

Grafos y Música

Juan David Martínez Mercado

2020-II

Introducción

Cuando se empieza a aprender teoría musical es común empezar con los tonos y semitonos, notas y silencios, intervalos, grados y las escalas musicales, todo esto para poder luego escribirlo y leerlo en un pentagrama, que suele ser difícil de comprender al principio; una alternativa a todo esto es convertir los conceptos básicos como semitonos, notas y escalas, en un grafo circular si representamos una octava en un piano o un camino cuando es una melodía o partitura; en este proyecto se buscara diseñar un modelo basado en teoría de grafos para entender de otra forma los conceptos básicos de teoría musical, con esto se podrá analizar una partitura de forma más orgánica calculando el centro del grafo generado, de esta forma cuando se quiere saber que notas son más frecuentes podremos usar una representación gráfica que dibuje los vértices con mayor grado mas grandes, o también crear un árbol de expansión de la partitura que muestre de una forma diferente la canción.

Descripción del problema

Se quiere visualizar y encontrar una relación entre la complejidad de la obra según la cantidad de vértices y la excentricidad usando los conceptos básicos de teoría musical, tales como notas, tonos, acordes y octavas representados en un grafo y todo esto visualizarlo en dos o tres dimensiones.

Objetivo general

Diseñar un modelo en grafos que represente las notas musicales básicas de una escala cromática, los semitonos o tonos y usarlo en los algoritmos de Backtracking y Floyd Warshall y luego visualizarlos en dos o tres dimensiones.

Objetivos específicos

- Definir que representan los acordes, tonos y las notas, para luego pasarlos al formato de un grafo.
- Crear el algoritmo que lea los archivos con extensión xml (los cuales son las partituras en un formato digital) para convertir la partitura en un grafo.
- Utilizar el algoritmo de Floyd para calcular la matriz de distancia mínima que se utilizara para calcular el centro.
- Crear el algoritmo de búsqueda en anchura que sera usado para simplificar la partitura.
- Probar el grafo que se obtuvo de la partitura y aplicarle el algoritmo de búsqueda en anchura para encontrar su árbol de expansión.

Marco teórico

A continuación se definirán los términos que se usaran en este proyecto.

Grafos

Definimos un grafo G como una terna que consiste en un conjunto de vértices llamado $V(G)$, un conjunto de aristas $E(G)$ y una relación que asocia a cada arista un par de vértices (extremos) no necesariamente distintos.[1]

Bucle

Es una arista cuyos extremos son iguales.

Aristas múltiples o paralelas

Son aristas que tienen el mismo par de extremos.

Grafo simple

Definimos un grafo simple $G = (V, E)$ como un grafo sin bucles ni aristas múltiples, siendo E los pares no ordenados de vértices.

Ciclo

Es un grafo simple con el mismo número de vértices y aristas cuyos vértices pueden ubicarse alrededor de un círculo de tal manera que dos vértices son adyacentes si aparecen de manera consecutiva sobre el círculo.

Grafo acíclico

Es un grafo sin ciclos.

Grafo conexo

Un grafo G es conexo si existe un u,v -camino entre cada par $uv \in V(G)$.

Subgrafo

Un subgrafo ($H \subseteq G$) de un grafo G es un grafo H tal que:

- $V(H) \subseteq V(G)$
- $E(H) \subseteq E(G)$
- La asignación de extremos a las aristas en H es la misma que en G .

Subgrafo inducido

de G es un subgrafo G' de G tal que contiene todas las aristas adyacentes al subconjunto de vértices de G . [2]

Subgrafo de expansión

Un subgrafo de expansión de G es un subgrafo con conjunto de vértices $V(G)$.

Caminata

Una caminata en un grafo G es una lista $v_0 e_1 v_1 \cdots e_k v_k$ de vértices y aristas tal que $\forall 1 \leq i \leq k$ la arista e_i tiene extremos v_{i-1} y v_i .

Sendero

Un sendero se define como una caminata sin aristas repetidas.

Camino

Un camino es un sendero sin vértices repetidos.

Árbol

Un árbol es un grafo simple, acíclico y conexo.

Árbol de expansión

Un árbol de expansión de un grafo G es un subgrafo de expansión que es un árbol.

Distancia

Si G tiene un u, v -camino, la distancia de u a v notada $d(u, v)$ es la longitud mínima de un u, v -camino. Si G no tiene dicho camino la distancia es ∞ .

