Aplicação do algoritmo de contraponto dissonante de Tenney na determinação de parciais em espectros de sons concretos

Raphael Sousa Santos (UFCG)
Liduino Pitombeira (UFRJ)

Resumo: Neste artigo, propomos um sistema para determinação de harmonias espectrais extraídas de sons concretos a partir da utilização do algoritmo de contraponto dissonante, desenvolvido por James Tenney. Com base nesse algoritmo, definimos um sistema composicional, descrevemos sua implementação como aplicativo computacional e o utilizamos no planejamento composicional de duas obras originais: o primeiro movimento de um quarteto de clarinetes e um quinteto de metais.

Palavras-chave: Música espectral. Contraponto dissonante. James Tenney.

Title: Applying Tenney's dissonant counterpoint algorithm in determining partial spectra in concrete sounds

Abstract: In this article we propose a system to determine spectral harmonies extracted from concrete sounds by using the dissonant counterpoint algorithm developed by James Tenney. Based on this algorithm, we define a compositional system, describe its implementation as a computer application, and then use it for the compositional planning of two original works: the first movement of a clarinet quartet and a brass quintet.

Keywords: Spectral Music. Dissonant Counterpoint. James Tenney.

SANTOS, Raphael; PITOMBEIRA, Liduino. Aplicação do algoritmo do contraponto dissonante de Tenney na determinação de parciais em espectros de sons concretos. *Opus*, Porto Alegre, v. 20, n. 1, p. 159-188, jun. 2014.

O presente artigo[1] descreve uma metodologia para determinação de harmonias espectrais controladas por um algoritmo desenvolvido por James Tenney (POLANSKY, 2010), o qual, por sua vez, baseou-se nos princípios do contraponto dissonante de Charles Seeger (1994). Segundo Polansky (2010: 21-22), esse algoritmo viria a se tornar o procedimento padrão do compositor James Tenney para a seleção aleatória de parâmetros e, de fato, ele também chegou a utilizá-lo em composições de caráter espectral, mais notadamente em sua série de obras *Spectrum*. Nessas obras, as alturas são extraídas de uma série harmônica ideal e controladas por esse algoritmo. Aqui, entretanto, utilizamos o princípio do algoritmo do contraponto dissonante predominantemente para selecionar alturas de espectros de sons concretos.

Faremos uma breve fundamentação sobre a técnica do contraponto dissonante e sobre o algoritmo de Tenney para, em seguida, propor um sistema composicional que, associado a três aplicativos desenvolvidos durante a pesquisa, nos permitiu planejar o primeiro movimento de um quarteto de clarinetes e um quinteto de metais.

Contraponto dissonante

A técnica do contraponto dissonante, teorizada pelo compositor Charles Seeger (1994) e utilizada por outros compositores da *American Atonal School* (POLANSKY, 2011: 64), tais como Henry Cowell, Carl Ruggles e Ruth Crawford Seeger, surge no contexto da primeira metade do século XX e é, em sua forma mais estrita, uma negação das regras do contraponto tonal (SPILKER, 2010: 4). Dentre as prescrições, estão a utilização da sétima maior, da segunda menor e da nona menor como intervalos essenciais (SPILKER, 2010: 26) e a não repetição de qualquer altura até que, no mínimo, outras seis tenham sido executadas (SEEGER, 1994: 174)[2]. Carl Ruggles, por exemplo, segundo Henry Cowell (1996: 42), escreve no mínimo sete ou oito notas antes de repetir alguma altura ou sua oitava, um fato que podemos constatar ao examinarmos (na figura 1) o início do primeiro movimento (Largo) de sua obra para piano intitulada *Evocations: Four Chants for Piano,* escrita entre 1937 e 1943 (e revisada em 1954). Observa-se que a primeira classe de alturas (Dó#), tocada pela mão esquerda, só reaparece depois de sete classes de alturas diferentes (Sol#, Ré, Mi, Fá, Dó, Fá# e Si) terem sido enunciadas.

[1] O presente artigo foi escrito como parte integrante dos resultados de um projeto de iniciação científica (PIBIC), financiado pelo CNPq, na Universidade Federal de Campina Grande (UFCG).

