ISSN 1870-4069

Modelado del problema de composición musical polifónica automática como un problema de optimización biobjetivo

Natan Vilchis, Adriana Lara

Instituto Politécnico Nacional,
Escuela Superior de Física y Matemáticas,
México

nvilchist1300@alumno.ipn.mx, alaral@ipn.mx

Resumen. En este trabajo se propone un modelo matemático que plantea la composición musical asistida por computadora como un problema de optimización. Se consideran dos funciones objetivo en conflicto para representar las características deseadas de la melodía a crear. Se muestran ejemplos de la aplicación de este modelo utilizando un algoritmo evolutivo multiobjetivo para aproximar las soluciones del problema biobjetivo. Los resultados de este trabajo pueden escucharse en la página web que proporcionamos para tal fin.

Palabras clave: Composición musical automática, algoritmo genético, optimización multiobjetivo.

Modeling the Automatic Polyphonic Music Composition Problem as a Bi-Objective Optimization Problem

Abstract. This work proposes a mathematical model that states computer-assisted musical composition as an optimization problem. Two conflicting objective functions represent the desired characteristics of the melody to be created. Examples of the application of this model using a multi-objective evolutionary algorithm to approximate the solutions of the biobjective problem can be heard on the web page that we provide for this purpose.

Keywords: Automatic music creation, genetic algorithm, multi-objective optimization.

1. Introducción

La inteligencia artificial aplicada a la música ha explorado diversas herramientas como el deep learning, música generativa, computación cuántica, entre otros [17]. En el presente trabajo se modela el problema de la composición musical asistida por computadora con el lenguaje de la optimización matemática, teniendo como objetivo generar canciones con una alegría máxima, mientras se busca un minimalismo en los adornos; para ello se utilizó la teoría musical emocional de Mauro de María [11], teoría musical clásica [6], estudio de patrones musicales [12, 5, 15, 14, 18, 7, 2, 13] y trabajos relacionados estado del arte como MetaCompose [20] y MorpheuS [10] que proporcionaron elementos valiosos para nuestro modelo.

El presente trabajo es una combinación de elementos existentes, tales como la definición 15 (basada en la ecuación 4 de [20]), las restricciones 17, 18, 19 y 20 del problema de optimización biobjetivo (basadas en [10] junto con combinación y simplificación de [12, 5, 15, 14, 18, 7, 2, 13]) y con aportes originales basados en teoría musical emocional [11] presentes en los demás elementos del problema de optimización.

2. Conceptos básicos y elementos fundamentales del modelo

Hemos representado una canción como un vector $x \in \mathbb{Z}^{b+2bp}$, donde las primeras b componentes denotan el índice del acorde para el correspondiente compás y las siguientes bp componentes representan las notas de la armonía; finalmente, las últimos bp componentes, representan las notas de la melodía.

De esta manera, cada melodía tiene $b \in \mathbb{N}$ compases y a su vez, cada compás está dividido en p particiones, con $p \in \{2, 4, 8, 16\}$; además, cada componente $41 \leq x_j \leq 88$, para $j \in \{b+1, \dots, 2bp\}$ es una nota MIDI [1]; donde la nota MIDI 41 será tomada como una alargación de la nota anterior, mientras que la nota MIDI 42 se utilizará como silencio. El siguiente par de pentagramas muestra la representación de una canción correspondiente al vector $x^0 \in \mathbb{Z}^{4+2(4)(8)}$ dado por:

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

La definición 2 usa la definición de una nota MIDI [1], las definiciones 24, 25, 26, 27, 28 y 29 fueron construidas a partir de la teoría musical clásica de adornos musicales [19], mientras que las definiciones restantes fueron creadas a partir del modelo musical emocional [11] y adaptadas para el problema abordado del presente trabajo. A continuación se presentan definiciones necesarias para el modelo:

Definición 1 (Nota musical) Una nota musical w será un elemento del conjunto W definido por: $W := \{C, C\sharp, D, D\sharp, E, F, F\sharp, G, G\sharp, A, A\sharp, B\}$. La correspondiente nota musical numerada de w está dada por la función:

$$\mathrm{noteToNumbered} : W \to \mathbb{Z};$$

Descrita en la siguiente tabla:

Nota musical	C	$C\sharp$	D	$D\sharp$	E	F	$F\sharp$	G	$G\sharp$	A	$A\sharp$	B
Notación latina	Do	Do♯	Re	Re♯	Mi	Fa	Fa♯	Sol	Sol♯	La	La♯	Si
Nota musical numerada	0	1	2	3	4	5	6	7	8	9	10	11

Definición 2 (Melodía) El vector $x = (x_1, x_2, \ldots, x_m)^T \in \mathbb{Z}^m$ se dice ser una melodía, si cada x_i es una nota MIDI, para $i = 1, \ldots, m$.

