

Harmonização musical usando busca heurística

Roque Anderson Saldanha Teixeira¹, Luiz Chaimowicz¹

¹Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)
MG - Brasil

roqueandersonst@ufmg.br, chaimo@dcc.ufmg.br

Abstract. *The harmonization process requires musical knowledge and creativity to choose and arrange the chords producing pleasing results. This article is aimed at demonstrating how to harmonize a melody using simple techniques given search in Artificial Intelligence. This work shows: the concepts necessary for an understanding of the system, the technology used, the development of a prototype and finally presents the results obtained in practical applications.*

Resumo. *O processo de harmonização musical exige conhecimento e criatividade para escolher e organizar os acordes produzindo, assim, resultados agradáveis. Este artigo tem como objetivo de demonstrar como harmonizar uma melodia dada utilizando técnicas simples de busca em Inteligência Artificial. Este trabalho apresenta: os conceitos necessários para o entendimento do sistema, a tecnologia utilizada, o desenvolvimento de um protótipo e finalmente apresenta os resultados obtidos em aplicações práticas.*

1. Introdução

Criar um algoritmo para compor não é uma ideia nova. Pitágoras (cerca de 500 a.C) associava a ideia de música com operações lógicas e matemáticas. Em 1787, Mozart escreveu um algoritmo que utilizava dois dados de 6 lados para compor minuetos de 16 compassos. Ao jogar os dois dados é consultada uma tabela que possui 11 variações para cada um dos 16 compassos e 6 motivos mozartianos. Teoricamente $11^{16} \times 6^{16} = 1,3 \times 10^{29}$ composições inéditas de Mozart [Chuang 1995]. Hiller e Isaacson (1959) foram provavelmente os pioneiros a utilizar um modelo computacional para composição musical, utilizando números aleatórios e cadeias de Markov. Fazer um modelo computacional para composição musical é interessante por vários aspectos: Para os músicos que podem utilizar programas para um “esboço” de suas peças e posteriormente editar manualmente, ajustando de acordo com suas necessidades. Para os cientistas da computação também é interessante pois os requisitos deste domínio podem levar a novos modelos de computação e linguagens [Anders and Miranda 2011]. Há vários outros campos para utilização de tais algoritmos como:

- Música Adaptativa para Jogos Eletrônicos [Morais 2009], [Francisco et al. 2013]
- Sistemas Tutor para Ensino de Música [de Oliveira Caminha 2000]
- Teclados Arranjadores e softwares para composição Musical
- Músicas para sistema de Telefonia [Langston 1989]

Um dos problemas para a composição computacional é a harmonização, que consiste em encadear sons simultâneos (acordes). Segundo [Papadopoulos and Wiggins 1999], modelos matemáticos como cadeias de Markov são úteis devido à sua simplicidade e rapidez, porém necessitam de modelos para gerar a tabela de probabilidades e a escolha das notas se baseia somente na nota anterior. O objetivo deste trabalho é utilizar um sistema de busca baseado em heurística para harmonizar uma melodia. Foi utilizada a busca heurística por dois motivos. Em primeiro lugar por ser um sistema simples que não considera apenas cada acorde, mas o caminho completo. Com isso, é capaz de gerar bons resultados, sem complexidade. Em segundo lugar por não ter sido encontrado seu uso em nenhuma das bibliografias analisadas.

Além desta seção introdutória este trabalho possui mais cinco seções: Seção 2 que trata de conceitos básicos de harmonização, para melhor entendimento do problema. Seção 3 descreve o uso do sistema de busca heurística, como foi modelado o problema e descrição da heurística usada. Na quarta seção, é descrito como foi desenvolvido o protótipo. E, finalmente, as quinta e sexta seções apresentam, respectivamente, os resultados da aplicação do protótipo e a conclusão.