[2] Seeger (2010: 26) enfatiza que "uma separação maior é melhor, mas tudo depende do caráter e da condução das alturas envolvidas no processo. O ouvido deve ser o juiz".

Spilker observa que, no arquivo de Seeger da Biblioteca do Congresso Americano, encontram-se onze cadernos de contraponto de Cowell. Um deles contém as diretrizes de realização do contraponto dissonante e inclui quarenta e três exercícios utilizando três *cantus firmi*, seguindo a pedagogia das espécies Fuxianas (SPILKER, 2010: 23-24). Na figura 2, temos um dos exercícios de primeira espécie (nota contra nota) realizado por Cowell, no qual se percebe claramente a abundância de sétimas maiores, utilizadas em 2/3 do exercício.

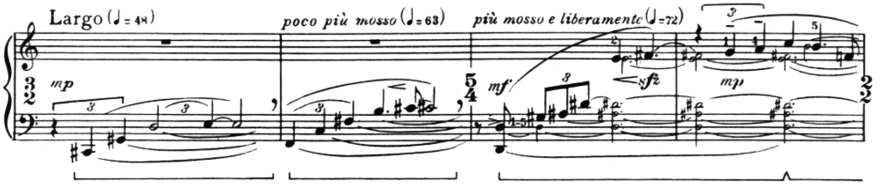


Fig. 1: Carl Ruggles, *Evocations*. Início do 1° movimento (Largo).

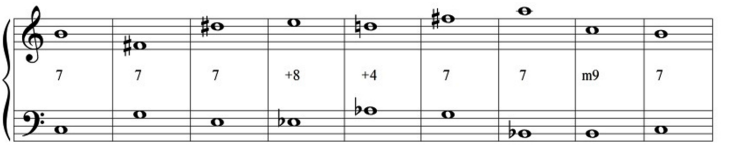


Fig. 2: Exemplo de contraponto dissonante de primeira espécie, realizado por Cowell.
Fonte: SPILKER, 2010.

Em 1977, Tenney publicou um artigo sobre a obra de Ruggles, no qual realizou um exame quantitativo minucioso da distribuição intervalar em diversas obras desse compositor. Por exemplo, no primeiro movimento de *Evocations*, cujo início é mostrado na figura 1, Tenney constata uma forte incidência de segundas menores (1 semitom), com 19,6%, e de trítonos (6 semitons), 18,5%. Os percentuais desses intervalos são mostrados na tabela 1 e um gráfico desse comportamento intervalar é mostrado na figura 3.

Intervalo	1	2	3	4	5	6	7	8	9	10	11	12	13	>13
%	19,6	12,0	7,6	3,3	13	18,5	10,9	3,3	1,1	1,1	4,3	0	2,2	3,3

Tab. 1: Incidência intervalar no primeiro movimento de *Evocations*.

Fonte: TENNEY, 1977.

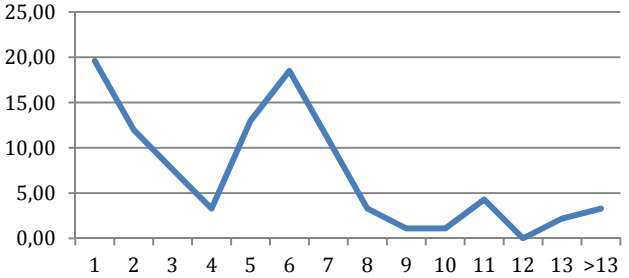


Fig. 3: Gráfico do percentual de incidência intervalar no primeiro movimento de *Evocations*.

Fonte: TENNEY, 1977.

A descrição estatística do comportamento intervalar na obra de Ruggles, realizada por Tenney, revela, como afirma Polansky (2011: 64), aspectos composicionais praticados, não só por Ruggles, mas também por toda a *American Atonal School*. Esses aspectos serão, mais tarde, na década de 1980, formalizados computacionalmente e "integrados em sua própria música". Assim, pode-se dizer que James Tenney desenvolveu o algoritmo a partir desses estudos. Passaremos a descrevê-lo na próxima seção.

