ISSN 2007-9737

Automatic Composition of Music Using a Genetic Algorithm, Emotional Musical Theory and Machine Learning

Adriana Lara[1], Giovanni Guzmán[2], Natan Vilchis[1]

[1] Instituto Politécnico Nacional,
Escuela Superior de Física y Matemáticas,
Mexico

[2] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

alaral@ipn.mx, jguzmanl@cic.ipn.mx, contacto@natanvilchis.org

Abstract. This work proposes a mathematical model for computer-aided music composition as a multi-objective optimization problem. This work aims to create a framework to automatically generate a set of songs with two melodies by combining a genetic algorithm with machine learning. Musical patterns were studied [16, 6, 18, 2] to simplify them and apply them for the construction of the optimization model. This work uses recent emotional music theory to construct the optimization problem [11]. Three conflicting objective functions represent the desired characteristics of the melody to be created: (1) song happiness, (2) song minimalism, and (3) song genre. Two of these objectives are analytically designed, fulfilling well-studied features like those in [14, 25, 11]. The third objective function was developed using a machine learning model like in [5, 8, 27]. The software JSymbolic is used [15] for extracting features in real-time and getting the score with the machine learning model trainer in the present work. The results obtained by this work can be listened to by test examples presented in a video format.

Keywords. Music composition, multiobjective optimization, evolutionary music.

1 Introduction

Musical composition using artificial intelligence is one of the great challenges due to the different characteristics that a song contains; in addition, the different types of properties of a song make its study difficult.

Some artificial intelligence tools used for this purpose are: deep learning, generative music, quantum computing, machine learning techniques, among others [21, 5, 8, 27]. In the present work, the problem of computer-assisted musical composition is modeled in terms of a multi-objective optimization problem.

To generate songs with a maximum happiness, while simultaneously looking for minimalism in the ornaments and looking for songs in two different genres as a third criterion.

For this, Mauro de María's emotional music theory [11], classical music theory [7], study of musical patterns [16, 6, 19, 18, 22, 10, 2, 17] were augmented and related state-of-the-art works such as MetaCompose [25] and MorpheuS [14] that provide valuable elements for our model. The present work is a combination of existing elements, such as the definition 9 (based on equation 4 of [25]).

The first four terms of the g_3 function of the optimization problem are based on [14] together with combination and simplification of [16, 6, 19, 18, 22, 10, 2, 17] and with original contributions based on emotional music theory [11] present in the other elements of the optimization problem.

In addition, to evaluate the third objective to optimize, a model generated with the help of machine learning through the extraction of features from songs of different genres is used.

2 Limitations

There are different limitations in the current work, which are described below:

- The songs generated by the present work have at most two notes playing at the same time. This is because songs have been defined as two melodies playing simultaneously. An example of the song representation will be detailed in section IV, basic concepts.
- The parameter p (bar partitions) will determine the number of notes within a bar. A larger number of p allows a larger number of musical figures to be written to each melody. For example, if $p = 2$, each melody will allow musical figures of value $\frac{1}{2}$ (𝅗𝅥) and $\frac{2}{2}$ (𝅝), if $p = 4$, each melody will allow musical figures of value $\frac{1}{4}$ (♩), $\frac{2}{4}$ (𝅗𝅥), $\frac{3}{4}$ (𝅗𝅥.) and $\frac{4}{4}$ (𝅝) and so on.
- A larger value of b (number of bars in the song) or p (bar partitions) implies more time in optimization process; that is, increasing the length of the song increases the time to find feasible solutions.
- The generated songs are based on the evocation of happy songs (objective g_1) and minimalist objective (objective g_2), other emotions and other musical aspects can be taken into account by adding other objectives in the optimization problem.

3 Related Work

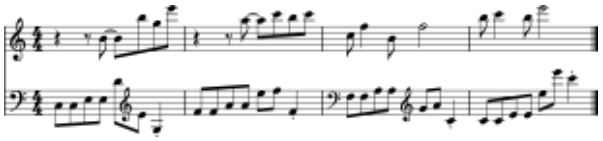
There are different related works that allowed to generate the model shown in the optimization problem section, each one of them and the way in which their valuable ideas were applied are described as follow:

- The musical emotional theory elaborated by Mauro de María [11] provides a great perspective about the different parts that a musical composition has and how each of its parts influences the evocation of emotions. For the elaboration of the present work, a simplification of Mauro de María's theory was carried out to model happy and minimalist songs. The main ideas of Mauro de María were applied mainly in the definitions 4 , 23 and 14.
- There are different related works [16, 6, 19, 18, 22, 10, 2, 17] that address the problem of finding musical patterns successfully. In the elaboration of the present work it was tried to use some of these algorithms in the optimization problem; however, it was decided to use a simplification of these given that they are very expensive and the generation of results could take maybe a few months. These simplifications appear in the first four terms of the g_3 function of the optimization problem.
- MetaCompose [25] and MorpheuS [14] propose structured music generation. In the case of MorpheuS, a tension pattern must be provided to ensure that the result is adequate. In our work, this tension is reflected in the functions g_1, g_2 and g_3, while the structure of the song is reflected in the constraints of the optimization problem.

4 Basic Concepts and Fundamental Elements of the Model

We have represented a song as a vector $x \in \mathbb{Z}^{b+2bp}$, where the first b components denote the chord index for the corresponding bar; the following bp components represent the notes of harmony; finally, the last bp components represent the notes of the melody. In this way, each melody $x^{(1)} \in \mathbb{Z}^{bp}$ has $b \in \mathbb{N}$ bars and, in turn, each bar is divided into p partitions, with $p \in \{2, 4, 8, 16\}$; furthermore, each component $41 \leq x_j \leq 88$, for $j \in \{b+1, \ldots, 2bp\}$ is a MIDI note [1]; where MIDI note 41 will be taken as a elongation of the previous note, while MIDI note 42 will be used as rest. The next score shows the representation of a song corresponding to the vector $x^0 \in \mathbb{Z}^{4+2(4)(8)}$ given by:

$$x^0 = \begin{matrix}[1, 11, 11, 1, 48, 48, 52, 52, 62, 64, 55, 42, 65, 65, 69, 69, 76,\\ 77, 65, 42, 53, 53, 57, 57, 67, 69, 60, 42, 60, 60, 64, 64, 76, 88,\\ 84, 42, 42, 41, 41, 71, 41, 83, 79, 88, 42, 41, 41, 81, 41, 84, 83,\\ 84, 72, 77, 41, 71, 77, 41, 41, 41, 83, 84, 41, 83, 88, 41, 41, 41]^T.\end{matrix} \quad (1)$$



That said, we consider a melody to be a vector of MIDI notes [1], the definitions 15, 16, 17, 18, 19 and 20 were built from the classical music ornaments theory [23], while the remaining definitions were created from the emotional musical model [11] and adapted to the optimization problem of this work. The necessary definitions for our model will be shown below.