2. Harmonização Musical

A música é a arte de combinar os sons simultaneamente e sucessivamente, com ordem, equilíbrio e proporção dentro do tempo [Med 1996]. Seus principais elementos são:

- MELODIA – Sons em ordem sucessiva. (horizontal) (Figura: 1- linha vermelha)
- HARMONIA – Sons em ordem simultânea (vertical). (Figura: 1- em azul)
- RITMO – Proporção em que estão dispostos os sons da harmonia e da melodia. (Figura: 1- em verde)

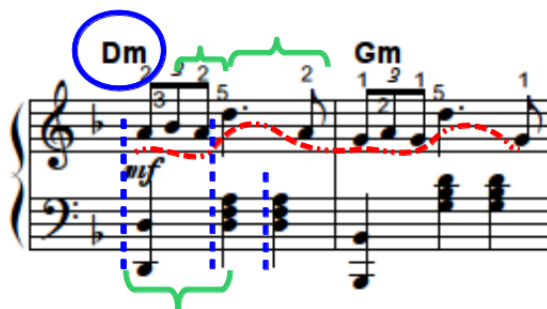


Figura 1. Elementos da Música - (Serenata, Franz Schubert).

As notas são agrupadas em conjuntos de tempos chamados de compasso. Dentro de um compasso, existem tempos fortes (têsis) e fracos (ársis). A nota pertencente ao tempo forte, que em geral é o primeiro tempo do compasso, será harmonizada pelo protótipo com um acorde que contenha esta nota. Aqui, será considerado o campo harmônico de dó maior natural (C, Dm, Em, F, G, Am, B^o) que é formado pelas notas da escala de dó maior (dó, ré, mi, fá, sol, lá, si) (Tabela:1). Apesar da simplicidade, os conceitos apresentados até aqui podem ser usados para aplicação em qualquer tonalidade. Vários trabalhos demonstram bons resultados tanto para criação de melodias [Langston 1989] como para harmonização [Papadopoulos and Wiggins 1999]. Alguns métodos têm sido utilizados como: Markovianos [Frankel-Goldwater 2006], Redes Neurais e Algoritmos Genéticos [Tuohy 2006].

Tabela 1. Formação de Acordes – Campo Harmônico de Dó Maior.

Nome do acorde	Cifra	Notas
Dó Maior	C	Dó, Mi, Sol
Ré Menor	Dm	Ré, Fá, Lá
Mi Menor	Em	Mi, Sol, Si
Fá Maior	F	Fá, Lá, Dó
Sol Maior	G	Sol, Si, Ré
Lá Menor	Am	Lá, Dó, Mi
Si Diminuto	B°	Si, Ré, Fá

3. Busca Heurística para Harmonização

Um problema de busca pode ser definido como encontrar uma sequência de ações que, partindo de um estado inicial, leve a um estado que satisfaça os objetivos do agente. Este problema, possui os seguintes componentes: estados, estado inicial, ação/modelo de transição, teste objetivo e custo. [Brewka 1996].

Em nosso problema específico temos os seguintes componentes:

Estados – São os acordes que harmonizam cada nota.

Estado Inicial – É dada pela tonalidade da música, que em geral, é definida pelo acorde inicial rege a escala, o campo harmônico e termina a frase musical. Diante de determinado tom, a música, inicia e termina com o acorde deste tom. Logo, conhecendo-se o tom da música, sabe-se também quais são o primeiro e o último acordes.

Ação / Modelo de transição – Sabe-se também pelo tom qual é o campo harmônico aplicável. Ao escolher uma nota da melodia, deve-se buscar quais são os acordes, dentro do campo harmônico, que contêm esta nota. Assim, se terá um grupo de acordes possíveis para cada nota. Portanto, a função sucessora consiste em determinar quais são os acordes do campo harmônico que possuem a nota a ser harmonizada.

Teste objetivo – O objetivo já é conhecido no início do problema, varrer todas as notas e terminar com o acorde do tom inicial. (desconsiderando modulações)

Custo/Heurística – O bom senso e apreciação musical são usados por músicos para a escolha de acordes, mas algumas regras de harmonia ajudam a saber qual caminho é mais agradável, ao passar de um acorde para outro.

A implementação do protótipo foi baseada no algoritmo *Greedy Search*. Sua função sucessora retorna todos os acordes que contêm a nota dada e pertencem ao campo harmônico escolhido. A escolha da sequência de acordes é baseada na função heurística mostrada na figura 5 discutida a seguir. O valor é calculado como *resistência* em se passar do acorde de origem aos acordes gerados pela função sucessora. Por fim é escolhida a sequência de acordes que produz o caminho de menor custo.

