

Composición algorítmica

Diana Garbanzo¹ y Oscar Esquivel²

CI-1441 Paradigmas computacionales

Escuela de Ciencias de la Computación e Informática

Facultad de Ingeniería

Universidad de Costa Rica

¹diana.garbanzo@ucr.ac.cr, ²oscar12esquivel@gmail.com

Julio de 2015

Resumen

La composición algorítmica ha sido objeto de estudio desde la década de los cincuentas. Múltiples paradigmas han sido utilizados para resolver este problema, algunos más naturales que otros para la representación musical. Para este estudio se busca generar canciones de rock, basadas en una estructura básica que predominó a partir de la década de los ochentas. Se propone el uso de un sistema de reglas para la generación de las canciones. Para ello, se utiliza el lenguaje Prolog, y para lograr la reproducción de la música, la biblioteca de sonido del lenguaje de programación Java. Debido a algunas dificultades las herramientas se utilizaron de forma desacoplada. Al finalizar el desarrollo y efectuar pruebas, las notas resultaron ser más secuencias de notas que una melodía agradable debido a la naturaleza aleatoria de las reglas. El programa desarrollado es también extensible, para componer canciones más complejas y/o de otros géneros musicales.

Palabras clave: composición, algorítmica, rock, ochentas, Prolog, Java, sistema, reglas.

1. Introducción

Este documento describe una solución dada al problema de composición algorítmica, en el cual se busca que la música pueda ser generada desde una compu-

tadora, a partir del uso de algoritmos variados según sea el caso.

Inicialmente se realiza una breve descripción de su historia, algunos proyectos significativos y diferentes formas o paradigmas con los que el problema ha sido resuelto con el pasar de los años. A continuación se realiza una descripción de la propuesta de la solución, incluyendo objetivos y planificación del proyecto. Luego se explica la forma en que se llevó a cabo para este caso particular. Posterior a esto, se describe la experimentación y el análisis de los resultados obtenidos. Finalmente, se mencionan algunos problemas abiertos y futuros.

Para llevar a cabo lo ya planteado, se utilizará el paradigma de investigación: prototipado rápido. El mismo “consiste en construir un artefacto y considerarlo como ejemplo de una clase más general de soluciones a una clase de problema dada” (de la Ossa, 2015). El mismo está dirigido por el método de la experimentación y puede conllevar a pérdidas de información en caso que no resulte exitoso el producto que se espera obtener.

Las limitaciones del estudio se encuentran en las siguientes características: falta de generalización de las estructuras de las canciones en el género rock (para el estudio particular), así como también existe falta de generalización en la composición para otros géneros musicales. Esto debido a la cantidad de tiempo con que se cuenta para llevar a cabo el proyecto.

2. Marco teórico

Desde la década de los cincuentas, ya existía un sistema de composición musical algorítmica (en adelante referida como composición algorítmica), el “Illiac Suite” creada por Hiller e Isaacson (1958). Dicho sistema dio paso a la biblioteca MUSICOMP; la cual es una biblioteca de subrutinas para computadoras, usadas por los colegas de Hiller y escrita por Baker, uno de los colegas de Hiller (Fernández, Vico, 2013).

Existe un trabajo previo al mencionado anteriormente, realizado por Caplin y Prinz en 1955, el cual no fue publicado. El mismo utilizó dos enfoques (Fernández, Vico, 2013): “primero una implementación del juego de dados de Mozart y un generador de líneas melódicas, usando probabilidades transicionales estocásticas para varios aspectos de composición”.

Algunas técnicas muy mencionadas en la investigación sobre composición musical, son: doce tonos de Schoenberg y las basadas en cadenas de Markov. Por ejemplo en el trabajo realizado por Olson en 1961 y el algoritmo de Gill, implementado en 1963, se utilizaron respectivamente dichas técnicas.

Por otra parte, existen algunos trabajos no publicados formalmente, como por ejemplo, el “Banal Tune-Maker” de Pinkerton en 1956, el cual utilizaba una simple cadena de Markov de canciones de cuna. Otro ejemplo es el trabajo de Kurzweil en 1965, el cual generaba música en un estilo de composición clásica (Rennie, 2010).

Existen varios métodos conocidos y utilizados en la composición musical, entre los cuales se encuentran: sistemas basados en reglas, cadenas de markov, gramáticas, redes neurales artificiales y métodos evolutivos.

2.1. Gramáticas

Una gramática formal puede ser definida como un conjunto de reglas para expandir símbolos de alto nivel a secuencias más detalladas de símbolos o palabras, que representan elementos de lenguajes formales. Estas palabras son generadas al aplicar de forma repetida, reglas de re-escritura en una secuencia de “pasos de derivación” (Fernandez and Vico, 2013).

