Método de la composición musical

Raúl Barba Rojas[1] Diego Guerrero del Pozo[1] Ismael Pérez Nieves[1]
Manuel Fernández del Campo[1] Jaime Camacho García[1]

[1]Universidad de Castilla-La Mancha
{raul.barba, diego.guerrero, Manuel.Fernandez17, jaime.camacho1,
Ismael.Perez3}@alu.uclm.es

Abstract

El método de la composición musical es un algoritmo metaheurístico, que intenta dibujar un simil entre el proceso de composición musical y el proceso de optimización en la resolución de un problema por medio de parámetros y paradigmas. Su aplicación en la edad contemporánea toca varios campos, desde la optimización de problemas NP-hard hasta la composición de verdaderas piezas musicales. En este documento, vamos a detallar cómo funciona este algoritmo por dentro así como las metáforas que utiliza en su desarrollo y su forma dentro de la implementación. También hablaremos de la historia del algoritmo, exponiendo sus orígenes, motivación y evolución, así como el estado del arte actual. Veremos las aplicaciones que tiene este algoritmo y las soluciones que brinda a situaciones cotidianas. Y por último, veremos la implementación que hemos realizado como ejemplo de su funcionamiento y explicaremos de forma detallada el proceso de desarrollo.
Keywords: optimización, metaheurísticas, composición musical.

Contents

1 Introducción

Este artículo pretende ser una descripción detallada sobre el método de la composición musical, inicialmente descrito en [6].

Aunque posteriormente se entrará en más detalles, podemos describir el método de la composición musical como un algoritmo metaheurístico, cuyo principal objetivo es la resolución de problemas de optimización, mediante el establecimiento de una similitud entre el proceso de composición musical y el proceso de optimización, tal y como se describe en [8] y [6].

Por tanto, el método de la composición musical (MCM[1]), es un algoritmo metaheurístico basado en metáforas, concretamente en la metáfora musical, puesto que el algoritmo se basa en simular una sociedad

[1]En inglés sus siglas cambian a MMC: *method of musical composition*

musical, donde los compositores intercambian información para crear la mejor composición musical [1]. Así, su principal objetivo es que mediante el intercambio de melodías entre compositores, el algoritmo sea capaz de encontrar la mejor melodía[2].

Se trata de una metaheurística que es similar a otras metaheurísticas basadas en la metáfora musical, como podría ser la búsqueda harmónica (*harmony search (HS)*, sin embargo, el método de composición musical (*MMC*) no es un derivado de la búsqueda harmónica, puesto que se basa en la idea anterior: un sistema social, donde el intercambio de conocimiento, el aprendizaje... son la clave para la búsqueda de buenas melodías, que se traducirían, idealmente, en los óptimos globales de un problema de optimización.

Por ello, tal y como se describe [11], se propone un algoritmo en que, gracias a la comunicación y al intercambio de información entre los compositores, se obtienen las mejores melodías (mejores soluciones), tal y como se puede apreciar en la figura 1.

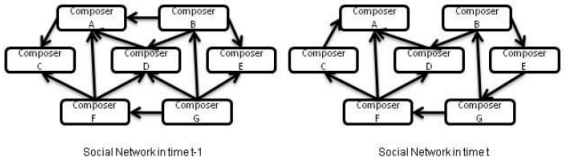

Figure 1: Sistema/Red social de compositores

Con respecto a este artículo, nuestro principal objetivo es hacer un estudio detallado sobre todo lo relacionado con esta metaheurística. Por ello, el resto del artículo quedará estructurado de la siguiente manera:

1. Historia y evolución: cuyo principal objetivo es alcanzar un mayor grado de conocimiento sobre el origen de este algoritmo metaheurístico, principalmente nos enfocaremos en describir cómo surgió y cómo evolucionó.

2. Estado del arte: su principal objetivo es describir cómo ha sido su popularidad hasta la actualidad, cómo es su popularidad actualmente y qué cabe esperar de este algoritmo metaheurístico en un futuro.

3. Descripción detallada: es la sección principal, en la que intentaremos explicar cómo funciona el método de la composición musical, el esquema que utiliza e incluso un posible pseudocódigo, que nos permitiría implementarlo para resolver cualquier problema de optimización[3].

4. Aplicaciones: se trata de una sección en la que intentaremos mostrar ejemplos de aplicación de esta metaheurística para resolver problemas reales.

5. Conclusiones: es la sección final en la que trataremos de resumir todo el artículo, para fijar las ideas principales que queremos transmitir.

2 Historia y evolución

El principal objetivo de esta sección es responder a importantes preguntas sobre esta metaheurística como:

- ¿Quién y cuándo la inventó?
- ¿Cuál fue su origen?
- ¿Cómo ha sido su evolución?

El método de la composición musical fue creado por Román Anselmo Mora-Gutiérrez, Javier Ramírez-Rodríguez y Eric Alfredo Rincón García en el año 2013, en [6], si bien los propios autores crearon una revisión de esta metaheurística tras poco más de un año de su creación, como se puede comprobar en [7].

[2]Aplicado al contexto de los problemas de optimización y algoritmos metaheurísticos, encontrar la solución óptima, aunque esto no es garantizado.