A heurística proposta faz uso de técnicas baseadas no conceito de *Espaço Harmônico* [Holland et al. 1990], que também é empregada em teoria da Inteligência Artificial da harmonia tonal [Lerdahl 1988] e teoria cognitiva [Balzano 1980]. O grupo C_{12} é representado pelo conjunto $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ (ou $\{C, C\#, D, D\#, E, F, F\#, G, G\#, A, A\#, B\}$ para o tom de C Maior) (figura:2) e pela tabela Cayley [Balzano 1980].

Apesar da representação e as demonstrações de Balzano serem motivadas por

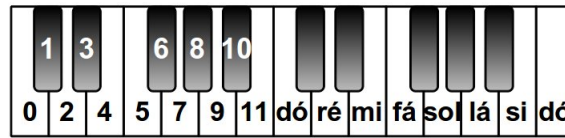


Figura 2. Representação de C_{12} no Teclado).

preocupação matemáticas, incluem importantes aspectos fundamentais da teoria musical, atributos melódicos e relações harmônicas presentes na música ocidental. Uma delas é a escala diatônica maior e menor ($N^7 = \{0,2,4,5,7,9,11\}$) [Balzano 1986]. Na tabela 2 existem algumas distribuições de intervalos formados pelos subgrupos de C_{12} .

Tabela 2. Subgrupos de C_{12} – [Pearce 2002].

Subgrupo	Conjunto de notas	n distintas	Nome da Escala
C_6	$\{0,2,4,6,8,10\}$	2	Tons Inteiros
C_4	$\{0,3,6,9\}$	3	Diminuto 7
C_3	$\{0,4,8\}$	4	Triade Aumentada
C_2	$\{0,6\}$	6	Tritono
C_7	$\{0,2,4,5,9,11\}$		Escala diatônica
C_5	$\{0,2,4,7,9\}$ $\{0,2,4,6,9\}$ $\{0,1,4,7,9\}$ $\{0,1,4,6,9\}$		Escla Pentatônica
C_4	$\{0,4,7,11\}$ $\{0,3,6,10\}$ $\{0,4,7,10\}$ $\{0,2,5,9\}$		Dominante 7, Maior 7 Aumentada 6, -
C_3	C_3		-
C_2	C_2		-
C_1	$\{0\}$		-

Basendo-se em representações com um único gerador que é isomórfico para o espaço de semitons, obtemos o “espaço de quinta” perfeita (N^7)¹² e seu inverso (N^5)¹². Pelo espaço de quintas, pode-se perceber o caminho com menor alteração entre uma tonalidade e outra, havendo apenas uma alteração para \sharp ou b em cada passo. Esta proximidade que proporciona um relação entre os tons (Tabela: 3).

Tal relação é confirmada na prática musical, onde “o acorde de V grau possui um tendência atrativa que pede resolução natural sobre o I grau.” (Priolli, 1983).

Outra representação isomórfica para C_{12} é gerada por subgrupos de C_3 e C_4 , considerando $(0,0)(0,1),(0,2),(0,3),(1,0),(1,1), \dots (2,2),(2,3)$ [Balzano 1980].

$C_4 \times C_3 \leftarrow \rightarrow C_{12} : (a,b) \leftarrow \rightarrow ([4a + 3b] \bmod 12)$. Esta representaç ao é mostrada na tabela 4.

No espaço de terças, as terças maiores ($4 \bmod 12$) são representadas pelo eixo horizontal e as terças menores ($3 \bmod 12$) pelo eixo vertical. A escala diatônica é representada pelo paralelograma (figura 3).

Partindo do tom de C em seu campo harmônico, pode-se definir custos para transição para seus vizinhos. Acordes com maior tendência a se encadear com outro possuem menor resistência; então, serão atribuídos valores menores. Já para acordes com maior resistência, serão atribuídos valores maiores (figura 4).

Tabela 3. Tonalidades –(modificado de [Pearce 2002]).

