

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

FELIPE ANDRÉ WIELER

**COMPARAÇÃO SINTÁTICA E SEMÂNTICA DE MÚSICAS ATRAVÉS DE
GRAFOS**

CURITIBA

2023

FELIPE ANDRÉ WIELER

**COMPARAÇÃO SINTÁTICA E SEMÂNTICA DE MÚSICAS ATRAVÉS DE
GRAFOS**

Syntactic and Semantic comparison of songs through graphs

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Licenciatura em Matemática do Curso
de Licenciatura em Matemática da Universidade
Tecnológica Federal do Paraná.

Orientador: Prof. Dr. João Luis Gonçalves

CURITIBA

2023



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

FELIPE ANDRÉ WIELER

**COMPARAÇÃO SINTÁTICA E SEMÂNTICA DE MÚSICAS ATRAVÉS DE
GRAFOS**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Licenciatura em Matemática do Curso
de Licenciatura em Matemática da Universidade
Tecnológica Federal do Paraná.

Data de aprovação: 15/junho/2023

André Fabiano Steklain Lisbôa
Doutorado
Universidade Tecnológica Federal do Paraná

João Luis Gonçalves
Doutorado
Universidade Tecnológica Federal do Paraná

Roy Wilhelm Probst
Doutorado
Universidade Tecnológica Federal do Paraná

**CURITIBA
2023**

RESUMO

Este trabalho tem por objetivo aprimorar o processo de análise musical, usando grafos como representações de músicas, como proposto pelo Leonardo Cruzato (2021). Vamos decompor a análise em duas partes, a sintática e a semântica. Na análise sintática usaremos o algoritmo VF2 para comparar estruturalmente os grafos, através da identificação de isomorfismos. Na análise semântica usamos os espectros dos grafos além de uma análise sobre os ciclos do grafos e seus formatos, caracterizados pelo desvio padrão dos pesos de suas arestas.

Palavras-chave: grafos, músicas, espectro e isomorfismo.

ABSTRACT

This work aims to improve the musical analysis process, using graphs as representations of songs, as proposed by Leonardo Cruzado (2021). We will decompose the analysis into two parts, the syntactic and the semantic. In the syntactic analysis we will use the VF2 algorithm to structurally compare the graphs, through the identification of isomorphisms. In the semantic analysis we used the spectra of the graphs in addition to an analysis of the cycles of the graphs and their formats, characterized by the standard deviation of the weights of their edges.

Keywords: graphs, music, spectrum and isomorphism.

LISTA DE FIGURAS

Figura 1 – Respresentação das pontes de <i>Köningsberg</i>	11
Figura 2 – Respresentação do grafo das pontes de <i>Köningsberg</i>	12
Figura 3 – Grafo Acíclico, Arvore e Conexo	12
Figura 4 – Grafo da música Piano Melody de Lemoncreme	14
Figura 5 – Grafo com 20 notas da música Piano Melody de Lemoncreme	28
Figura 6 – Grafo para alteração de F para A#	30
Figura 7 – Grafo para alteração de C para D	33
Figura 8 – Diferença nos espectros quando nota alterada de C para C#	37
Figura 9 – Diferença nos espectros quando nota alterada de C para G	38
Figura 10 – Grafo G	40
Figura 11 – Grafo H	41
Figura 12 – Espectro dos grafos G e H	41
Figura 13 – Grafo com 20 Notas de Piano Melody	43
Figura 14 – Grafo com 20 Notas da 5 sinfonia	43
Figura 15 – Espectro dos grafos	44

LISTA DE TABELAS

Tabela 1 – Desvios-padrão dos Exemplos 3,4 e 5.	42
Tabela 2 – Desvios-padrão das Músicas do Lemoncreme e do Beethoven.	44

SUMÁRIO

1	INTRODUÇÃO	7
2	REVISÃO DE LITERATURA	9
3	TEORIA DE GRAFOS	11
3.1	Espectro de Grafos	16
4	SISTEMA DE INTERVALOS GENERALIZADO - GIS	18
5	COMPARAÇÃO ENTRE GRAFOS	20
5.1	Análise do algoritmo VF2	20
5.2	Viabilidade Sintática	21
5.2.1	Regras de viabilidade sintáticas	21
5.3	Viabilidade Semântica	25
5.3.1	Algoritmo modificado de Dijkstra	25
5.3.2	Comparação do Espectro do Grafo	36
5.3.3	Desvio-Padrão dos Ciclos	39
6	CONCLUSÕES	46
	REFERÊNCIAS	47

1 INTRODUÇÃO

Esse trabalho foi motivado pelo meu interesse em juntar as áreas de programação e a área da matemática, além de ser requisito obrigatório para a conclusão do curso de Licenciatura em Matemática da UTFPR. O Professor João Luis Gonçalves do Departamento Acadêmico de Matemática (DAMAT) da UTFPR-CT e o graduando Leonardo Cruzato, desenvolveram um trabalho sobre Indexação Musical via Espectro de Grafos (CRUZATO, 2021) e após uma conversa com o professor, achamos que seria interessante ampliar a comparação entre grafos usando maiores recursos computacionais e outros aspectos matemáticos dos grafos.

Os grafos são utilizados em matemática para estudar as relações entre objetos de um conjunto. Sua aplicação à música implica em associar um grafo a cada música, em que os vértices são as notas e as arestas descrevem a sequência de notas da melodia. Podemos associar ao grafo diversas características da música como melodia, harmonia e ritmo através do ponderamento das arestas. O impacto destas características são refletidas, por exemplo, no espectro da matriz laplaciana do grafo.

Em computação os grafos são muito utilizados para resolver problemas de armazenamento e busca, e utilizados também em inteligência artificial. Por exemplo, muitas aplicações de grafos na computação usam grafos que tem forma de árvore, que são grafos acíclicos e conexos para a criação de arranjos dinâmicos não lineares de elementos. Este modelo de grafo permite que dados sejam agrupados com uma regra de ordenamento definida, admitindo inserções e remoções de elementos como mostrado em (KOCH; PARDAL; SANTOS, 2022)

O tratamento matemático de características sonoras tem sido tema de muitas pesquisas e tem sido aplicado em muitas áreas como reconhecimento de fala e sintetização de voz. É possível observar isto ao realizar uma pesquisa sobre "*speech recognition and synthesis using graph theory*" e obtendo 110.000 resultados. Em inteligência artificial atende aplicações como comandos de voz e a capacidade de gerar uma resposta falada, facilitando a interação do público com estas tecnologias. Nossos objetivos são diferentes dessas aplicações, mas os resultados estão relacionados.

A finalidade principal deste trabalho é ampliar as possibilidades de comparação, e consequente caracterização, de músicas por meio da aplicação da Teoria de Grafos. Para alcançar esse objetivo, serão adotados métodos já utilizados anteriormente para representar músicas por meio de estruturas de grafos, a exemplo das propostas por Alberto Pinto e Paolo Tagliolato (2008).

Para atingir tal meta, vamos realizar um estudo das características matemáticas dos grafos, tais como as propriedades das matrizes de adjacência e laplaciana, assim como seus respectivos espectros associados. Esses fundamentos matemáticos serão empregados para fundamentar a análise musical. Adicionalmente, a aplicação do conceito de isomorfismo de grafo e subgrafo será explorada para identificar padrões e similaridades entre as diferentes composições

musicais. Nesse contexto, distintas métricas serão discutidas para a análise musical, cada uma delas com o propósito de capturar aspectos específicos das músicas.

2 REVISÃO DE LITERATURA

Nossa revisão de literatura começou com o trabalho de Alberto Pinto e Paolo Tagliolato (2008), intitulado *A Generalized Graph-Spectral Approach to Melodic Modeling and Retrieval*, que aborda a teoria musical (Pitch Classes) e a teoria espectral de grafos. Em seguida estudamos o trabalho *A statistical generalization of music graph spectral mining*, também de Alberto Pinto (2009), que aborda a construção de boas representações para músicas que contenham características musicais específicas, características essas fundamentais para as noções de similaridade entre músicas. Neste trabalho é apresentado o uso do espectro de grafo na modelagem da sequência melódica através de uma representação da superfície rítmica de melodias.

No trabalho de 2007 dos autores (PINTO *et al.*, 2007) intitulado *INDEXING MUSIC COLLECTIONS THROUGH GRAPH SPECTRA*, eles propõem um método de indexação de músicas usando o espectro de grafos, onde os autores codificam o intervalo da estrutura melódica da música no grafo. Para cada grafo os autores associam uma estrutura algébrica que é uma matriz $n \times n$, onde n são os números de vértices no grafo. Para verificar a similaridade entre as duas músicas os autores se utilizam da matriz laplaciana, onde são calculados os autovalores desta matriz e a organizam por ordem de tamanho. Este espectro pode ser visto como um ponto em uma dimensão espacial e para ver a semelhança entre dois grafos, é calculado a distância Euclidiana entre estes dois pontos. Ao comparar esta nova abordagem de indexação musical em uma base de dados de músicas folclóricas holandesas, contra abordagens existentes, este novo método mostrou vantagens frente às outras abordagens.

Recentemente iniciamos o estudo do trabalho intitulado *Teoria dos grafos aplicada à análise musical* de (ATHAYDE, 2020), esse trabalho é muito completo e traz o estado da arte da análise musical usando grafos.

Durante o processo de revisão bibliográfica, foi identificado um artigo relevante, intitulado *A (sub)graph isomorphism algorithm for matching large graphs* de (CORDELLA *et al.*, 2004), que toma uma abordagem diferente para a comparação de grafos, utilizando o isomorfismo de grafos e subgrafos. Essa descoberta proporcionou uma nova perspectiva sobre o problema abordado neste trabalho, ampliando o horizonte de possibilidades de análise.

No referido artigo, foi apresentado um algoritmo que permite a identificação do isomorfismo entre grafos, levando em consideração não apenas a estrutura do grafo, mas também a estrutura específica do problema modelado no grafo. Essa flexibilidade é de grande relevância para este trabalho, uma vez que nos permite adaptar o algoritmo às necessidades específicas e características das músicas em análise.

Ao considerar a contribuição desse artigo, será possível explorar uma abordagem mais abrangente e refinada para a comparação de grafos, a fim de identificar similaridades entre as músicas representadas por eles. A utilização do algoritmo proposto, adaptado conforme apropriado, permitirá uma análise mais precisa e personalizada, considerando tanto a estrutura geral do grafo quanto os elementos específicos relacionados à música.

A inclusão desse artigo no trabalho acrescentará uma dimensão adicional à análise dos grafos musicais, ampliando as possibilidades de comparação e fornecendo uma base sólida para a identificação de padrões e relações entre as músicas estudadas.

3 TEORIA DE GRAFOS

Muitos problemas e eventos podem ser descritos com o auxílio de diagramas que consistem em um conjunto de pontos, que representam objetos, juntamente com as linhas que ligam estes pontos e que representam alguma relação entre os objetos.

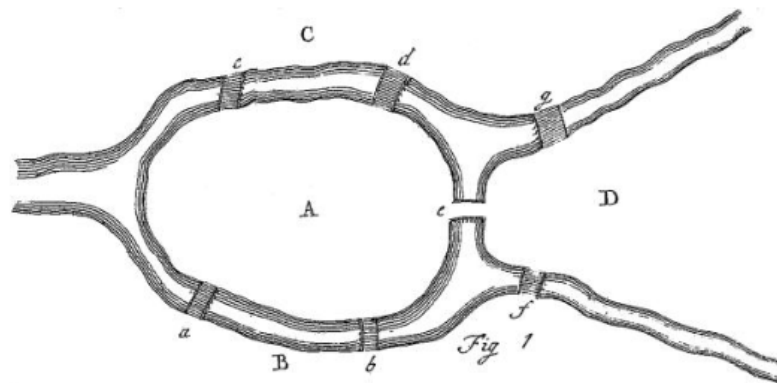
Por exemplo, os pontos podem ser os computadores na web e as linhas as conexões que estes computadores fazem, os pontos também podem representar as notas musicais e as linhas representam a sequência de execução das notas.

A abstração matemática desses diagramas definem os grafos, sendo os pontos chamados de vértices e as linhas de arestas. A teoria de grafos estuda as características e propriedades dos grafos. E portanto essa teoria ajuda na interpretação de fenômenos e resolução de problemas modelados por grafos.