[3]Al ser un algoritmo metaheurístico, no resuelve un problema concreto, sino que es un algoritmo general que debe especificarse para resolver problemas concretos

El origen de este algoritmo fue crear una metaheurística que replicase el proceso creativo de la composición musical, tal y como especifican los autores en [7]. De esta manera[4], los autores proponen un algoritmo en el que mediante un sistema social en el que los compositores intercambian información sobre sus melodías, se intenta llegar a la mejor melodía: la que tiene mejor aceptación por parte del público.

De esta forma, el origen del método de la composición musical se encuentra en la analogía, mencionada en 1, entre el proceso de optimización y la composición musical. Según los creadores de la metaheurística en [6], la analogía[5] es clara y puede entenderse como se expresa en la tabla 1.

Table 1:

Pregunta comparativa	Optimización	Composición musical
¿Qué pretende?	Óptimos globales	Música con la mayor aceptación del público
¿Cómo se determina la calidad de la solución?	Función objetivo	Grado de aceptación del público
¿Qué determina la calidad de la solución?	Valores de las variables del problema	Características de sonido de la meolodía
¿Es un proceso finito?	Sí	Sí
¿Qué determina el progreso?	Iteración	Cada refinamiento de la melodía

Sin embargo, otro aspecto a tener en cuenta es la *motivación*, es decir, ¿por qué decidieron los autores crear esta metaheurística y no otra? La respuesta a dicha pregunta se encuentra en [7], en cuya introducción hablan de la importancia que ha tenido en los últimos años una metaheurística que también está asociada a la metáfora musical: la búsqueda harmónica (*HS*). Dicha metaheurística demostró ser capaz de resolver problemas de optimización reales, dando mejores resultados que otros métodos heurísticos o deterministas, como se muestra en [3].

Por ese motivo, los autores del método de composición musical decidieron crear una nueva metaheurística, que llevaría por nombre *MMC* (method of the musical composition), o *MCM* en castellano (método de la composión musical), cuyo objetivo fuera utilizar esa misma metáfora de la creación musical para resolver problemas de optimización, pero a través de un sistema social[6], que permita a los compositores (creadores de melodías) intercambiar información sobre qué melodías están usando, para que se pueda llegar a la mejor solución, surgiendo así el método de la composición musical tal y como lo conocemos hoy.

Con lo explicado hasta este momento, hemos dado respuesta a dos de las tres preguntas formuladas al inicio de la sección, sin embargo, todavía falta explicar cómo ha sido la evolución de esta metaheurística. En primer lugar, cabe mencionar que justo después de su creación, los autores intentaron llevar a cabo una demostración de que el método de la composición musical puede ser muy útil para proporcionar buenas soluciones a problemas de optimización, en algunos casos mejores que los métodos existentes hasta la fecha.

De esta forma, en [6] los autores muestran la validación del método de la composición musical mediante un conjunto de 5 problemas de optimización por restricciones no lineales (*CNOP*)[7]. Esto es interesante desde el punto de vista de la evolución de la metaheurística, puesto que nos interesa saber cómo se comporta respecto a problemas reales. Los resultados de los autores muestran que el método de la composición musical puede encontrar mejores soluciones que otras metaheurísticas como aquellas basadas en el sistema inmunológico[8]

Posteriormente, en [11] el método de la composición musical fue probado satisfactoriamente[9], a través de un experimento sobre un conjunto de problemas de optimización más amplio, concretamente un *benchmark* de 13 problemas de optimización de diferente índole, incluyendo problemas resueltos por métodos exactos, e.g. método de Newton, pero también otros problemas únicamente resueltos

[4]Será explicado con mayor detalle en la sección 4

[5]La tabla que proponemos es una versión compacta y enfocada, que la versión original que se encuentra en [6]

[6]Los autores la describen como una *heurística multiagente*

[7]CNOP: Constrained nonlinear optimization problem

[8]Los autores no especifican qué metaheurística, basada en la metáfora del sistema inmunológico, fue utilizada para la prueba.

[9]Proporciona, según los autores, buenas soluciones aproximadas, respecto de otras metaheurísticas conocidas como la búsqueda harmónica (*HS*)

por metaheurísticas como la búsqueda tabú (TS), tal y como especifican los creadores del método de composición musical en [11].

De esta manera, los creadores de algoritmo fueron capaces de demostrar empíricamente que su metaheurística es realmente útil para resolver diferentes problemas de optimización, y es por ello que es una metaheurística cuya evolución es completamente creciente, pasando de ser una metaheurística poco conocida a una metaheurística utilizada en diferentes problemas de optimización, gracias a su capacidad para proporcionar buenas soluciones en problemas complejos. De esta forma, el método de la composición musical tiene aplicaciones para resolver problemas de optimización reales[10], tal y como se muestra en [13] (2014) y en [9] (2017).

Por tanto, estamos ante una metaheurística que ha tenido una gran evolución desde su aparición, con aplicaciones para resolver problemas reales, que son relativamente recientes, como los mencionados anteriormente, e incluso otros más recientes como su aplicación para resolver el problema de ruteo de vehículos, tal y como se describe en [5] (2020).