Grupo	n de #/b	notas #/b	Tom Maior	Relativo Menor
0			C maior	A menor
7	1 #	(F)	G maior	E menor
2	3 #	(F C)	D maior	B menor
9	3 #	(F C G)	A maior	F# menor
4	4 #	(F C G D)	E maior	C#menor
11	5 #	(F C G D A)	B maior	G#menor
	7 b	(B E A D G C F)	Cb maior	Ab menor
6	6 #	(F C G D A E)	F# maior	D# menor
	6 b	(B E A D G C)	Gb maior	Eb menor
1	7 #	(F C G D A E B)	C# maior	A# menor
	5 b	(F B E A D G)	Db maior	Bb menor
8	4 b	(B E A D)	Ab maior	F menor
3	3 b	(B E A)	Eb maior	C menor
10	2 b	(B E)	Bb maior	G menor
5	1 b	(B)	F maior	D menor

Tabela 4. Tonalidades – (modificado de [Pearce 2002]).

$C_4 \times C_3$	(0,0)	(0,1)	(0,2)	(0,3)	(1,0)	(1,1)	(1,2)	(1,3)	(2,0)	(2,1)	(2,2)	(2,3)
C_{12}	0	3	6	9	4	7	10	1	8	11	2	5

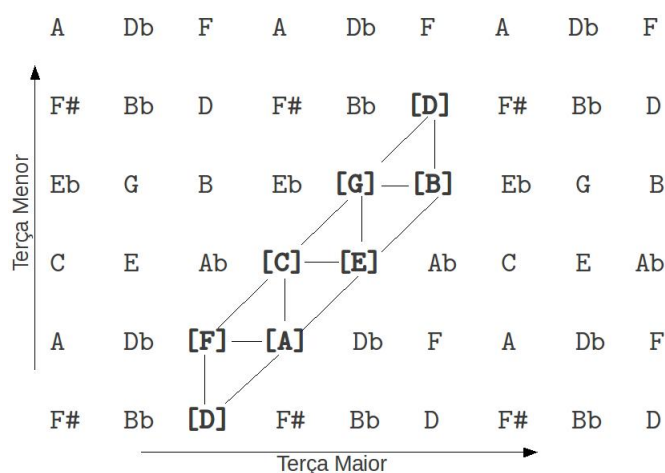


Figura 3. Representação do espaço de terças [Balzano 1980].

Não há custo em se manter o mesmo acorde (custo zero). Para o acorde diminuto, não foi dado custo zero ao se manter o acorde, devido à sua característica de tensão, que pede uma resolução rápida. Aqui, foi adotada uma distância múltipla por 2 entre os acordes, para haver um espaço para variação, que será discutido mais tarde. A tabela 5 mostra os custos utilizados no protópo para o campo Harmônico de C Maior.

O exemplo apresentado na Figura 5 ilustra o funcionamento da harmonização. Para a primeira nota, MI é atribuído o acorde de C (tom já definido neste problema). Para

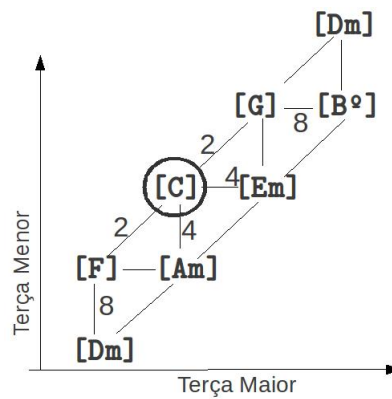


Figura 4. Espaço de terças para C maior.

	C	Dm	Em	F	G	Am	B°
C	0	8	4	2	2	4	8
Dm	8	0	8	4	2	2	4
Em	4	8	0	8	4	2	2
F	2	4	8	0	8	4	2
G	2	2	4	8	0	8	4
Am	4	2	2	4	8	0	8
B°	8	4	2	2	4	8	0

Figura 5. Tabela de custos.

a segunda nota são atribuídos os possíveis acorde e os custos baseados na tabela de custos (tabela: 5). Ao protótipo, foi atribuído um parâmetro x , para possibilitar um certo grau de improvisação ao programa. A variável s possui valor aleatório, limitado pela valor de x , onde: $1 \leq s \leq x$. Atribuindo-se $x = 1$, o caminho será determinado pelos acordes de menor custo. Com $x \geq 1$, o caminho sofre alterações que podem ser úteis para testar o efeito de outros acordes, ao harmonizar a mesma melodia (figura: 6).