Excentricidad

La excentricidad de un vértice u es:

$$\epsilon(u) = \max_{v \in V(G)} d(u, v)$$

Centro

El centro de un grafo G es el subgrafo inducido por los vértices de mínima excentricidad.

Música

En este proyecto usaremos el sistema de afinación construido mediante la división de la octava en doce partes iguales llamadas semitonos temperados, es decir el temperamento igual.[3]

Frecuencia

La frecuencia es el número de repeticiones por unidad de tiempo de cualquier evento periódico.[4]

Nota

Una nota es un monosílabo (en el sistema inglés de notación) que representa una frecuencia de sonido.

Acorde

Tres o mas notas tocadas al mismo tiempo se define como acorde.

Intervalo

Un intervalo es la distancia que existe entre dos notas. Esta distancia se mide en unidades de semitonos o tonos (un tono equivale a dos semitonos). El intervalo mínimo de medio tono es siempre igual entre una nota y su consecutiva en el temperamento igual, de modo que:

$$F_{n+1} = F_n 2^{\frac{1}{12}}$$

Donde F es la frecuencia.[5]

Escala

Una escala es una serie de sonidos que van desde la frecuencia más baja a la más alta siguiendo intervalos de frecuencia definidos.[6]

Escalas mayores

Las escalas mayores son escalas diatónicas (todos los intervalos sucesivos son de uno o medio tono) y heptatónicas (se construyen con siete notas).[5]

Escalas menores

Las escalas menores son también escalas diatónicas y heptatónicas. Un aspecto importante es que si bien solo existe un tipo de escala mayor, existen tres variaciones de la escala menor. Estas tres variaciones son la escala menor natural, la armónica y la melódica.[5]

Escalas pentatónicas

Las escalas pentatónicas, como su nombre indica, se construyen a partir de solo cinco notas. Estas cinco notas pueden pertenecer o bien a la escala mayor o a la escala menor. Según el caso puede distinguirse entre escalas pentatónicas mayores o escalas pentatónicas menores.[5]

Grados musicales

Los grados musicales son los nombres que reciben las notas de una escala según su posición con respecto a la nota principal de la escala.[5]

Octava

Se denomina octava al intervalo de ocho grados entre dos notas de la escala musical en un piano.[6]

Partitura

Una partitura es un documento que indica como debe interpretarse una composición musical.

Modelamiento del problema

Para poder resolver el problema se dará a cada concepto una interpretación; así pues los nodos de los grafos que se presentaran en este proyecto se definen como las notas o acordes, las aristas serán, ya sea los intervalos en una octava o la siguiente nota en una partitura, su peso dependerá en cual de los dos estemos, si es una escala cromática entonces el peso de cada arista sera de $\frac{1}{2}$, es decir un semitono, y si estamos en una partitura el peso de la arista sera de 1, con esto tendremos dos tipos de grafos, un grafo circular que representa la octava, y un grafo dirigido que representa la partitura; cabe aclarar que las escalas pentatónicas son subgrafos de las escalas diatónicas que a su vez son subgrafos de una escala cromática.

Solución propuesta

Se utilizara el algoritmo de búsqueda en profundidad o Backtracking para encontrar un árbol de expansión desde cualquier nodo, también se usara el algoritmo de Floyd Warshall para encontrar el centro del grafo a partir de la excentricidad mínima en todos los vértices, estos dos algoritmos se utilizaran en las partituras que se encuentran en archivos xml y en cuanto a la visualización se usaran herramientas de Python y html.

Para generar las escalas mayores y menores a partir de la escala cromática, se creara un archivo xml el cual pueda ser leído por el programa y luego convertido en un grafo circular que muestre los colores de las teclas de un piano, los nombres de cada vértice y el valor de los intervalos o aristas, para poder guardarlo en un pdf, la idea de este algoritmo es mostrar que la tercera nota de una escala comparte las mismas notas en una escala ya sea mayor o menor, dependiendo desde cual escala se escoja la nota.