Definition 1 (Melody of the j-th bar of a melody)**.** Let $x \in \mathbb{Z}^{bp}$ be a melody, with $b, p \in \mathbb{N}$. The melody of the j-th bar $x_{\mathcal{B}(j)}$ corresponds to the subvector of x given as:

$$\mathcal{B}(j) := x_{(1+p(j-1):jp)}, \quad j = 1, \ldots, b. \quad (2)$$

Definition 2 (Chord)**.** The vector $x = (x_1, x_2, \ldots, x_q)^T \in \mathbb{Z}^q$ is said to be a chord, if each x_i is a numbered musical note, for $i = 1, \ldots, q$ and furthermore it is verified that $x_i \neq x_j \quad \forall i \neq j$.

Definition 3 (Melody with a good start)**.** Let $x \in \mathbb{Z}^k$ be a melody. The tune x is said to have a good start if the function $NS : \mathbb{Z}^k \to \{0, 1\}$ evaluated in the tune x is equal to one, where the function NS is defined as follows:

$$NS(x) := \begin{cases} 1 & \text{if } \nexists j \in \{1, \ldots, k\} : x_j = 41, \\ 1 & \text{si } \min\limits_{\substack{j=1,\ldots,k \\ x_j > 41}} j < \min\limits_{\substack{\ell=1,\ldots,k \\ x_\ell = 41}} \ell, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4 (Total real notes in melody)**.** Let $x \in \mathbb{Z}^k$ be a melody, the total number of real notes in a melody is the result of $TN(x)$, where the function $TN : \mathbb{Z}^k \to \mathbb{Z}$ is defined as follows:

$\text{TN}(x) :=$ number of elements x_j greater than 42, $j = 1, \ldots, k$.

Definition 5 (Melody repetition of a melody)**.** Let $x \in \mathbb{Z}^k$ be a melody such that $\exists x_j > 42$. The repeat melody $y \in \mathbb{Z}^k$ of melody x is the result of $\text{repeatMelody}(x)$, where the function $\text{repeatMelody} : \mathbb{Z}^k \to \mathbb{Z}^k$ is described in the algorithm 1.

Algorithm 1 repeatMelody function to get the repetition melody of a melody.

procedure repeatMelody(x)
  Let $m, j \in \mathbb{N}, \text{saveNote} \in \mathbb{Z}, y \in \mathbb{Z}^k$.
  $m \leftarrow \min\limits_{\substack{j=1,\ldots,k \\ x_j > 41}} (j)$.
  $\text{saveNote} \leftarrow x_m$.
  $j \leftarrow m + 1$.
  $y \leftarrow x$.
  while $j \leq k$ **do**
    if $x_j == 41 \vee x_j == 42$ **then**
      $y_j \leftarrow \text{saveNote}$.
    else
      $\text{saveNote} \leftarrow x_j$.
    end if
    $j \leftarrow j + 1$.
  end while
  return y.
end procedure

Definition 6 (Simultaneously Pressed Notes)**.** Let $x, w \in \mathbb{N}$ be two MIDI notes, the notes x, w are said to be simultaneously pressed notes if the result of $PSN(x, w)$ is one, where the function PSN is defined as follows:

$$\text{PSN}(x, w) := \begin{cases} 1, & \text{if } \quad (x > 42) \wedge (w > 42), \\ 0, & \text{otherwise.} \end{cases}$$

Definition 7 (Proper note for the melody)**.** Let x be a MIDI note, let $w \in \mathbb{Z}^k$ be a melody such that $\exists w_j > 42$.

Let $\text{ND}(w) = \{(x_i - 1) \bmod 12 \; \forall x_i > 42\} \bigcup \{(x_i + 1) \bmod 12 \; \forall x_i > 42\}$. x is said to be a suitable note

for the melody w if the result of $\mathrm{MF}(x, w)$ is equal to one, where the function MF is defined as follows:

$$\mathrm{MF}(x, w) := \begin{cases} 1, \text{ if } x \notin \mathrm{ND}(w), \\ 0, \text{ otherwise.} \end{cases}$$

Using the definition 5, the following definition is built, which will return a vector, where each of its components will be $-1, 0$ or 1.

Definition 8 (SDR function)**.** Let $x \in \mathbb{Z}^t$ be a melody with $t \geq 2$, such that $\exists x_j > 42$. Let $D : \mathbb{Z}^{t-1} \to \mathbb{Z}^t$ the first difference function. Let $S : \mathbb{Z}^{t-1} \to \mathbb{Z}^{t-1}$ a vector with the signs of the corresponding elements of argument. The function $SDR : \mathbb{Z}^t \to \mathbb{Z}^{t-1}$ is defined as follows:

$$\mathrm{SDR}(x) := S(D(\mathrm{repeatMelody}(x))).$$

Definition 9 (First note of the melody is the root note of the chord)**.** Let $x \in \mathbb{Z}^k$ be a melody such that $\exists x_j > 42$, let $w \in \mathbb{Z}^q$ be a chord. The melody x is said to have as its first note the root note of the chord w if the result of $\mathrm{FirstNoteFC}(x, w)$ is equal to one, where the function $\mathrm{FirstNoteFC}$ is described below:

$$\mathrm{FirstNoteFC}(x, w) := \begin{cases} 1, & \text{if } \quad x_r \bmod 12 = w_1, \\ & \quad\; r = \min\limits_{\substack{x_j > 42 \\ j=1,\ldots,k}} (j), \\ 0, & \text{otherwise.} \end{cases}$$

Definition 10 (MIDI note on chord)**.** Let x be a MIDI note, let $w \in \mathbb{Z}^q$ be a chord. The MIDI note x will be a note of the chord w if the result of $\mathrm{NInChord}(x, w)$ is equal to one, where the function $\mathrm{NInChord}$ is defined as follows:

$$\mathrm{NInChord}(x, w) := \begin{cases} 1, & \text{if } \quad x \bmod 12 \; = w_j, \\ & \quad\; \text{for some } j = 1, \ldots, q, \\ 0, & \text{otherwise.} \end{cases}$$

Using the definition 10, the following definition is constructed as follows.