3 Estado del arte

El método de composición musical ha sido utilizado en numerosos trabajos en la actualidad. El algoritmo es relativamente nuevo, pero aún así ha logrado hacerse un hueco dentro del mundo de los algoritmos para proporcionar nuevas formas de resolver problemas.

3.1 Aplicaciones en el Reconocimiento de Patrones

Las investigaciones contemporáneas en Composición Musical se centran en encontrar patrones dentro de la música introducida como entrada. Una de las propuestas fue la utilización de técnicas de minería de datos para buscar patrones y descubrir reglas de composición musical que puede que ni el propio autor se haya dado cuenta de su existencia [14].

Los doctores en Ciencias de la Computación Man-Kwan Shan y Shih-Chuan Chiu de la Universidad de National Chiao-Tung University crearon un sistema que hacía esto mismo. El sistema busca patrones en un repositorio de canciones que luego serían pasados por un proceso de Extracción de la Melodía Principal. De este procesamiento se harían tres análisis para detectar patrones: búsqueda de motifs, análisis de la estructura musical y análisis de melodías. Después de todo este aprendizaje, se comenzaría el proceso de generar acordes y melodías con los patrones encontrados.

Otra propuesta fue desarrollada por los doctores Ning-Han Liu, Yi-Hung Wu y Arbee L. P. Chen en su paper "Identifying Prototypical Melodies by Extracting Approximate Repeating Patterns from Music Works" [4]. Su idea consiste en buscar patrones en segmentos musicales para componer melodías. Su idea consiste en considerar cada segmento musical como un posible candidato y buscar similitudes entre otros candidatos.

3.2 Aplicaciones de Algoritmos Genéticos

3.2.1 GenJam

GenJam está basado en una simulación de un improvisador de jazz mediante el uso del algoritmo genético. Su salida son composiciones únicas e irrepetibles, como lo haría el improvisador. Este algoritmo llama "eventos" a componentes musicales como notas individuales, frases melódicas o piezas musicales enteras [14]. En el algoritmo, estos eventos forman pequeñas frases llamadas "licks" que se unen para formar la improvisación del software.

GenJam se encarga de este proceso por medio de algoritmos genéticos. Va mejorando poblaciones de frases melódicas por medio de operadores previamente definidos La función fitness de cada frase está determinada por el usuario, que tendrá que evaluar manualmente generación en generación hasta obtener una población con frases melódicas con sentido. Esta evaluación consiste en marcar con 'g' si la porción que se esta juzgando es buena o con 'b' si es mala. La función fitness es un acumulador que se incrementa cuando detecta 'g' y decrementa cuando detecta 'b'.

GenJam tiene tres modos [2]: learning, breeding o demo. El modo learning está pensado para construir valores de fitness y no usar operadores genéticos, seleccionando frases melódicas aleatoriamente y presentadas al mentor. El modo demo está mas orientado a simular un show, con un proceso de selección por comparación y eliminación similar a un torneo teniendo en cuenta el valor fitness. En el

[10] Estos problemas, y algunos más, serán detallados en la sección 5

modo breeding se utilizan operadores genéticos y se sustituye a la mitad de la población por nuevos hijos antes de la presentación para recibir la evaluación del mentor.

3.2.2 Algoritmo genético de Drangan Mátic

La representación de Drangan Mátic consiste en una representación flexible para poder almacenar la composición musical, que a su vez almacena el ritmo y el timbre de cada nota. Para lograr esto, se realiza previamente una enseñanza de ritmos predeterminados, a partir de los cuales se empezará a asignar timbres a las notas para poder completar la melodía. El resultado normalmente presenta ritmos muy homogéneos, encontrándose la variedad en las tonalidades.

Esta solución presenta una función fitness diferente a las anteriores que le permite distinguir entre una melodía buena o mala. La realiza por medio de una serie de criterios que se encuentran localizados en los intervalos que forma una nota con su sucesora. Estos criterios pueden ser parámetros como el espacio entre notas o la diferencia de escalas entre ellas. El resultado son armonías agradables al oído humano, pero llegan a estar muy alejadas del nivel de creatividad que poseemos los humanos. Sin embargo, se está estudiando la posibilidad de emplear métodos como la metaheurística para mejorar este aspecto de la solución.

3.3 Aplicaciones de las Cadenas de Márkov

A la hora de llevar el uso de las Cadenas de Márkov a la Teoría Musical, se preguntó si era posible analizar las composiciones musicales en busca de señales que permitan predecir "qué es lo siguiente que puede suceder". O, dicho de otra manera, que permitan realizar una predicción sobre el siguiente paso que se debería realizar. Por ejemplo, dado un solo, se podrían llegar a realizar acompañamientos musicales que lo eleve.

Cristopher Rafael, oboísta profesional y matemático en la Universidad de Indiana, intentó afrontar este problema mediante el uso de las Cadenas de Márkov Hidden y modelos bayasianos. Su sistema fue bautizado como "Music Plus One" [12]. Dentro de su blog, podemos encontrar vídeos ejemplos sobre cómo su sistema puede "predecir el futuro", analizando la melodía que está sonando para predecir cuándo reproducir notas orquestales.