2.2. Sistemas simbólicos basados en conocimiento

Dicho término abarca varios sistemas de reglas basados en diferentes paradigmas, con el factor común de representar el conocimiento con símbolos estructurados. Ya que el conocimiento sobre composición musical ha sido tradicionalmente estructurado como grupos de reglas formales para manipular símbolos musicales, este tipo de sistemas resultan ser naturales al implementar la composición algorítmica (Fernandez and Vico, 2013).

2.3. Aprendizaje de reglas

Aunque el conocimiento en los sistemas basados en reglas es usualmente estático, parte de éste puede ser dinámicamente modificado. El término para esto es “machine learning”, el cual abarca muchos métodos, como redes neuronales y cadenas de markov. Un ejemplo de esto es el sistema MUSE, un sistema basado en reglas que resuelve varias tareas de armonización en cuatro partes, implementado por Schwanauer en el 1993 (Fernandez and Vico, 2013).

2.4. Cadenas de Markov

Las cadenas de Markov consisten en un proceso estocástico, en el cual se transiciona de un estado a otro de acuerdo con una probabilidad asociada y la transición depende sólo del estado anterior. Las probabilidades asociadas pueden ser representadas en una matriz de transiciones o como un grafo dirigido y con pesos equivalentes a dichas probabilidades. Cuando son aplicadas a la composición musical, las matrices pueden provenir de composiciones previas o de derivaciones manuales de teoría musical. En herramientas de desarrollo de software se utiliza con más frecuencia la segunda opción (Fernandez and Vico, 2013).

2.5. Redes neuronales artificiales (RNA)

Son modelos computacionales que consisten en un conjunto de neuronas artificiales: dispositivos computacionales muy simples que agregan entradas numéri-

cas a una salida numérica usando, generalmente, una simple función no lineal. Normalmente las neuronas están organizadas en redes recurrentes con varias capas interconectadas entre sí, y se pueden encontrar muchas variaciones de implementación en la literatura. Usualmente son utilizadas como un método de aprendizaje automático, en el cual se utiliza un conjunto de ejemplos para entrenar la red, con el fin de utilizarlo para reconocer o generar patrones similares (Fernandez and Vico, 2013).

Debido a esto, se necesita que exista un conjunto de composiciones de música, para poder imitar su comportamiento. Un aspecto importante es la manera en que las composiciones se alimentan a los RNAs: se pueden presentar como patrones temporales en las entradas de red o alimentados a la vez (como totalidades).

2.6. Métodos evolutivos

Los métodos evolutivos están basados en poblaciones de soluciones. Estos métodos se basan en generar, seleccionar, combinar y reemplazar un conjunto de soluciones. Las mismas se relacionan entre ellas y compiten entre sí. El objetivo es que las mejores “mutaciones”, sobrevivan a lo largo del proceso, obteniendo cada vez mejores resultados (Alba et al., 2015).

3. El problema

¿Cómo crear una máquina que a partir de valores numéricos aleatorios pueda componer música, sin ayuda de una persona? Este espacio se redujo a componer canciones de rock, según la estructura básica que predominó en el género a partir de la década de 1980.

4. Objetivos y cronograma

4.1. Objetivo general

Implementar un programa que, utilizando un sistema de reglas, el cual use valores numéricos aleatorios generados en tiempo de ejecución, pueda generar

secuencias de notas musicales agradables, según el género musical: rock, secuencias que sigan la estructura típica de las canciones de este género.

4.2. Objetivos específicos

Para implementar el programa compositor se deben implementar dos componentes principales: el sistema de reglas y una interfaz de usuario.

4.2.1. Sistema de reglas

Implementar un sistema de reglas que contenga la lógica de la composición algorítmica, en un lenguaje que permita un encadenamiento de reglas.

4.2.2. Interfaz

Implementar una interfaz que interactúe con el usuario, que permita iniciar la composición y que una vez que ésta finalice, permita reproducir la canción obtenida; en un lenguaje que posea facilidad para la elaboración de interfaces y reproducción musical básica.

4.3. El cronograma

Para el desarrollo del proyecto se planificaron dos etapas: la primera referente a investigación del paradigma más idóneo para llevar a cabo la implementación, y la segunda sobre el desarrollo del producto. La primera fue planificada del primero al treinta y uno de mayo del año dos mil quince, y la segunda, durante el mes de junio del mismo año. Además, se planificó la redacción del informe del proyecto para el mes de julio, previo a la fecha de entrega.