Definition 11 (At least fifty percent of MIDI notes are chord notes)**.** Let $x \in \mathbb{Z}^k$ be a melody, let $w \in \mathbb{Z}^q$ be a chord. At least fifty percent of the notes of the melody x are said to be notes of the chord w if the result of $\mathrm{FPN}(x, w)$ is equal to one, where the function FPN is defined as follows:

$$\mathrm{FPN}(x, w) := \begin{cases} 1 & \text{if } \dfrac{\sum\limits_{x_j > 42} \mathrm{NinChord}(x_j, w)}{TN(x)} \geq 0.5, \\ 0 & \text{otherwise.} \end{cases}$$

Using the definitions 10 and 4, the following definition is constructed as follows, which allows to know if the MIDI notes of a melody are notes of a given chord.

Definition 12 (Most MIDI notes in the melody are chord notes)**.** Let $x \in \mathbb{Z}^k$ be a melody, let $w \in \mathbb{Z}^q$ be a chord. The MIDI notes of the melody x are said to be mostly chord notes if the result of $\mathrm{MPN}(x, w)$ is equal to one, where:

$$\mathrm{MPN}(x, w) := \begin{cases} 1, \text{ if } \sum\limits_{x_i > 42} \mathrm{NinChord}(x_i, w) \geq \mathrm{TN}(x) - 1, \\ 0, \text{ otherwise.} \end{cases}$$

Definition 13 (Beats of a melody)**.** Let $x \in \mathbb{Z}^k$, the beats of the melody x will be the result of $\mathrm{BT}(x)$, where the function $BT : \mathbb{Z}^k \to \mathbb{Z}^k$ is defined below:

$$\mathrm{BT}(x) := y, \quad y \in \mathbb{Z}^k,$$

where y_j, para $j = 1, \ldots, k$, it's subject to:

$$y_j = \begin{cases} 1 & \text{if } x_j > 42, \\ 0 & \text{if } x_j = 42, \\ -1 & \text{if } x_j = 41. \end{cases}$$

Definition 14 (Total notes of the melody in the C Major scale)**.** Let $x \in \mathbb{Z}^k$ be a melody, the total number of notes of the melody x that are in the C Major scale $\mathcal{S} = \{0, 2, 4, 5, 7, 9, 11\}$ is the result of $\mathrm{TNS}(x)$, where the function TNS is defined as follows:

$$\mathrm{TNS}(x) := \sum_{\substack{x_j > 42 \\ x_j \bmod 12 \in \mathcal{S}}} 1. \tag{3}$$

Definition 15 (Neighbor Ornament)**.** Let $x \in \mathbb{Z}^3$ be a melody, let $u, w \in \mathbb{Z}^q$ chords. x is said to be an embroidery motif for the chords u, w if the result of $isNeighbor(x, u, w)$ is equal to one, where the function $isNeighbor$ is defined as follows:

$$\text{isNeighbor}(x,u,w) := \begin{cases} 1, & \text{if} \quad \begin{array}{l} (x_1 > 42) \wedge (x_2 > 42) \\ \wedge\, (x_3 > 42) \\ \wedge\, (1 \leq |x_1 - x_2| \leq 2) \\ \wedge\, (x_1 = x_3) \\ \wedge\, (x_1 \bmod 12 = u_j, \\ \quad \text{for some j=1,...,q}), \\ \wedge\, (x_3 \bmod 12 = w_\ell, \\ \quad \text{for some } \ell\text{=1,...,q}), \end{array} \\ 0, & \text{otherwise.} \end{cases}$$

Definition 16 (Escape ornament)**.** Let $x \in \mathbb{Z}^3$ be a melody, let $u, w \in \mathbb{Z}^q$ be two chords. x is said to be an escape ornament for the chords u, w if the result of $isEscape(x,u,w)$ is equal to one, where the function $isEscape$ is defined as follows:

$$\text{isEscape}(x,u,w) := \begin{cases} 1, & \text{if} \quad \begin{array}{l} (x_1 > 42) \wedge (x_2 > 42) \\ \wedge\, (x_3 > 42) \\ \wedge\, (1 \leq |x_1 - x_2| \leq 2) \\ \wedge\, (|x_2 - x_3| > 2) \\ \wedge\, ((x_2 - x_1)(x_3 - x_2) < 0) \\ \wedge\, (x_1 \bmod 12 = u_j, \\ \quad \text{for some j=1,...,q}) \\ \wedge\, (x_3 \bmod 12 = w_\ell, \\ \quad \text{for some } \ell\text{=1,...,q}), \end{array} \\ 0, & \text{otherwise.} \end{cases}$$

Definition 17 (Cambiata ornament)**.** Let $x \in \mathbb{Z}^3$ be a melody, let $u, w \in \mathbb{Z}^q$ be two chords. x is said to be a cambiata ornament for the chords u, w if the result of $isCambiata(x,u,w)$ is equal to one, where the function $isCambiata$ is defined as follows:

$$\text{isCambiata}(x,u,w) = \begin{cases} 1, \text{if} & \begin{array}{l} (x_1 > 42) \wedge (x_2 > 42) \\ \wedge\, (x_3 > 42) \\ \wedge\, (1 \leq |x_2 - x_3| \leq 2) \\ \wedge\, (|x_1 - x_2| > 2) \\ \wedge\, ((x_2 - x_1)(x_3 - x_2) < 0) \\ \wedge\, (x_1 \bmod 12 = u_j, \\ \quad \text{for some j=1,...,q}) \\ \wedge\, (x_3 \bmod 12 = w_\ell, \\ \quad \text{for some } \ell\text{=1,...,q}), \end{array} \\ 0, & \text{otherwise.} \end{cases}$$

Definition 18 (Passing tone ornament)**.** Let $x \in \mathbb{Z}^3$ be a melody, let $u, w \in \mathbb{Z}^q$ be two chords. x is said to be a passing ornament for the chords u, w if the result of $\text{isPassingTone}(x,u,w)$ is equal to one, where the function isPassingTone is defined as follows:

Algorithm 2 countAppoggiatura function to obtain the total number of appoggiatura ornaments given a list of L chords.