Algoritmo do contraponto dissonante

Esse algoritmo, criado e empregado por James Tenney em suas composições, é descrito por Polansky (2011) com uma rigorosa formalização matemática. O nome dado ao algoritmo, pelo próprio Polansky, foi motivado pela conexão desse procedimento computacional com a prática do contraponto dissonante, discutida na seção anterior. É interessante esclarecer que, apesar do nome, esse algoritmo não se restringe ao contexto contrapontístico tradicional.

O procedimento do algoritmo de Tenney pode ser descrito, superficialmente, como uma sequência de sorteios em que a probabilidade de um elemento ser sorteado está associada à quantidade de rodadas passadas em que este elemento não ocorreu. Assim, por exemplo, um elemento que não é sorteado a cinco rodadas tem probabilidade maior de ocorrer do que um que não é sorteado a duas rodadas. O elemento sorteado na rodada imediatamente anterior, por sua vez, tem sua probabilidade de ocorrer reduzida a zero.

Polansky (2011: 64) afirma que esse algoritmo é um caso especial do que Charles Ames denomina de *feedback* estatístico. Segundo Ames (1987: 1), essa técnica "é provavelmente a mais presente em seus programas composicionais" e consiste basicamente "em manter a estatística descrevendo como cada opção foi utilizada no passado, polarizando as decisões para favorecer as opções que mais se distanciam de sua representação ideal". Essa representação, no caso do contraponto dissonante, é uma distribuição completamente uniforme, isto é, em que todos os elementos aparecem o mesmo número de vezes. No entanto, o *feedback* estatístico poderia ser utilizado, em outro contexto, para privilegiar um determinado conjunto de elementos. Isso tem uma repercussão direta nos aspectos estéticos de uma obra musical, no sentido de que se pode controlar a hierarquia de determinados elementos, evitando ou favorecendo certas centricidades.

Informalmente, o algoritmo do contraponto dissonante pode ser descrito a partir do estabelecimento de uma lista de contagem com *n* valores, um para cada classe de altura (POLANSKY, 2011: 65-66). Antes do algoritmo ser posto em funcionamento, ou seja, antes da primeira rodada de sorteio, os valores de contagem assumem o mesmo valor inicial (diferente de zero, comumente 1). O núcleo do algoritmo consiste de quatro diretivas: (1) selecione aleatoriamente um elemento de uma lista, utilizando os valores de contagem como probabilidades para escolher cada elemento; (2) atribua zero ao valor de contagem do elemento selecionado; (3) Incremente os valores de contagem de todos os outros elementos, de alguma forma determinística, por exemplo, adicionando uma constante; e (4) repita (volte ao passo 1).

Antes de exemplificarmos uma aplicação desse algoritmo é interessante observarmos a operação de um sorteio puramente aleatório, isto é, de um sorteio em que as probabilidades de ocorrências são sempre as mesmas a cada rodada. Para isso, suponhamos um conjunto hexacordal A, formado pelas seguintes classes de alturas:

{0,1,4,5,8,9}[3]. Construiremos uma linha melódica exclusivamente a partir de dez sorteios das classes de alturas do conjunto A[4]. Como se trata de um sorteio uniforme, ou seja, um sorteio em que todos os elementos têm a mesma chance de ocorrer, não teremos o mínimo controle sobre a probabilidade de ocorrerem repetições. A tabela 2 mostra uma das possibilidades resultantes do sorteio ao utilizarmos o seguinte algoritmo escrito em Python 3[5]:

```
import random
conjunto = [0,1,4,5,8,9]
for i in range (10):
        valor_sorteado = random.choice (conjunto)
        print (valor_sorteado)
```

Classe de alturas sorteada	0	0	1	9	0	4	0	0	1	4
Quantidade de semicolcheias	1	1	2	6	1	3	1	1	2	3

Tab. 2: Sorteio de classes de alturas do conjunto A= {0,1,4,5,8,9}.