Na ciência da computação, os grafos são usados para representar redes de comunicação, organização de dados, dispositivos computacionais, fluxo de computação, etc. A estrutura de sites pode ser representada por um grafo orientado, onde os vértices são as páginas da internet e as arestas são os links entre as páginas.

Em 1736 o matemático Leonhard Euler (1707-1783) escreveu o primeiro artigo relacionado a grafos. Ele começou discutindo um problema hoje conhecido como “O Problema das Pontes de Königsberg”, onde se questionava a possibilidade de partindo de uma região qualquer da cidade de Königsberg, construir um caminho onde cada ponte fosse atravessada uma única vez e que o fim do caminho fosse o ponto de partida.

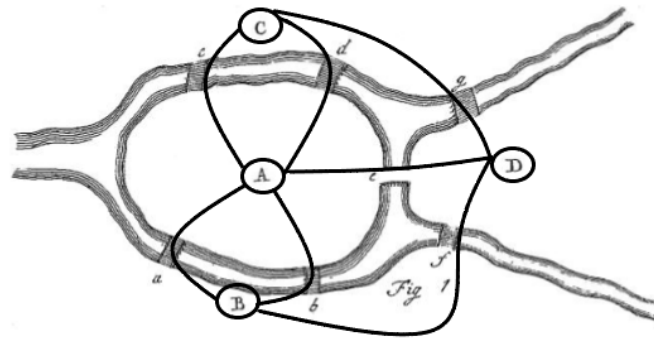
Figura 1 – Representação das pontes de Königsberg



Fonte: Frederico José Andries Lopes e Plínio Zornoff Táboas.

Euler propôs uma representação alternativa do problema, onde ele transformava as regiões em vértices e os caminhos que, através das pontes, ligam as regiões em arestas, criando um grafo, como mostra a Figura 2.

Figura 2 – Representação do grafo das pontes de Königsberg

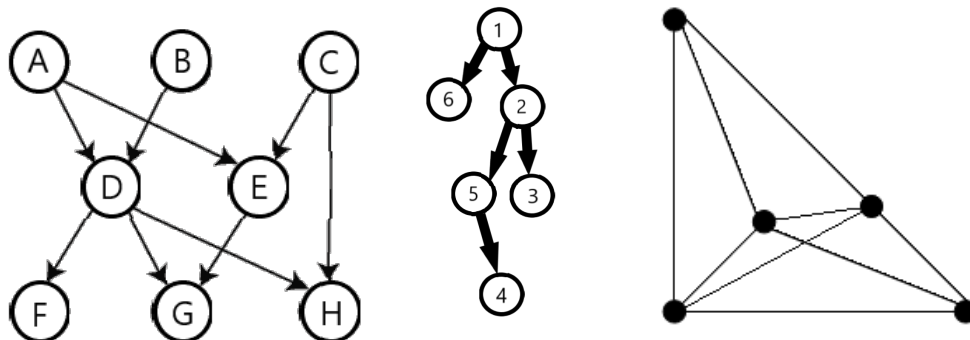


Fonte: Autoria Própria.

Rigorosamente um grafo G é formado por um par $(V(G), E(G))$ onde $V(G)$ é um conjunto finito, cujos elementos são os vértices, e $E(G)$ um conjunto de pares, ordenados ou não, de elementos, não necessariamente distintos, de $V(G)$.

Os grafos podem ser caracterizados de diferentes formas. Um grafo simples é aquele em que cada vértice tem o mesmo número de arestas incidindo sobre si. O grafo completo é um grafo simples em que cada vértice do grafo tem uma aresta conectando este vértice a cada um dos outros vértices. Grafo nulo é o grafo em que o conjunto de vértices é vazio. Em um grafo chamamos de caminho uma sequência de arestas tal que cada aresta que tem uma aresta antecessora compartilha com ela um vértice. Um grafo é conexo se para qualquer par $\{v, w\}$ de vértices, existe um caminho com extremos v e w . Um grafo acíclico é um tipo especial de grafo que não possui ciclos, ou seja, não há caminhos que retornem a um vértice inicial. Em outras palavras, não é possível percorrer um caminho sequencial de arestas no grafo e voltar ao ponto de partida. Grafo árvore é um grafo simples acíclico e conexo ou seja onde cada par de vértices é conectado por um único caminho.

Figura 3 – Grafo Acíclico, Arvore e Conexos



Fonte: Autoria Própria.

Um grafo pode ser orientado, e nesse caso é denominado dígrafo, se suas arestas forem orientadas, ou seja, a ordem no par de vértices que determina a aresta é importante. Se em um dígrafo temos uma aresta $\{v, w\}$, dizemos que a aresta tem sentido de v para w .

Além de orientação/sentido as arestas podem também ter associada a elas um peso, cujo significado varia conforme o contexto. Quando esses pesos são atribuídos as arestas dizemos que o grafo é ponderado.

Neste trabalho, iremos utilizar grafos orientados e ponderados como base para a representação e análise de músicas. Em particular, os vértices dos grafos serão as notas musicais e as arestas representarão a sequência de execução das notas.

Neste caso o conjunto $E(G)$ passa a ser formado uma tripla em que as duas primeiras entradas são elementos de $V(G)$ e a terceira é o peso. No nosso contexto, esses pesos podem representar características específicas das músicas, como ritmo, harmonia, melodia ou qualquer outro aspecto relevante para a análise musical. Dessa forma, os grafos ponderados permitem a incorporação de informações detalhadas sobre as músicas em suas arestas, enriquecendo a representação e possibilitando uma análise mais refinada.

A utilização de grafos orientados e ponderados oferece diversas vantagens para o nosso trabalho. Em relação aos grafos orientados, eles nos permitem capturar relações direcionais entre as diferentes partes de uma música, como a progressão de acordes, a sequência de notas ou o fluxo melódico. Já os grafos ponderados possibilitam a consideração de diferentes aspectos musicais, permitindo uma análise mais precisa e contextualizada.

Ao incorporar essas características em nossa abordagem, estaremos aptos a explorar e extrair informações relevantes sobre as músicas representadas por meio dos grafos. Essas informações de peso e direção serão fundamentais para a criação de métricas de similaridade, identificação de padrões musicais e outras análises que visem compreender e organizar as composições musicais de forma mais efetiva.

Exemplo 1. O grafo G , da música *Piano Melody* de Lemoncreme (MASTRIPOLITO, 2013) com vértices

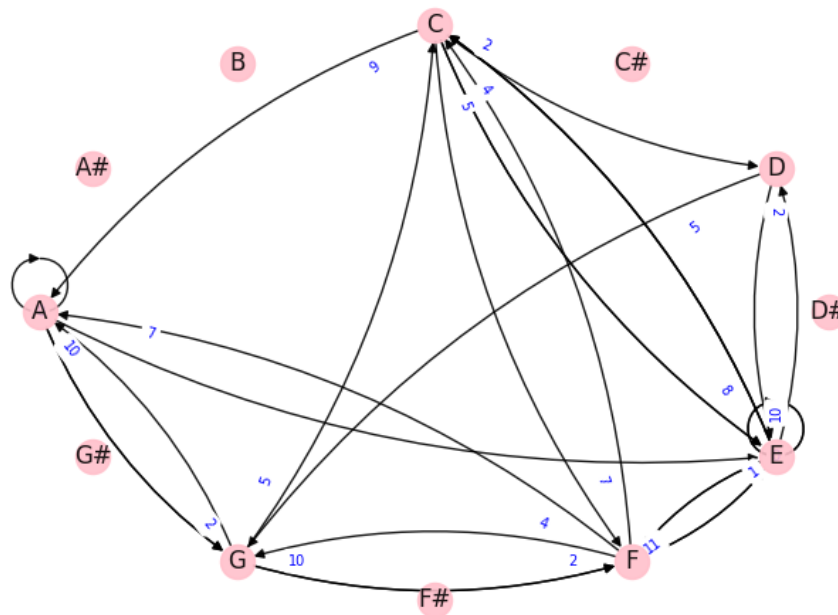
$$V(G) = \{C, C\#, D, D\#, E, F, F\#, G, G\#, A, A\#, B\}$$

e arestas

$$\begin{aligned} E(G) = \{ & \{C, E, 4\}, \{E, F, 1\}, \{F, A, 4\}, \{A, G, 10\}, \{G, F, 10\}, \{F, E, 11\}, \{E, E, 0\}, \{E, F, 1\}, \\ & \{F, G, 2\}, \{G, C, 5\}, \{C, D, 2\}, \{D, E, 2\}, \{E, C, 8\}, \{C, E, 4\}, \{E, D, 10\}, \{D, G, 5\}, \\ & \{G, A, 2\}, \{A, G, 10\}, \{G, F, 10\}, \{F, E, 11\}, \{E, C, 8\}, \{C, A, 9\}, \{A, A, 0\}, \{A, E, 7\}, \\ & \{E, E, 0\}, \{E, C, 8\}, \{C, F, 5\}, \{F, C, 7\}, \{C, F, 5\} \} \end{aligned}$$

é representado na Figura 4.

Figura 4 – Grafo da música Piano Melody de Lemoncreme



Fonte: Autoria Própria

A matriz de adjacência A é usada para indicar quando pares de vértices são adjacentes ou não em um grafo. Podemos definir a matriz de adjacência como:

$$A_{\{i,j\}} = \begin{cases} p_{i,j} & \text{se } (v_i, v_j) \in E(G) \\ 0 & \text{se } (v_i, v_j) \notin E(G) \end{cases}$$

em que $p_{i,j}$ é o peso da aresta (v_i, v_j) , caso essa exista.

O grau de um vértice v_i de um grafo não orientado é o número de arestas que estão conectadas a este vértice. Já em um grafo orientado o grau de um vértice v_i é o número de arestas que têm v_i como seu primeiro vértice, dizemos que a aresta parte de v_i . A matriz de graus, D de um grafo $G = (V, E)$ é definida como

$$D_{\{i,j\}} = \begin{cases} \text{grau}(v_i) & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$$

Da definição segue que a matriz de graus D é diagonal.

A matriz laplaciana L pode ser usada para encontrar propriedades importantes em grafo e ela é obtida subtraindo da matriz de graus a matriz de adjacência, isto é, $L = D - A$.

A seguir temos as três matrizes do grafo da Figura 4.

3.1 Espectro de Grafos

A teoria espectral de grafos é o estudo das propriedades do grafo em relação aos autovalores, autovetores e polinômio característico da matriz laplaciana associada ao grafo, isto é, do espectro da matriz laplaciana.

Um autovalor da matriz M é um escalar λ para o qual existe um vetor não-nulo x , denominado autovetor, que satisfaz $Mx = \lambda x$. Desta equação temos que o par autovalor e autovetor deve satisfazer $(M - \lambda I)x = 0$ com I sendo a matriz identidade, ou seja, como x é não-nulo a matriz $M - \lambda I$ deve ser singular. De forma equivalente os λ tais que $\det(M - \lambda I) = 0$ são autovalores de M . O conjunto de autovalores de M é denominado espectro de M e denotado por $S_p(M) = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$.

O determinante de $M - \lambda I$, denotado por $\det(M - \lambda I)$ ou por $|M - \lambda I|$ é chamado de polinômio característico de M . Note que $S_p(M)$ é o conjunto das raízes do polinômio característico de M .

Para um grafo G chamamos de espectro de G , e denotamos $S_p(G)$, ao espectro da matriz laplaciana de G . Por exemplo, para o grafo G da Figura 4 temos

$$S_p(G) = \{0, 0, 0, 0, 0, 0, 12.79, 20.03, 39.46, 59.37, 76.57, 103.74\}.$$

O espectro de uma matriz, e consequentemente o espectro de um grafo, servem como uma medida da transformação que a matriz em questão aplica nos vetores do seu espaço. Consequentemente, espera-se que transformações similares tenham espectros similares. Contudo, a recíproca dessa expectativa não é verdadeira, ou seja, existem matrizes que representam transformações distintas e que têm o mesmo espectro. Apesar disso, a proximidade entre espectros pode ser usada como uma medida de similaridade entre grafos, principalmente se incluirmos também uma análise sobre os autovetores associados a cada espectro.

No contexto deste trabalho, propomos utilizar a norma da diferença entre os espectros de dois grafos como uma medida da diferença entre eles.