Implementación de la solución

El lenguaje de programación que se uso para la implementación de la solución fue Python, así pues, se crearon los siguientes archivos, carpetas y funciones:

- ***cell1.py***
 - **pasar_na_num(G):**
Recibe un grafo de la libreria networkx. Retorna una lista con los extremos de las aristas como números.
 - **c_2D():**
Convierte un archivo .xml en un grafo 2 dimensional agrupando las notas y acordes pertenecientes a cada compás en una lista siendo estos los vértices. Muestra un grafo dirigido con los vértices pintados con una escala de color que va desde los azules a los verdes, siendo los azules las notas o acordes con menor grado y los verdes con mayor grado; muestra también las aristas con una escala de color que va desde los blancos hasta los azules, así la arista mas blanca representa el inicio de la partitura y el más azul el final. Retorna el grafo y el nombre del archivo.
 - **c_3D(G, nombre):**
Recibe un grafo en 2 dimensiones y el nombre de la partitura y transforma el grafo en uno de tres dimensiones.
 - **c_csv(grafo1):**
Recibe lo que retorna la función c_2D y crea un archivo csv que contiene las columnas Source y Target, que representan las aristas del grafo.
- ***cell2.py***
 - **song_to_dict():**
Convierte la partitura en un diccionario que incluye los instrumentos numerados del 0 hasta el total de instrumentos-1, compases, notas y acordes de todo el documento, retornando una tripla con el diccionario, una lista con todas las notas y otra con los acordes y por último el nombre.
 - **m_graph(n, nombre):**
Recibe una lista con los vértices del grafo y el nombre del archivo, lo convierte en un grafo 2 dimensional. Retorna el grafo y el nombre del archivo.
- ***cell3.py***
 - **Clase: GrafoSimple**
 - **crear_multigrafo(self, g2D):**
Recibe como parámetros una tripla de la función song_to_dict y lo que retorna la función m_graph. Crea un pdf o un png o un svg del grafo.
- ***BA.py***

- **Clase: BuquedaProfundidad**

- **tG(self, G):**
Recibe un grafo G y retorna un diccionario de adyacencia.
- **dfs(self, visitados, graph, node, e_eT):**
Es la función recursiva de Backtracking, recibe un conjunto de visitados, un diccionario de adyacencia, el nodo donde empieza y una lista para llenar de aristas.
- **arbol_de_expan(self, g, g2D):**
Esta función recibe un diccionario de la función song_to_dict y una tripla de la función m_graph, y utiliza estos parámetros para crear ya sea una imagen o un pdf del árbol de expansión.

- **FW.py**

- **print_solution(distance,s):**
Recibe una matriz distance y el tamaño de esta. Imprime la matriz de en consola.
- **centro_FW(G,nombre):**
Recibe un grafo de la librería networkx y el nombre del grafo. Ejecuta el algoritmo de Floyd Warshall y muestra el centro en un png.

- **main.py**

- **Carpeta: Escalas**

- **Escalas.py**

- **getMusicProperties(x):**
Recibe ya sea una nota o un acorde y lo convierte en una cadena de texto.
- **crear_escala(mM,nota):**
Recibe si la escala es mayor o menor y desde que nota empezar a crear la escala. Guarda las escalas en forma de grafo en un png.

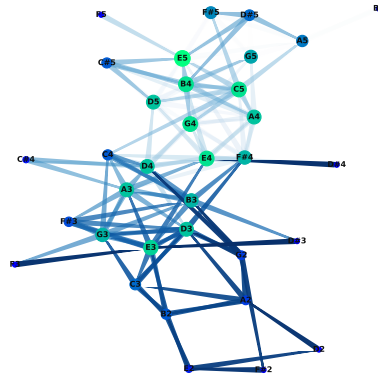
- **main.py**

Así pues para crear las imágenes se usó la librería NetworkX y para generar el html en el que se encuentra el grafo en formato tridimensional se usó la librería plotly. Los archivos xml se encuentran en las páginas que recomienda la página de MusicXml[7]

Resultados obtenidos

Usando partituras de canciones

A continuación se mostrara el grafo que se genera a partir de la partitura 988 Aria de Johann Sebastian Bach [Figura 1].



Continuando con los algoritmos si aplicamos el algoritmo de Floyd Warshall y un algoritmo para hallar el centro en el grafo 1 obtenemos este grafo [Figura 3].

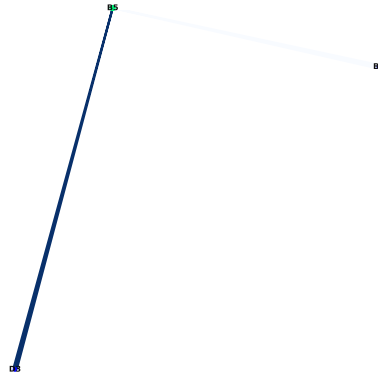


Figura 3: Centro del grafo 1

Generando grafos de escalas

A continuación se muestra la escala cromática que empieza desde C, convertida en un grafo [Figura 4].

