J., Hsu, A. J., Hspt, A. J.: Fractal Geometry of Music. In: Proc. Natl. Acad. Sci. United States Am. Phys., vol. 87, pp. 938–941 (1990)
- Munoz, E., Cadenas, J. M., Ong, Y. S., Acampora, G.: Memetic Music Composition. IEEE Trans. Evol. Comput., vol. 20, no. 1, pp. 1–15 (2016)
- 14. Kumar, C., Dutta, S., Chakraborty, S.: Hiding messages using musical notes: A fuzzy logic approach. Int. J. Secur. its Appl., vol. 9, no. 1, pp. 237–248 (2015)
- Rodrigues, A., Costa, E., Cardoso, A., Machado, P.: Evolving L-systems with musical notes. In: Lecture Notes in Computer Science, vol. 9596, pp. 186–201 (2016)
- 16. GarageBand: [Online]. Available: https://www.apple.com/mx/mac/garageband/.
- 17. GuitarPro: [Online]. Available: https://www.guitar-pro.com/en/index.php.
- 18. ChordBot: [Online]. Available: http://www.chordbot.com/.
- 19. TonePad: [Online]. Available: https://appadvice.com/app/tonepad/.
- 20. Iamus: [Online]. Available: http://www.melomicsrecords.com/.
- Liu, C.-H., Ting, C.-K.: Computational Intelligence in Music Composition: A Survey. IEEE Trans. Emerg. Top. Comput. Intell., vol. 1, no. 1, pp. 2–15 (2017)
- 22. Miller, B. L., Goldberg, D. E.: Genetic Algorithms, Tournament Selection, and the Effects of Noise. Complex Syst., vol. 9, no. 3, pp. 193–212 (1995)
- 23. Gestal, M.: Introducción a los Algoritmos Genéticos. (2010)
- Moujahid, A., Inza, I.: Algoritmos genéticos. Metod. Mat. en Ciencias la Comput., pp. 1– 34 (2008)