Definición 3 (Melodía del compás j-ésimo de una melodía) Sea $x \in \mathbb{Z}^{bp}$ una melodía, con $b, p \in \mathbb{N}$. La melodía del compás j-ésimo $x_{\mathcal{B}(j)}$ corresponde al subvector de x dado como: $x_{\mathcal{B}(j)} := x_{(1+p(j-1):jp)}, \quad j = 1, \ldots, b.$

Definición 4 (Acorde) El vector $x = (x_1, x_2, \ldots, x_q)^T \in \mathbb{Z}^q$ se dice ser un acorde, si cada x_i es una nota musical numerada, para $i = 1, \ldots, q$ y además se verifica que $x_i \neq x_j \quad \forall i \neq j$.

Definición 5 (Melodía con buen inicio) Sea $x \in \mathbb{Z}^k$ una melodía. La melodía x se dice tener un buen inicio si la función $NS : \mathbb{Z}^k \to \{0, 1\}$ evaluada en la melodía x es igual igual a uno, donde la función NS se define como sigue:

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

Definición 6 (Melodía de notas reales) Sea $x \in \mathbb{Z}^k$ una melodía tal que $\exists x_j > 42$. La melodía de notas reales de la melodía x es el resultado de $MR(x)$, donde la función $MR : \mathbb{Z}^k \to \mathbb{Z}^q, q \leq k$, se define como sigue:

$$MR(x) := [x_1, x_2, \ldots, x_j, \ldots, x_{k-1}, x_k]^T, \quad x_j > 42, j = 1, \ldots, k.$$

Definición 7 (Total de notas reales en melodía) Sea $x \in \mathbb{Z}^k$ una melodía, el número total de notas reales de una melodía es el resultado de $TN(x)$, donde la función $TN : \mathbb{Z}^k \to \mathbb{Z}$ está definida como sigue:

$$TN(x) := \text{número de elementos } x_j \text{ mayores a 42}, \quad j = 1, \ldots, k.$$

Definición 8 (Vector diferencia) Sea $x \in \mathbb{Z}^n, n \geq 2$, el vector diferencia correspondiente a x está determinado por la función $D : \mathbb{Z}^n \to \mathbb{Z}^{n-1}$, donde:

$$D(x) := x_{(2:n)} - x_{(1:n-1)}.$$

Algoritmo 1 Función repeatMelody para obtener la melodía repetición de una melodía.

Función repeatMelody(x)
    Sean $m, j \in \mathbb{N}, \text{saveNote} \in \mathbb{Z}, y \in \mathbb{Z}^k$.
    Hacer $m \leftarrow \min\limits_{\substack{x_j > 42 \\ 1 \leq j \leq k}} (j)$.
    Hacer $\text{saveNote} \leftarrow x_m$.
    Hacer $j \leftarrow m + 1$.
    Hacer $y \leftarrow x$.
    Mientras $j \leq k$ **Hacer**
        Si $x_j == 41 \vee x_j == 42$ **Entonces**
            Hacer $y_j \leftarrow \text{saveNote}$.
        En caso contrario
            Hacer $\text{saveNote} \leftarrow x_j$.
        Terminar Si
        Hacer $j \leftarrow j + 1$.
    Terminar Mientras
    Devolver y.
Terminar Función

Definición 9 (Máximo salto de una melodía) Sea $x \in \mathbb{Z}^k$ una melodía. El máximo salto de la melodía x es el resultado de $MS(x)$, donde la función $MS : \mathbb{Z}^k \to \mathbb{Z}$ se define como sigue:

$$MS(x) := \begin{cases} 0, \text{ si } \quad TN(x) < 2, \\ \|D(MR(x))\|_\infty, \quad \text{en caso contrario.} \end{cases}$$

Definición 10 (Vector signo) Sea $x \in \mathbb{Z}^r$, el vector signo correspondiente a x es el resultado de $S(x)$, donde la función $S : \mathbb{Z}^r \to \mathbb{Z}^r$ se define como sigue:

$$S(x) := [\text{sgn}(x_1),\ \text{sgn}(x_2), \ldots,\ \text{sgn}(x_r)]^T.$$

Definición 11 (Melodía repetición de una melodía) Sea $x \in \mathbb{Z}^k$ una melodía tal que $\exists x_j > 42$. La melodía repetición $y \in \mathbb{Z}^k$ de la melodía x es el resultado de $\text{repeatMelody}(x)$, donde la función $\text{repeatMelody} : \mathbb{Z}^k \to \mathbb{Z}^k$ se describe en el algoritmo 1.

Definición 12 (Conjunto de notas disonantes de una melodía) Sea $x \in \mathbb{Z}^k$ una melodía, tal que $\exists x_j > 42$. El conjunto de notas disonantes para la melodía x es el resultado de $ND(x)$, donde la función ND se define como:

$$ND(x) := \{\text{MIDIToNumbered}(x_i - 1) \forall x_i > 42\} \bigcup \{\text{MIDIToNumbered}(x_i + 1) \forall x_i > 42\}.$$

Definición 13 (Nota adecuada para la melodía) Sea x una nota MIDI, sea $w \in \mathbb{Z}^k$ una melodía tal que $\exists w_j > 42$. Se dice que x es una nota adecuada para la melodía w si el resultado de $MF(x, w)$ es igual a uno, donde la función MF se define como sigue:

$$MF(x, w) := \begin{cases} 1, \text{ si } x \notin ND(w), \\ 0, \text{ en caso contrario.} \end{cases}$$