procedure countAppoggiatura(w,x,L)
  Let $\text{possible}, j, \text{found}, C_1, k, m \in \mathbb{Z}, s \in \mathbb{R}$.
  $\text{found} \leftarrow 0$.
  $\text{possible} \leftarrow \frac{bp}{2}$.
  $s \leftarrow 0$.
  $j \leftarrow 0$.
  while $j < \text{possible}$ **do**
    $k \leftarrow \lfloor 2j/p \rfloor + 1$.
    $m \leftarrow w_k$.
    $C_1 \leftarrow L_m$.
    $y \leftarrow x_{(2j+1:\ 2j+2)}$.
    if $\text{isAppoggiatura}(y, C_1) == 1$ **then**
      $\text{found} \leftarrow \text{found} + 1$.
    end if
    $j \leftarrow j + 1$.
  end while
  $s \leftarrow \dfrac{\text{found}}{\text{possible}}$.
  return s.
end procedure

$$\text{isPassingTone}(x,u,w) = \begin{cases} 1, \text{if} & \begin{array}{l} (x_1 > 42) \wedge (x_2 > 42) \\ \wedge\, (x_3 > 42) \\ \wedge\, (1 \leq |x_2 - x_3| \leq 2) \\ \wedge\, (1 \leq |x_1 - x_2| \leq 2) \\ \wedge\, ((x_2 - x_1)(x_3 - x_2) > 0) \\ \wedge\, (x_1 \bmod 12 = u_j, \\ \quad \text{for some j=1,...,q}) \\ \wedge\, (x_3 \bmod 12 = w_\ell, \\ \quad \text{for some } \ell\text{=1,...,q}), \end{array} \\ 0, & \text{otherwise.} \end{cases}$$

Definition 19 (Appoggiatura ornament)**.** Let $x \in \mathbb{Z}^2$ be a melody, let $w \in \mathbb{Z}^q$ be a chord. The melody x is said to be an appoggiatura ornament for the chord w if the result of $\text{isAppoggiatura}(x,w)$ is equal to one, where the function isAppoggiatura is defined as follows:

$$\text{isAppoggiatura}(x,w) := \begin{cases} 1, & \text{if} \quad \begin{array}{l} (x_1 > 42) \wedge (x_2 > 42) \\ \wedge\, (1 \leq |x_1 - x_2| \leq 2) \\ \wedge\, (x_2 \bmod 12 = w_j, \\ \quad \text{for some j=1,...,q}) \end{array} \\ 0, & \text{otherwise.} \end{cases}$$

Using definition 19, algorithm 2 is defined, which will be used to count the total appoggiatura ornaments of a given melody.

Algorithm 3 countThree function to get the total of three note embellishments given a function J and a list of chords L.

procedure countThree(w, x, J, L)
  Let possible$, j,$ found$, k, \ell, m_1, m_2 \in \mathbb{Z}, s \in \mathbb{R}, y \in \mathbb{Z}^3, C_1, C_2 \in \mathbb{Z}^q$.
  found $\leftarrow 0$.
  possible $\leftarrow \frac{pb-2}{2}$.
  $s \leftarrow 0$.
  if possible > 0 **then**
    $j \leftarrow 0$.
    while $j <$ possible **do**
      $k \leftarrow \lfloor 2j/p \rfloor + 1$.
      $\ell \leftarrow \lfloor (2j+2)/p \rfloor + 1$.
      $m_1 \leftarrow w_k$.
      $m_2 \leftarrow w_\ell$
      $C_1 \leftarrow L_{m_1}$.
      $C_2 \leftarrow L_{m_2}$.
      $y \leftarrow x_{(2j+1:2j+3)}$.
      if $J(y, C_1, C_2) == 1$ **then**
        found $\leftarrow$ found $+1$.
      end if
      $j \leftarrow j + 1$.
    end while
    $s \leftarrow \dfrac{\text{found}}{\text{possible}}$.
  end if
  return s.
end procedure

Definition 20 (Total anticipation ornaments)**.** Let $x, y \in \mathbb{Z}^{bp}$ be two melodies. The total anticipation ornaments for the melody x and the harmony y is the result of the countAnticipation function, where the countAnticipation function is defined as follows:

$$\text{countAnticipation}(x, y) := \sum_{j=1}^{b-1} 1 \quad \text{such that:}$$
$$\left(\text{PSN}(x_{\mathcal{B}(j)_p}, y_{\mathcal{B}(j+1)_1}) = 1\right)$$
$$\wedge \left(x_{\mathcal{B}(j)_p} = y_{\mathcal{B}(j+1)_1}\right).$$

Definition 21 (Melody ornaments)**.** Let: $x, y \in \mathbb{Z}^{bp}$ be two melodies, let $w \in \mathbb{Z}^b$ be a vector containing the chord indices for each bar. Let: L be a chord list. The total ornaments for the melody x is the result of $\text{MOrnaments}(w, x, L)$, which uses definitions 15, 16, 17, 18 20 and algorithms 2, 3. The function MOrnaments is defined as follows:

$$\begin{aligned}\text{MOrnaments}(w, x, y, L) := \frac{1}{6}\Big(&\text{countThree}(w, x,\\ &\text{isNeighbor}, L) + \text{countThree}(w,\\ &x, \text{isEscape}, L) + \text{arg1}(w,\\ &x, \text{isCambiata}, L)\\ &+ \text{countThree}(w, x,\\ &\text{isPassingTone}, L)\\ &+ \text{countAppoggiatura}(w, x, L)\\ &+ \text{countAnticipacion}(x, y)\Big).\end{aligned}$$

Definition 22 (Elongation melody)**.** Let $x \in \mathbb{Z}^k$ be a melody, x is said to be a elongation melody if elongation(x) is equal to one, where the elongation function is defined as follows:

$$\text{elongation}(x) := \begin{cases} 1 & \text{if } x_1 > 42 \\ & \wedge\, (x_j = 41, \forall j = 2, \ldots, k), \\ 0 & \text{otherwise.} \end{cases}$$

Definition 23 (Happiness per bar of the song)**.** Let:

$$\begin{aligned}L =&([0, 4, 7]^T, [2, 5, 9]^T, [4, 7, 11]^T, [5, 9, 0]^T,\\ &[7, 11, 2]^T, [9, 0, 4]^T, [11, 2, 5]^T, [0, 4, 7, 10]^T,\\ &[2, 5, 9, 0]^T, [4, 7, 11, 2]^T, [5, 9, 0, 3]^T, [7, 11, 2, 5]^T,\\ &[9, 0, 4, 7]^T, [11, 2, 5, 8]^T)\end{aligned}$$

Be a list with major and seventh chords. Let:

$$\begin{aligned}M =&[1,\ -2,\ -3,\ 2,\ 3,\ -1,\ -0.5,\ 1,\\ &-2,\ -3,\ 2,\ 3,\ -1,\ -0.5]^T,\end{aligned}$$