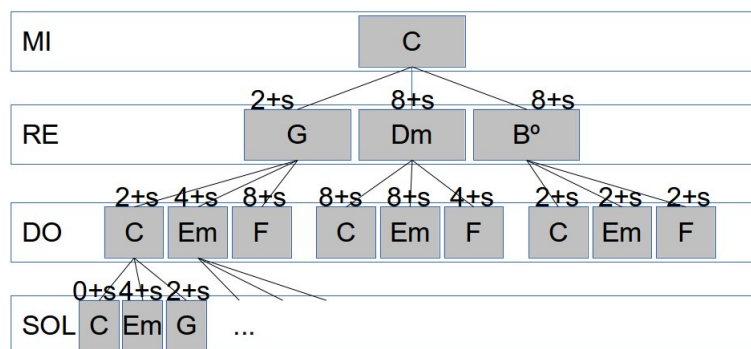


Figura 6. Árvore de decisões para o arquivo de entrada (figura: 7).

4. Desenvolvimento do Sistema

O protótipo foi desenvolvido em linguagem JAVA sem interface gráfica. O programa recebe como entrada um arquivo MIDI com melodia em um único canal (figura: 7). Sua saída gera 3 arquivos; destes, dois são arquivos de log, sendo um com todos os passos dados descritos detalhadamente e outro, com o resultado final da harmonização. Os logs mostram os caminhos e os custos calculados para cada acorde da árvore de decisões. O terceiro arquivo de saída contém a harmonização do arquivo MIDI de entrada, resultado da introdução de uma trilha com os acordes. Este arquivo possibilita ouvir o resultado da harmonização (figura: 8).



Figura 7. Exemplo de entrada.

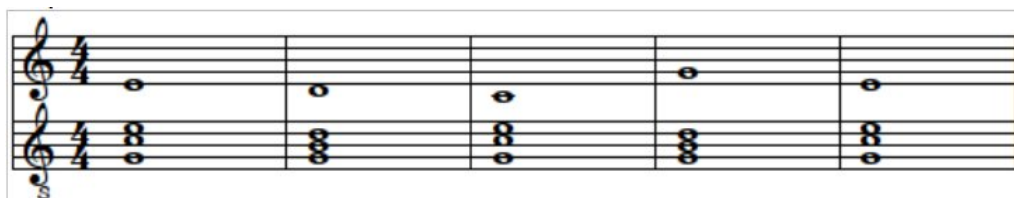


Figura 8. Exemplo de saída.

4.1. Tecnologia utilizada (Tecnologia MIDI)

No protótipo, optou-se por utilizar a tecnologia MIDI, devido à facilidade de manipulação dos dados, uma vez que ela permite manipular áudio na forma de tabela numérica. A palavra MIDI (*Musical Instrument Digital Interface*) pode representar:

- Um conector físico para controlar instrumentos em "tempo real",
- Formato de mensagem que é enviado por esta interface,
- Um padrão de arquivo para música digital (.mid)[Association and others 2007].

De uma forma geral, a tecnologia MIDI é um protocolo de dados para transmissão de informação de áudio. Estes dados não contêm o áudio, como é o caso dos arquivos WAV ou MP3, mas sim informações numéricas a respeito de como o som deve ser produzido. Esta informação é enviada para um instrumento ou interface do padrão MIDI, que a transforma em som. Atualmente, este padrão é muito utilizado também por programas de Tecnologia de Estúdio Virtual, VST (*Virtual Studio Technology*), que recebem ordem de dados MIDI e substituem por sons de instrumentos gravados em formato WAV em tempo real [Miletto et al. 2004]

Abaixo seguem algumas informações carregas pelos arquivos MIDI:

- *CMD (ShortMessage)* – O evento disparado no instrumento (tocar/parar nota)
- *Channel* – Canal que será usado para reprodução do som. MIDI permitem reprodução de vários instrumentos em canais independentes simultaneamente.
- *Key* – Número MIDI da nota que será reproduzida (de 0 a 127).
- *Velocity* – Intensidade de reprodução ou força (de 0 a 127).

A mensagem é enviada por um Evento que carrega, além da mensagem, o tempo do evento para controle da reprodução ou interrupção do som.

