

Un compositor musical automático basado en una metaheurística cooperativa

G. Acampora
V. Loia

Dpto. di matematica
e Informatica
Università di Salerno

J.M. Cadenas
E. Muñoz

Dpto. de Ingeniería de la
Información y las Comunicaciones
Universidad de Murcia

R. De Prisco
R. Zaccagnino

Dpto. di Informatica
ed Applicazioni
Università di Salerno

Resumen

Desde los comienzos de la ciencia de la computación, la generación de artefactos artísticos ha atraído la atención de numerosos investigadores. Una de las artes que más se han beneficiado de este interés es la música, habiendo aparecido numerosos trabajos que demuestran la sinergia existente entre ambas disciplinas. En este trabajo nos centraremos en el problema del bajo no continuo, que consiste en generar de un modo automático una pieza de música de 4 voces, recibiendo como entrada la línea del bajo (en ocasiones acompañada de indicaciones acerca de que acordes usar). Para resolver este problema se propone un sistema metaheurístico cooperativo, en el que distintas metaheurísticas son ejecutadas en paralelo mientras intercambian información.

Palabras clave: Problemas dinámicos, Metaheurísticas cooperativas.

1. Introducción

Desde los primeros tiempos de la informática, tanto informáticos como artistas han estado considerando el uso de ordenadores en la producción de artefactos. La música es una de las artes que más se han beneficiado de la aparición de los ordenadores. La música computacional es tan antigua como los propios ordenadores: la suite de ILLIAC¹ [7, 8] es la

primera pieza musical compuesta por un ordenador digital y data del 1957. El programa que la generó fue escrito por Hiller.

El termino “música computacional” abarca varios aspectos en las dos disciplinas, incluyendo programas para notación musical, síntesis de sonido, performances en tiempo real, instrumentos digitales, etc. Nosotros nos centramos en la *composición algorítmica*, es decir, el problema de componer música mediante un programa de ordenador, sin ninguna intervención humana.

La composición de música algorítmica ha fascinado tanto a informáticos como a músicos. La literatura está llena de trabajos en este sentido, empezando con el trabajo de Hiller y llegando a los más recientes, como, por ejemplo los trabajos propuestos por D. Cope [2, 3, 4]. De hecho existen trabajos previos a la era de los ordenadores que pueden ser considerados “algorítmicos” (como por ejemplo el juego de dados de Mozart).

Se han usado varias herramientas para obtener compositores algorítmicos: números aleatorios, gramáticas formales, autómatas celulares, fractales, inteligencia artificial, algoritmos evolutivos, música genética (música basada en ADN y proteínas). El libro de Miranda [11] contiene una buena revisión.

La música occidental, a partir del llamado periodo de la práctica común, está basada en el sistema de música “Temperamento igual”. Este sistema de música es en el que se escribe toda la música occidental y está basado en reglas ampliamente aceptadas. El problema que

¹ILLIAC - un ordenador construido en los 50.

trataremos es uno muy específico, conocido como el problema del bajo continuo. Este problema surgió en el siglo XVI aunque encontró su auge durante el barroco. En este problema el compositor solamente crea la voz del bajo, pero no especifica el contrapunto o los acordes del ripieno (es decir, el resto de voces) que deja a cargo del interprete, el cual debe improvisar aquellos que mejor se avienen a la armonía del conjunto. En ocasiones la composición incluye *cifras* o indicaciones para guiar al intérprete en la elección de los acordes, en este caso, el problema se conoce como problema del bajo continuo cifrado. Por tanto, en este problema se tiene una línea de bajo como entrada y el objetivo es componer las otras 3 voces para completar una pieza completa de música a 4 voces. Ejemplos inalcanzables de estas composiciones son los corales de Bach.

A menudo los algoritmos se utilizan para encontrar un óptimo o soluciones buenas, donde óptimo y bueno están definidos de acuerdo a una métrica definida precisamente. Para composiciones musicales esta métrica precisa no existe. Esto es un serio problema porque no existe una forma fácil de decir que una solución es “mejor” que otra. La bondad de una composición musical (como cualquier otra producción artística) es una opinión subjetiva. En el contexto del problema que estamos considerando, las reglas armónicas bien conocidas son de ayuda, aunque no constituyen una métrica.