Definición 14 (Función SDR) Sea $x \in \mathbb{Z}^t$ una melodía con $t \geq 2$, tal que $\exists x_j > 42$, La función $SDR : \mathbb{Z}^t \to \mathbb{Z}^{t-1}$ se define a continuación:

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

Definición 15 (Primera nota de la melodía es nota fundamental del acorde) Sea $x \in \mathbb{Z}^k$ una melodía tal que $\exists x_j > 42$, sea $w \in \mathbb{Z}^q$ un acorde. Se dice que la melodía x tiene como primer nota a la nota fundamental del acorde w si el resultado de $\text{FirstNoteFC}(x, w)$ es igual a uno, donde la función FirstNoteFC se describe a continuación:

$$\text{FirstNoteFC}(x, w) := \begin{cases} 1, & \text{si} \quad \text{MIDIToNumbered}(x_r) = w_1, \quad r = \min\limits_{\substack{x_j > 42 \\ j=1,\ldots,k}} (j), \\ 0, & \text{en otro caso.} \end{cases}$$

Definición 16 (Nota MIDI en acorde) Sea x una nota MIDI, sea $w \in \mathbb{Z}^q$ un acorde. La nota MIDI x será una nota del acorde w si el resultado de $\text{NInChord}(x, w)$ es igual a uno, donde la función NInChord se define a continuación:

$$\text{NInChord}(x, w) := \begin{cases} 1, & \text{si } \text{MIDIToNumbered}(x) = w_j, \text{ para algún } j = 1, \ldots, p, \\ 0, & \text{en otro caso.} \end{cases}$$

Definición 17 (Al menos el 50 % de las notas MIDI son notas del acorde) Sea $x \in \mathbb{Z}^k$ una melodía, sea $w \in \mathbb{Z}^q$ un acorde. Se dice que al menos el cincuenta porciento de las notas de la melodía x son notas del acorde w si el resultado de $FPN(x, w)$ es igual a uno, donde la función FPN se define a continuación:

$$FPN(x, w) := \begin{cases} 1, \text{ si} & \left(\sum\limits_{x_j > 42} \text{NinChord}(x_j, w) \right) / \, TN(x) \geq 0.5, \\ 0, & \text{en caso contrario.} \end{cases}$$

Definición 18 (Mayoría de las notas MIDI de la melodía son notas del acorde) Sea $x \in \mathbb{Z}^k$ una melodía, sea $w \in \mathbb{Z}^q$ un acorde. Se dice que las notas MIDI de la melodía x son mayormente notas del acorde si el resultado de $MPN(x, w)$ es igual a uno, donde:

$$MPN(x, w) := \begin{cases} 1, \text{ si} & \sum\limits_{x_i > 42} \text{NinChord}(x_i, w) \geq TN(x) - 1, \\ 0, & \text{en caso contrario.} \end{cases}$$

Definición 19 (Pulsaciones de una melodía) Sea $x \in \mathbb{Z}^k$, las pulsaciones de la melodía x será el resultado de $BT(x)$, donde la función $BT : \mathbb{Z}^k \to \mathbb{Z}^k$ se define a continuación:

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

donde y_j, para $j = 1, \ldots, k$, está sujeto a:

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

Definición 20 (Nota MIDI en la escala de Do Mayor) Sea $x \in \mathbb{Z}$ una nota MIDI, se dice que la nota x está en la escala de Do Mayor (denotada como $\mathcal{S} = \{0, 2, 4, 5, 7, 9, 11\}$) si el resultado de NEC(x) es igual a uno, donde la función NEC se define como sigue:

$$NEC(x) := \begin{cases} 1, & \text{si} \quad \text{MIDIToNumbered}(x) \in \mathcal{S}, \\ 0, & \text{en caso contrario.} \end{cases}$$

Definición 21 (Total de notas de la melodía en la escala de Do Mayor) Sea $x \in \mathbb{Z}^k$ una melodía, el número total de notas de la melodía x que se encuentran en la escala de Do Mayor es el resultado de $TNS(x)$, donde la función TNS se define como sigue:

$$TNS(x) := \sum_{x_j > 42} NEC(x_j).$$

Definición 22 (Notas pulsadas de forma simultánea) Sean $x, w \in \mathbb{N}$ dos notas MIDI, se dice que las notas x, w son pulsadas de forma simultánea si el resultado de $PSN(x, w)$ es uno, donde la función PSN está definida de la siguiente manera:

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

Definición 23 (Total de notas pulsadas simultáneamente de dos melodías) Sean $x, w \in \mathbb{Z}^{bp}$ dos melodías, el número total de notas pulsadas simultáneamente de las melodías x, w es el resultado de $TNC(x, w)$, donde la función TNC está definida de la siguiente manera:

$$TNC(x, w) := \sum_{j=0}^{b-1} \sum_{k=1}^{p} PSN(x_{(jp+k)}, w_{(jp+k)}).$$

Definición 24 (Adorno bordadura) Sea $x \in \mathbb{Z}^3$ una melodía, sean $u, w \in \mathbb{Z}^q$ acordes. Se dice que x es un adorno de bordadura para los acordes u, w si el resultado de $\text{esBordadura}(x, u, w)$ es igual a uno, donde la función esBordadura se define como sigue:

$$\text{esBordadura}(x, u, w) := \begin{cases} 1, & \text{si} \quad \begin{array}{l} (x_1 > 42) \wedge (x_2 > 42) \wedge \\ (x_3 > 42) \wedge (1 \leq |x_1 - x_2| \leq 2) \\ \wedge\, (x_1 = x_3) \wedge (\text{MIDIToNumbered}(x_1) = u_j, \text{para} \\ \text{algún j=1,\ldots,p}), \wedge(\text{MIDIToNumbered}(x_3) = w_\ell, \text{para} \\ \text{algún } \ell\text{=1,\ldots,p}), \end{array} \\ 0, & \text{en caso contrario.} \end{cases}$$

Definición 25 (Adorno echapée) Sea $x \in \mathbb{Z}^3$ una melodía, sean $u, w \in \mathbb{Z}^q$ dos acordes. Se dice que x es un adorno echapée para los acordes u, w si el resultado de $\text{esEchapee}(x, u, w)$ es igual a uno, donde la función esEchapee se define como sigue:

$$\text{esEchapee}(x,u,w) := \begin{cases} 1, & \text{si} \quad \begin{array}{l} (x_1 > 42) \wedge (x_2 > 42) \wedge (x_3 > 42) \\ \wedge\, (1 \leq |x_1 - x_2| \leq 2) \wedge (|x_2 - x_3| > 2) \\ \wedge\, ((x_2 - x_1)(x_3 - x_2) < 0) \\ \wedge\, (\text{MIDIToNumbered}(x_1) = u_j, \text{para} \\ \text{algún j=1,\dots,p}) \\ \wedge\, (\text{MIDIToNumbered}(x_3) = w_\ell, \text{para} \\ \text{algún } \ell\text{=1,\dots,p}), \end{array} \\ 0, & \text{en caso contrario.} \end{cases}$$

Definición 26 (Adorno cambiata) Sea $x \in \mathbb{Z}^3$ una melodía, sean $u, w \in \mathbb{Z}^q$ dos acordes. Se dice que x es un adorno cambiata para los acordes u, w si el resultado de $\text{esCambiata}(x, u, w)$ es igual a uno, donde la función esCambiata se define como sigue:

$$\text{esCambiata}(x,u,w) = \begin{cases} 1, & \text{si} \quad \begin{array}{l} (x_1 > 42) \wedge (x_2 > 42) \wedge (x_3 > 42) \\ \wedge\, (1 \leq |x_2 - x_3| \leq 2) \wedge (|x_1 - x_2| > 2) \\ \wedge\, ((x_2 - x_1)(x_3 - x_2) < 0) \\ \wedge\, (\text{MIDIToNumbered}(x_1) = u_j, \text{para} \\ \text{algún j=1,\dots,p}) \\ \wedge\, (\text{MIDIToNumbered}(x_3) = w_\ell, \text{para} \\ \text{algún } \ell\text{=1,\dots,p}), \end{array} \\ 0, & \text{en caso contrario.} \end{cases}$$

Definición 27 (Adorno de paso) Sea $x \in \mathbb{Z}^3$ una melodía, sean $u, w \in \mathbb{Z}^q$ dos acordes. Se dice que x es un adorno de paso para los acordes u, w si el resultado de $\text{esDePaso}(x, u, w)$ es igual a uno, donde la función esDePaso se define como sigue:

$$\text{esDePaso}(x,u,w) = \begin{cases} 1, & \text{si} \quad \begin{array}{l} (x_1 > 42) \wedge (x_2 > 42) \wedge (x_3 > 42) \\ \wedge\, (1 \leq |x_2 - x_3| \leq 2) \wedge (1 \leq |x_1 - x_2| \leq 2) \\ \wedge\, ((x_2 - x_1)(x_3 - x_2) > 0) \\ \wedge\, (\text{MIDIToNumbered}(x_1) = u_j, \text{para} \\ \text{algún j=1,\dots,p}) \\ \wedge\, (\text{MIDIToNumbered}(x_3) = w_\ell, \text{para} \\ \text{algún } \ell\text{=1,\dots,p}), \end{array} \\ 0, & \text{en caso contrario.} \end{cases}$$

Definición 28 (Adorno apoyatura) Sea $x \in \mathbb{Z}^2$ una melodía, sea $w \in \mathbb{Z}^q$ un acorde. Se dice que la melodía x es un adorno apoyatura para el acorde w si el resultado de $\text{esApoyatura}(x, w)$ es igual a uno, donde la función esApoyatura se define como sigue:

$$\text{esApoyatura}(x,w) := \begin{cases} 1, & \text{si} \quad \begin{array}{l} (x_1 > 42) \wedge (x_2 > 42) \wedge (1 \leq |x_1 - x_2| \leq 2) \\ \wedge\, (\text{MIDIToNumbered}(x_2) = w_j, \text{para} \\ \text{algún j=1,\dots,p}), \end{array} \\ 0, & \text{en caso contrario.} \end{cases}$$

Algoritmo 2 Función countThree para obtener el total de adornos de tres notas dada una función J y una lista de acordes L.