Otro intento fue realizado de la mano de los profesores de State University of New York at New Paltz en su paper "Extracting Patterns in Music for Composition via Markov Chains" [2]. Esta solución intenta encontrar patrones en un conjunto de melodías musicales por medio de procesamiento de datos como la Minería de Datos o el Machine Learning, y utilizar estos patrones para realizar una composición por medio de las Cadenas de Márkov.

Sin embargo, la naturaleza probabilística de estas soluciones conlleva una serie de molestias a largo plazo, debido a que esta aleatoriedad se va acumulando, llevando a composiciones sin mucho sentido.

3.4 Conclusiones sobre el estado del arte

Ahora mismo, este algoritmo sigue estando en proceso de mejorar y obtener resultados realmente útiles. Las mejores soluciones que existen dejan de lado la teoría musical para centrarse en la aleatoriedad hasta encontrar una solución óptima. Aún queda mucho por descubrir, pero ya se están empezando a ver algoritmos de Deep Learning que generan canciones después de pasarle como entrada múltiples canciones del mismo artista [10]. Si somos capaces de empezar a generar música de forma automática será un gran descubrimiento que marcará para siempre la historia de la música.

4 Descripción detallada

En esta sección, se explicará y comentará detalladamente el procedimiento de este algoritmo para conseguir el fin perseguido, en este caso, de optimización[11]; así como las fases necesarias para ello.

Tal y como se indica en [8] (2012), cada compositor o agente que forma parte de este algoritmo, registrará sus melodías como vectores n-dimensionales, siendo cada uno de estos una posible solución, en función de si pasan la prueba de aptitud o no.

Según [6] (2013), este algoritmo consta de las siguientes fases:

[11] Puede ser utilizado para conseguir que vectores con valores numéricos cumplan ciertas condiciones matemáticas donde necesitamos calcular el mínimo en funciones complejas

1. Modificación de los enlaces entre compositores.
2. Intercambio de información (melodías) entre compositores.
3. Proceso de decisión de cada compositor para determinar si utiliza o no la información recibida de su entorno; es decir, si incluye nuevas melodías en su trabajo.
4. Cada compositor genera una nueva matriz de conocimiento gracias a la información de su entorno.
5. La nueva matriz se utiliza para generar una melodía nueva.
6. El compositor decide si incluir la nueva melodía en su obra. En caso afirmativo, sustituirá a la peor melodía que almacene hasta el momento.

Basándonos en este información, podemos desarrollar un pseudocódigo similar al citado en [8], que muestre como debería funcionar nuestro algoritmo, así como su estructura. Este algoritmo se presenta en el algoritmo 1.

Algorithm 1: Estructura general de los algoritmos MMC

```
Input: parámetros de MMC y compositores creados
Output: mejores melodías generadas por los compositores
Crear una sociedad de compositores con sus reglas de interacción;
for compositor i do
    inicializar aleatoriamente obra del compositor;
end for
repeat
    Actualizar sociedad de compositores;
    Intercambiar información entre compositores;
    for compositor i do
        Actualizar matriz de conocimiento;
        Generar y evaluar nueva melodía;
        if nueva melodía es mejor que la peor melodía del compositor then
            Reemplazar la peor melodía con la nueva en la obra;
        end if
    end for
until se cumpla el criterio de finalización;
```

El criterio de finalización suele ser un máximo número de intercambio de información entre agentes o un número de iteraciones[12].

Respecto a los elementos del algoritmo MMC, tenemos, como acabamos de mencionar, el criterio de finalización (Que llamaremos numIter); así como una serie de factores, que serán descritos a continuación:

- Factor de genio sobre innovador (ifg): es la probabilidad de generar una nueva melodía sin tener en cuenta la matriz de conocimiento.
- Factor de genio sobre cambio (cfg): es la probabilidad de generar una valor para una variable de decisión sin tener en cuenta la matriz de conocimiento.
- Factor de intercambio entre agentes (fcla): es la probabilidad de que un agente modifique sus enlaces con otros compositores de la sociedad.
- Número de compositores (Nc).
- Número de acordes de una melodía (Ns): representan la longitud de los vectores que simulan una melodía.

Todos estos detalles se mencionan en [7] (2014), donde, además, se especifican los valores que estos factores deben tener, mostrados en la tabla 2.

Además, es conveniente tener en cuenta que el número de evaluaciones del algoritmo (o Ne), es igual al número de compositores por el número de iteraciones que deseemos.

[12] Si lo deseasemos, también podríamos hacer que el criterio de finalización se cumpla cuando alguna melodía cumpla un requisito impuesto, aunque así no podamos controlar el tiempo de ejecución

Table 2: Tabla con los valores válidos de cada factor

Factor	Valores posibles
numIter	$[0,\infty) \cap \mathbb{N}$
ifg	$[0,1] \cap \mathbb{R}$
cfg	$[0,1] \cap \mathbb{R}$
fcla	$[0,1] \cap \mathbb{R}$
Nc	$[2,\infty) \cap \mathbb{N}$
Ns	$[3,\infty) \cap \mathbb{N}$

También es importante destacar que cada compositor debe poseer una matriz P para almacenar las variables de decisión necesarias, y que estas se encontrarán en un rango descrito previamente. Esta matriz será referenciada como $P_{*,*,i}$[13].