La existencia de una métrica precisa reduciría el problema de composición de música a un problema en un espacio de búsqueda muy grande: la composición de música se podría ver como un problema combinatorio que consiste en colocar un número finito de notas en un pentagrama finito. Desafortunadamente (¿o afortunadamente?), como se ha dicho, esta métrica no existe. Para “comparar” soluciones usamos reglas de armonía.

Como se ha comentado anteriormente, existen muchas herramientas y técnicas que pueden usarse para diseñar un compositor algorítmico. Las metaheurísticas son una de ellas. Las metaheurísticas [9] son estrategias inteligentes usadas para diseñar y mejorar pro-

cedimientos heurísticos muy generales con un alto rendimiento. Dentro de ellas un campo que está obteniendo muy buenos resultados son las metaheurísticas cooperativas, que son estrategias en las que metaheurísticas simples cooperan para encontrar una solución. En este trabajo se presenta una metaheurística cooperativa adaptativa para resolver el problema del bajo continuo. Esta estrategia se adapta a las distintas instancias del problema que puedan aparecer mediante la utilización de conocimiento adquirido a través de ejecuciones anteriores de las metaheurísticas que la componen.

El resto del trabajo se organiza de la siguiente manera. En la Sección 2 se introducen algunos conceptos de música necesarios para entender el problema del bajo continuo. En la Sección 3 se presenta el compositor propuesto. En la Sección 4 se muestran algunos resultados obtenidos con el mismo y finalmente en la Sección 5 se presentan las conclusiones.

2. Antecedentes musicales

En este trabajo se considera el sistema musical de temperamento igual, usado en los países occidentales. Este sistema, define octavas empezando por un sonido de referencia (una frecuencia dada, normalmente 440Hz) que son aquellos sonidos cuyas frecuencias pueden ser obtenidas repetidamente doblando/reduciendo a la mitad la frecuencia de referencia. Cada octava está dividida en 12 notas espaciadas igualmente², denotadas por las letras A, A# o Bb, C, C# o Db, D, D# o Eb, E, F, F# o Gb, G y G# o Ab. La música de temperamento igual está basada en la noción de tonalidad. En un sentido poco estricto, una tonalidad es un grupo de notas que forman una escala. Empezando por cada una de las 12 notas de una octava se puede tener una tonalidad (hay varios tipos para cada nota, como mayor, menor, etc.). Por ejemplo, la escala mayor de C es C, D, E, F, G, A, B, mientras que la escala mayor de D es D, E, F#, G, A, B, C#. Las notas de una escala se denotan a menudo

²Para ser precisos si una nota tiene frecuencia f la “siguiente” nota tiene frecuencia $f \cdot 2^{\frac{1}{12}}$.

también como I, II, III, IV, V, VI, VII, VIII especialmente cuando se quiere enfatizar sólo el grado de la escala y no la nota particular, que depende de la tonalidad. Una pieza escrita en una tonalidad dada puede ser fácilmente transpuesta a otra tonalidad moviendo simplemente todas las notas. Esto es posible puesto que todas las notas están espaciadas igualmente. Una buena explicación de la música de temperamento igual se puede encontrar en [10].

Usualmente una tonalidad se considera la tonalidad principal de una pieza, y por tanto, las notas de la escala correspondiente se consideran más importantes que las notas fuera de la misma. La música occidental, empezando por el periodo de la práctica común, está basada en reglas armónicas y melódicas bien establecidas para el sistema de temperamento igual. Para una discusión acerca de estas reglas puede verse el libro de armonía [12].

El interés de este trabajo se centra en la música compuesta para 4 voces, como los corales de Bach. Un coral está compuesto de 4 voces independientes, llamadas *bajo*, *tenor*, *contralto* y *soprano*. Una pieza de música consiste en una secuencia de *compases*, cada uno consistente en un número de unidades de tiempo dadas. En cada unidad de tiempo las cuatro voces interpretan una nota ³. El conjunto vertical de notas en una unidad de tiempo es un *acorde*. La noción de acorde es fundamental. Hay numerosos tipos de acorde. En este trabajo se consideran sólo aquellos compuestos de 3 y 4 notas. Los acordes se construyen sobre cada grado de la escala, es decir, I, II, III, IV, V, VI, VII, VIII. A menudo la nota sobre la que se construye el acorde, llamada *nota raíz* del acorde, es dada al bajo; sin embargo el bajo puede interpretar cualquier otra nota (inversiones de acorde). Las inversiones de acorde son denotadas usualmente con un superíndice que indica la inversión (por ejemplo, III⁶, V⁴⁶, I³⁵⁷). Los acordes considerados se muestran en la Tabla 1 (para cada uno se da un ejemplo en la tonalidad de C).