5. La propuesta de solución

Para llevar a cabo el desarrollo del proyecto se planificó la implementación del sistema de reglas en el lenguaje Prolog, mientras que para la interfaz se usa Java, lenguaje que posee una biblioteca de sonido que permite reproducir música en formato MIDI. El sistema de reglas estaría basado en una estructura de canción de rock (determinada a partir del estudio de

varias canciones de la década de 1980 en adelante), y generaría notas con una duración asociada, a partir de una escala escogida al principio.

Se propone completar las partes de la canción definidas en la estructura antes mencionada, mediante el uso de dichas notas. Para ello se representa la información de cada nota como pares de números: el primero correspondiente al sonido (un sonido único según la enumeración de teclas en un piano) y el segundo a su duración (en tiempos, por ejemplo 4 tiempos si es una redonda, o 2 si es una blanca, etc.). Por otra parte, la representación del conocimiento se da mediante las reglas mismas en Prolog. El flujo de trabajo esperado en cada ejecución del programa es entonces el siguiente:

- Escoger una escala
- Generar una melodía
- Generar una transición
- Generar versos
- Generar coros

6. Desarrollo, prueba y validación

Para el desarrollo del proyecto se utilizaron las herramientas: Swi-prolog y Netbeans. En la primera se establecieron las reglas con la siguiente estructura: una regla general que usa a otras subreglas, las cuales componen las diferentes secciones de la canción y pueden usar otras reglas auxiliares, hasta llegar a una regla que recursivamente genere notas hasta alcanzar la duración escogida para esa sección de la canción. Al final se unen y reutilizan las secciones generadas para terminar con una lista de pares de números que se imprime en un archivo de texto, cuyo contenido luego será leído desde Java.

Netbeans fue utilizado para implementar una interfaz en Java que permite escuchar la canción que es producida desde Prolog. En primera instancia se planeó el uso de NetLogo, dada su mayor facilidad para reproducir música en comparación con Java, agregando un acoplamiento con Prolog mediante la librería

"NetProLogo". Sin embargo, a pesar de la facilidad de la misma, no se logró desreferenciar la lista resultante desde NetLogo, por lo se optó por utilizar Java.

En este nuevo enfoque se buscó hacer el llamado a Prolog desde la aplicación mediante el uso de la biblioteca "jpl" provista por Swi-prolog. Al principio la conexión fue exitosa, por lo que se planeó que desde la aplicación el usuario escogiera el tipo de canción que deseaba generar (aleatoria o con una melodía ingresada por el usuario) y en el caso de ser la segunda opción, pudiera escoger las notas y su duración. Sin embargo, al intentar generar el archivo ejecutable portable de Java, no se logró que éste funcionara según lo esperado, ya que el llamado a Prolog nunca ocurría, por lo que en última instancia que desacoplaron las funcionalidades: desde el intérprete de Swi-prolog se genera la canción y desde la aplicación el usuario selecciona el archivo resultante para ser escuchado. Debido a este desacoplamiento, la posibilidad de ingresar una melodía para que el programa componga una canción a partir de la misma, tuvo que ser eliminada del alcance del programa.

7. Experimentación y análisis

El programa se probó múltiples veces, evaluando en cada iteración la longitud en minutos, el apego a la escala escogida, y la variedad de duraciones de las notas en el resultado. Uno de los principales aspectos que debieron mejorarse después de cada iteración fue la longitud de las secciones de la canción, estas longitudes se escogen aleatoriamente pero dentro de un rango predeterminado; éstos rangos debieron modificarse ya que al principio las canciones eran demasiado cortas.

Otro aspecto que se mejoró fueron las escalas que el programa tenía para escoger, al principio se permitía usar desde las notas más graves hasta las más agudas, pero las notas graves no se apreciaban lo suficiente, por lo que se eliminaron de las escalas para generar notas en general más agudas.

Después de los aspectos anteriores, lo siguiente que se notó fue que no siempre las canciones resultaban agradables. Dada la naturaleza aleatoria del progra-

ma en la mayoría de los casos la secuencia de notas sonaba como sólo eso, una secuencia de notas; en muy pocos casos se produjo una secuencia que resultara agradable e interesante en su mayor parte.

8. Problemas abiertos y problemas futuros

El programa, al trabajar con un sistema de reglas, termina siendo bastante extendible. Queda abierto a la posibilidad de agregar muchas más reglas, sea para componer canciones más complejas, o para componer canciones de otros géneros musicales. Además, también queda abierto a la posibilidad de componer canciones en uno o varios géneros dada una melodía ingresada por el usuario, funcionalidad que tuvo que eliminarse durante el proceso de desarrollo.

Referencias

- Alba, E. et al.
2015. Métodos evolutivos.
- de la Ossa, A.
2015. La investigación científica en las ciencias de la computación.
- Fernandez, J. D. and F. Vico
2013. Ai methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, (48).