Una vez descrito todas las variables necesarias para el funcionamiento del algoritmo de MMC, podemos detallar con mayor cuidado todas las fases que lo componen, comenzando por el intercambio de información entre agentes.

Este interambio, por su parte, requiere de 3 fases principales:

1. El compositor i determina cual es sus melodías es la menos deseada, de acuerdo a la función de aptitud. Este melodía se designará como $x_{i-worst}$.

2. El compositor i compara $x_{i-worst}$ con $x_{k-worst}$ del compositor k, siempre y cuando exista un enlace entre ambos compositores.

3. El compositor i decide si considerar información del compositor k solo si $x_{k-worst}$ es mejor que $x_{i-worst}$.

Para que esta parte funcione, es importante la forma en la que se establecen estos enlaces entre compositores, detallado en el algoritmo 2.

Una vez estos enlaces están establecidos, se puede proceder al intercambio de información entre compositores. La idea de esta parte es que cada uno de estos compositores creará una matriz denominada $ISC_{*,*,i}$[14], consistente en melodías obtenidas de compositores vecinos, con la cuál se decidirá si se sustituye alguna melodía de nuestra propia obra con alguna de las adquiridas, de acuerdo al algoritmo 3.

Despúes de estas interacciones entre compositores, disponemos ahora de una matriz de conocimiento que puede ser utilizada por cada compositor para generar aún mejores melodías. Dicho esto, es importante hablar de como, efectivamente, se generan nuevos vectores.

El primer paso para ello es generar otra nueva matriz, denominada $KM_{*,*,i}$[15], que es la unión de la matriz de conocimiento adquirido y la score matrix de cada compositor. Con esta matriz generada, queda evaluar la aptitud de cada vector de $KM_{*,*,i}$, como se muestra en el algoritmo 4.

[13] También llamada score matrix
[14] O matriz de conocimiento adquirido
[15] O matriz de conocimiento del entorno

Algorithm 2: Actualización de los enlaces entre compositores

Input: Nc, fcla, sociedad de compositores previa
Output: sociedad de compositores actualizada
if *t = 1* **then**
  for *i = 1 : Nc - 1* **do**
    for *k = i + 1 : Nc* **do**
      if *valor aleatorio < 0.5* **then**
        Crear enlace entre compositores i y k;
      end if
    end for
  end for
else
  for *i = 1 : Nc* **do**
    if *valor aleatorio < fcla* **then**
      Elegir aleatoriamente un compositor k tal que i $\neq$ k;
      Cambiar la relación entre compositores;
    end if
  end for
end if
Comprobar que cada agente tiene al menos un enlace;

Algorithm 3: Intercambio de información con el entorno

Input: $P_{*,*,i}$, Nc, función a optimizar (f)
Output: $ISC_{*,*,i}$
for *i = 1 : Nc* **do**
  $x_{i-worst} \leftarrow$ peor vector de $P_{*,*,i}$;
  for *k = 1 : Nc tal que i* $\neq$ *k* **do**
    $x_{k-worst} \leftarrow$ peor vector de $P_{*,*,k}$;
    if *existe enlace entre i y k* **then**
      if $x_{i-worst}$ *es peor que* $x_{k-worst}$ **then**
        Compositor i elige aleatoriamente una melodía de $P_{*,*,k}$ y la añade a $ISC_{*,*,i}$;
      end if
    end if
  end for
end for

Algorithm 4: Creación y evaluación de la matriz de conocimiento del entorno

Input: $P_{*,*,i}$, $ISC_{*,*,i}$, función a optimizar (f)
Output: aptitud de $KM_{*,*,i}$
for *i = 1 : Nc* **do**
  $KM_{*,*,i} = P_{*,*,i} \cup ISC_{*,*,i}$;
  for *k = 1 : Nc tal que i* $\neq$ *k* **do**
    $x_{k-worst} \leftarrow$ peor vector de $P_{*,*,k}$;
    $a_i = \sum_{j=1}^{r} f(KM_{j,*,i})$;
    for *j = 1 : r* **do**
      aptitud$(KM_{j,*,i}) = \frac{a_i - f(KM_{j,*,i})}{a_i * (Nc-1)}$
    end for
  end for
end for

Continuamos con la descripción en profundidad del MMC, para ello analizaremos la parte del algoritmo que se encarga de generar nuevos tonos para los compositores que forman parte del entorno, como se generan, y como son utilizados para actualizar los trabajos de otros compositores, y como finalmente una vez esta todo el proceso explicado y desarrollado en detalle, veremos como se construye el set de soluciones para cada uno de los compositores.

En este paso los compositores usan sus ideas innovadoras para así poder decidir el usar o no la información provista por su matriz de conocimiento $KM*, *, i$, para asi construir una nueva solución.