³Esto es una simplificación de lo que sucede en realidad, puesto que una composición puede tener también notas de paso que no son parte de la armonía

Tabla 1: Acordes - con ejemplos en la tonalidad de C

Acorde	Nota raíz	Conjunto de notas
Triada mayor	C	C, E, G
Triada menor	E	E, G, B
Séptima mayor	C	C, E, G, B
Séptima menor	D	D, F, A, C
Séptima dominante	G	G, B, D, F
Séptima semidisminuida	B	B, D, F, A

Una de las reglas fundamentales del sistema de música de temperamento igual es que el conjunto vertical de notas debe ser uno de los acordes permitidos. Otras reglas afectan a secuencias de acordes. Algunas secuencias son “mejores” que otras, donde mejores es difícil de definir puesto que es una evaluación subjetiva. En cualquier modo está ampliamente aceptado que secuencias particulares de acordes funcionan mejor que otras. Algunos acordes son “más importantes” que otros porque sugieren, preparan o refuerzan centros tonales. El arte de la música tonal consiste precisamente en colocar los acordes de tal manera que su interacción sea agradable y significativa. Del lado práctico, esto se traduce en reglas simples que indican cosas como (tomada de [12]):

I es seguido de IV o V, algunas veces VI, menos a menudo II o III

II es seguido de V, algunas veces de VI, menos a menudo I, III o IV

Además de reglas acerca de secuencias de acordes, también existen reglas acerca de líneas melódicas. Una línea melódica es simplemente la secuencia de notas tocadas por cada voz. Las reglas acerca de líneas melódicas se pueden referir a el movimiento de una sola voz (por ejemplo, normalmente un salto mayor que una octava no está permitido) o también a los movimientos de dos voces (por ejemplo dos voces, que continúan con una quinta paralela no están permitidas). Para una descripción más detallada puede leerse el libro [12].

En la Figura 1 se presenta un ejemplo del problema considerado (y una posible solución). La parte *a* de la figura proporciona la



Figura 1: Ejemplo del problema

entrada: sólo una línea de bajo. La parte *b* muestra la misma línea pero con indicaciones acerca de los acordes a usar, mientras que la parte *c* proporciona una solución.

3. Un compositor de música automático

El problema del bajo continuo se puede resolver en dos fases diferentes. En la primera se deben obtener las cifras que guíen en la elección de los acordes, o lo que es lo mismo, se debe pasar del problema del bajo continuo al del bajo continuo cifrado. Para esta fase se utilizará un algoritmo basado en programación dinámica propuesto en [6]. Para la segunda fase, que trata de resolver el problema del bajo continuo cifrado, se usará una metaheurística cooperativa adaptativa. Para describirla es necesario introducir tres aspectos principales: la arquitectura del sistema, el esquema de cooperación que permite crear una sinergia entre las diferentes estrategias, y el proceso de extracción del conocimiento que otorga adaptabilidad al sistema.

Previamente se realizará una pequeña introducción a las estrategias metaheurísticas cooperativas. El desarrollo de metaheurísticas cooperativas consiste en la combinación de metaheurísticas existentes para obtener estrategias híbridas, en las que las metaheurísticas cooperan de modo paralelo. Debido a su capacidad de cooperar paralelamente para resolver una instancia de un problema, las metaheurísticas cooperativas están estrechamente relacionadas con las metaheurísticas paralelas. Muchos estudios [5] han demostrado que las metaheurísticas paralelas obtienen mejores resultados que sus componentes secuenciales, incluso cuando

el tiempo disponible es menor. Estos estudios también han mostrado que la combinación de distintos hilos, usando cada uno una metaheurística diferente, incrementa la robustez de la búsqueda global con respecto a los cambios en la instancia del problema. La estrategia propuesta combina distintas metaheurísticas que se ejecutan en paralelo mientras intercambian información. El intercambio de información se controla utilizando conocimiento previo acerca del rendimiento de las mismas.