Os Eventos, por sua vez, são adicionados às trilhas (*Tracks*) como filas de reprodução.

As trilhas são adicionadas em sequências que podem conter várias trilhas, que serão executados em paralelo. Por fim, uma sequência é adicionado a um sequenciador, que é responsável por controlar a reprodução de todo conteúdo (*Play, Pause, Stop, tempo-trick*, etc..) A seguir (Figura 9), seguem algumas informações carregadas pelo arquivo MIDI:

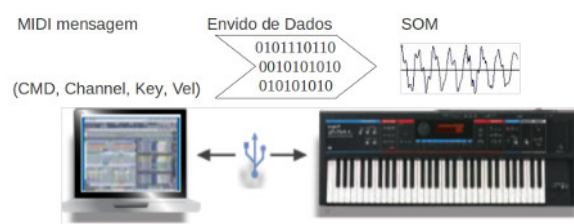


Figura 9. Interface MIDI (MIDI Manufacturers Association, 2009).

5. Resultados

Para melhor análise dos dados, serão comparadas as cifras gerada pelo algoritmo com cifras originais de pequenas melodias.

Nas figuras (10 a 12) estão alguns exemplos de resultados obtidos pelo algoritmo. A linha com *Org* apresenta a harmonização original. A linha com *Alg* apresenta a harmonização feita pelo algoritmo. Abaixo de cada teste, são comentadas as diferenças ocorridas.

Neste trabalho, serão mostrados em detalhes apenas três dos vários testes realizados. As melodias harmonizadas foram:

1. Noite Feliz - (Franz Grouber) (compasso 3, s= 1) (Figura:10)
2. Jingle Bells - (James Lord Pierpont) (compasso 4, s=1) (Figura: 11)
3. Asa Branca - (Luiz Gonzaga/Humberto Teixeira) (compasso 2 s=1) (Figura:12)

Como pode ser observado, os resultados obtidos foram muito bons, mas alguns acordes encontrados pelo algoritmo são diferentes dos originais. A seguir são realizadas algumas consideração acerca destas divergências.

Org:	C	C	C	C	G	G	C	C
Alg:	C	C	C	C	G	G	C	C

Org:	F	F	C	C	F	F	C
Alg:	F	F	C	C	F	F	C

Org:	C	G	G#dim	Am	C	C	G	C
Alg:	C	G	Bdim	C	C	C	G	C

Figura 10. Noite feliz - Franz Grouber.

Na melodia de *Noite feliz* (Figura 10), os acordes de G#dim e Bdim têm a mesma função harmônica, mas são de campos harmônicos diferentes. O acorde de Am é "relativo menor" de C maior.

Org:	C	C	C	C	F	C	C	G
Alg:	C	C	C	C	F	C	G	G



Org:	C	C	C	C	F	C	G	C
Alg:	C	C	C	C	F	C	G	C



Figura 11. Jingle Bells - James Lord Pierpont.

O problema encontrado no 7º compasso, que deveria ser harmonizado com o acorde de G, é devido ao retardo da melodia, que inicialmente está na nota MI, que é harmonizado por C.

Org:	C	C	C	C	F	C	C	G
Alg:	C	C	C	C	F	C	G	G



Org:	C	C	C	C	F	C	G	C
Alg:	C	C	C	C	F	C	G	C



Figura 12. Asa branca - Luiz Gonzaga/Humberto Teixeira.

Aqui, os compassos acéfalos (compassos sem nota no primeiro tempo) não foram harmonizados.

6. Conclusão

A harmonização de melodia possui um vasto campo possibilidades de mercado, devido ao grande avanço da tecnologia digital para dispositivos de áudio, jogos eletrônicos, instrumentos digitais e interatividade. Os algoritmos como *Greedy Search* não são mencionados para este fim nas biografias analisadas. Nelas utilizam CSPs, Algoritmos Genéticos entre outros. No entanto foi escolhido este tipo de busca devido sua simplicidade e para análise de seus resultados neste tipo de aplicação. Os sistemas de busca sem informação não puderam ser implementados pois o problema não busca um *estado final* definido e sim cada estado até o fim da melodia. Assim os sistemas de busca com informação se mostram úteis para tal fim. Os sistemas de custo uniforme e A* não foram implementados pois não há custo real entre os estados gerados, somente heurísticas encontradas em estudos de harmonia. Empregando-se as teorias de harmonia analisadas e o algoritmo *Greedy Search* obteve-se resultados satisfatórios para pequenas melodias e harmonização simples. Várias