Be a vector with the happiness level of each chord corresponding to L. Let: $w \in \mathbb{Z}^b$ Be a vector containing the indices of the corresponding chord for bar j, for $j = 1, \ldots, b$. The happiness per bar of the song is the result of $\text{Happy}(w)$, where the function Happy is defined as follows:

$$(\,w) := \frac{1}{b} \sum_{j=1}^{b} M_{w_j}.$$

Definition 24 (Harmony ornaments)**.** Let $y \in \mathbb{Z}^{bp}$ be a melody, let $w \in \mathbb{Z}^b$ be a vector containing the chord indices for each bar, let L be a list of chords. The total ornaments for the harmony y is the result of $\mathrm{HOrnaments}(w, x, L)$, where the function $\mathrm{HOrnaments}$ is defined as follows:

$$\begin{aligned}\mathrm{HOrnaments}(w, y, L) := \frac{1}{5}\Big(&\mathrm{countThree}(w, y,\\ &\mathrm{isNeighbor}, L)\\ &+ \mathrm{countThree}(w, y, \mathrm{isEscape}, L)\\ &+ \mathrm{countThree}(w, y,\\ &\mathrm{isCambiata}, L)\\ &+ \mathrm{countThree}(w, y,\\ &\mathrm{isPassingTone}, L)\\ &+ \mathrm{countAppoggiatura}(w, y, L)\Big).\end{aligned}$$

5 Machine Learning

The third component of the F function was built from a machine learning model. To make the model, 262 songs divided into four genres (classic, film, pop and rock) were used. Each of these songs have two melodies (one for the treble and one for the bass).

Subsequently, 5 features (repeated notes $\mathcal{C}_1$, most common pitch class prevalence $\mathcal{C}_2$, melodic interval histogram $\mathcal{C}_3$, pitch class variety $\mathcal{C}_4$ and pitch class distribution $\mathcal{C}_5$) were extracted from each of these songs using music21 [9] with the JSymbolic [15] submodule.

Later, K-Means was used to obtain two clusters of the features obtained, with this classification, musical songs were divided into two genres: classical genre in terms of repetition of notes and contemporary genre in terms of freedom of movement.

Then, an AdaBoost regressor $h_1(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5)$ with 1 estimator was performed to model these two clusters, where 0 indicates that it corresponds to the first cluster (classic) and 1 to the second cluster (freedom of movement).

Now, since the music21 function to obtain the five features requires a MIDI file or a stream object, a function $h_2(y, z) = [\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5]^T$, $y, z \in \mathbb{Z}^{bp}$ was used to convert the song from our representation to a stream file. Since the obtained AdaBoost regressor model yields values greater than one and less than zero, the function g_4 was used as follows:

$$g_4(y, z) = \begin{cases} 0.5 & \text{if} \quad 0 < h_1\left(h_2(y, z)\right) \vee h_1\left(h_2(y, z)\right) > 1,\\ h_1\left(h_2(y, z)\right) & \text{otherwise.}\end{cases}$$

6 Optimization Problem

The generation of the desired melodies has been modeled using the following problem:

$$\mathrm{Minimize}\, F(x) = [g_1, g_2, g_3]^T, \text{ with } x \in Z^{b+2bp}.$$

For convenience, the vector x is presented as the concatenation of three vectors $w \in \mathbb{Z}^b$ and $y, z \in \mathbb{Z}^{bp}$ as $x = (w, y, z)$.

For convenience in writing for some parts of the optimization problem, the notation described in the 1 definition will be used. To model g_1, the definitions 4, 14, 21, 23 and 24 were used.

To model g_2 the definitions 4, 21, 22 and 24 were used. To model g_3 the definitions 8, 13 and g_4 function from section V were used. The functions g_1, g_2 and g_3 are minimized simultaneously and are defined as:

$$\begin{aligned}g_1(w, y, z) = 3 &- \frac{\mathrm{TNS}(y) + \mathrm{TNS}(z)}{\mathrm{TN}(y) + \mathrm{TN}(z)}\\ &- \mathrm{MOrnaments}(z, y) - \mathrm{Happy}(w)\\ &- \mathrm{HOrnaments}(y).\end{aligned}$$

$$\begin{aligned}g_2(y, z) = 2 &+ \mathrm{MOrnaments}(z, y)\\ &+ \mathrm{HOrnaments}(y)\\ &+ \frac{\mathrm{TN}(y) + \mathrm{TN}(z)}{2bp}\\ &- \sum_{j=1}^{bp-\frac{p}{2}+1} \mathrm{elongation}(z_{(j:j+\frac{p}{2}-1)})\\ &- \sum_{j=1}^{bp-\frac{p}{2}+1} \mathrm{elongation}(y_{(j:j+\frac{p}{2}-1)}).\end{aligned}$$

$$g_3(w,y,z) = \sum_{j=2}^{b} \|BT(y_{\mathcal{B}(j)}) - \mathrm{BT}(y_{\mathcal{B}(1)})\| + \sum_{\substack{j=2 \\ j \text{ par}}}^{b} \|\,\mathrm{BT}(z_{\mathcal{B}(j)}) - \mathrm{BT}(z_{\mathcal{B}(j-1)})\| + \sum_{j=2}^{b} \|\,\mathrm{SDR}(y_{\mathcal{B}(j)}) - \mathrm{SDR}(y_{\mathcal{B}(1)})\| + \sum_{\substack{j=2 \\ j \text{ par}}}^{b} \|\,\mathrm{SDR}(z_{\mathcal{B}(j)}) - \mathrm{SDR}(z_{\mathcal{B}(j-1)})\| + g_4(y,z).$$

In this way, the vector w will have as components the indices of the chords of each bar, the vector y will have the notes of the harmony and the vector z will have the notes of the melody of the song. Using the definitions 2, 3, 4, 7, 9, 11, 12 and 6, the optimization problem is presented subject to the following constraints:

- $b \in \mathbb{N}, p \in \{2,4,8,16\}$.
- $\exists y_j > 42$ for $j = 1,\ldots,bp, \exists z_\ell > 42$, for $\ell = 1,\ldots,bp$.
- $L = (L_1,\ldots,L_r)$, where L_j is a chord, for $j = 1,\ldots,r$.
- $w_1 = 1, w_b = 1$.
- $1 \leq w_j \leq r$, for $j = 2,\ldots,b-1$.
- $41 \leq y_j \leq 88$, for $j = 1,\ldots,bp$.
- $41 \leq z_j \leq 88$, for $j = 1,\ldots,bp$.
- $y_j < z_j, \quad \forall y_j, z_j$ such that $\mathrm{PSN}(y_j, z_j) = 1$.
- $\mathrm{NS}(y) + \mathrm{NS}(z) - 2 = 0$.
- $\sum_{j=0}^{b-1}\sum_{k=1}^{p} \mathrm{MF}(y_{(jp+k)}, z_{\mathcal{B}(j)}) - TN(z) = 0$
- $\sum_{j=1}^{b} \mathrm{FirstNoteFC}(y_{\mathcal{B}(j)}, L_j) - b = 0$.
- $\sum_{j=1}^{b} \mathrm{FPN}(z_{\mathcal{B}(j)}, L_j) - b = 0$.
- $\sum_{j=1}^{b} \mathrm{MPN}(y_{\mathcal{B}(j)}, L_j) - b = 0$.

7 Experimental Results

The NSGA-II multi-objective evolutionary algorithm [12] was used to solve the constrained bio-objective optimization problem described in the previous section.

For the parameters of the NSGA-II algorithm, a population of 100 individuals was used, a cross of $2b$ points [26] with crossover probability $p_c = 0.75$, a mutation with the simulated binary crossover distribution [3, 13], with a mutation probability of $p_m = 0.05$ and $\eta = 1$.

For the implementation of the code, C++, Python 3, pymoo [4] scikit-learn [24] and music21 [9] were used. For all the results, the list of chords L presented in the definition 23 was used, where L has the major and seventh chords of the C Major scale.

Different tests were carried out varying the parameters b and p, the results obtained have been placed on the next page [1], where some of the solutions found in the Pareto front are shown[2] obtained by the algorithm for each execution, where it is possible to appreciate the difference between greater happiness, greater minimalism and different genre with patron described in the functions g_1, g_2 and g_3 of the optimization problem.

The previous page will also include a video with the Pareto fronts obtained in each execution after optimization process, rotating the graph for better viewing. It should be noted that the songs obtained in the Pareto fronts of each execution are very interesting and creative, so the approach to the problem of musical composition using musical patterns and musical dimensions seems to be a promising way to expand the results and research.

[1] natanvilchis.org/micai2022/

[2] The Pareto front [20] is the set that gives a solution to a multi-objective problem.

For reasons of space, these definitions have been left out of this text.

ISSN 2007-9737

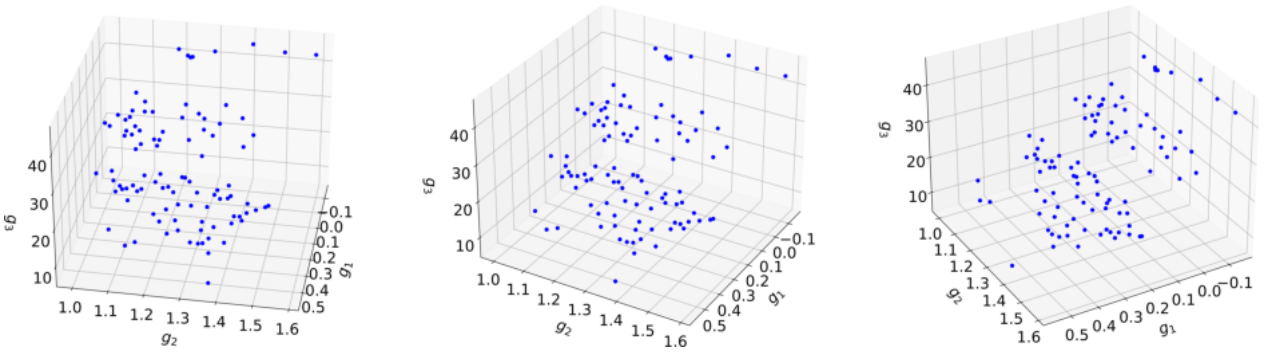


Fig. 1. Execution 1: Pareto front obtained, different views

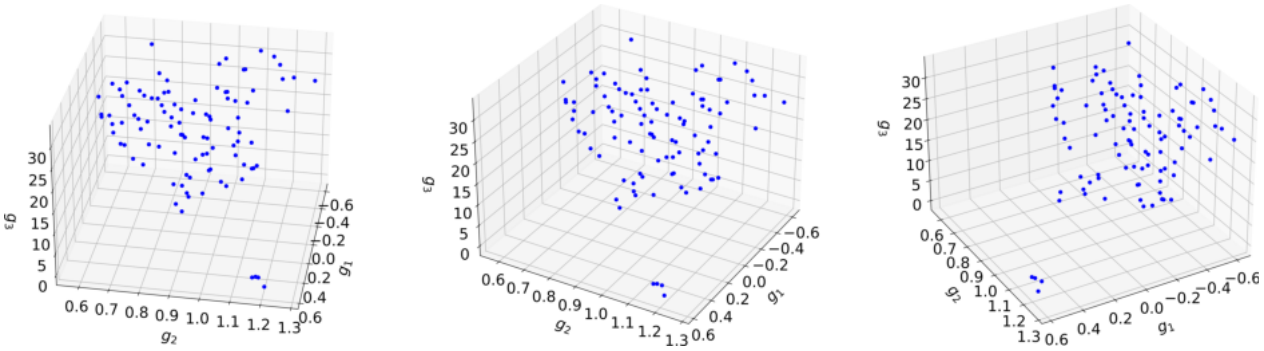


Fig. 2. Execution 2: Pareto front obtained, different views

It should be noted that given the task of converting each of the songs in the population to a stream object in real time in order to extract the features using music21 with JSymbolic, it is relatively time consuming, taking between 10 to 40 seconds to complete each generation and its evaluations. corresponding.

For example, for execution 1, the parameters $b = p = 8$ and 6600 generations were used. In the results shown on the page you can see very interesting results, for example, the extreme corresponding to the minimization of the third component of the function F shows a clear harmonic and rhythmic pattern.

Now, the extreme corresponding to the maximization of happiness shows a melodic play between the fourth and fifth bars. Finally, the extreme corresponding to the maximization of minimalism can be seen that the previously defined ornaments are avoided in a greater way.

The figure 1 shows the Pareto front obtained for execution 1 after the optimization process with the mentioned parameters and generations.

It can be seen that there are high values for the g_3 function, where it indicates that the genre can be freer with respect to rhythmic and harmonic patterns.