Algorithm 5: Creación de una nueva melodía

Input: $KM*,*,i$, i, f, g, Nc, $f(x)$, $fitness(KM*,*,i)$, x_i^U for all l=1,2....,n and x_l^L, for all l=1,2.....,n
Output: A new tune $(x*, new)$
for *i = 1 : Nc* **do**
  if *rand < (1 - ifg)* **then**
    for *l = 1: n* **do**
      $x_l^{max} \leftarrow$ máximo valor de x_l en $KM*,l,i$;
      $x_l^{min} \leftarrow$ máximo valor de x_l en $KM*,l,i$;
      $KMj,l,i \leftarrow$ obtiene aleatoriamente la melodía j de $KM*,l,i$ considerando $fitness(KMj,*,i)$;
      $KMj^{'},l,i \leftarrow$ obtiene aleatoriamente la melodía $j^{'}$ de $KM*,l,i$ considerando $fitness(KMj^{'},*,i)$;
      if *rand < (1-cfg)* **then**
        $x_lnew \leftarrow$ creando un nuevo motivo basado en la información seleccionada;
      else
        if *rand < 0.5* **then**
          x_lnew=x_l^{min}+(rand*(KMj,l,i-x_l^{min}))
        else
          x_lnew=x_l^{max}-(rand*(x_l^{max}-KMj,l,i))
        end if
      end if
    end for
  else
    for *l=1 : n* **do**
      x_lnew=x_l^U - (rand*(x_l^U-x_l^L)
    end for
  end if
  Computa el valor de la función objetivo de x_*new (f(x_*new)
end for

Si la respuesta es SI a utilizar la matriz de conocimiento por parte del compositor, el algoritmo número 6 describe una de las posibles estrategias para implementar este proceso de creación, en otro caso la melodía sera generada de manera completamente aleatoria. Esta primera estrategia asume que la informacion contenida en $KM*,*,i$ esta completa y todas esas posibles melodías puede ser creadas desde elementos de $KM*,*,i$.

Algorithm 6: Método utilizado por el MMC original para crear un nuevo motivo

Input: $KMj^{'},l,i$
and KMj,l,i **Output:** Xl,new
Xl,new=KMj,l,i+(rand*(($KMj^{'},l,i$-KMj,l,i)-(2*x_l^{rand})))

En este paso cada uno de los compositores toma una decisión, sustituir o no sustituir la peor melodía en su matriz de puntuación $P*,*,i$ con $x_{i,new}$. La decisión se basa en valor de la función objetivo: si la nueva melodía tiene un valor mejor que $x_{i-worst}$, $x_{i,new}$ reemplaza la peor melodía en $P*,*,i$.

Algorithm 7: Actualizando el trabajo de los compositores

Input: $P*,*,i$
, f(x), and Nc **Output:** matriz $P*,*,i$ actualizada
for *i=1 : Nc* **do**
  $x_{x-worst} \leftarrow$ elementos dentro de P con el peor valor de aptitud;
  if *f($x_{x-worst}$ es peor que f($x_{*,new}$* **then**
    Reemplaza $x_{x-worst}$ por $x_{*,new}$ en $P*,*,i$
  end if
end for

Y para finalizar el MMC, el algoritmo selecciona la melodía contenida en el trabajo de cada uno de los compositores que logra el mejor valor de función objetivo. Todo esto más lo mostrado en lo anterior forman el proceso integro y completo del MMC para la creación y selección de melodías.

Algorithm 8: Construyendo un set de soluciones

Input: $P*, *, i$
, f(x), and Nc **Output:** Set con las mejores soluciones encotradas por los compositores ($S_{*,*}$)
for *i=1 : Nc* **do**
  Si, * $\leftarrow$ elementos dentro de $P*, *, i$ con los mejores valores de f(x);
end for

5 Aplicaciones

Como se ha comentado anteriormente, el método de la composición musical (MCM) es usado para la resolución de problemas de optimización, ya que la utilización de técnicas heurísticas como esta ha llegado a ser una herramienta muy útil para poder obtener soluciones a dichos problemas, siendo estas soluciones de una gran calidad, pero no tienen poque ser las más optimas, ya que esto puede derivar en no obtener unos tiempos de cómputo razonables, que es otro de los objetivos de este tipo de algoritmos. A su vez, también se debe mencionar que para este tipo de problemas no hay un método que destaque o supere drásticamente sobre el resto.

Algunos problemas que ha conseguido resolver este tipo de método los podemos ver en nuestro día a día como son el caso de:

- La generación de turnos de los enfermeros.
- Los horarios de los diferentes medios de transporte.
- El establecimiento de grupos de trabajo.

Cabe destacar que su uso es también muy común en centros de educación superior para poder llevar a cabo la asignación de los alumnos, las clases, los exámenes e incluso los profesores a los horarios académicos establecidos, aunque estas no son las únicas variables que se pueden implementar en este tipo de problemas.

Relacionado con el problema académico anterior, explicaremos más en detalle el uso de este algoritmo metaheurístico para la creación de un currículo académico donde queremos ubicar unidades de enseñanza y aprendizaje (UEA) en la menor cantidad posible de trimestres, mientras se cumplen las condiciones que ha impuesto la universidad y que así los estudiantes puedan apuntarse a estas, al mismo tiempo que se consigue minimizar la cantidad de trimestres para superar todas las unidades de enseñanza y aprendizaje.