melodias foram analisadas além das descritas neste artigo. Diferentemente de outros trabalhos, a metodologia proposta foi utilizada para harmonizar melodias já existentes, e não cria novas melodias. O sistema harmoniza a melodia de forma rápida e bem parecida com harmonizações feitas por humanos. Para trabalhos futuros, além do uso de outros algoritmos para comparar resultados, pretende-se analisar não somente o primeiro tempo do compasso, mas também fazer uma busca dentro do compasso para notas mais relevantes, evitando alguns problemas que foram detectados como: i) Não harmonização de compassos acéfalos; ii) Erro ao harmonizar compassos com retardo e iii) erro ao harmonizar compassos com síncope. Acredita-se que essas melhorias permitirão a harmonização de melodias mais complexas, mantendo-se os bons resultados.

Agradecimentos

Os autores gostariam de agradecer à CAPES, ao CNPq e à Fapemig pelo apoio ao desenvolvimento do trabalho.

Referências

- Anders, T. and Miranda, E. R. (2011). Constraint programming systems for modeling music theories and composition. *ACM Computing Surveys (CSUR)*, 43(4):30.
- Association and others, M. M. (2007). Tutorial on midi and music synthesis.
- Balzano, G. J. (1980). The group-theoretic description of 12-fold and microtonal pitch systems. *Computer Music Journal*, 4(4):66–84.
- Balzano, G. J. (1986). What are musical pitch and timbre? *Music Perception*, pages 297–314.
- Brewka, G. (1996). Artificial intelligence: a modern approach by stuart russell and peter norvig, prentice hall. series in artificial intelligence, englewood cliffs, nj. *The Knowledge Engineering Review*, 11(01):78–79.
- Chuang, J. (1995). Mozarts musikalisches würfelspiel. URL: <http://sunsite.univie.ac.at/Mozart/dice>.
- de Oliveira Caminha, A. (2000). Mhits - um sistema tutor inteligente em harmonia musical.
- Francisco, K. N., Ruste, J.-P. L., Sagum, R. A., Song, Y. S., and Sy, D. S. G. (2013). Incongruity theory applied in dynamic adaptive game artificial intelligence. *International Journal of Future Computer and Communication*, 2(5).
- Frankel-Goldwater, L. (2006). Computers composing music: An artistic utilization of hidden markov models for music composition. *Journal of Undergraduate Research*, 5(1-2):2007.
- Holland, S., Elsom-Cook, M., and Holland, S. (1990). Architecture of a knowledge-based tutor. *Guided discovery tutoring*. London: Paul Chapman Publishing Ltd.
- Langston, P. (1989). Six techniques for algorithmic music composition. In *15th International Computer Music Conference (ICMC)*, Columbus, Ohio, November, pages 2–5. Citeseer.
- Lerdahl, F. (1988). Tonal pitch space. *Music Perception*, pages 315–349.

- Med, B. (1996). Teoria da música. 4ª edição revista e ampliada. *Brasília-DF, Musimed*.
- Miletto, E. M., Costalonga, L., Flores, L., Fritsch, E., Pimenta, M., and Vicari, R. (2004). Introdução à computação musical. In *IV Congresso Brasileiro de Computação*.
- Morais, L. L. F. (2009). Música adaptativa para jogos eletrônicos. Monografia (Bacharel em Sistemas da Informação), UFMG (Universidade Federal de Minas Gerais), MG, Brazil.
- Papadopoulos, G. and Wiggins, G. (1999). Ai methods for algorithmic composition: A survey, a critical view and future prospects. In *AISB Symposium on Musical Creativity*, pages 110–117. Citeseer.
- Pearce, M. (2002). The group-theoretic description of musical pitch systems. URL: <http://webprojects.eecs.qmul.ac.uk/marcusp/>.
- Tuohy, D. R. (2006). *Creating tablature and arranging music for guitar with genetic algorithms and artificial neural networks*. PhD thesis, University of Georgia.