Función countThree(w,x,J,L)
  Sean $\text{possible}, j, \text{found}, k, \ell, m_1, m_2 \in \mathbb{Z}, s \in \mathbb{R}, y \in \mathbb{Z}^3, C_1, C_2 \in \mathbb{Z}^q$.
  Hacer $\text{found} \leftarrow 0$.
  Hacer $\text{possible} \leftarrow \frac{pb-2}{2}$.
  Hacer $s \leftarrow 0$.
  Si $\text{possible} > 0$ **Entonces**
    Hacer $j \leftarrow 0$.
    Mientras $j < \text{possible}$ **Hacer**
      Hacer $k \leftarrow \lfloor \frac{2j}{p} \rfloor + 1$.
      Hacer $\ell \leftarrow \lfloor \frac{2j+2}{p} \rfloor + 1$.
      Hacer $m_1 \leftarrow w_k$.
      Hacer $m_2 \leftarrow w_\ell$
      Hacer $C_1 \leftarrow L_{m_1}$.
      Hacer $C_2 \leftarrow L_{m_2}$.
      Hacer $y \leftarrow x_{(2j+1:2j+3)}$.
      Si $J(y, C_1, C_2) == 1$ **Entonces**
        Hacer $\text{found} \leftarrow \text{found} + 1$.
      Terminar Si
      Hacer $j \leftarrow j + 1$.
    Terminar Mientras
    Hacer $s \leftarrow \text{found} / \text{possible}$.
  Terminar Si
  Devolver s.
Terminar Función

Definición 29 (Total de adornos de anticipación) Sea $x, y \in \mathbb{Z}^{bp}$ dos melodías. El total de adornos de anticipación para la melodía x y la armonía y es el resultado de la función $\text{contarAnticipacion}$, donde la función $\text{contarAnticipacion}$ se define como sigue:

$$\text{contarAnticipacion}(x, y) := \sum_{j=1}^{b-1} 1 \quad \text{tales que} \quad \begin{array}{l} \left(PSN(x_{\mathcal{B}(j)_p}, y_{\mathcal{B}(j+1)_1}) = 1\right) \\ \wedge \left(x_{\mathcal{B}(j)_p} = y_{\mathcal{B}(j+1)_1}\right). \end{array}$$

Definición 30 (Total de adornos de tres notas de una melodía) Sea $x \in \mathbb{Z}^{bp}$ una melodía, sea $w \in \mathbb{Z}^b$ un vector con los índices de acordes para cada compás de la melodía x, sea J una función de adorno de tres notas, sea L una lista de acordes. El total de adornos de tres notas presentes en la melodía x es el resultado de $\text{countThree}(w, x, J, L)$, donde la función countThree queda descrita en el algoritmo 2.

Definición 31 (Adornos para la melodía) Sean $x, y \in \mathbb{Z}^{bp}$ dos melodías, sea $w \in \mathbb{Z}^b$ un vector que contiene los índices de acordes para cada compás, sea L una lista de acordes. El total de adornos para la melodía x es el resultado de OrnamentsM(w,x,L), donde la función OrnamentsM se define de la siguiente manera:

$$\begin{aligned}\mathrm{OrnamentsM}(w,x,y,L) := \frac{1}{6}\Big(&\mathrm{countThree}(w,x,\mathrm{esBordadura},L)\\ &+\mathrm{countThree}(w,x,\mathrm{esEchapee},L)\\ &+\mathrm{countThree}(w,x,\mathrm{esCambiata},L)\\ &+\mathrm{countThree}(w,x,\mathrm{esDePaso},L)\\ &+\mathrm{contarApoyatura}(w,x,L)\\ &+\mathrm{contarAnticipacion}(x,y)\Big).\end{aligned}$$

Definición 32 (Melodía alargación) Sea $x \in \mathbb{Z}^k$ una melodía, se dice que x es una melodía alargación si alargacion(x) es igual a uno, donde la función alargacion se define como sigue:

$$\mathrm{alargacion}(x) := \begin{cases} 1, & \text{si} \quad (x_1 > 42) \wedge (x_j = 41), j = 2, \ldots, k, \\ 0, & \text{en otro caso.} \end{cases}$$

Definición 33 (Total de notas blancas en la melodía) Sea $x \in \mathbb{Z}^{bp}$ una melodía, la cantidad total de notas blancas en la melodía es el resultado de $\mathrm{Half}(x)$, donde la función Half se define como sigue:

$$\mathrm{Half}(x) := \sum_{j=1}^{bp-\frac{p}{2}+1} \mathrm{alargacion}(x_{(j:j+\frac{p}{2}-1)}).$$

Definición 34 (Alegría por compás de la canción) Sea:

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

Una lista con acordes de mayores y séptimos, sea:

$$M = [1, -2, -3, 2, 3, -1, -0.5, 1, -2, -3, 2, 3, -1, -0.5]^T.$$

Un vector con el nivel de alegría de cada acorde correspondiente a L. Sea $w \in \mathbb{Z}^b$ un vector que contiene los índices del acorde correspondiente para el compás j, para $j = 1, \ldots, b$. La alegría por compás de la canción es el resultado de $\mathrm{Happy}(w)$, donde la función Happy se define como sigue:

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

Algoritmo 3 Función $\mathrm{contarApoyatura}$ para obtener el total de adornos de apoyatura dada una lista de acordes L.