Unas de las razones del uso de este algoritmo metaheurístico para la resolución de este problema, es como ya sabemos, sus tiempos de computo razonables y sus soluciones de buena calidad, pero también se deben mencionar las siguientes características:

- La técnica en el ámbito de la resolución de problemas de optimización continua ha tenido un desempeño excepcional.
- Otorga la capacidad de explorar simultáneamente diferentes semiespacios del espacio de búsqueda, ya que es un algoritmo poblacional.
- Se comprobó que el empleo de métodos de resolución exacta no otorgaban una solución decente en un lapso de tiempo razonable.
- Tiene la capacidad de centrar la búsqueda en las regiones del espacio de búsqueda que sean más prometedoras, ya que es una heurística social y cuenta con inteligencia colectiva.
- La presencia de dinámicas de escalas, en concreto tres, local, personal y social. Esto está relacionado directamente con las dos fases que tiene el algoritmo, la exploración y la diversificación, dando así con un mecanismo de herencia triple.

Antes de seguir desarrollando el problema debemos de entender y saber que una UEA es cursada por un estudiante, además de tener una cantidad determinada de créditos y puede que para cursarla se tenga que cumplir algún requisito, como puede ser estar inscrito con anterioridad a otras unidades o haber superado un mínimo de créditos. A su vez el estudiante tiene un máximo de créditos en el primer trimestre de 46 créditos y en los demás de 60 créditos, para así poder evitar una carga académica excesiva para los estudiantes.

Una vez visto y entendido cual es el problema a resolver, podemos ver el pseudocódigo:

Crear una sociedad artificial con una red de interacción entre ellos.

Para cada uno de los compositores hacer

Generar aleatoriamente un conjunto de soluciones

Fin Para

Mientras no se cumpla el criterio de paro hacer

Intercambiar información entre los compositores

Para cada uno de los compositores hacer

Actualizar la matriz de conocimiento

Generar y evaluar una nueva solución

Actualizar el conocimiento

Fin Para

Actualizar los vínculos de la red

Fin Mientras

[13] (2014)

Además de la implementación, donde entendemos que una melodía es la representación de una solución, es decir 66 entradas o, dicho de otra forma, una entrada por UEA. Por otra parte, se generará una sociedad de NC compositores, que tendrá NS melodías, lo que quiere decir que habrá NC*NS soluciones. También se debe saber que se por cada iteración, cada compositor generara una melodía nueva, es decir, NC melodías por iteración. Con todo esto, podemos ver cómo funcionan las etapas:

- Lo primero que hará el algoritmo es generar una red social por cada compositor, donde un compositor compartirá información con el compositor i+1 y el compositor i-1, salvo que este sea el compositor 1 o el NC-ésimo, por lo que solo compartirá información con otro compositor.
- Ahora se generará un conjunto de soluciones aleatoriamente para cada uno de los compositore.
- Seguidamente cada compositor comparara su peor melodía con lo compositores que pertenezcan a su red social, adquiriendo de esta las melodías que la mejoran.
- Procede el algoritmo por lo tanto a actualizar la matriz de conocimiento.
- Para seguidamente crear nuevas soluciones, ya que cada compositor elegirá 3 melodías, la primera será la mejor solución que haya en su matriz de conocimiento, la siguiente será una al azar de las restantes en su matriz de conocimiento y la última será una propia del compositor. Durante este proceso, la participación de los compositores es aleatoria, favoreciendo así la diversificación.
- Finalizada la etapa anterior, se actualizará la matriz de conocimiento y si la solución fuese mejorada se reemplazaría, pero en caso contrario la nueva solución se desecharía.

Después de toda la implementación, se llevaron a cabo pruebas con diferentes casos como podemos ver en la siguiente tabla:

		Melodías		
		3	6	9
Compositores	5	18	39	82,3
	10	25,7	77,2	173,7
	15	31	117,1	187,3
	20	43,4	115,1	188,3
	25	52,8	127	139,9

Figure 2: Porcentaje de soluciones factibles con 11 trimestres.

[13] (2014)

En cada combinación se realizaron 100 ejecuciones, obteniendo así que la mejor solución es la combinación de 15 compositores y 3 melodías, ya que es una de las que menos recursos usa y de las que mayor porcentaje de soluciones factibles tiene, siendo estas 99 soluciones de 11 trimestres y una de 12 trimestres.

Por último, podemos ver también en la siguiente tabla el tiempo promedio en segundos empleado por MMC.

		Melodías		
		3	6	9
Compositores	5	18	39	82,3
	10	25,7	77,2	173,7
	15	31	117,1	187,3
	20	43,4	115,1	188,3
	25	52,8	127	139,9

Figure 3: Tiempo en segundos promedio empleado en cada MMC.