Observando os resultados da tabela 2[6], verificamos que a classe de alturas 0 ocorreu em 50% dos casos, as classes de altura 1 e 4 ocorreram em 20% dos casos, cada uma, e a classe de alturas 9 ocorreu em apenas 10% dos casos. Não houve ocorrências das

[3] Neste trabalho, seguindo uma tradição já consolidada no campo da Teoria Pós-tonal, representamos as classes de alturas por números inteiros. Assim, Dó=0, Dó#=1 e assim por diante. Para evitar ambiguidades quando essas classes de alturas forem justapostas, por exemplo, no conjunto 1086, representaremos o Lá# por A e o Si por B. Essa nomenclatura se inspira na representação numérica de base hexadecimal utilizada na ciência da computação. Desta forma, o conjunto 1086 tem quatro classes de alturas (1,0,8,6) e não três (10,8,6) uma vez que o 10 teria que ser representado pela letra A.

[4] Com o intuito de concentrar nossa observação no fenômeno estatístico, as decisões relativas ao aspecto rítmico serão oriundas também do mesmo hexacorde, utilizando uma simples escala de durações. Assim, 0 = 1 semicolcheia, 1 = 2 semicolcheias, 4 = 3 semicolcheias, 5 = 4 semicolcheias, 8 = 5 semicolcheias e 9 = 6 semicolcheias.

[5] Os exemplos são fornecidos em Python 3, por ser uma linguagem de fácil acesso e gratuita. Nenhum dos exemplos tem alguma dependência externa, além das bibliotecas já inclusas no Python. Portanto, todos os exemplos aqui apresentados podem ser facilmente reproduzidos, bastando para isso a instalação dos pacotes disponíveis em http://www.python.org.

[6] Em virtude do algoritmo envolver um procedimento aleatório, outras execuções poderão gerar resultados diferentes.

classes de alturas 5 e 8. A repetição demasiada de uma classe de alturas é indesejável, se o objetivo é emular um resultado estético próximo ao encontrado na música de Ruggles, por exemplo. Esse fato se agrava mais ainda quando essa repetição é imediata, ou seja, quando as classes de alturas se repetem consecutivamente, como ocorre duas vezes, com a classe de alturas 0, na tabela 2, cujos resultados são mostrados em notação musical na figura 4.

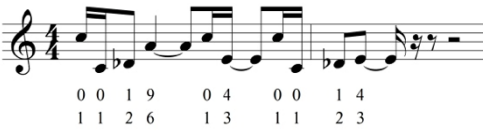


Fig. 4: Representação em notação musical dos resultados obtidos na tabela 1.

Para que possamos evoluir em direção ao próximo passo - aplicação do algoritmo do contraponto dissonante - é necessário que examinemos mais profundamente o conceito de contagem. Como dissemos anteriormente, a contagem nos informa há quantas rodadas um determinado elemento não ocorre. Dessa maneira, partindo dos dados obtidos na tabela 2, podemos mapear a contagem de cada elemento do conjunto A, durante as dez rodadas do sorteio. Por exemplo, na primeira rodada, a classe de alturas 0 foi sorteada. Isso significa que ela ocorreu há 0 rodadas (porque ocorreu imediatamente), enquanto as demais classes de altura do conjunto A estão sem ocorrer há 1 rodada. Na segunda rodada, a classe de alturas 0 ocorre novamente. Então, nessa rodada ela está há 0 rodadas sem ocorrer e as demais estão há duas rodadas sem ocorrer. Na terceira rodada a classe de alturas 1 é sorteada. A contagem para a classe de alturas 1 é 0, para a classe de alturas 0 é 1 (porque está há uma rodada sem ocorrer) e para as demais classes é 3, uma vez que estão há três rodadas sem ocorrer. Esse procedimento de contagem continua até a décima rodada. Observemos que, como se trata de um sorteio uniforme, todas as classes de altura têm a mesma probabilidade de ocorrer[7]. Nesse caso, como temos seis elementos, e todos têm a mesma probabilidade de ocorrer, o valor de probabilidade para cada elemento é

[7] Embora pareça trivial e até mesmo desnecessária, a inclusão desses valores de probabilidade na tabela 3 prepara o leitor para compreender mais profundamente o algoritmo do contraponto dissonante, no qual os valores de probabilidade são diretamente afetados pelos valores de contagem. Observando a tabela 3, verificamos que, no caso do sorteio uniforme, as probabilidades são inteiramente independentes da contagem.

sempre 1/6. As contagens e os valores de probabilidade, para cada uma das classes de alturas do conjunto A, são mostrados na tabela 3.