3.1. Arquitectura

La estrategia se modela usando un sistema multiagente que contiene dos tipos de agentes, *agentes de optimización* y *agente coordinador*. Cada agente de optimización ejecuta una metaheurística y el agente coordinador es el responsable de iniciar el proceso activando paralelamente los agentes y coordinando la cooperación entre los agentes de optimización eligiendo los valores de sus parámetros y controlando su intercambio de soluciones de un modo adaptativo. (Figura 2).

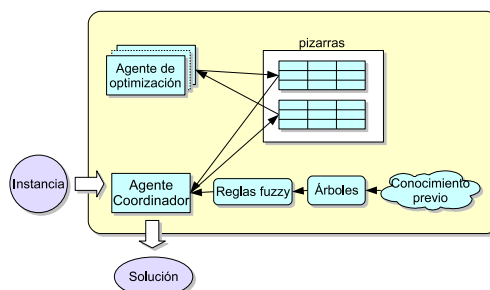


Figura 2: Un sistema multiagente cooperativo

La adaptabilidad del coordinador se con-

sigue con el conocimiento obtenido por medio de aprendizaje computacional y codificado como una colección árboles. En particular, los árboles ayudan al coordinador a: 1) elegir los mejores valores de parámetros para cada metaheurística y 2) ordenar la idoneidad de cada metaheurística para resolver la instancia que se esté considerando, utilizando un conjunto de pesos en el rango $[0, 1]$. A través del análisis de pesos, el coordinador comprobará si alguna metaheurística está comportándose pobremente y, en consecuencia, puede actualizar sus soluciones añadiendo un conjunto de soluciones mejores obtenido por otra metaheurística con un mejor comportamiento. Los agentes de optimización ejecutan asincrónicamente sus estrategias de optimización mientras envían y reciben información. El intercambio de información se realiza a través de un esquema de pizarra. Concretamente se utilizan dos pizarras, la primera la utilizan los agentes de optimización para publicar información sobre su comportamiento, la segunda la utiliza el coordinador para indicar las instrucciones para cada agente. Específicamente, después de un periodo especificado, cada metaheurística para y publica su solución/población actual (dependiendo de si es basada en trayectorias o en poblaciones). Entonces el agente coordinador analiza el comportamiento de cada metaheurística, teniendo en cuenta el conocimiento previo, y decide si alguna estrategia necesita mejorar su rendimiento. En este caso, el coordinador escribe en la segunda pizarra la nueva solución/población que las metaheurísticas tendrán que usar.

3.2. Esquema de cooperación

El esquema de cooperación de esta estrategia está definido principalmente por las reglas de control usadas por el coordinador y los árboles que guían su mecanismo de adaptación. Estos componentes tienen que ayudar al agente coordinador a:

- Elegir los mejores valores de parámetros para cada metaheurística.

- Descubrir cuando una metaheurística con un comportamiento pobre tiene que recibir nuevas soluciones de otras metaheurísticas con mejor comportamiento.

La primera tarea se realiza utilizando un conjunto de árboles producto de la aplicación de un proceso de aprendizaje supervisado. Cada uno de estos árboles está relacionado con un parámetro de una metaheurística. El agente coordinador realiza la segunda tarea considerando la información contenida en los árboles restantes. Estos árboles analizan la instancia que se va a resolver y ordenan las diferentes metaheurísticas cooperantes, devolviendo una colección de pesos $\Omega = \{\omega_i, i = 1, \dots, nm\}$ donde nm es el número de metaheurísticas, $\omega_i \in [0, 1]$ and $\sum_{i=1}^{nm} \omega_i = 1$. Cada peso ω_i está asociado con una metaheurística m_i y representa su idoneidad para resolver la instancia actual.