A norma da diferença entre os espectros é uma medida que quantifica a distância ou diferença entre dois espectros, sendo neste caso estes vetores do $\mathbb{R}^{12}$. Existem várias formas de calcular a norma, sendo a norma Euclidiana uma das mais comumente utilizadas. A norma Euclidiana, também conhecida como norma L2, é calculada como a raiz quadrada da soma dos quadrados das diferenças entre os elementos correspondentes dos dois espectros.

Ao utilizar a norma da diferença entre os espectros dos grafos, poderemos obter uma medida numérica que reflete a diferença entre os padrões de conexões e características musicais dos grafos. Quanto maior a diferença entre os espectros, maior será a distância entre os grafos em termos de suas propriedades musicais.

Portanto, ao calcular a norma da diferença entre os espectros dos grafos, iremos quantificar e comparar a diferença entre os grafos de forma objetiva e numérica, fornecendo uma medida

de distância entre eles. Isso pode ser útil para identificar similaridades ou diferenças significativas entre as estruturas musicais representadas pelos grafos. Por exemplo, se $\mathbf{v}$ e $\mathbf{w}$ são dois vetores de autovalores de dois grafos então a medida da diferença entre eles é dada por:

$$\|\mathbf{v} - \mathbf{w}\| = \sqrt{(v_1 - w_1)^2 + (v_2 - w_2)^2 + \dots + (v_n - w_n)^2}.$$

4 SISTEMA DE INTERVALOS GENERALIZADO - GIS

O Sistema de Intervalos Generalizado (GIS, do inglês Generalized Interval System) do livro de (LEWIN, 2007) é uma abordagem teórica no contexto da teoria musical. Ele permite a modelagem de espaços musicais relacionados a tons e ritmos, a partir da aplicação de estruturas algébricas particulares.

No GIS, os intervalos não são apenas a distância entre duas notas musicais em termos de altura, mas uma relação mais abrangente entre objetos musicais. O intervalo entre dois objetos em um espaço musical é considerado um elemento de um grupo, que pode agir sobre o primeiro objeto para obter o segundo.

Esses grupos são usados para modelar características musicais específicas, permitindo uma análise mais completa dos relacionamentos entre os elementos musicais. Cada escolha de um GIS diferente permite modelar uma característica particular da música, como tons, ritmos e outras características musicais.

Portanto, o GIS oferece uma estrutura teórica para a modelagem de espaços musicais com base em intervalos generalizados. Isso vai além da simples relação entre frequências ou alturas e permite uma análise mais abrangente e sofisticada da música, levando em consideração diferentes aspectos musicais.

Definição 1. Segundo Lewin (2007), um Sistema Generalizado de Intervalos (GIS) é uma tripla ordenada $(S, IVLS, int)$, onde: S é o espaço do GIS, ou conjunto de elementos, $IVLS$ é o grupo de intervalos para o GIS, um grupo matemático e int é a função que mapeia $S \times S$ no $IVLS$.

Todos obedecem as seguintes condições:

(A): Para todo r, s , e t , em S , $int(r,s)int(s,t) = int(r,t)$.

(B): Para todo s em S e todo i em $IVLS$, existe um único t em S que pertence ao intervalo i para s tal que $int(s,t) = i$.

Neste trabalho iremos utilizar um GIS definido no livro do (LEWIN, 2007), que tem sólida base teórica e é amplamente reconhecido e usado. Esse GIS é apresentado no Exemplo 2, a seguir.

Exemplo 2. Seja S o espaço musical formado pelas doze classes de notas, isto é,

$$S = \{C, C\#, D, D\#, E, F, F\#, G, G\#, A, A\#, B\}.$$

Se distribuirmos as notas de S sobre as marcas das horas de um relógio circular de 12 horas, de forma que no sentido horário as notas estão em ordem crescente de tons (veja a Figura 4), então $int(s, t)$ é definido como o número de horas que percorremos ao prosseguir no sentido horário de s para t . Por exemplo, se s está em 8 horas e t está na 1 hora, $int(s, t) = 5$. Observe que o número $int(s, t)$ não depende de qual classe de tonalidade está posicionada às 12 horas. Em

todo o caso, $\text{int}(E, E) = 0$, $\text{int}(E, F) = 1$ e $\text{int}(F, E) = 11$. Embora tenhamos optado por descrever a função int usando a ideia de um relógio, poderíamos fazê-lo descrevendo o valor da função para quaisquer pares de elementos de S . Seja $IVLS$ o grupo dos inteiros sob módulo de 12. Considere o GIS $(S, IVLS, \text{int})$.

A escolha por esse GIS deve-se a ele estabelecer relações entre as classes de tons musicais e a perfeita adequação a representação de grafo que estamos utilizando.

Aplicaremos o GIS para atribuir pesos às arestas do grafo, representando as distâncias ou intervalos musicais entre as classes de tons. Essa informação enriquecerá nossa análise, permitindo-nos considerar as características tonais das músicas de forma mais precisa e abrangente.

5 COMPARAÇÃO ENTRE GRAFOS

Vamos abordar neste capítulo a comparação entre grafos, através de isomorfismos. Dois objetos são ditos isomorfos se são essencialmente iguais, isto é se possuem a mesma estrutura matemática, sob alguma ótica.

Os grafos G e H são ditos isomorfos se existe um isomorfismo de G em H . Um isomorfismo de G em H é uma bijeção f entre os conjuntos de vértices de G e H , $f : V(G) \rightarrow V(H)$, tal que quaisquer dois vértices u e v de G são adjacentes em G se e somente se $f(u)$ e $f(v)$ são adjacentes em H . Esse tipo de bijeção é comumente descrito como “bijeção com preservação de arestas”, pois preserva a estrutura das relações entre os vértices, que são justamente descritas pelas arestas.

Uma possibilidade de verificar a existência de um isomorfismo entre dois grafos é checar se eles passam pelo crivo de cinco regras, chamadas de regras de viabilidade sintáticas, definidas no algoritmo VF2, criado por (CORDELLA *et al.*, 2004) e amplamente utilizado.

No artigo de (CORDELLA *et al.*, 2004) também é apresentada a ideia de uma viabilidade semântica que deve ser proposta e implementada conforme as características do problema que é abordado através do uso de grafos. Para o nosso contexto, que é a comparação de músicas, vamos usar a teoria espectral de grafos para implementar uma regra de viabilidade semântica. Outra alternativa que propomos para verificar a viabilidade semântica em conjunto com o espectro dos grafos associados as músicas é usar parcialmente o algoritmo de distância de Dijkstra's, que serve para calcular a menor distância entre dois vértices quaisquer, para caracterizar ciclos dentro do grafo.

Para dois grafos G e H dados, propomos que sejam detectados os maiores subgrafos que sejam isomorfos. Idealmente, se os subgrafos forem os próprios grafos então os grafos são isomorfos e as músicas associadas são sintaticamente iguais. Quanto maior o tamanho dos subgrafos, relativamente ao grafo, maior é a similaridade entre os grafos e consequentemente entre as músicas.

Eventualmente pode ser interessante relaxar o isomorfismo permitindo um certo número de transformações entre os subgrafos em questão, obtendo desta forma subgrafos que são quase isomorfos. Por exemplo, se a condição $\{u, v\} \in E(G) \equiv \{f(u), f(v)\} \in E(H)$ não é satisfeita para algumas arestas. Desta forma temos uma informação que pode ser convertida em uma métrica de similaridade entre os grafos.

5.1 Análise do algoritmo VF2

No artigo (CORDELLA *et al.*, 2004) foi introduzido o algoritmo VF2, onde são estabelecidas uma série de regras de viabilidade que determinam se dois grafos parciais, $G_1(s)$ e $G_2(s)$, associados a um estado s , são isomorfos. Essas regras de viabilidade são classificadas em duas

categorias: viabilidade sintática e viabilidade semântica. A primeira checa exclusivamente da estrutura do grafo, enquanto a segunda trata das características inerentes a modelagem.

O algoritmo VF2, estabelece uma função de viabilidade $F(s, n, m)$, cujos valores possíveis são verdadeiro ou falso, em que s representa o estado atual do mapeamento de vértices, n é o vértice pertencente a G_1 e m é o vértice pertencente a G_2 , onde n e m são vértices candidatos ao mapeamento.

Como já mencionamos, a viabilidade tem uma parte sintática e outra semântica. Isso fica claro na definição de F como

$$F(s, n, m) = F_{syn}(s, n, m) \wedge F_{sem}(s, n, m),$$

ou seja, a viabilidade é a conjunção de duas funções de viabilidade, uma delas sintática, F_{syn} e outra semântica F_{sem} .

5.2 Viabilidade Sintática

Vamos analisar a estrutura de F_{syn} . Essencialmente a função de viabilidade sintática é a conjunção de 5 regras, a saber

$$F_{syn}(s, n, m) = R_{pred} \wedge R_{succ} \wedge R_{in} \wedge R_{out} \wedge R_{new}.$$

Essas regras sintáticas buscam garantir que o mapeamento entre os grafos parciais, no estado s , seja válido e respeite as propriedades estruturais dos grafos.

É importante observar que essas regras sintáticas são apenas uma parte do algoritmo VF2. As regras semânticas também desempenham um papel fundamental na verificação da consistência dos grafos parciais e na determinação do isomorfismo entre eles, estas regras serão discutidas posteriormente.

5.2.1 Regras de viabilidade sintáticas

Para descrever melhor as regras vamos estabelecer inicialmente as nomenclaturas que utilizaremos nas definições das regras:

- N_1 denota o conjunto de vértices no grafo G_1 ;
- N_2 denota o conjunto de vértices no grafo G_2 ;
- B_1 denota o conjunto de arestas no grafo G_1 ;
- B_2 denota o conjunto de arestas no grafo G_2 ;

- $P(s)$ é o conjunto de todos os possíveis pares candidatos a serem adicionados ao estado atual obtidos considerando os conjuntos de vértices diretamente conectados a $G_1(s)$ e $G_2(s)$;
- $M_1(s)$ é conjunto que contem o vértice $n \in G_1$ do mapeamento parcial dos vértices de G_1 a G_2 que foi definido até o estado s ;
- $M_2(s)$ é conjunto que contem o vértice $m \in G_2$ do mapeamento parcial dos vértices de G_1 a G_2 que foi definido até o estado s ;
- $M(s)$ é a solução de mapeamento parcial;
- $Pred(G_1, n)$ é o conjunto de vértices predecessores do vértice n no grafo G_1 ;
- $Succ(G_1, n)$ é o conjunto de vértices sucessores do vértice n no grafo G_1 ;
- T_1^{in} é o conjunto de vértices que não estão no mapeamento parcial e que são origem de arestas que terminam em $G_1(s)$;
- T_1^{out} é o conjunto de vértices que não estão no mapeamento parcial e que são destino de arestas que começam em $G_1(s)$;
- $T_1(s)$ é o conjunto de vértices que estão em T_1^{in} ou T_1^{out} ;
- $\tilde{N}_1(s)$ são vértices que não estão no mapeamento parcial nem nos conjuntos de vértices que são origem ou destino de arestas que terminam ou começam em $G_1(s)$. Ou seja, $\tilde{N}_1(s) = N_1 - M_1(s) - T_1(s)$.

Vamos apresentar na sequência as definições das regras seguidas de uma breve interpretação. Cada uma dessas regras retorna um valor lógico, verdadeiro ou falso.

A regra R_{pred} é definida por:

$$R_{pred}(s, n, m) \Leftrightarrow (\forall n' \in M_1(s) \cap Pred(G_1, n) \exists m' \in Pred(G_2, m) | (n', m') \in M(s)) \\ \wedge (\forall m' \in M_2(s) \cap Pred(G_2, m) \exists n' \in Pred(G_1, n) | (n', m') \in M(s)).$$

A regra R_{pred} checka se para um determinado estado s , para cada vértice n' na interseção de $M_1(s)$ e o conjunto de predecessores de n em G_1 , existe um vértice m' no conjunto de predecessores de m em G_2 tal que (n', m') pertencem a $M(s)$, e se para cada vértice m' na interseção de $M_2(s)$ e o conjunto de predecessores de m em G_2 , existe um vértice n' no conjunto de predecessores de n em G_1 tal que (n', m') pertencem a $M(s)$.

Em outras palavras, a proposição requer que para todo n' pertencente ao conjunto de soluções parciais (mapeamento parcial) que também esteja no conjunto de predecessores de n em G_1 , deve existir um vértice correspondente no conjunto de predecessores de m em G_2 tal

que seu par pertença a $M(s)$, e vice-versa, para todo m' pertencente ao conjunto de soluções parciais que também esteja no conjunto de predecessores de m em G_2 , deve existir um vértice correspondente no conjunto de predecessores de n em G_1 de modo que seu par pertença a $M(s)$. As condições impostas por R_{pred} são necessárias para que (n, m) pertença a $M(s)$.