Rodada	Contagem						Valor sorteado	Probabilidade					
	0	1	4	5	8	9		0	1	4	5	8	9
1	0	1	1	1	1	1	0	1/6	1/6	1/6	1/6	1/6	1/6
2	0	2	2	2	2	2	0	1/6	1/6	1/6	1/6	1/6	1/6
3	1	0	3	3	3	3	1	1/6	1/6	1/6	1/6	1/6	1/6
4	2	1	4	4	4	0	9	1/6	1/6	1/6	1/6	1/6	1/6
5	0	2	5	5	5	1	0	1/6	1/6	1/6	1/6	1/6	1/6
6	1	3	0	6	6	2	4	1/6	1/6	1/6	1/6	1/6	1/6
7	0	4	1	7	7	3	0	1/6	1/6	1/6	1/6	1/6	1/6
8	0	5	2	8	8	4	0	1/6	1/6	1/6	1/6	1/6	1/6
9	1	0	3	9	9	5	1	1/6	1/6	1/6	1/6	1/6	1/6
10	2	1	0	10	10	6	4	1/6	1/6	1/6	1/6	1/6	1/6

Tab. 3: Contagem do sorteio do conjunto A= {0,1,4,5,8,9}, de acordo com os resultados mostrados na tabela 2 e probabilidade de ocorrência de cada um de seus elementos.

Realizemos, agora, o sorteio de elementos do mesmo conjunto A, aplicando uma versão simplificada do algoritmo do contraponto dissonante.[8] Isso implica em utilizar os valores obtidos nas contagens, a cada rodada, para o cálculo das probabilidades (P) de ocorrência de um determinando elemento *i*, inserido em um conjunto de *n* elementos, de acordo com a equação1.

[8] Veremos a seguir que na versão do algoritmo de Tenney fornecida por Polansky (2011: 67) aplica-se uma função ao valor da contagem a cada rodada do algoritmo. No entanto, o algoritmo que gerou os dados constantes na tabela 4 não incorpora ainda essa função e, por isso, é uma versão simplificada.

$$P_i = \frac{c_i}{\sum_{i=1}^{n} c_i} \quad (1)$$

Como dissemos anteriormente, antes de iniciarmos o sorteio, os valores de contagem devem ser inicializados para um mesmo valor. Escolhemos o valor 1[9]. Como a probabilidade depende da contagem, antes da primeira rodada todos os elementos têm a mesma probabilidade de ocorrência, ou seja, 1/6[10], uma vez que estão em uma condição de igualdade. Ao realizarmos nosso experimento, a primeira classe de alturas sorteada foi a 5. Assim, essa classe de alturas recebe o valor de contagem 0, porque aconteceu há 0 rodadas e as demais classes de alturas recebem o valor 2, significando que estão há duas rodadas sem ocorrer[11]. Como a contagem influencia diretamente a probabilidade de ocorrência dos elementos, a classe de alturas 5 tem probabilidade nula de acontecer no próximo sorteio e as demais classes de altura têm probabilidade de 2/10 cada uma. Na segunda rodada a classe de alturas 4 é sorteada. Isso reduz sua probabilidade para 0. A classe de alturas 5 tem probabilidade 1/3, enquanto as demais (que ainda não ocorreram) têm uma maior probabilidade, 3/13. Esse processo de ajuste das probabilidades acontece a cada sorteio. Os resultados são mostrados na tabela 4. Deve-se observar que ainda estamos lidando com um procedimento aleatório, ou seja, um maior valor de probabilidade não implica necessariamente no sorteio da classe de alturas associada a esse valor. Assim, por exemplo, na sexta rodada temos que a classe de alturas com maior probabilidade de ocorrer na rodada seguinte é a 1 (com probabilidade de 7/18). No entanto, a classe de alturas sorteada é a 9 (com probabilidade 3/18).

[9] Esse valor não poderia ser 0, uma vez que a probabilidade para a primeira rodada seria indefinida, porque ao aplicarmos a equação 1 obteríamos uma divisão 0/0.

[10] Obtém-se esse valor também pela equação 1, na qual $c_0 = 1$ e $(c_1+c_2+c_3+c_4+c_5+c_6) = 6$.