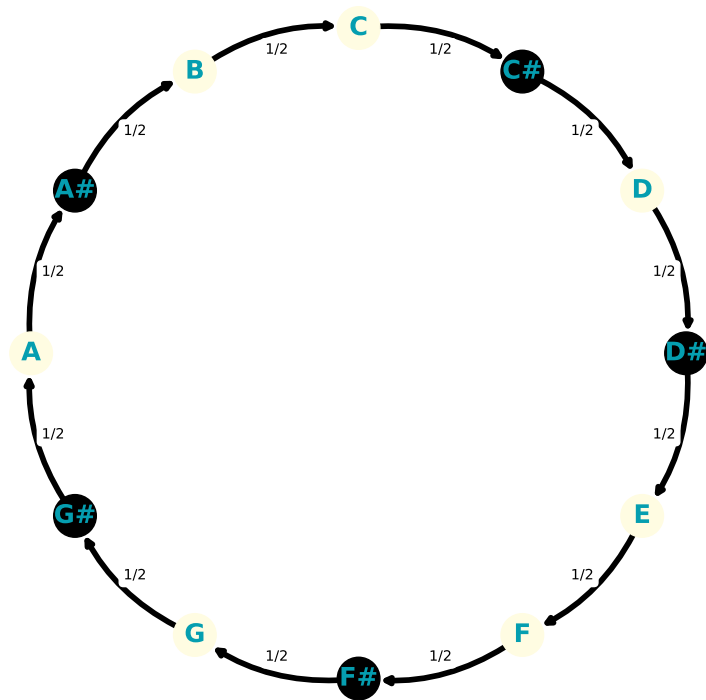


Figura 4: Escala cromática que inicia desde de C

Teniendo en cuenta el grafo de la Figura 4 podremos crear la escala mayor de C [Figura 5] y la escala menor natural de C [Figura 6].

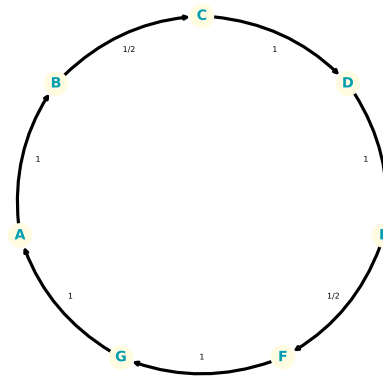


Figura 5: Escala mayor de C

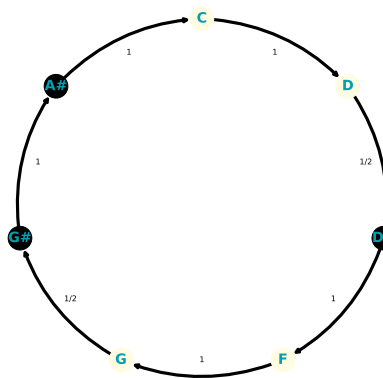


Figura 6: Escala menor natural de C

Conclusiones

En el proyecto se logro hacer un acercamiento a la teoría musical gracias a los grafos, así pues, el uso de estos mapas mentales pueden ayudar a entender de otra manera las escalas y la música.

Se puede decir que este proyecto solo toma en cuenta las cosas básicas de teoría musical, pero si se puede profundizar mas agregando funciones al programa o estudiando otras cuestiones en la teoría musical, con esto en mente el proyecto es un pequeño paso a una nueva manera de ver la música.

Dado al tiempo no se pudo crear una interfaz gráfica por lo cual el programa solo funciona desde una terminal y un computador con todas las librerías necesarias, seria bueno si en un futuro se pudiera crear una interfaz gráfica y todo el programa meterlo en un contenedor.

Dada la falta de datos, no se puede sacar una conclusión de si la complejidad de la obra depende del centro o de otro factor, con mas datos y presupuesto si se lograría el objetivo.

Bibliografía

- [1] Douglas B. West, Introduction to graph theory, 2005.
- [2] http://www.unipamplona.edu.co/unipamplona/portalIG/home_23/recursos/general/11072012/graf3.pdf
- [3] https://es.wikipedia.org/wiki/Temperamento_igual
- [4] <https://es.wikipedia.org/wiki/Frecuencia>
- [5] https://www.acordespiano.com/escalas/#Escalas_cromaticas
- [6] <http://www.ehu.eus/acustica/espanol/musica/somues/somues.html>
- [7] <https://www.musicxml.com/music-in-musicxml/>