A regra R_{succ} é definida por:

$$R_{succ}(s, n, m) \Leftrightarrow (\forall n' \in M_1(s) \cap Succ(G_1, n) \exists m' \in Succ(G_2, m) | (n', m') \in M(s)) \\ \wedge (\forall m' \in M_2(s) \cap Succ(G_2, m) \exists n' \in Succ(G_1, n) | (n', m') \in M(s)).$$

Para um dado estado s , R_{succ} checa se para cada vértice n' na interseção de $M_1(s)$ e o conjunto de sucessores de n em G_1 , existe um vértice m' no conjunto de sucessores de m em G_2 tal que (n', m') pertence a $M(s)$, e se para cada vértice m' na interseção de $M_2(s)$ e o conjunto de sucessores de m em G_2 , existe um vértice n' no conjunto de sucessores de n em G_1 tal que (n', m') pertence a $M(s)$. Tais condições são necessárias para (n, m) pertencer a $M(s)$.

Esta regra requer que para todo n' pertencente ao conjunto de soluções parciais que também esteja no conjunto de sucessores de n em G_1 , deve existir um vértice correspondente no conjunto de sucessores de m em G_2 tal que seu par pertença a $M(s)$, e vice-versa para todo m' pertencente ao conjunto de soluções parciais que também esteja no conjunto de sucessores de m em G_2 , deve existir um vértice correspondente no conjunto de sucessores de n em G_1 tal que seu par pertença a $M(s)$. Assim podemos concluir que o par (n, m) satisfaz mais uma condição necessária para pertencer a $M(s)$.

A regra R_{in} é definida por:

$$R_{in}(s, n, m) \Leftrightarrow (Card(Succ(G_1, n) \cap T_1^{in}(s)) \geq Card(Succ(G_2, m) \cap T_2^{in}(s))) \\ \wedge (Card(Pred(G_1, n) \cap T_1^{in}(s)) \geq Card(Pred(G_2, m) \cap T_2^{in}(s))).$$

R_{in} verifica se para um dado estado s , a cardinalidade da interseção entre o conjunto de sucessores do vértice n em G_1 que também estão no conjunto de vértices que ainda não estão na solução parcial e que terminam no vértice n , é maior ou igual à cardinalidade da interseção entre o conjunto de sucessores do vértice m em G_2 que também estão no conjunto de vértices que ainda não estão na solução parcial e que terminam no vértice m , e também a cardinalidade da interseção entre o conjunto de predecessores do vértice n em G_1 que também estão no conjunto de vértices que ainda não estão na solução parcial e que terminam no vértice n , é maior ou igual à cardinalidade da interseção entre o conjunto de predecessores do vértice m em G_2 que também estão no conjunto de vértices que ainda não estão na solução parcial e que terminam no vértice m .

Em outras palavras, a proposição exige que o número de sucessores de n em G_1 que também estão no conjunto de vértices que ainda não estão na solução parcial e que terminam no vértice n seja maior ou igual ao número de sucessores do vértice m em G_2 , e da mesma forma para o número de predecessores. Se ambas as condições forem válidas, então o par (n, m) pode pertencer a $M(s)$.

Analogamente, a regra R_{out} é definida por:

$$R_{out}(s, n, m) \Leftrightarrow (Card(Succ(G_1, n) \cap T_1^{out}(s)) \geq Card(Succ(G_2, m) \cap T_2^{out}(s))) \\ \wedge (Card(Pred(G_1, n) \cap T_1^{out}(s)) \geq Card(Pred(G_2, m) \cap T_2^{out}(s))).$$

Para um dado estado s , R_{out} checa se a cardinalidade da interseção entre o conjunto de sucessores do vértice n em G_1 que também estão no conjunto de vértices que ainda não estão na solução parcial e que iniciam no vértice n , é maior ou igual à cardinalidade da interseção entre o conjunto de sucessores do vértice m em G_2 que também estão no conjunto de vértices que ainda não estão na solução parcial e que iniciam no vértice m , e também se a cardinalidade da interseção entre o conjunto de predecessores do vértice n em G_1 que também estão no conjunto de vértices que ainda não estão na solução parcial e que iniciam no vértice n , é maior ou igual à cardinalidade da interseção entre o conjunto de predecessores do vértice m em G_2 que também estão no conjunto de vértices que ainda não estão na solução parcial e que iniciam no vértice m .

Ou seja, se os vértices sucessores do vértice n em G_1 interseção com o conjunto de vértices que são destino do vértice n ainda não inclusos no mapeamento parcial devem ter pelo menos tantos alvos quanto os vértices sucessores do vértice m em G_2 interseção com o conjunto vértices que são destino do vértice m ainda não inclusos no mapeamento parcial. E também os vértices predecessores do vértice n em G_1 interseção com o conjunto vértices que são destino do vértice n ainda não inclusos no mapeamento parcial devem ter pelo menos tantos alvos quanto os vértices predecessores do vértice m em G_2 interseção com o conjunto vértices que são destino do vértice m ainda não inclusos no mapeamento parcial. Sendo isto verdade então podemos concluir que o par (n, m) pode pertencer a $M(s)$.

A última regra de viabilidade sintática, R_{new} pode ser definida por:

$$R_{new}(s, n, m) \Leftrightarrow Card(\tilde{N}_1(s) \cap Pred(G_1, n)) \geq Card(\tilde{N}_2(s) \cap Pred(G_2, m)) \\ \wedge Card(\tilde{N}_1(s) \cap Succ(G_1, n)) \geq Card(\tilde{N}_2(s) \cap Succ(G_2, m)).$$

Para um dado estado s , R_{new} verifica se a cardinalidade da interseção $\tilde{N}_1(s)$ com os predecessores de n em G_1 é maior ou igual ao número de vértices na interseção de $\tilde{N}_2(s)$ com os predecessores de m em G_2 e a cardinalidade da interseção de $\tilde{N}_1(s)$ com os sucessores de n em G_1 é maior ou igual ao número de vértices na interseção de $\tilde{N}_2(s)$ com os sucessores de m em G_2 .

Em outras palavras, $R_{new}(s, n, m)$ indica que adicionar o par (n, m) ao mapeamento parcial atual $M(s)$ resultaria em um mapeamento que é pelo menos tão promissor em termos de mapeamento de vértices adicionais em G_1 para vértices em G_2 como qualquer outro par de vértices ainda não mapeados.

5.3 Viabilidade Semântica

No âmbito da viabilidade semântica, faremos uso da análise do espectro dos grafos, com ênfase na diferença entre os autovalores das matrizes laplacianas dos respectivos grafos. Essa abordagem é detalhada no Capítulo 3, no qual se explora o espectro de grafos e também foi aplicada de forma extensiva no trabalho (CRUZATO, 2021). Por meio dessa análise, será possível avaliar a semelhança entre os espectros dos grafos associados às músicas em comparação.

Além da abordagem usando a teoria espectral, queremos propor a utilização dos ciclos dos grafos para comparação entre eles. Nossos grafos têm sempre 12 vértices que representam as 12 notas musicais e estão dispostos sobre um círculo em ordem crescente de tom no sentido horário. E as arestas desses grafos formam um caminho, e portanto sem bifurcações, mas eventualmente com ciclos que se completam quando uma nota tem sua execução repetida.

Utilizaremos a ponderação da classes de tons, que é definida e discutida no Capítulo 4. Essa ponderação permitirá considerar as características tonais e melódicas das músicas.

Ao incorporar essas propriedades na viabilidade semânticas, estaremos levando em consideração aspectos mais abstratos e simbólicos das músicas, indo além da parte puramente estrutural dos grafos. Veja que nas regras de viabilidade sintática os pesos das arestas não foram considerados.

A seguir apresentamos como computaremos os ciclos, através de uma modificação do algoritmo de Dijkstra, e como os usaremos na viabilidade semântica.

5.3.1 Algoritmo modificado de Dijkstra

O algoritmo de Dijkstra é um algoritmo clássico utilizado para encontrar o caminho mais curto em um grafo ponderado e direcionado. Ele recebe como entrada um grafo, um vértice de origem e encontra o caminho mais curto a partir desse vértice para todos os outros vértices do grafo.

O algoritmo de Dijkstra segue os seguintes passos:

- Inicializa todas as distâncias do vértice de origem aos demais vértices do grafo como infinito, e a distância do vértice de origem, que é definida como zero;
- Cria um conjunto vazio chamado “não visitados” para armazenar os vértices ainda não visitados;

- Enquanto o conjunto “não visitados” não estiver vazio, faça o seguinte:
 - Escolha o vértice com a menor distância do conjunto “não visitados” e o marque como visitado;
 - Para cada vértice adjacente ao vértice atual que ainda não foi visitado, faça o seguinte:
 - * Calcule a distância total até esse vértice somando a distância do vértice atual com o peso da aresta entre eles;
 - * Se essa distância total for menor que a distância atual do vértice adjacente, atualize a distância do vértice adjacente.

Ao final do algoritmo, todas as distâncias dos vértices serão atualizadas com os caminhos mais curtos a partir do vértice de origem.

O algoritmo de Dijkstra utiliza uma abordagem gulosa, selecionando sempre o vértice com a menor distância para ser visitado a cada iteração. Dessa forma, ele garante que a distância encontrada para cada vértice é a menor possível.

Propomos uma modificação do algoritmo de Dijkstra, na qual ao invés de buscar um caminho mais curto entre dois vértices de um grafo ponderado, busque-se o ciclo mais curto.

Iniciamos pela definição de ciclos fechados em um grafo como uma sequência de notas que começa e termina na mesma nota.

Consideramos a ponderação das arestas usando o GIS definido no Capítulo 4 no qual, por exemplo, $\text{int}(C, A) = 9$ e conseqüentemente esse será o peso de uma aresta (C, A) . Ou seja, o tamanho de uma aresta será a distância tonal entre os vértices, a menos de oitavas.

Ao invés de buscar o caminho mais curto entre dois vértices específicos, adaptamos o algoritmo de Dijkstra para buscar o ciclo mais curto que começa e termina em cada vértice. Isso é feito introduzindo uma condição adicional de parada no algoritmo. Essa condição adicional de parada é verificada se o vértice do ciclo em questão foi alcançado novamente na sequência de vértices. Essa verificação é realizada para garantir que o ciclo tenha sido percorrido completamente e que não haja mais arestas a serem exploradas.

Ao percorrer o ciclo no grafo, o algoritmo registra cada vértice visitado em uma sequência. Após completar o ciclo, ele verifica se o vértice inicial do ciclo foi alcançado novamente na sequência de vértices. Se isso ocorrer, significa que o ciclo foi percorrido completamente, e o algoritmo pode ser encerrado.

Entre duas notas (vértices) consecutivas temos um semitom. Ao todo temos 12 semitons se percorrermos as notas em ordem crescente e somarmos suas distâncias/semitons. Desta forma, dizemos que um caminho deu k voltas inteiras se seu comprimento, considerando a ponderação, é maior $12k$ e menor que $13k$.

Também utilizamos o algoritmo de Dijkstra com a finalidade de contabilizar quantas vezes voltas inteiras são percorridas até que se forme um ciclo fechado para cada nota.

Vejam que devido a ponderação das arestas a comparação entre os ciclos, por exemplo usando o desvio-padrão dos pesos das arestas que os formam, traz uma análise qualitativa das semelhanças entre músicas, apesar de ciclos aparentemente serem um aspecto estrutural dos grafos.