Para controlar el intercambio de soluciones, el coordinador utiliza el siguiente conjunto de reglas fuzzy (este conjunto de reglas se replica una vez por cada metaheurística):

si $(\omega_1 \cdot \delta_1)$ es *suficiente* **entonces**
 enviar soluciones de m_1 a m_h

si $(\omega_{nm} \cdot \delta_{nm})$ es *suficiente* **entonces**
 enviar soluciones de m_{nm} a m_h

donde:

- nm es el número de metaheurísticas.
- m_h es la metaheurística evaluada.
- $\delta_i = (\xi_{m_i} - \xi_{m_h}) / \text{maximum}(\xi_{m_i}, \xi_{m_h})$, donde ξ es una medida de rendimiento.
- $\omega_{m_i} \in [0, 1]$ donde $\sum_{i=1}^{nm} \omega_{m_i} = 1$ y representa el peso de la metaheurística m_i .
- *suficiente* es un conjunto fuzzy con función de pertenencia trapezoidal (con soporte $\in [0, 1]$) definido por:

$$\mu(x, a, b, c, d) = \begin{cases} 0 & x \leq a \text{ or } x \geq d \\ \frac{x-a}{b-a} & x \in (a, b] \\ \frac{d-x}{d-c} & x \in [c, d) \\ 1 & x \in [b, c] \end{cases}$$

con $(a, b, c, d) = (0, 0, 1, 1)$.

Este conjunto de reglas indica al coordinador cuando tiene que enviar nuevas soluciones a una metaheurística dada. En concreto, cambia la posición en el espacio de búsqueda de una metaheurística que está mostrando un comportamiento pobre por una posición cercana a la posición de otra metaheurística con un comportamiento mejor. El disparo de la regla está controlado por un α -corte definido por el usuario. En el caso de que se dispare más de una regla el coordinador aplica todas ellas.

Para cambiar la solución de una metaheurística se tiene en cuenta lo siguiente:

- $m_{receptor}$ es basada en trayectorias. Se envía una solución “cercana” a la mejor solución obtenida entre las metaheurísticas que han disparado la regla. “cercana” se obtiene aplicando $\frac{1}{2 * \omega_{m_{emisor}}}$ veces un operador de mutación.
- $m_{receptor}$ es basado en poblaciones.
 - ◊ m_{emisor} es basada en trayectorias. Una proporción de los peores individuos de $m_{receptor}$ igual a $\omega_{m_{emisor}}$ se sustituye por un conjunto de soluciones cercanas a la mejor solución de m_{emisor} .
 - ◊ m_{emisor} es basada en poblaciones. Una proporción de los peores individuos de $m_{receptor}$ igual a $\omega_{m_{emisor}}$ se sustituye por un conjunto de los mejores miembros (soluciones) de la población de m_{emisor} .
 - ◊ Distintas metaheurísticas tienen que enviar soluciones. Una proporción de los peores individuos de $m_{receptor}$ igual a la suma de los ω_i de los emisores se sustituye por un conjunto de soluciones donde cada m_{emisor} elige sus soluciones como se describió anteriormente.

3.3. Obteniendo los árboles

Para obtener los modelos de pesos y parámetros se utiliza un proceso de extracción como el que se muestra en la Figura 3, compuesto de dos fases diferentes, preparación de datos y

minería de datos. Se debe notar que este proceso es supervisado y previo a la ejecución de la estrategia.

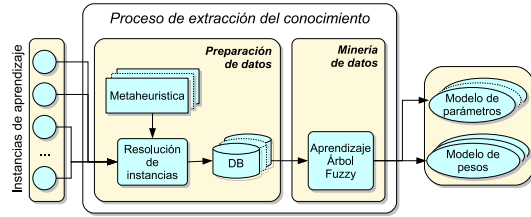


Figura 3: Proceso de extracción del conocimiento

Fase de preparación de datos

En esta fase las distintas metaheurísticas se ejecutan varias veces sobre distintas instancias de entrenamiento, usando distintos valores de parámetros para obtener una colección de conjuntos de datos que describan el comportamiento de cada metaheurística. Existen dos tipos diferentes de conjuntos de datos, de parámetros y de pesos. El primer tipo está relacionado con la elección de los valores de parámetros de las metaheurísticas y existe un conjunto de datos para cada parámetro de cada metaheurística, el segundo tipo está relacionado con la importancia de cada metaheurística para una instancia dada, y existe un conjunto de datos para cada metaheurística.