[11] Esse valor, como se percebe, é artificial: todas as classes de altura, com exceção da 5, que ocorreu imediatamente, estão sem acontecer há somente uma rodada. No entanto, como ajustamos todos os valores de contagem em 1 antes do início do sorteio, as contagens já começam com esse incremento.

Rodada	Contagem						Valor sorteado	Probabilidade					
	0	1	4	5	8	9		0	1	4	5	8	9
0	1	1	1	1	1	1		1/6	1/6	1/6	1/6	1/6	1/6
1	2	2	2	0	2	2	5	2/10	2/10	2/10	0	2/10	2/10
2	3	3	0	1	3	3	4	3/13	3/13	0	1/13	3/13	3/13
3	4	4	1	2	4	0	9	4/15	4/15	1/15	2/15	4/15	0
4	0	5	2	3	5	1	0	0	5/16	2/16	3/16	5/16	1/16
5	1	6	3	4	0	2	8	1/16	6/16	3/16	4/16	0	2/16
6	2	7	0	5	1	3	4	2/18	7/18	0	5/18	1/18	3/18
7	3	8	1	6	2	0	9	3/20	8/20	1/20	6/20	2/20	0
8	4	9	2	0	3	1	5	4/19	9/19	2/19	0	3/19	1/19
9	5	0	3	1	4	2	1	5/15	0	3/15	1/15	4/15	2/15
10	6	1	0	2	5	3	4	6/17	1/17	0	2/17	5/17	3/17

Tab. 4: Contagem do sorteio do conjunto A= {0,1,4,5,8,9}, utilizando o algoritmo do contraponto dissonante e probabilidade de ocorrência de cada um de seus elementos.

Para a realização desse sorteio utilizamos o seguinte algoritmo escrito em Python 3:

```
import random

conjunto = [0,1,4,5,8,9]
contagem = [1,1,1,1,1,1]

def weighted_choice(choices):
        total = 0
        for value in conjunto:
                total = total + choices[value]
                r = random.uniform(0, total)
                upto = 0
        for value in conjunto:
                prob = choices[value]
                if upto + prob> r:
                        returnvalue
                upto += prob
```

```
def dicionario_pesos(conjunto, contagem):
        pesos = {}
        for i in range(0,len(conjunto)):
                pesos[conjunto[i]] = contagem[i]
          return pesos

vetor_valor_sorteado=[]

for k in range(10):
        pesos = dicionario_pesos(conjunto,contagem)
        valor_sorteado = weighted_choice(pesos)
        for m in range(0, len(conjunto)):
                if conjunto[m]==valor_sorteado:
                        contagem[m]=0
                else:
                        contagem[m]=contagem[m]+1
        vetor_valor_sorteado.append(valor_sorteado)

print (vetor_valor_sorteado)
```

Classe de alturas sorteada	5	4	9	0	8	4	9	5	1	4
Quantidade de semicolcheias	4	3	6	1	5	3	6	4	2	3

Tab. 5: Classes de alturas do conjunto {0,1,4,5,8,9} sorteadas pelo contraponto dissonante e suas respectivas figuras rítmicas.

Observando os resultados da tabela 5, verificamos que as classes de altura 4, 5, 8 e 9 apareceram em 20% dos casos, cada uma, e as classes de altura 0 e 1 apareceram em 10% dos casos, cada uma. Isso mostra que os resultados obtidos nesse sorteio são mais próximos de uma distribuição uniforme em comparação com o sorteio anterior. Esses resultados são mostrados em notação musical na figura 5.

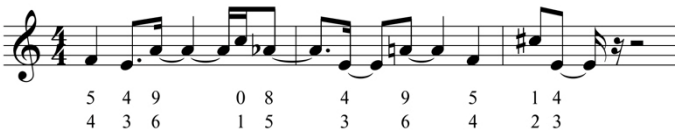


Fig. 5: Representação em notação musical dos resultados obtidos na tabela 3.