Para um grafo G e $v \in V(G)$ supomos que existem n ciclos que começam e terminam em v , os quais têm, respectivamente, $k_1, k_2, \dots, k_n$ arestas. Podemos denotar esses ciclos por:

$$\begin{aligned} ciclo_{1,v} &= \{(v, v_1), (v_1, v_2), \dots, (v_{k_1-1}, v)\} \\ ciclo_{2,v} &= \{(v, v_{k_1+1}), (v_{k_1+1}, v_{k_1+2}), \dots, (v_{k_1+k_2-1}, v)\} \\ &\vdots \\ ciclo_{n,v} &= \{(v, v_{k_1+\dots+k_{n-1}+1}), (v_{k_1+\dots+k_{n-1}+1}, v_{k_1+\dots+k_{n-1}+2}), \dots, (v_{k_1+\dots+k_n-1}, v)\} \end{aligned}$$

O peso de cada ciclo descrito acima é definido pela soma dos pesos de suas arestas, isto é,

$$\begin{aligned} peso_{1,v} &= int(v, v_1) + int(v_1, v_2) + \dots + int(v_{k_1-1}, v) \\ peso_{2,v} &= int(v, v_{k_1+1}) + int(v_{k_1+1}, v_{k_1+2}) + \dots + int(v_{k_1+k_2-1}, v) \\ &\vdots \\ peso_{n,v} &= int(v, v_{k_1+\dots+k_{n-1}+1}) + int(v_{k_1+\dots+k_{n-1}+1}, v_{k_1+\dots+k_{n-1}+2}) + \dots + int(v_{k_1+\dots+k_n-1}, v). \end{aligned}$$

A soma dos pesos de todos os ciclos associados a um vértices v é denotado por $peso_v$, isto é,

$$peso_v = \sum_{i=1}^n peso_{i,v}$$

sendo n o número de ciclos associados a v .

Definimos o peso dos ciclos da música M como:

$$peso(M) = \sum_{j=1}^{12} peso_{v_j}.$$

Note que o peso dos ciclos de uma música não é necessariamente igual a soma dos pesos das arestas do grafo associado a música em questão.

Observamos que o peso de cada ciclo, $peso_{i,v}$, é múltiplo de 12, o que é consequência direta do GIS que escolhemos para ponderar as arestas. Consequentemente, para qualquer arestas v , temos que $peso_v$ é também múltiplo de 12 e o mesmo vale para $peso(M)$.

Os ciclos 13 ciclos formados para as notas da música, têm seus pesos descritos a seguir:

$$\begin{aligned} peso_{1,C} &= int(C, E) + int(E, F) + int(F, A) + int(A, G) + int(G, F) \\ &+ int(F, E) + int(E, E) + int(E, F) + int(F, G) + int(G, C) = 48 \end{aligned}$$

$$peso_{2,C} = int(C, D) + int(D, E) + int(E, C) = 12$$

$$peso_{1,D} = int(D, E) + int(E, C) + int(C, E) + int(E, D) = 24$$

$$peso_{1,E} = int(E, F) + int(F, A) + int(A, G) + int(G, F) + int(F, E) = 36$$

$$peso_{2,E} = int(E, E) = 0$$

$$peso_{3,E} = int(E, F) + int(F, G) + int(G, C) + int(C, D) + int(D, E) = 12$$

$$peso_{4,E} = int(E, C) + int(C, E) = 12$$

$$peso_{1,F} = int(F, A) + int(A, G) + int(G, F) = 24$$

$$peso_{2,F} = int(F, E) + int(E, E) + int(E, F) = 12$$

$$\begin{aligned} peso_{3,F} &= int(F, G) + int(G, C) + int(C, D) + int(D, E) + int(E, C) \\ &+ int(C, E) + int(E, D) + int(D, G) + int(G, A) + int(A, G) \\ &+ int(G, F) = 60 \end{aligned}$$

$$peso_{1,G} = int(G, F) + int(F, E) + int(E, E) + int(E, F) + int(F, G) = 24$$

$$\begin{aligned} peso_{2,G} &= int(G, C) + int(C, D) + int(D, E) + int(E, C) + int(C, E) \\ &+ int(E, D) + int(D, G) = 36 \end{aligned}$$

$$peso_{3,G} = int(G, A) + int(A, G) = 12$$

$$\begin{aligned} peso_{1,A} &= int(A, G) + int(G, F) + int(F, E) + int(E, E) + int(E, F) \\ &+ int(F, G) + int(G, C) + int(C, D) + int(D, E) + int(E, C) \\ &+ int(C, E) + int(E, D) + int(D, G) + int(G, A) = 72. \end{aligned}$$

Os pesos dos ciclos associados a cada nota são:

$$peso_C = 48 + 12 = 60$$

$$peso_D = 24$$

$$peso_E = 36 + 0 + 12 + 12 = 60$$

$$peso_F = 24 + 12 + 60 = 96$$

$$peso_G = 24 + 36 + 12 = 72$$

$$peso_A = 72.$$

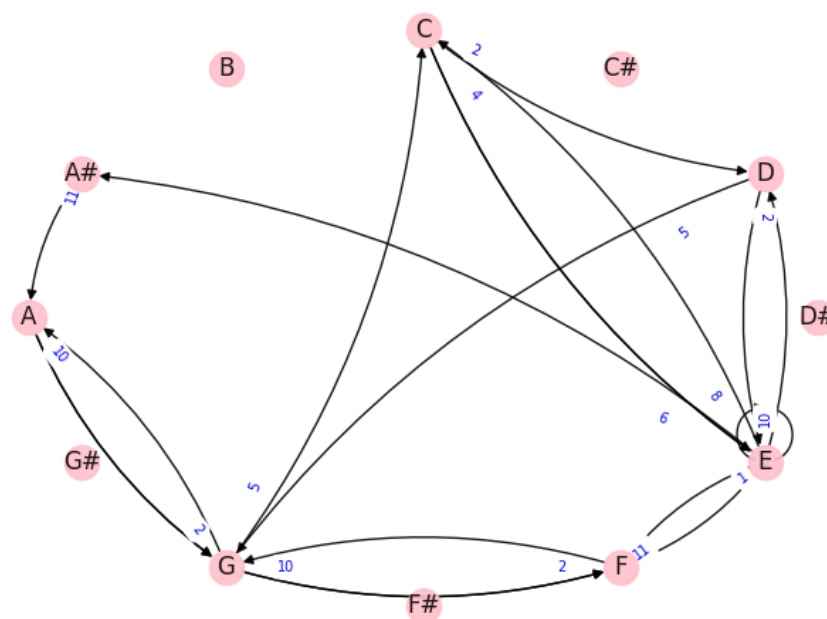
Consequentemente,

$$peso(M) = 60 + 24 + 60 + 96 + 72 + 72 = 384.$$

Podemos ver que nessas primeiras 20 notas da música temos um peso total dos ciclos de 384 e embora $384 = 32 * 12$ não podemos dizer que a música tem 32 voltas inteiras, pois os diferentes ciclos têm arestas comuns.

Exemplo 4. Partindo da mesma música do Exemplo 3 e substituindo somente a primeira nota F pela nota $A\#$ e aplicando o algoritmo novamente, temos:

Figura 6 – Grafo para alteração de F para A#



Fonte: Autoria Própria

$$\begin{aligned}
&int(C, E) = 4, \quad int(E, A\#) = 6, \quad int(A\#, A) = 11, \quad int(A, G) = 10, \\
&int(G, F) = 10, \quad int(F, E) = 11, \quad int(E, E) = 0, \quad int(E, F) = 1, \\
&int(F, G) = 2, \quad int(G, C) = 5, \quad int(C, D) = 2, \quad int(D, E) = 2, \\
&int(E, C) = 8, \quad int(C, E) = 4, \quad int(E, D) = 10, \quad int(D, G) = 5, \\
&int(G, A) = 2, \quad int(A, G) = 10, \quad int(G, F) = 10
\end{aligned}$$

A soma dos pesos das arestas é de $113 = 5 + 9 * 12$, o que nos dá um total de nove voltas inteiras, uma diferença de uma volta inteira se comparado ao Exemplo 3. Podemos observar que esta mudança se dá pelo fato de $(int(E, A\#) + int(A\#, A)) - (int(E, F) + int(F, A)) = 12$ o que é exatamente o peso de uma volta inteira.

$$\begin{aligned}
peso_{1,C} &= int(C, E) + int(E, A\#) + int(A\#, A) + int(A, G) + int(G, F) \\
&\quad + int(F, E) + int(E, E) + int(E, F) + int(F, G) + int(G, C) = 60 \\
peso_{2,C} &= int(C, D) + int(D, E) + int(E, C) = 12 \\
peso_{1,D} &= int(D, E) + int(E, C) + int(C, E) + int(E, D) = 24 \\
peso_{1,E} &= int(E, A\#) + int(A\#, A) + int(A, G) + int(G, F) + int(F, E) = 48 \\
peso_{2,E} &= int(E, E) = 0 \\
peso_{3,E} &= int(E, F) + int(F, G) + int(G, C) + int(C, D) + int(D, E) = 12 \\
peso_{4,E} &= int(E, C) + int(C, E) = 12 \\
peso_{1,F} &= int(F, E) + int(E, E) + int(E, F) = 12 \\
peso_{2,F} &= int(F, G) + int(G, C) + int(C, D) + int(D, E) + int(E, C) \\
&\quad + int(C, E) + int(E, D) + int(D, G) + int(G, A) + int(A, G) \\
&\quad + int(G, F) = 60 \\
peso_{1,G} &= int(G, F) + int(F, E) + int(E, E) + int(E, F) + int(F, G) = 24 \\
peso_{2,G} &= int(G, C) + int(C, D) + int(D, E) + int(E, C) + int(C, E) \\
&\quad + int(E, D) + int(D, G) = 36 \\
peso_{3,G} &= int(G, A) + int(A, G) = 12 \\
peso_{1,A} &= int(A, G) + int(G, F) + int(F, E) + int(E, E) + int(E, F) \\
&\quad + int(F, G) + int(G, C) + int(C, D) + int(D, E) + int(E, C) \\
&\quad + int(C, E) + int(E, D) + int(D, G) + int(G, A) = 72.
\end{aligned}$$

Ao observar a soma dos pesos dos ciclos para cada nota desta música com a alteração de uma única nota, temos que $\text{peso}(M) = 384$, não sofrendo alteração, era o esperado para uma alteração de uma única nota.

Aplicando o algoritmo VF2 verificamos que a estrutura sintática não se manteve. Com base na alteração realizada na primeira instância da nota F para A#, que ainda não era utilizada na música original, podemos tirar as seguintes conclusões:

Ao contabilizar as voltas da música, observamos que houve uma alteração de apenas uma volta, ou seja, a música continua tendo uma estrutura similar à original. Além disso, o peso dos ciclos se manteve igual, indicando uma preservação das características musicais relacionadas aos ciclos.

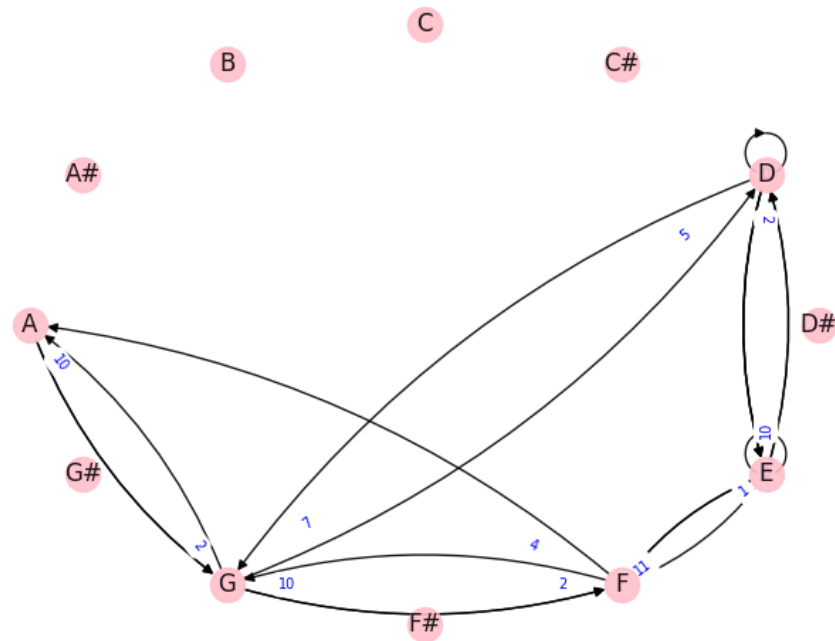
No entanto, ao aplicar a verificação sintática, ou seja, analisar a estrutura do grafo, identificamos que ela foi modificada devido à substituição da nota F por A#. Isso implica em uma alteração na organização e nas conexões dos vértices no grafo.

Apesar da diferença na estrutura do grafo, a análise dos ciclos permite verificar que as músicas ainda possuem similaridades. Isso indica que, mesmo com a mudança na organização geral dos vértices, as características musicais relacionadas aos ciclos permanecem semelhantes.