[13] (2014)

6 Conclusiones

El método de la composición musical es un algoritmo, cuanto menos, curioso. Resulta bastante interesante ver cómo se puede llevar una metáfora tan aparentemente alejada de la informática como puede ser la composición de piezas musicales y basar una metodología enteramente en ella. Dice mucho del campo de la Informática, y de cómo los informáticos, aún habiéndose formado oficialmente únicamente en esta materia, pueden emplear un abanico de conocimientos fuera del campo para la resolución de nuevos problemas. Resulta de interés general para cualquier ingeniero informático al menos informarse acerca del MMC para ver cómo se pueden resolver multitud de problemas con soluciones creativas.

Es un algoritmo relativamente nuevo, por lo que aún falta mucha documentación y volumen de profesionales que estén experimentados en ello. Pero los resultados actuales ya empiezan a vislumbrar un futuro posiblemente brillante.

References

[1] Mohamed Abdel-Basset, Laila Abdel-Fatah, and Arun Kumar Sangaiah. Chapter 10 - metaheuristic algorithms: A comprehensive review. In Arun Kumar Sangaiah, Michael Sheng, and Zhiyong Zhang, editors, *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, Intelligent Data-Centric Systems, pages 185–231. Academic Press, 2018. `isbn: 978-0-12-813314-9. doi:https://doi.org/10.1016/B978-0-12-813314-9.00010-4.` URL `https://www.sciencedirect.com/science/article/pii/B9780128133149000104`.

[2] John A Biles. Genjam in transition: From genetic jammer to generative jammer. 2002.

[3] Zong Woo Geem. Recent advances in harmony search algorithm. In Springer-Verlag Berlin and Heidelberg GmbH & Co. KG, editors, *Recent Advances in Harmony Search Algorithm*, Studies in Computational Intelligence. 2010.

[4] Ning-Han Liu, Yi-HungWu, and Arbee L. P. Chen. Identifying prototypical melodies by extracting approximate repeating pattern from music works. 2010.

[5] Edwin Montes-Orozco, Roman Mora-Gutiérrez, Bibiana Obregon, Sergio de-los-cobos silva, ERIC RINCÓN-GARCÍA, MIGUEL GUTIÉRREZ-ANDRADE, and PEDRO LARA-VELÁZQUEZ. MatheurÍsticas para resolver el problema de ruteo de vehÍculos con ventanas de tiempo. *Revista de Matemática: Teoría y Aplicaciones*, 27:325–352, 06 2020. `doi:10.15517/rmta.v27i2.37889`.

[6] Roman Anselmo Mora-Gutierrez, Javier Ramirez-Rodriguez, Eric Alfredo Rinc-Garcia, Antonin Ponsich, Oscar Herrera-Alcantara, and Pedro Lara-Velazquez. An optimization algorithm inspired by musical composition in constrained optimization problems. *Revista de Matematica Teoria y*

Aplicaciones, 20:183 – 202, 12 2013. ISSN 1409-2433. URL http://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S1409-24332013000200006&nrm=iso.

[7] Roman Anselmo Mora-Gutierrez, Javier Ramirez-Rodriguez, and Eric Alfredo Rincon-Garcia. An optimization algorithm inspired by musical composition. *Artif. Intell. Rev.*, 41(3):301–315, mar 2014. ISSN 0269-2821. doi:10.1007/s10462-011-9309-8. URL https://doi.org/10.1007/s10462-011-9309-8.

[8] Roman Mora-Gutiérrez, Javier Ramirez, Eric Rincón-García, Antonin Ponsich, and Oscar Herrera-Alcántara. An optimization algorithm inspired by social creativity systems. *Computing*, 94, 11 2012. doi:10.1007/s00607-012-0205-0.

[9] Roman Mora-Gutiérrez, Antonin Ponsich, Eric García, Sergio de-los-cobos silva, Miguel Andrade, and Pedro Lara-Velázquez. Method of musical composition for the portfolio optimization problem. pages 365–376, 10 2017. isbn:978-3-319-62427-3. doi:10.1007/978-3-319-62428-0_29.

[10] OpenAI. Pop, in the style of katy perry. URL https://soundcloud.com/openai_audio/jukebox-novel_lyrics-78968609?utm_source=www.technologyreview.com&utm_campaign=wtshare&utm_medium=widget&utm_content=https%253A%252F%252Fsoundcloud.com%252Fopenai_audio%252Fjukebox-novel_lyrics-78968609.

[11] E.A. Rincón-García R.A: Mora-Gutierrez, J. Ramírez-Rodríguez. An optimization algorithm inspired by musical composition. Transparencias de MMC, 2012.

[12] Cristopher Rafael. Music plus one. URL https://music.informatics.indiana.edu/~craphael/music_plus_one/.

[13] Rafaela Blanca Silva Lopez, Rosa Elena Cruz Miguel, Eric Alfredo Rincon Garcia, Roman Anselmo Mora Gutierrez, and Antonin Ponsich. Aplicación del método de composición musical al problema de asignación de unidades de enseñanza y aprendizaje. *Ingeniare. Revista chilena de ingenieria*, 22:292 – 299, 04 2014. ISSN 0718-3305. URL http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-33052014000200014&nrm=iso.

[14] Francisco Vilchez Vargas and José Astuvilca Fuster. Composición musical artificial con algoritmos genéticos. pages 18–21, 05 2014.