O próximo passo na elaboração do algoritmo do contraponto dissonante, da forma que se aproxima da descrição de Polansky (2011), consiste em generalizarmos o núcleo do algoritmo (equação I) pela inserção de uma função que atue sobre os valores de contagem aplicados ao cálculo da probabilidade. Assim, a equação I é modificada e transformada na equação II, em que os valores de contagem (c_i) passam por uma função *f*, durante o cálculo das probabilidades de ocorrência de cada elemento.

$$P_i = \frac{f(c_i)}{\sum_{i=1}^{n} f(c_i)} \qquad \text{(II)}$$

Suponhamos agora a aplicação desse princípio para a geração de uma linha melódica partindo do mesmo conjunto utilizado nos casos anteriores, de tal forma que possamos comparar o efeito da função no cálculo das probabilidades. Assim sendo, seguindo o mesmo procedimento utilizado nos casos anteriores, para o cálculo da probabilidade de cada elemento, conta-se há quantas rodadas esse elemento não é sorteado e aplica-se esse número (*c*) em uma determinada função. No nosso caso, a função utilizada será $f(c) = c^a$, onde *a* é uma constante. Observemos que, se essa constante *a* for igual a 1, teremos um caso similar ao anterior (tabela 4), ou seja, a função resulta no próprio valor da contagem. Se *a* for menor do que 1, teremos uma menor diferença entre as probabilidades ocasionando uma maior instabilidade no processo de escolha, mesmo que não haja repetição imediata, uma vez que o valor imediatamente escolhido tem probabilidade igual a zero. Se, por outro lado, a constante *a* for maior do que 1, teremos um maior distanciamento entre as probabilidades, o que torna maior a chance de evitarmos repetições nas proximidades. A tabela 6 mostra o sorteio do conjunto A = {0,1,4,5,8,9}, utilizando o algoritmo do contraponto dissonante com a função $f(c) = c^2$.

Rodada	Contagem						Valor sorteado	Probabilidade					
	0	1	4	5	8	9		0	1	4	5	8	9
0	1	1	1	1	1	1		0,17	0,17	0,17	0,17	0,17	0,17
1	2	0	2	2	2	2	1	0,20	0	0,20	0,20	0,20	0,20
2	3	1	3	3	3	0	9	0,24	0,03	0,24	0,24	0,24	0
3	4	2	4	0	4	1	5	0,30	0,08	0,30	0	0,30	0,02
4	5	3	0	1	5	2	4	0,39	0,14	0	0,02	0,39	0,06
5	0	4	1	2	6	3	0	0	0,24	0,02	0,06	0,55	0,14
6	1	5	2	3	0	4	8	0,02	0,45	0,07	0,16	0	0,29
7	2	6	3	4	1	0	9	0,06	0,55	0,14	0,24	0,02	0
8	0	7	4	5	2	1	0	0	0,52	0,17	0,26	0,04	0,01
9	1	0	5	6	3	2	1	0,01	0	0,33	0,48	0,12	0,05
10	2	1	6	0	4	3	5	0,06	0,02	0,55	0	0,24	0,14

Tab. 6: Contagem do sorteio do conjunto A= {0,1,4,5,8,9}, utilizando o algoritmo do contraponto dissonante, com a função $f(c) = c^2$, e probabilidade de ocorrência de cada um de seus elementos.

Para a realização desse sorteio, com a inclusão de uma função, utilizamos o seguinte algoritmo escrito em Python 3:

```
import random

conjunto = [0,1,4,5,8,9]
contagem = [1,1,1,1,1,1]
alpha = 2

def weighted_choice(choices):
        total = 0
        for value in conjunto:
                total = total + choices[value]
        r = random.uniform(0, total)
        upto = 0
        for value in conjunto:
```

```
                    prob = choices[value]
                    if upto + prob > r:
                        return value
                    upto += prob

def funcao_probabilidade(x, alpha_arg):
    return x**alpha_arg

def dicionario_pesos(conjunto, contagem):
    pesos = {}
    for i in range(0,len(conjunto)):
        pesos[conjunto[i]] = funcao_probabilidade(contagem[i],alpha)
    return pesos

vetor_valor_sorteado = []
print(contagem,end="   ")

for k in range(10):
    pesos = dicionario_pesos(conjunto,contagem)
    valor_sorteado = weighted_choice(pesos)
    for m in range(0, len(conjunto)):
        if conjunto[m]==valor_sorteado:
            contagem[m]=0
        else:
            contagem[m]=contagem[m]+1
    soma_pesos = sum(p for x,p in pesos.items())
    print(["{0:.2f}".format(pesos[x]/soma_pesos) for x in conjunto])
    print(contagem, valor_sorteado, end=" ")
    vetor_valor_sorteado.append(valor_sorteado)

pesos = dicionario_pesos(conjunto,contagem)
soma_pesos = sum(p for x,p in pesos.items())
print(["{0:.2f}".format(pesos[x]/soma_pesos) for x in conjunto])
print(vetor_valor_sorteado)
```