Essa conclusão ressalta a importância de considerar diferentes aspectos na análise musical. Enquanto a verificação sintática se concentra na estrutura global do grafo, a análise dos ciclos permite examinar características mais específicas e sua relação com a preservação das propriedades relacionadas a representação da música escolhida assim como do GIS definido para a música.

Exemplo 5. *Neste exemplo partimos da música do Exemplo 3 e substituímos todas as notas C por notas D, dessa forma temos:*

Figura 7 – Grafo para alteração de C para D



Fonte: Autoria Própria

$$\begin{aligned}
 &int(D, E) = 2, \quad int(E, F) = 1, \quad int(F, A) = 4, \quad int(A, G) = 10, \\
 &int(G, F) = 10, \quad int(F, E) = 11, \quad int(E, E) = 0, \quad int(E, F) = 1, \\
 &int(F, G) = 2, \quad int(G, D) = 7, \quad int(D, D) = 0, \quad int(D, E) = 2, \\
 &int(E, D) = 10, \quad int(D, E) = 2, \quad int(E, D) = 10, \quad int(D, G) = 5, \\
 &int(G, A) = 2, \quad int(A, G) = 10, \quad int(G, F) = 10.
 \end{aligned}$$

$$\begin{aligned} peso_{1,D} &= int(D, E) + int(E, F) + int(F, A) + int(A, G) + int(G, F) \\ &\quad + int(F, E) + int(E, E) + int(E, F) + int(F, G) + int(G, D) = 48 \end{aligned}$$

$$peso_{2,D} = int(D, D) = 0$$

$$peso_{3,D} = int(D, E) + int(E, D) = 12$$

$$peso_{4,D} = int(D, E) + int(E, D) = 12$$

$$peso_{1,E} = int(E, F) + int(F, A) + int(A, G) + int(G, F) + int(F, E) = 36$$

$$peso_{2,E} = int(E, E) = 0$$

$$peso_{3,E} = int(E, F) + int(F, G) + int(G, D) + int(D, D) + int(D, E) = 12$$

$$peso_{4,E} = int(E, D) + int(D, E) = 12$$

$$peso_{1,F} = int(F, A) + int(A, G) + int(G, F) = 24$$

$$peso_{2,F} = int(F, E) + int(E, E) + int(E, F) = 12$$

$$\begin{aligned} peso_{3,F} &= int(F, G) + int(G, D) + int(D, D) + int(D, E) + int(E, D) \\ &\quad + int(D, E) + int(E, D) + int(D, G) + int(G, A) + int(A, G) \\ &\quad + int(G, F) = 60 \end{aligned}$$

$$peso_{1,G} = int(G, F) + int(F, E) + int(E, E) + int(E, F) + int(F, G) = 24$$

$$\begin{aligned} peso_{2,G} &= int(G, D) + int(D, D) + int(D, E) + int(E, D) + int(D, E) \\ &\quad + int(E, D) + int(D, G) = 36 \end{aligned}$$

$$peso_{3,G} = int(G, A) + int(A, G) = 12$$

$$\begin{aligned} peso_{1,A} &= int(A, G) + int(G, F) + int(F, E) + int(E, E) + int(E, F) \\ &\quad + int(F, G) + int(G, D) + int(D, D) + int(D, E) + int(E, D) \\ &\quad + int(D, E) + int(E, D) + int(D, G) + int(G, A) = 72. \end{aligned}$$

Desta vez a soma dos pesos das arestas é $99 = 3 + 8 * 12$ nos dando ainda as mesmas 8 voltas inteiras do Exemplo 3. Ao observar a soma dos pesos dos ciclos temos que: $peso(M) = 372$, que é ligeiramente diferente do valor obtido nos dois exemplos anteriores.

Ao aplicarmos o algoritmo VF2 para verificar se há um isomorfismo entre o grafo deste exemplo e o grafo do Exemplo 3, observamos que não foi possível obter um isomorfismo entre eles.

Analogamente ao Exemplo 5, ao trocarmos C por outras notas na música do Exemplo 3, obtivemos os seguintes resultados:

- Substituindo C por $C\#$: A soma dos pesos das arestas foi $100 = 4 + 8 \times 12$, $\text{peso}(M) = 384$ e há um isomorfismo entre os grafos;
- Substituindo C por $D\#$: A soma dos pesos das arestas foi $110 = 2 + 9 \times 12$, $\text{peso}(M) = 444$ e há um isomorfismo entre os grafos;
- Substituindo C por E : A soma dos pesos das arestas foi $97 = 1 + 8 \times 12$, $\text{peso}(M) = 312$ e **não** há um isomorfismo entre os grafos;
- Substituindo C por F : A soma dos pesos das arestas foi $120 = 0 + 10 \times 12$, $\text{peso}(M) = 384$ e **não** há um isomorfismo entre os grafos;
- Substituindo C por $F\#$: A soma dos pesos das arestas foi $119 = 11 + 9 \times 12$, $\text{peso}(M) = 444$ e há um isomorfismo entre os grafos;
- Substituindo C por G : A soma dos pesos das arestas foi $106 = 10 + 8 \times 12$, $\text{peso}(M) = 348$ e **não** há um isomorfismo entre os grafos;
- Substituindo C por $G\#$: A soma dos pesos das arestas foi $105 = 9 + 8 \times 12$, $\text{peso}(M) = 384$ e há um isomorfismo entre os grafos;
- Substituindo C por A : A soma dos pesos das arestas foi $104 = 7 + 8 \times 12$, $\text{peso}(M) = 336$ e **não** há um isomorfismo entre os grafos;
- Substituindo C por $A\#$: A soma dos pesos das arestas foi $103 = 6 + 8 \times 12$, $\text{peso}(M) = 384$ e há um isomorfismo entre os grafos;
- Substituindo C por B : A soma dos pesos das arestas foi $102 = 5 + 8 \times 12$, $\text{peso}(M) = 384$ e há um isomorfismo entre os grafos.

Ao examinar os resultados, constatamos que há isomorfismo entre os grafos quando uma nota é substituída por uma nota que não era utilizada no grafo original. Ou seja, a estrutura do grafo permaneceu a mesma, mesmo que semanticamente haja uma mudança significativa.

No caso de substituição por notas que já estavam na música, as alterações no número de voltas inteiras foi pequeno. Isso sugere que as mudanças introduzidas não afetaram significativamente a sequência ou a direção das conexões musicais no grafo. Portanto, é possível afirmar que as modificações realizadas preservaram a coerência estrutural da música, refletindo uma continuidade nos padrões musicais e nas relações entre as notas.

Essa observação é importante, pois reforça a noção de que a substituição de notas por uma nova nota, mesmo quando algumas notas já são repetidas, não altera de forma drástica a

organização musical e as características do grafo. Isso nos permite realizar análises e comparações, levando em consideração a estrutura fundamental da música, independentemente das notas específicas utilizadas.

Essa constatação reforça a utilidade da abordagem baseada em grafos para a análise musical, permitindo-nos identificar padrões, estruturas e similaridades musicais de forma mais abrangente e independente das escolhas de notas específicas feitas pelos compositores. Contudo, a soma dos pesos das arestas e o pesos dos ciclos mostraram-se limitados na análise semântica. Embora tenhamos computado os ciclos, pouco exploramos suas propriedades quanto a forma, até aqui.

5.3.2 Comparação do Espectro do Grafo

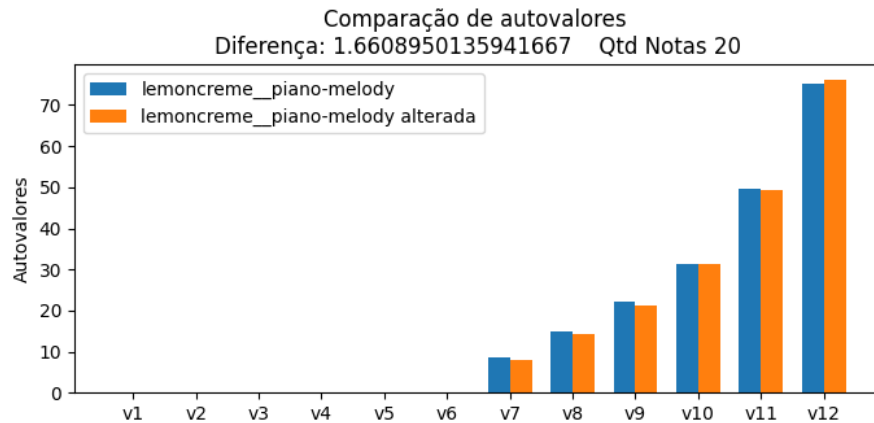
Para prosseguir com a análise de viabilidade semântica, planejamos conduzir experimentos que envolvam a comparação dos espectros dos grafos associados a músicas. Utilizaremos a música Piano Melody de Lemoncreme (MASTRIPOLITO, 2013), novamente, como base para comparação dos espectros.

A partir do grafo musical gerado anteriormente, que representa a estrutura da música em termos de conexões entre as notas, iremos extrair os espectros correspondentes. Esses espectros são os autovalores da matriz laplacianas dos grafos em questão.

Em seguida, realizaremos as mesmas modificações aplicadas anteriormente, substituindo todas as notas por uma nova nota na música. A intenção é verificar como essas alterações afetam os espectros dos grafos. Ao manter as mesmas 20 notas e aplicar as mesmas modificações, teremos um cenário controlado para a análise comparativa.

Ao comparar o espectro do grafo original com os espectros dos grafos modificados, poderemos identificar diferenças significativas e avaliar o impacto das modificações nas propriedades estruturais das músicas. Essa análise nos permitirá verificar se as alterações realizadas preservam a coerência semântica e estrutural das composições musicais.

Por meio desses experimentos, esperamos obter insights valiosos sobre a viabilidade semântica das modificações introduzidas nos grafos musicais. Essa abordagem nos auxiliará a compreender melhor a influência das mudanças nas notas na estrutura e nas características musicais representadas pelos grafos.

Figura 8 – Diferença nos espectros quando nota alterada de C para C#

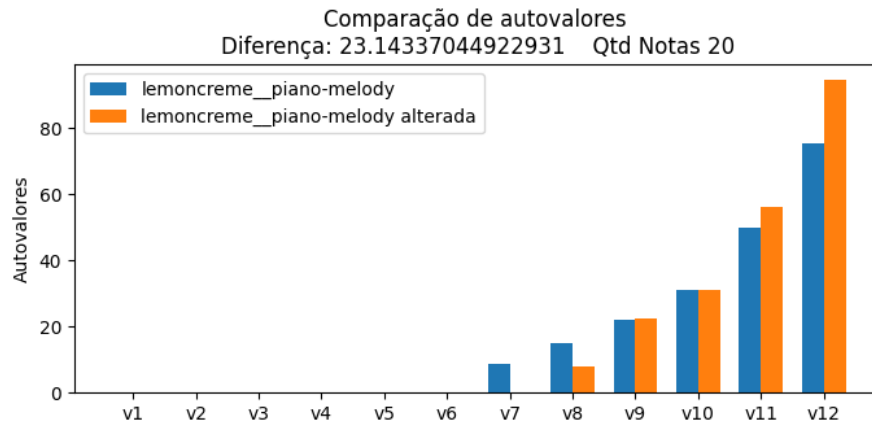
Fonte: Autoria Própria

Após realizar a substituição das notas C por notas C#, observamos que essa modificação não teve um impacto significativo no espectro do grafo musical, como podemos ver na Figura 8, que compara os doze autovalores associados aos dois grafos. Essa constatação é consistente com as observações anteriores, que indicaram a preservação da estrutura do grafo e a manutenção da quantidade de voltas inteiras.

A substituição de uma nota por outra próxima em termos de altura tonal, como C# no lugar de C, parece não ter afetado drasticamente as relações e as conexões musicais representadas pelo grafo. Essa constância no espectro do grafo indica uma continuidade nas características tonais e melódicas da música, apesar da troca de uma nota específica.

Esses resultados fortalecem a ideia de que as modificações introduzidas nas notas não comprometeram a coerência semântica e estrutural da música, refletindo a robustez da abordagem baseada em grafos para a análise musical.

Portanto, podemos afirmar que a alteração da nota C para C# preservou a estrutura do grafo, isto é, há isomorfia entre os grafos original e modificado. Além disso, do ponto de vista semântico, a observação de um impacto mínimo no espectro do grafo reforça a continuidade das características musicais representadas pelos grafos, apesar da alteração de uma nota específica.