In addition, the figure 1 highlights the conflict between the functions g_1, g_2 and g_3, showing that there is a relationship between happiness, minimalism and the choice of musical genre. Now, for execution 2, the parameters $b = p = 8$ were used, with $11,200$ generations.

Very promising results are shown in the results shown on the previous page; for example, with respect to function g_3, a clear difference is verified between the genre of the two songs, where one has a pattern at each bar and the other is freer. In the figure 2 it can be seen that there are four songs whose value in the objective function F are

minimum for the third function; in addition, these songs have a value between 0.4 and 0.6 for the function g_1; that is, the emotion level of happiness is low, since the higher g_1 value, less happiness level in the song.

For execution 3, the parameters $b = 8$ and $p = 4$ with $21,700$ generations were used. As you can see in the videos on the previous page, they are different between the genres, where one is calmer in the harmonic part and the other has much more movement.

Now, the difference between decorations between the extremes of happiness and minimalism are also remarkable.

In figure 3 you can see a less disperse Pareto front compared to previous executions. In addition, the compromise between the functions g_1, g_2 and g_3 can be observed.

In figure 4, you can see the Parallel Coordinate Plots of each execution, where each line represents a solution and its respective evaluation in functions g_1, g_2 and g_3.

It can be seen that between the functions g_1 and g_2 there is a conflict: if the value of g_1 increases, then the value of g_2 decreases and vice versa; in other words, if the level of happiness increases then the level of minimalism decreases and vice versa.

Now, an interesting behavior can be observed between a) and b) of figure 4, where, despite having the same parameters ($b = p = 8$), there are different densities for the lines between g_2 and g_3.

This suggests that you can have songs from a different musical genre but with the same levels of happiness and minimalism.

Finally, in c), figure 4, there is an remarkable inverse relationship between happiness and minimalism. In another way, the lines between the function g_2 and g_3 have a lower density compared to a) and b).

This suggests that for songs with similar levels of happiness and minimalism, different variations of musical genres are possible.

8 Conclusions

In this work, the automatic musical composition problem was modeled through optimization problem. The aim was to create melodies where a set of compromise solutions (Pareto front) was established regarding the level of happiness, as the first objective function, simplicity of ornaments as the second optimization function and the musical genre as a third objective. For the creation of the model, emotional musical theory and related works were used, promising results were obtained by finding auditory diverse elements on the Pareto front in each execution.

The graphs obtained from the Pareto fronts shown in figures 1, 2 and 3 mainly show that there is a relationship between the functions g_1, g_2 and g_3; In addition, the larger the values of b,p (as in figures 1 and 2), the dispersion of the points is greater, since the search space is expanding, compared to figure 3, where the dispersion of the dots is smaller, which appears to form a path. The dispersion of the points in the Pareto fronts is a good indicator that the functions g_1, g_2 and g_3 have a relationship between them and also explore different musical aspects.

In figure 4 it can be seen that there is a conflict between the functions g_1, g_2 and g_3; In other words, to compose a song, it must be considered that increasing one musical aspect (happiness, minimalism or musical genre) will affect the values of other aspects.

We sought to simplify the most important aspects of music, trying to maintain a balance between the creative capacity of the results and avoiding complexity in the problem; for example, for the musical patterns the first four terms of the function were made.

Now, the use of machine learning shows an improvement compared to using the two objectives of the F function only; however, the runtime feature extraction of the optimization algorithm is a costly task since you have to convert each song to a stream object or MIDI file to use JSymbolic.

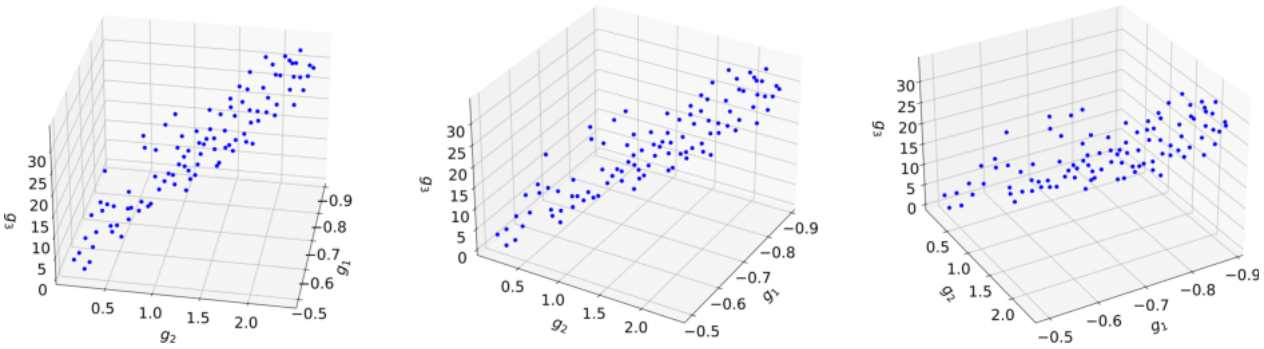


Fig. 3. Execution 3: Pareto front obtained, different views

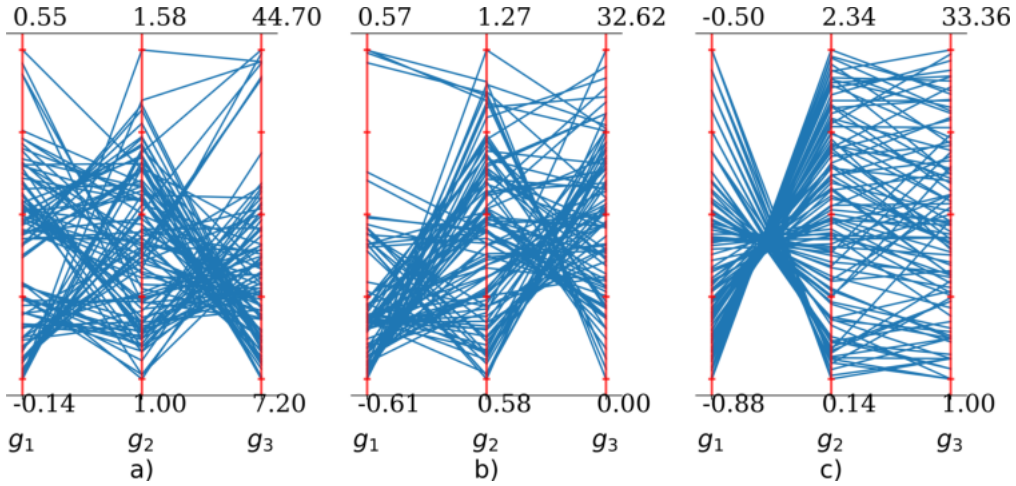


Fig. 4. Parallel Coordinate Plots of the three executions: a) execution 1, b) execution 2, c) execution 3

9 Future Work

The results obtained in the present work are interesting and promising, for this purpose three ways are proposed as future work to improve the results given the learning trajectory with this work:

– Change the song representation to one that allows multiple melodies, for example multidimensional ordered set used in [17]. With this improvement proposal, it is intended to generalize a song in a better way, allowing to explore more about the space of polyphonic songs to improve the results. However, changing the representation may have to solve other problems, such as identifying and separating the notes corresponding to each melody for individual analysis (melody structure, melody patterns, ornaments, emotional level, among others).