En particular cada conjunto de datos de parámetros, relacionado con un parámetro de una metaheurística, almacena para cada instancia una descripción de esta y el valor del parámetro que obtuvo un mejor rendimiento en la resolución de la misma. Por otro lado cada conjunto de datos de pesos, relacionado con una metaheurística, almacena para cada instancia una descripción de esta y un peso obtenido como $\omega_{m_i} = \frac{fit_{best}^{q,i}}{\sum_{l=1}^{nm} fit_{best}^{q,l}}$, donde:

$fit_{best}^{q,i}$ es el mejor valor de la función objetivo asociado a la metaheurística i y la instancia q .

Fase de minería de datos

Una vez se ha reunido la información y se han obtenido los conjuntos de datos, la fase de minería de datos extrae la colección de ár-

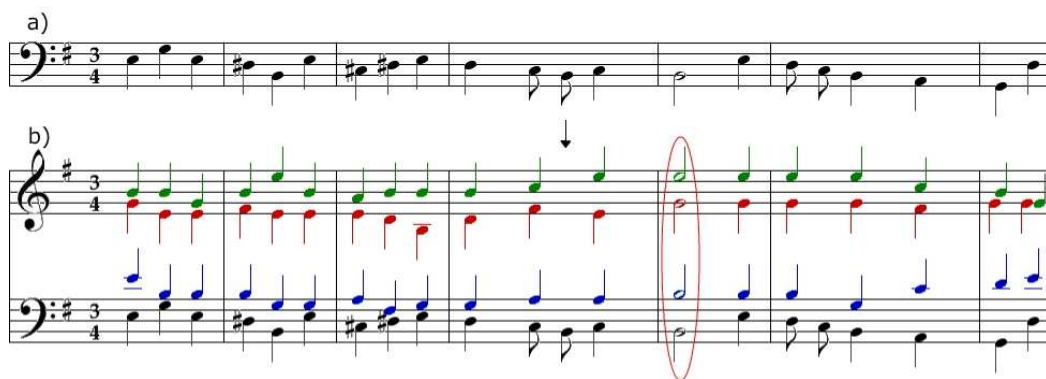


Figura 4: Un ejemplo de entrada (parte a), y solución final (parte b)

boles fuzzy. Se pueden distinguir dos tipos de modelos, *modelos de parámetros* y *modelos de pesos*. El primer tipo está relacionado con la elección de valores de parámetros realizada por el coordinador y existe un modelo por cada parámetro de cada metaheurística. Por otro lado el segundo tipo de modelos está relacionado con la obtención del conjunto de pesos Ω , y existe un modelo por cada metaheurística.

Una vez finalizada esta fase, obtenemos un conjunto de árboles que guían al coordinador, y este está disponible para controlar la ejecución de los agentes de optimización.

4. Análisis experimental

En esta sección se realizarán tests para comprobar la validez de la estrategia cooperativa propuesta.

4.1. Configuración del sistema

Usaremos un sistema compuesto por cuatro metaheurísticas diferentes: un algoritmo genético, una búsqueda tabú, un temple simulado y una optimización por enjambre de partículas.

El siguiente aspecto que debe ser tratado es el proceso de extracción del conocimiento. Como instancias de entrenamiento se han usado los corales de J.S. Bach, en particular 36 de ellos. Cada una de estas instancias se resolvió usando diferentes combinaciones de valores de

parámetros con cada una de las metaheurísticas con el objetivo de obtener los conjuntos de datos antes mencionados. Estos últimos son la entrada de la fase de minería de datos, cuya salida son los árboles fuzzy tanto de parámetros como de pesos.

4.2. Resultados

Se han ejecutado varias pruebas con distintas instancias diferentes de las utilizadas en el proceso de extracción del conocimiento. La Figura 4 muestra un ejemplo de los resultados obtenidos por la técnica, al aplicarla al coral BWV 266 de J.S. Bach. La parte *a* de la figura muestra la línea del bajo (entrada), la parte *b* muestra la solución generada por el sistema.