Figura 9 – Diferença nos espectros quando nota alterada de C para G

Fonte: Autoria Própria

Ao realizar a substituição das notas C por notas G, observamos que essa alteração resultou em diferenças mais significativas nos espectros dos grafos musicais, conforme Figura 9. Essa constatação está de acordo com as observações anteriores, que indicaram a falta de isomorfismo entre o grafo da música original e o grafo da música modificada, bem como uma diferença na quantidade de voltas inteiras.

A substituição das notas C por G envolve uma alteração tonal mais drástica, resultando em mudanças nas relações e nas conexões musicais representadas pelo grafo. Lembrando que a nota G já era utilizada na música. Essas mudanças se refletem na diferença nos espectros dos grafos, indicando uma distinção nas características tonais e melódicas das composições musicais, isto é, uma diferença semântica.

A ausência de isomorfismo entre os grafos original e modificado demonstra a perda de equivalência estrutural entre as músicas após a modificação das notas. Essa falta de isomorfismo indica uma divergência nas relações entre os elementos musicais representados pelos vértices dos grafos, evidenciando a alteração significativa na estrutura musical.

Além disso, a diferença de 3 voltas inteiras entre os grafos original e modificado reforça a mudança na organização temporal e estrutural das composições. Essa diferença indica que a modificação das notas C para G resultou em uma variação no número de sequências musicais fechadas presentes nas composições.

Esses resultados apontam para a importância das notas específicas na construção e na estruturação das composições musicais. Alterações tonais mais significativas, como a substituição de notas que possuem relações musicais distintas, podem resultar em diferenças mais expressivas nos espectros dos grafos e na estrutura geral das músicas.

Portanto, ao modificar as notas C para G, verificamos diferenças mais significativas nos espectros dos grafos, a falta de isomorfismo entre os grafos original e modificado, bem como uma diferença na quantidade total de ciclos das músicas. Essas constatações ressaltam a influência das notas específicas na estrutura musical e destacam a sensibilidade da abordagem baseada em grafos para capturar essas mudanças.

5.3.3 Desvio-Padrão dos Ciclos

O desvio-padrão é uma medida estatística que quantifica a dispersão ou variabilidade dos valores em um conjunto de dados. Ele mede o quanto os valores individuais do conjunto estão afastados da média.

$$\mu = \frac{ciclo_m}{N}$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

σ representa o desvio-padrão.

μ é a média dos *GIS* das notas.

x_i são os valores do *GIS* de cada par de notas.

N é a quantidade de arestas.

Assim o desvio-padrão das notas pode ser um indicador útil para verificar similaridades entre duas músicas. No contexto musical, o desvio-padrão das notas pode refletir a variação ou a diversidade tonal presente em uma música. Quando ocorrem mudanças na sequência das notas, mesmo mantendo a mesma quantidade de ciclos, as alturas tonais se modificam, levando a um desvio-padrão diferente.

O desvio-padrão também pode ser útil para encontrar diferenças no formato de ciclos do grafo em casos onde outras formas de mensurar a viabilidades semânticas não são capazes de capturar essas diferenças. O desvio-padrão dos pesos das arestas de um ciclo podem, eventualmente com métricas auxiliares, indicar propriedades geométricas dos ciclos e portanto ajudar na comparação entre ciclos.

Se as músicas tiverem desvios padrão semelhantes, isso pode indicar uma relativa uniformidade tonal e uma estrutura melódica mais consistente entre elas. Por outro lado, desvios padrão significativamente diferentes podem sugerir uma maior variação tonal e uma diversidade melódica mais pronunciada.

Se os pesos das arestas de um ciclo têm um desvio-padrão alto, isso indica que há uma grande dispersão e variação tonal no ciclo, ao que pode corresponder diferenças significativas no formato do grafo, ainda que estruturalmente essas diferenças sem pouco perceptíveis.

Por outro lado, o desvio-padrão é pequeno, isso indica que as alturas tonais estão próximas umas das outras e que há uma consistência na estrutura melódica. Nesse caso, é provável que o formato do ciclo seja o de um polígono regular se ele contiver apenas uma volta inteira.

É importante ressaltar que o desvio-padrão deve ser usado em conjunto com outras métricas e técnicas de análise musical para uma avaliação abrangente das características do grafo e das similaridades entre as músicas.

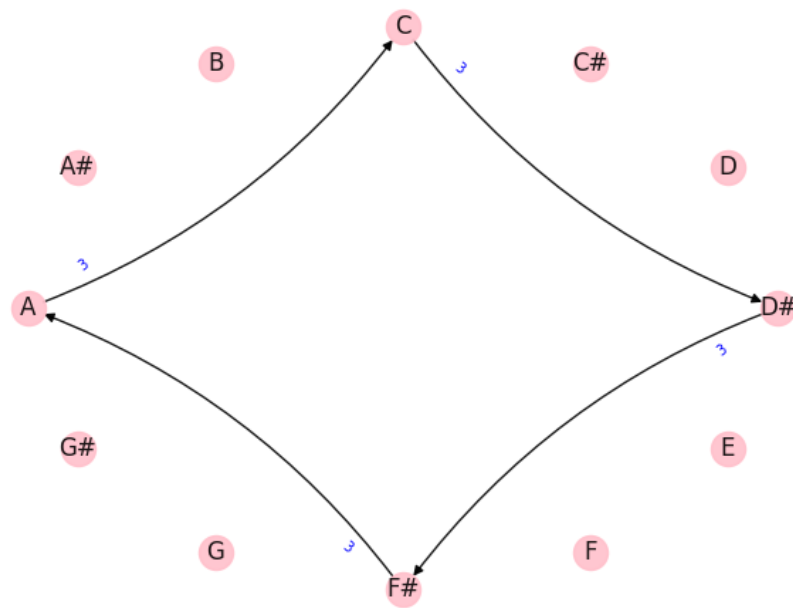
Segue o Exemplo 6 onde as regras aqui definidas para viabilidade semântica usando espectro do grafo e ciclos da música não capturam a diferença no formato do grafo.

Exemplo 6. Tomemos dois Grafos G e H , tais que

$$E(G) = \{(C, D\#, 3), (D\#, F\#, 3), (F\#, A, 3), (A, C, 3)\}$$

e é representado pela Figura 10.

Figura 10 – Grafo G



Fonte: Autoria Própria

G é um ciclo e o desvio-padrão desse ciclo é dado por:

$$\sigma = \sqrt{\frac{1}{4}((3 - 3)^2 + (3 - 3)^2 + (3 - 3)^2 + (3 - 3)^2)} = 0$$

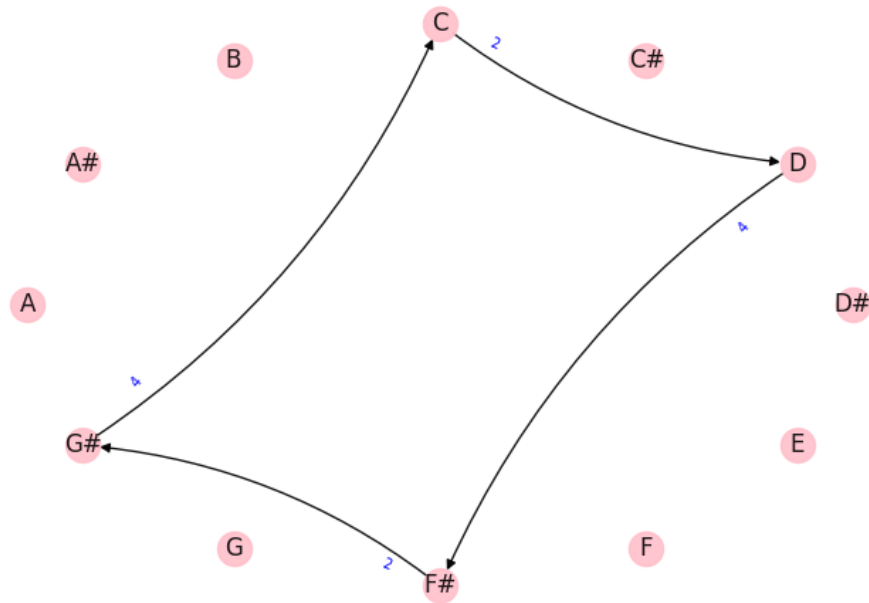
Um desvio-padrão 0 indica que as variações tonais estão todas próximas umas das outras, sem uma grande dispersão. Isso sugere uma constância na estrutura melódica, o que condiz com o ciclo ter o formato de um polígono regular.

Por outro lado seja

$$E(H) = \{(C, D, 2), (D, F\#, 4), (F\#, G\#, 2), (G\#, C, 4)\},$$

representado pela Figura 11.

Figura 11 – Grafo H



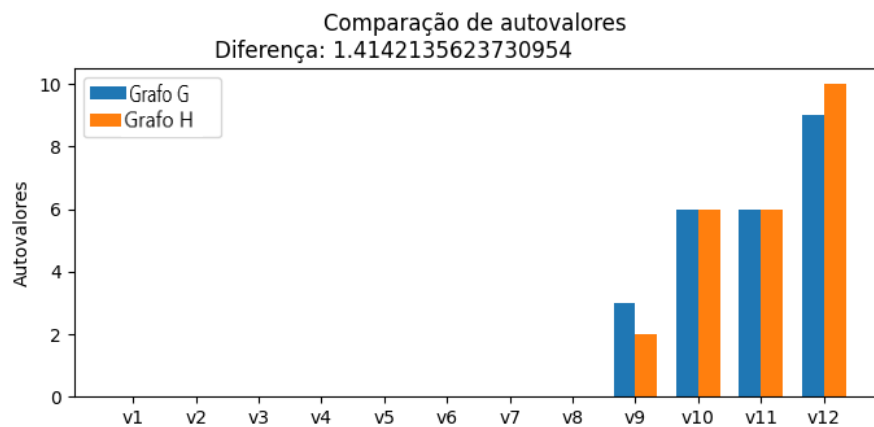
Fonte: Autoria Própria

H também é um ciclo e tem desvio-padrão dado por:

$$\sigma = \sqrt{\frac{1}{4}((2-3)^2 + (4-3)^2 + (2-3)^2 + (4-3)^2)} = 1.$$

Esse desvio-padrão, 1 indica que as variações tonais apresentam uma dispersão maior, com notas mais distantes e/ou mais próximas umas das outras. Essa diferença na dispersão sugere uma variação tonal maior e, conseqüentemente, uma estrutura mais irregular do ciclo.

Figura 12 – Espectro dos grafos G e H



Fonte: Autoria Própria

Podemos ver na comparação dos espectros dos dois grafos, apresentados na Figura 12, que uma diferença entre os dois grafos também foi capturada e portanto reforça a constatação obtida através dos desvios padrão dos grafos G e H .

Estruturalmente é bastante claro que os dois grafos são isomorfos sendo o mapeamento do isomorfismo o seguinte:

$$M(s) = \{(C, C), (D, D\#), (F\#, F\#), (A, G\#)\}.$$

Exemplo 7. Aqui serão analisado os desvios-padrão dos ciclos dos Exemplos 3,4 e 5, apresentados na Tabela 1.

	Ex.3	Ex.4	Ex. 5
$ciclo_{1,C}$	3.9192	4.0497	–
$ciclo_{2,C}$	2.8284	2.8284	–
$ciclo_{1,D}$	3.1623	3.1623	4.0694
$ciclo_{2,D}$	–	–	0.0
$ciclo_{3,D}$	–	–	4.0
$ciclo_{4,D}$	–	–	4.0
$ciclo_{1,E}$	3.9699	1.8547	3.9699
$ciclo_{2,E}$	0.0	0.0	0.0
$ciclo_{3,E}$	1.3565	1.3565	2.4166
$ciclo_{4,E}$	2.0	2.0	4.0
$ciclo_{1,F}$	2.8284	4.9666	2.8284
$ciclo_{2,F}$	4.9666	3.2853	4.9666
$ciclo_{3,F}$	3.2853	–	3.8462
$ciclo_{1,G}$	4.7074	4.7074	4.7074
$ciclo_{2,G}$	2.7479	2.7479	3.7198
$ciclo_{3,G}$	4.0	4.0	4.0
$ciclo_{1,A}$	3.758	3.758	4.1552
$ciclo_m$	3.7283	3.7623	4.0856

Tabela 1 – Desvios-padrão dos Exemplos 3,4 e 5.