Función $\mathrm{contarApoyatura}$(w,x,L)
    Sean $\mathrm{possible}, j, \mathrm{found}, C_1, k, m \in \mathbb{Z}, s \in \mathbb{R}$.
    Hacer $\mathrm{found} \leftarrow 0$.
    Hacer $\mathrm{possible} \leftarrow \frac{bp}{2}$.
    Hacer $s \leftarrow 0$.
    Hacer $j \leftarrow 0$.
    Mientras $j < \mathrm{possible}$ **Hacer**
        Hacer $k \leftarrow \lfloor \frac{2j}{p} \rfloor + 1$.
        Hacer $m \leftarrow w_k$.
        Hacer $C_1 \leftarrow L_m$.
        Hacer $y \leftarrow x_{(2j+1:2j+2)}$.
        Si $\mathrm{esApoyatura}(y, C_1) == 1$ **Entonces**
            Hacer $\mathrm{found} \leftarrow \mathrm{found} + 1$.
        Terminar Si
        Hacer $j \leftarrow j + 1$.
    Terminar Mientras
    Hacer $s \leftarrow \dfrac{\mathrm{found}}{\mathrm{possible}}$.
    Devolver s.
Terminar Función

Definición 35 (Adornos para la armonía) Sean $y \in \mathbb{Z}^{bp}$ una melodía, sea $w \in \mathbb{Z}^b$ un vector que contiene los índices de acordes para cada compás, sea L una lista de acordes. El total de adornos para la armonía y es el resultado de OrnamentsA(w,x,L), donde la función $\mathrm{OrnamentsA}$ se define de la siguiente manera:

$$\begin{aligned}\mathrm{OrnamentsA}(w, y, L) := \frac{1}{5}\Big(&\mathrm{countThree}(w, y, \mathrm{esBordadura}, L)\\ &+ \mathrm{countThree}(w, y, \mathrm{esEchapee}, L)\\ &+ \mathrm{countThree}(w, y, \mathrm{esCambiata}, L)\\ &+ \mathrm{countThree}(w, y, \mathrm{esDePaso}, L)\\ &+ \mathrm{contarApoyatura}(w, y, L)\Big).\end{aligned}$$

3. El problema de optimización biobjetivo

La generación de las melodías deseadas se ha modelado mediante el siguiente problema:

$$\text{Minimizar } F(x) = [g_1, g_2]^T, \quad \text{con } x \in Z^{b+2bp}.$$

Por comodidad el vector x se presenta como la concatenación de tres vectores $w \in \mathbb{Z}^b$ y $y, z \in \mathbb{Z}^{bp}$ como $x = (w, y, z)$. Las funciones g_1 y g_2 se minimizan de manera simultánea y están definidas como:

$$
\begin{aligned}
g_1(w,y,z) = 3 - \frac{TNS(y)+TNS(z)}{TN(y)+TN(z)} - \mathrm{OrnamentsM}(z,y) - \mathrm{OrnamentsA}(y) \\
- \mathrm{Happy}(w),
\end{aligned}
$$

$$
\begin{aligned}
g_2(y,z) = 2 + \mathrm{OrnamentsM}(z,y) + \mathrm{OrnamentsA}(y) - \mathrm{Half}(y) - \mathrm{Half}(z) \\
+ \frac{TN(y)+TN(z)}{2bp}.
\end{aligned}
$$

De esta manera, el vector w tendrá como componentes a los índices de los acordes de cada compás, el vector y tendrá las notas de la armonía y el vector z tendrá las notas de la melodía de la canción. El problema de optimización se presenta sujeto a las siguientes restricciones:

1. $b \in \mathbb{N}, p \in \{2,4,8,16\}$,
2. $\exists y_j > 42$ para $j = 1,\ldots,bp, \exists z_\ell > 42$, para $\ell = 1,\ldots,bp$,
3. $L = (L_1,\ldots,L_r)$, donde L_j es un acorde, para $j = 1,\ldots,r$,
4. $w_1 = 1, w_b = 1$,
5. $1 \le w_j \le r$, para $j = 2,\ldots,b-1$,
6. $41 \le y_j \le 88$, para $j = 1,\ldots,bp$,
7. $41 \le z_j \le 88$, para $j = 1,\ldots,bp$,
8. $y_j < z_j, \quad \forall y_j, z_j$ tales que $PSN(y_j,z_j) = 1$,
9. $z_j - y_j \le 24, \quad \forall y_j, z_j$ tales que $PSN(y_j,z_j) = 1$,
10. $MS(y) \le 12$,
11. $MS(z) \le 12$,
12. $GB(y,z) = NS(y) + NS(z) - 2 = 0$,
13. $\mathrm{Motif}(y,z) = \sum_{j=0}^{b-1}\sum_{k=1}^{p} MF(y_{(jp+k)}, z_{\mathcal{B}(j)}) - TN(z) = 0$,
14. $\mathrm{FirstNoteChord}(y,L) = \sum_{j=1}^{b} \mathrm{FirstNoteFC}(y_{\mathcal{B}(j)}, L_j) - b = 0$,
15. $\mathrm{MajorityMelody}(z,L) = \sum_{j=1}^{b} FPN(z_{\mathcal{B}(j)}, L_j) - b = 0$,
16. $\mathrm{MajorityArmony}(y,L) = \sum_{j=1}^{b} MPN(y_{\mathcal{B}(j)}, L_j) - b = 0$,
17. $\mathrm{BTRhythm}(y) = \sum_{j=2}^{b} \|BT(y_{\mathcal{B}(j)}) - BT(y_{\mathcal{B}(1)})\| = 0$, si $b > 1$,
18. $\mathrm{SDRRhythm}(y) = \sum_{j=2}^{b} \|SDR(y_{\mathcal{B}(j)}) - SDR(y_{\mathcal{B}(1)})\| = 0$, si $b > 1$,
19. $\mathrm{BTMelody}(z) = \sum_{\substack{j=2 \\ j \text{ par}}}^{b} \|BT(z_{\mathcal{B}(j)}) - BT(z_{\mathcal{B}(j-1)})\| = 0$, si $b > 1$,
20. $\mathrm{SDRMelody}(z) = \sum_{\substack{j=2 \\ j \text{ par}}}^{b} \|SDR(z_{\mathcal{B}(j)}) - SDR(z_{\mathcal{B}(j-1)})\| = 0$, si $b > 1$.

4. Resultados experimentales

Se utilizó el algoritmo evolutivo multiobjetivo NSGA-II [8] para resolver el problema de optimización biobjetivo con restricciones descrito en la sección anterior. Para los parámetros del algoritmo NSGA-II se usó una población de 100 individuos, una cruza de $2b$ puntos [21] con probabilidad de cruza $p_c = 0.75$, una mutación con la distribución de cruza binaria simulada [3, 9], con una probabilidad de mutación de $p_m = 0.05$ y $\eta = 1$.

Para la implementación del código se utilizó C++, Python 3 y la plataforma pymoo [4]. Cabe mencionar que las restricciones del problema mostradas en la sección anterior hacen difícil para varios algoritmos encontrar soluciones, dado que primero se debe llegar a la región factible para posteriormente optimizar g_1 y g_2; de esta manera, después de algunas pruebas con algunos algoritmos, se decidió utilizar el algoritmo NSGA-II por su poder de exploración, siendo el que llegaba más rápidamente a la región factible.

Para todos los resultados se utilizó la lista de acordes L presentada en la definición 34, donde L tiene los acordes mayores y séptimos de la escala de Do Mayor. Se realizaron distintas pruebas variando los parámetros b y p, los resultados obtenidos se han colocado en la siguiente página[1], donde se muestran algunas de las soluciones encontradas en el frente de Pareto[2] obtenido por el algoritmo para cada ejecución, donde se logra apreciar la diferencia entre mayor alegría y mayor minimalismo descritas en las funciones g_1 y g_2 del problema de optimización.

En la página web se presentan cuatro ejecuciones diferentes. Por ejemplo, para la ejecución 2, se utilizaron los parámetros $b = 10, p = 8$, tardando aproximadamente 18 horas en completar el proceso de optimización. Por otra parte, para la ejecución 3, se utilizaron los parámetros $b = 10$, $p = 4$ y se puede apreciar en los compases 7 y 8 que las notas son mayormente más graves que el resto de la canción, permitiendo tener mayor impacto en esos compases. Mientras tanto, para la ejecución 4, se utilizaron los parámetros $b = 4$, $p = 8$, a pesar que es una canción de cuatro compases, se logra apreciar que no acabó de forma abrupta y se puede identificar a la melodía.

Dado que la población del algoritmo evolutivo multiobjetivo es de 100, al terminar la ejecución se cuenta con 100 individuos óptimos aproximando el frente de Pareto. Cabe destacar que dadas las restricciones del problema, para ejecuciones distintas con los mismos parámetros b, p las canciones resultantes son distintas; esto es debido a que el algoritmo busca canciones que cumplan el mismo ritmo para la melodía y armonía (ver las restricciones 17, 18, 19 y 20 del problema de optimización biobjetivo), por lo que se pueden encontrar distintos patrones rítmicos para canciones con misma cantidad de compases b y divisiones de particiones p.

Finalmente, los resultados obtenidos muestran diversidad en las figuras de las notas, tal y como se muestra en la figura presentada como ejemplo x^0 en la sección 2, donde tal figura fue generada por el algoritmo evolutivo multiobjetivo.