– Use machine learning to extract rhythmic patterns, melodic patterns, and patterns that evoke some emotion. With this improvement proposal it is intended to improve the structure of the song. To make this possible, it is necessary to have a fast pattern detection algorithm, in terms of complexity. To make this possible, it is recommended to study in depth the existing algorithms [16, 6, 18, 2] to make an improvement that allows reducing their complexity.

– Study emotional music theory [11] in depth to model each dimension mathematically and its interaction in the evocation of emotions.

References

1. **Association of Musical Electronics Industry AMEI and MIDI Manufacturers Association MMA (2020).** Universal MIDI packet (UMP) format and MIDI 2.0 protocol.
2. **Bertin-Mahieux, T. (2013).** Large-scale pattern discovery in music. Ph.D. thesis, Columbia University.
3. **Blank, J. (2020).** pymoo - mutation.
4. **Blank, J. (2020).** pymoo: Multi-objective optimization in python.
5. **Cataltepe, Z., Yaslan, Y., Sonmez, A. (2007).** Music genre classification using MIDI and audio features. European Association for Signal Processing, Journal on Advances in Signal Processing, Vol. 2007, No. 1. DOI: 10 .1155/2007/36409.
6. **Collins, T. E. (2011).** Improved methods for pattern discovery in music, with applications in automated stylistic composition. Ph.D. thesis, Faculty of Mathematics, Computing and Technology, The Open University.
7. **Cooke, D. (1959).** The language of music. Oxford University Press.
8. **Cuenca-Rodríguez, M. E., McKay, C. (2021).** Exploring musical style in the anonymous and doubtfully attributed mass movements of the Coimbra manuscripts: A statistical and machine learning approach. Journal of New Music Research, Vol. 50, No. 3, pp. 199–219. DOI: 10.1080/09298215.2020.1870505.
9. **Cuthbert, M. S. (2022).** music21: A toolkit for computer-aided musicology.
10. **Dannenberg, R. B., Hu, N. (2003).** Pattern discovery techniques for music audio. Conference Proceedings: Third International Conference on Music Information Retrieval, Taylor and Francis, pp. 63–70.
11. **de María, M. (2021).** 18 dimensiones emocionales, decodificando las emociones de la música desde 18 dimensiones técnicas.
12. **Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002).** A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 182–197. DOI: 10.1109/4235.996017.
13. **Deb, K., Sindhya, K., Okabe, T. (2007).** Self-adaptive simulated binary crossover for real-parameter optimization. Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, Association for Computing Machinery, pp. 1187–1194. DOI: 10.1145/1276958.1277190.
14. **Herremans, D., Chew, E. (2017).** Morpheus: Generating structured music with constrained patterns and tension. IEEE Transactions on Affective Computing, Vol. 10, No. 4, pp. 510–523. DOI: 10.1109/taffc.2017.2737 984.
15. **McKay, C., Fujinaga, I. (2006).** jSymbolic: A feature extractor for midi files. International Conference on Mathematics and Computing.
16. **Meredith, D. (2013).** COSIATEC and SIATECCompress: Pattern discovery by geometric compression. International Society for Music Information Retrieval Conference, , No. 14.
17. **Meredith, D. (2014).** Compression-based geometric pattern discovery in music. 4th International Workshop on Cognitive Information Processing (CIP), pp. 1–6. DOI: 10.1109/CIP.2014.6844503.
18. **Meredith, D. (2019).** RECURSIA-RRT: Recursive translatable point-set pattern discovery with removal of redundant translators. Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 485–493.
19. **Meredith, D., Lemström, K., Wiggins, G. A. (2002).** Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. Journal of New Music Research, Vol. 31, No. 4, pp. 321–345. DOI: 10.1076/jnmr.31.4.321.14162.

20. **Miettinen, K. (2012).** Nonlinear multiobjective optimization. Springer Science and Business Media, Vol. 12. DOI: 10.1007/978-1-4615-5563-6.

21. **Miranda-Reck, E. (2021).** Handbook of artificial intelligence for music: Foundations, advanced approaches, and developments for creativity. Springer Nature. DOI: 10.1007/978-3-030-72116-9.

22. **Ren, I., Volk, A., Swierstra, W., Veltkamp, R. C. (2020).** A computational evaluation of musical pattern discovery algorithms. DOI: 10.48550/ARXIV.2010.12325.

23. **Rodríguez Alvira, J. (2022).** Funciones armónicas: Notas de adorno. www.teoria.com/es/aprendizaje/funciones/adorno/index.php.

24. **Scikit-learn (2022).** Machine learning in Python. scikit-learn 1.1.1, documentation. www.teoria.com/es/aprendizaje/funciones/adorno/index.php.

25. **Scirea, M., Togelius, J., Eklund, P., Risi, S. (2017).** Affective evolutionary music composition with MetaCompose. Genetic Programming and Evolvable Machines, Vol. 18, No. 4, pp. 433–465. DOI: 10.1007/s10710-017-9307-y.

26. **Umbarkar, A. J., Sheth, P. D. (2015).** Crossover operators in genetic algorithms: A review. ICTACT Journal on Soft Computing, Vol. 6, No. 1, pp. 1083–1092. DOI: 10.21917/ijsc.2015.0150.

27. **Vatolkin, I., McKay, C. (2022).** Multi-objective investigation of six feature source types for multi-modal music classification. Transactions of the International Society for Music Information Retrieval, Ubiquity Press, Vol. 5, No. 1, pp. 1–19. DOI: 10.5334/tismir.67.

Article received on 14/06/2022; accepted on 18/09/2022.
Corresponding author is Giovanni Guzmán.