O primeiro ciclo associado à nota F na música original desapareceu no Exemplo 4 após a modificação. Agora, o ciclo que era o segundo na sequência se tornou o primeiro, além disso temos uma variação no primeiro ciclo da nota C . Isso indica uma alteração na estrutura dos ciclos da música. No entanto, ao analisar o desvio-padrão da música como um todo, observamos uma diferença muito pequena, na ordem de 0.034. Isso sugere que, apesar das alterações nos ciclos específicos, a música ainda mantém uma similaridade geral.

No Exemplo 5, a modificação realizada na música foi mais significativa, substituindo todas as ocorrências da nota C pela nota D .

Apesar dessa alteração generalizada, vemos que ao calcular o desvio-padrão da música resultante, obtivemos uma diferença de 0.3573. Embora essa diferença seja maior em comparação ao exemplo anterior, ainda é consideravelmente pequena.

Isso sugere que, apesar da substituição das notas, a estrutura global da música foi preservada em certa medida. Além disso, ao analisar os pesos dos ciclos, observamos que eles também não sofreram uma grande alteração. Isso indica que, mesmo com a substituição das

notas, a estrutura cíclica da música se manteve em grande parte. Portanto, apesar da modificação significativa na música, tanto o desvio-padrão quanto os pesos dos ciclos indicam que ainda existem similaridades com a versão original. Esses resultados reforçam que mesmo diante de alterações substanciais na sequência de notas, vemos similaridades entre as músicas.

Exemplo 8. Comparação entre a música Piano Melody do Lemoncreme e a 5 sinfonia de Beethoven.

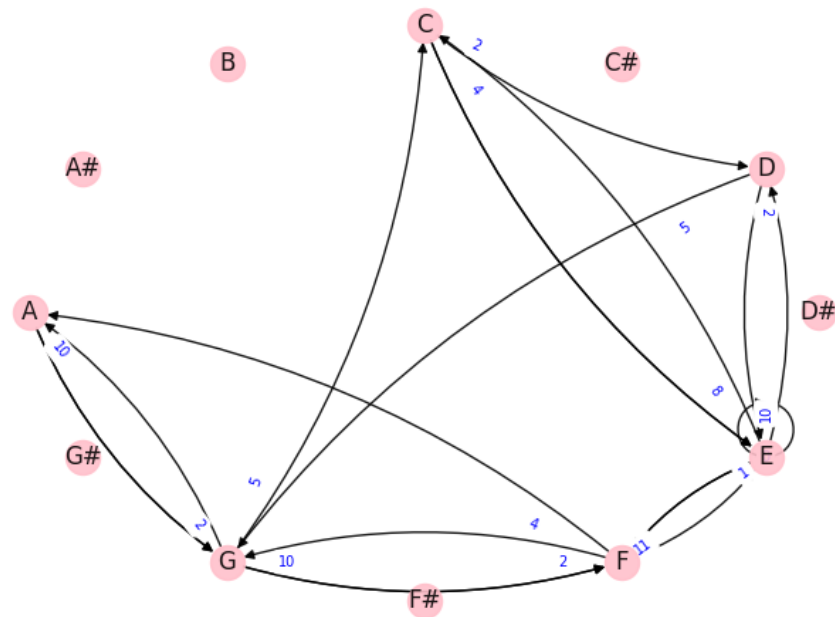


Figura 13 – Grafo com 20 Notas de Piano Melody

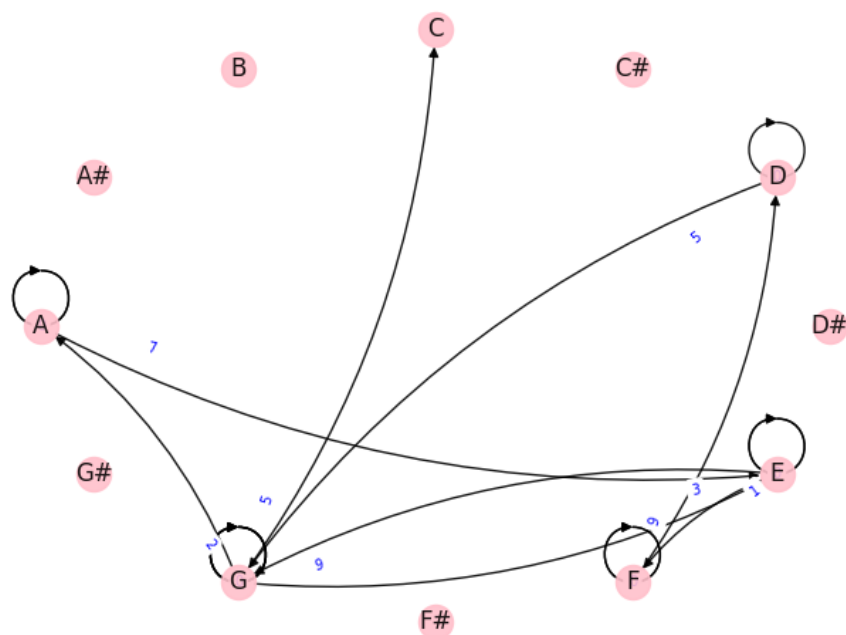


Figura 14 – Grafo com 20 Notas da 5 sinfonia

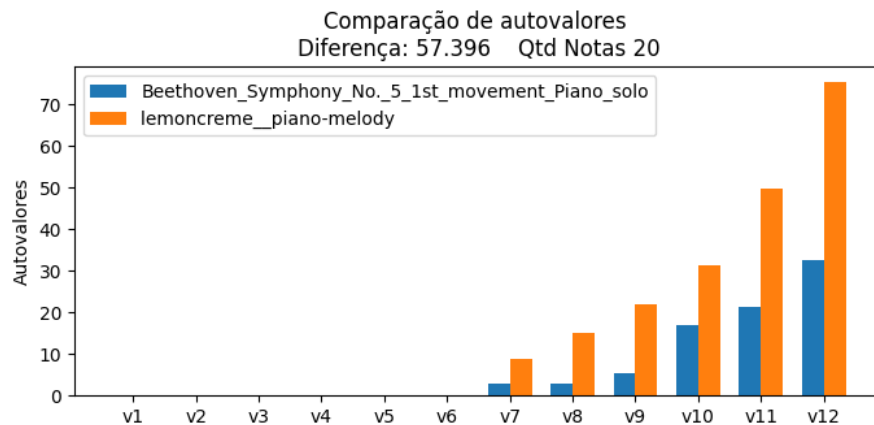


Figura 15 – Espectro dos grafos

Na comparação da música do Lemoncreme com a do Beethoven não foi possível encontrar um isomorfismo ao aplicar o algoritmo VF2. E na comparação do espectro de Beethoven e de Lemoncreme vemos uma diferença grande, mostrando que a estrutura musical é bastante diferente.

	Lemoncreme	Beethoven
<i>ciclo_{1,C}</i>	3.9192	–
<i>ciclo_{2,C}</i>	2.8284	–
<i>ciclo_{1,D}</i>	3.1623	0.0
<i>ciclo_{1,E}</i>	3.9699	3.0551
<i>ciclo_{2,E}</i>	0.0	0.0
<i>ciclo_{3,E}</i>	1.3565	0.0
<i>ciclo_{4,E}</i>	2.0	–
<i>ciclo_{1,F}</i>	2.8284	0.0
<i>ciclo_{2,F}</i>	4.9666	0.0
<i>ciclo_{3,F}</i>	3.2853	–
<i>ciclo_{1,G}</i>	4.7074	0.0
<i>ciclo_{2,G}</i>	2.7479	0.0
<i>ciclo_{3,G}</i>	4.0	3.8861
<i>ciclo_{4,G}</i>	–	0.0
<i>ciclo_{5,G}</i>	–	0.0
<i>ciclo_{6,G}</i>	–	2.4328
<i>ciclo_{1,A}</i>	3.758	0.0
<i>ciclo_{2,A}</i>	–	0.0
<i>ciclo_m</i>	3.7283	3.1332

Tabela 2 – Desvios-padrão das Músicas do Lemoncreme e do Beethoven.

Ao analisarmos os ciclos, os desvios padrões dos ciclos e da música como um todo, vemos que as duas músicas quase não compartilham ciclos, somente o *ciclo_{1,E}* e o *ciclo_{3,G}* que são compartilhados entre estas duas músicas e não tem o desvio padrão zerado, e ao comparar o desvio padrão destes ciclos vemos uma diferença significativa entre o *ciclo_{1,E}* e não tão significativa no *ciclo_{3,G}* mostrando uma similaridade somente neste ciclo. Analisando o desvio padrão da música como um todo vemos também uma grande diferença de 0.5951 em comparação com as diferenças vistas no exemplo anterior onde as alterações foram realizadas

em uma mesma música. Estes resultados podem nos mostrar que a estrutura das duas músicas é bem diferente.

Após as observações realizadas, podemos concluir que a utilização do desvio-padrão das notas nos permite identificar e quantificar as diferenças no formato do grafo musical. Mesmo quando outras viabilidades semânticas, como o isomorfismo, não são capazes de capturar essas alterações, o desvio-padrão acrescenta uma dimensão de análise que considera a dispersão das notas e dos ciclos.

Ao levar em conta o desvio-padrão, podemos analisar não apenas as notas individuais, mas também a variação e a distribuição dessas notas no contexto do grafo musical. Isso proporciona uma compreensão mais abrangente da estrutura e do formato da composição, permitindo-nos detectar alterações sutis ou significativas.

Além disso, ao considerar o desvio-padrão dos ciclos, podemos avaliar a variação no número e na configuração dos ciclos musicais, adicionando uma camada adicional de análise à compreensão da estrutura musical.

Dessa forma, a inclusão do desvio-padrão como uma métrica na análise de grafos musicais nos possibilita uma compreensão mais rica e abrangente das diferenças no formato e na estrutura musical. Isso contribui para a identificação de similaridades e diferenças entre composições.

6 CONCLUSÕES

Com o desenvolvimento deste trabalho, pudemos realizar uma análise estrutural das composições musicais utilizando a teoria de grafos como base. Exploramos diferentes aspectos das músicas, como a existência de isomorfismo entre grafos e subgrafos, um aspecto sintático, e comparação entre espectros e desvio padrão de ciclos, aspectos semânticos.

Ao longo do trabalho desenvolvemos ferramentas computacionais. Essas ferramentas permitem uma rápida extração das matrizes de adjacência e Laplaciana, a criação dos grafos correspondentes. Ainda essas ferramentas favorecem a aplicação de algoritmos já estabelecidos como o VF2 e o de Dijkstra, que nos permitiram, reciprocamente, verificar a existência de isomorfismo entre grafos e computar os ciclos presentes nas músicas.

Essas contribuições ampliam a análise de músicas através de grafos, permitindo uma compreensão mais profunda de suas características estruturais.

No geral, esse trabalho oferece uma abordagem inovadora para a análise musical, combinando a teoria de grafos, a teoria dos sistemas generalizados de intervalos e o processamento computacional. Essa integração de diferentes disciplinas e técnicas resulta em uma ferramenta poderosa para a análise e compreensão das composições musicais, contribuindo para o avanço no campo da música e das ciências computacionais aplicadas à música.

REFERÊNCIAS

- ATHAYDE, E. Teoria dos grafos aplicada à análise musical. p. 161, 2020.
- CORDELLA, L. *et al.* A (sub)graph isomorphism algorithm for matching large graphs. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 26, n. 10, p. 1367–1372, 2004.
- CRUZATO, L. D. C. Indexação musical via espectro de grafos. 2021.
- KOCH, I.; PARDAL, N.; SANTOS, V. F. dos. Edge deletion to tree-like graph classes. 10 2022.
- LEWIN, D. **Generalized Musical Intervals and Transformations (2007)**. 2. ed. Cidade: Oxford University Press, 2007.
- MASTRIPOLITO, B. **Piano Melody**. 2013. Disponível em: <https://benpm.github.io>.
- PINTO, A. A statistical generalization of music graph spectral mining. 01 2009.
- PINTO, A. *et al.* Indexing music collections through graph spectra. 2007. Disponível em: https://www.academia.edu/2690129/Indexing_music_collections_through_graph_spectra.
- PINTO, A.; TAGLIOLATO, P. A generalized graph-spectral approach to melodic modeling and retrieval. p. 8, 2008.