[1] natanvilchis.org/comia2022/

[2] El frente de Pareto [16] es el conjunto que da solución a un problema multiobjetivo. Por razones de espacio estas definiciones se han dejado fuera de este texto.

5. Conclusiones

En este trabajo se modeló el problema de composición musical automática mediante optimización biobjetivo. El fin fue crear melodías donde se estableciera un conjunto de soluciones compromiso (frente de Pareto) respecto al nivel de alegría, como primera función objetivo, y del minimalismo o simplicidad de adornos como segunda función optimizar.

Para la creación el modelo se empleó teoría musical sobre emociones y trabajos relacionados, se obtuvieron resultados satisfactorios al encontrar elementos auditivamente diversos sobre el frente de Pareto en cada ejecución. La construcción del modelo matemático presentado en este trabajo puede servir como base para el abordaje de otras emociones y aspectos diversos de forma dentro de la composición automática de melodías. La composición musical involucra distintos aspectos que necesitan ser estudiados de forma particular y su interacción entre los mismos, tales como ritmo, melodía, arpegios, entre otros.

Para el presente trabajo se buscó simplificar los aspectos más importantes de la música, tratando de mantener un equilibrio entre la capacidad creativa de los resultados y evitar la complejidad en el problema; por ejemplo, para los patrones musicales (en la figura melódica y la figura armónica) se utilizaron las funciones $\mathrm{BTRhythm}$, $\mathrm{SDRRhythm}$, $\mathrm{BTMelody}$, $\mathrm{SDRMelody}$. Como trabajo a futuro se propone cambiar la lista de acordes utilizadas en el presente trabajo para otorgar emociones alternativas.

Referencias

1. Association of Musical Electronics Industry AMEI and MIDI Manufacturers Association MMA: Universal midi packet (ump) format and midi 2.0 protocol (2020), version 1.0
2. Bertin-Mahieux, T.: Large-Scale pattern discovery in music. Ph.D. thesis, Columbia University (2013)
3. Blank, J.: pymoo - mutation (2020), pymoo.org/operators/mutation.html
4. Blank, J.: pymoo: Multi-objective optimization in python (2020), pymoo.org/
5. Collins, T. E.: 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 (2011)
6. Cooke, D.: The language of music. Oxford University Press, 1 edn. (1959)
7. Dannenberg, R. B., Hu, N.: Pattern discovery techniques for music audio. Journal of New Music Research, vol. 32, no. 2, pp. 153–163 (2003) doi: 10.1076/jnmr.32.2.153.16738
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197 (2002) doi: 10.1109/4235.996017
9. Deb, K., Sindhya, K., Okabe, T.: Self-adaptive simulated binary crossover for real-parameter optimization. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation. pp. 1187–1194 (2007) doi: 10.1145/1276958.1277190

10. Herremans, D., Chew, E.: MorpheuS: generating structured music with constrained patterns and tension. IEEE Transactions on Affective Computing, vol. 10, no. 4, pp. 510–523 (2017) doi: 10.1109/TAFFC.2017.2737984
11. de María, M.: 18 dimensiones emocionales, decodificando las emociones de la música desde 18 dimensiones técnicas (2021)
12. Meredith, D.: COSIATEC and SIATECCompress: Pattern discovery by geometric compression. In: International society for music information retrieval conference. pp. 6. No. 14, International Society for Music Information Retrieval (2013)
13. Meredith, D.: Compression-based geometric pattern discovery in music. In: 2014 4th International Workshop on Cognitive Information Processing (CIP). pp. 1–6 (2014) doi: 10.1109/CIP.2014.6844503
14. Meredith, D.: Recur sia-rrt: Recursive translatable point-set pattern discovery with removal of redundant translators. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. vol. 1168, pp. 485–493 (2019) doi: 10.1007/978-3-030-43887-6_42
15. Meredith, D., Lemström, K., Wiggins, G. A.: Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. Journal of New Music Research, vol. 31, no. 4, pp. 321–345 (2002)
16. Miettinen, K.: Nonlinear multiobjective optimization, vol. 12. Springer Science & Business Media (2012)
17. Miranda, E. R. (ed): Handbook of artificial intelligence for music: foundations, advanced approaches, and developments for creativity. Springer Nature (2021)
18. Ren, I., Volk, A., Swierstra, W., Veltkamp, R. C.: A computational evaluation of musical pattern discovery algorithms, (2020) doi: 10.48550/arXiv.2010.12325
19. Rodríguez-Alvira, J.: Funciones armónicas : Notas de adorno (2022)
20. Scirea, M., Togelius, J., Eklund, P., Risi, S.: Affective evolutionary music composition with metacompose. Genetic Programming and Evolvable Machines, vol. 18, no. 4, pp. 433–465 (2017) doi: 10.1007/s10710-017-9307-y
21. Umbarkar, A. J., Sheth, P. D.: Crossover operators in genetic algorithms: a review. ICTACT journal on soft computing, vol. 6, no. 1 (2015) doi: 10.21917/ijsc.2015.0150