Los resultados obtenidos cumplen las expectativas, devolviendo soluciones de una calidad aceptable y en ningún caso discordantes. Esto es así porque, ayudada por las reglas que definen la función objetivo, la metaheurística cooperativa es capaz de eliminar todos los errores críticos así como la mayoría de los permitidos por las reglas armónicas más utilizadas. En particular, en la solución mostrada se han eliminado todos los errores críticos quedando solamente algunos errores permitidos. En la parte de la solución que se muestra en la figura se puede ver uno de estos errores. El error está localizado en el quinto compás y resaltado por una elipse roja. Este tipo de error se conoce como octavas paralelas y se produce cuando

dos voces están separadas por un intervalo de octava, y se mueven en la misma dirección terminando de nuevo con la misma separación. Para preservar la independencia de las voces se evitan las octavas paralelas, puesto que su utilización anula una de las voces.

Por último resaltar que el tiempo utilizado en la generación de las soluciones es pequeño, aproximadamente un minuto, lo que permite al decisor repetir varias veces el proceso pudiendo seleccionar aquella solución que más le guste. Esto es importante puesto que, como se resaltó en la introducción, la bondad de una composición musical es una opinión subjetiva.

5. Conclusiones

En este trabajo se ha presentado un compositor automático basado en una estrategia metaheurística cooperativa. Entendiéndose por automático que genera la composición sin intervención humana. El problema específico considerado es el del bajo continuo, en el que se debe generar una pieza de 4 voces a partir de la línea del bajo. Para obtener la composición final, el compositor debe transformar el problema en el problema del bajo continuo cifrado y una vez obtenido resolverlo. Este trabajo se centra en la segunda parte, la resolución del problema del bajo continuo cifrado, que se realiza utilizando una estrategia metaheurística cooperativa y adaptativa.

En la estrategia presentada un conjunto de metaheurísticas diferentes cooperan bajo la supervisión de un coordinador. Este coordinador es modelado usando el conocimiento extraído de un proceso de aprendizaje supervisado aplicado a datos históricos de la ejecución de las metaheurísticas individuales. Para comprobar la validez de la estrategia se han realizado distintas pruebas obteniéndose resultados interesantes.

Como trabajos futuros se pretende profundizar en la manera de transformar el problema en su versión cifrada así como ampliar las reglas y acordes que se pueden utilizar en su posterior resolución.

Agradecimientos

Los autores agradecen al MICINN y al FEDER por el financiación dada a este trabajo

bajo el proyecto TIN2008-06872-C04-03. También agradecen el programa para grupos de investigación de excelencia 04552/GERM/06 de la “Fundación Séneca”. También se agradece a la “Fundación Séneca” por la financiación ofrecida a E. Muñoz a través de un programa FPI.

Referencias

- [1] Cadenas, J.M., Garrido, M.C., Muñoz, E., *Using machine learning in a cooperative hybrid parallel strategy of metaheuristics*, Inf. Sci. 179(19), 3255–3267, 2009.
- [2] Cope, D., *Experiments in Musical Intelligence*, A-R Editions, 1996.
- [3] Cope, D., *The Algorithmic Composer*, A-R Editions, 2000.
- [4] Cope, D., *Virtual Music*, MIT Press, Cambridge, 2004.
- [5] Crainic, T.G., Toulouse, M., *Parallel Strategies for Metaheuristics*, Handbook of Metaheuristics, Kluwer Academic Publisher, 475–513, 2003.
- [6] De Prisco, R., Zaccagnino, R., *An Evolutionary Music Composer Algorithm for Bass Harmonization*, EvoWorkshops, 567–572, 2009.
- [7] Hiller, L., *Computer music*, Scientific American, 201 (6), 109–120, 1959.
- [8] Hiller, L., Isaacson, L.M., *Experimental music*, McGraw-Hill, New York, 1959.
- [9] Melián, B., Moreno, J.A., Moreno, J.M., *Metaheurísticas: una visión global*, Inteligencia Artificial 19, 7–28, 2003.
- [10] Loy, G., *Musimathics*, 1, The MIT Press, 2006.
- [11] Miranda, E.R., *Composing Music with Computers*, Focal Press, 2001.
- [12] Piston, W., DeVoto, M., *Harmony*, W.W. Norton, 1987.