Classe de alturas sorteada	1	9	5	4	0	8	9	0	1	5
Quantidade de semicolcheias	2	6	4	3	1	5	6	1	2	4

Tab. 7: Classes de alturas do conjunto {0,1,4,5,8,9} sorteadas pelo contraponto dissonante e suas respectivas figuras rítmicas.

Observando os resultados da tabela 7, verificamos que as classes de altura 1, 9, 5 e 0 apareceram em 20% dos casos, cada uma, e as classes de altura 4 e 8 apareceram em 10% dos casos, cada uma. Isso mostra que os resultados obtidos nesse sorteio são similares, em termos de distribuição, aos resultados do sorteio anterior. Se a quantidade de rodadas fosse maior, o sorteio com função seria mais próximo da distribuição uniforme do que o sorteio anterior. Esses resultados são mostrados em notação musical na figura 6.

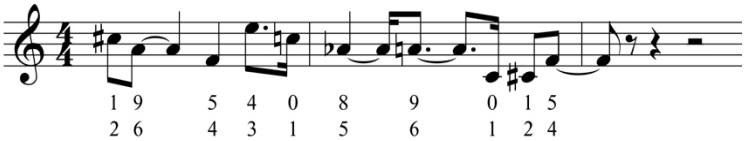


Fig. 6: Representação em notação musical dos resultados obtidos na tabela 7.

Vimos, assim, que as vantagens de utilização do algoritmo do contraponto dissonante tornam-se evidentes ao compará-lo com uma série de sorteios simples, a qual foi exemplificada na tabela 2, onde todas as classes de alturas tiveram a mesma probabilidade de ocorrer durante todo o processo e o resultando anterior não interferiu no resultado seguinte. Assim, numa série de sorteios simples, há probabilidade de ocorrer repetições em curto prazo, enquanto no algoritmo do contraponto dissonante essa probabilidade pode ser reduzida ou completamente eliminada. Uma extensão simples para este algoritmo é a utilização de pesos independentes de forma a gerar tendências à seleção de certos elementos.

Sistema composicional

O sistema composicional[12] aqui proposto e utilizado no planejamento de *Espectros,* primeiro movimento de *Berimbau*, para quarteto de clarinetes, de Liduino Pitombeira e de

[12] Segundo Flávio Lima (2011: 62), "um sistema composicional é um conjunto de diretrizes, formando um todo coerente, que coordena a utilização e interconexão de parâmetros musicais". Em nossa metodologia, consideramos que um sistema composicional se articula em níveis musicais profundos, arquetípicos, enquanto os aspectos mais superficiais são tratados em uma fase que denominamos

Dirge, para quinteto de metais, de Raphael Santos, utiliza o algoritmo do contraponto dissonante associado a técnicas e conceitos da música espectral e foi implementado computacionalmente. Para cada obra foi elaborado um aplicativo diferenciado, embora no núcleo desses aplicativos, o algoritmo que realiza o contraponto dissonante seja essencialmente o mesmo. No aplicativo utilizado para *Espectros,* elaborado em Python 3, o aspecto rítmico é desconsiderado e o resultado é fornecido no formato Lilypond[13] (a partir do qual se gera um arquivo PDF). Esse resultado é somente um repositório de acordes, cuja utilização deve ser concebida pelo compositor na fase de planejamento.

Por outro lado, para *Dirge,* o aplicativo, elaborado em Ruby, considera o aspecto rítmico e produz a obra pronta no formato Fomus[14], que pode ser convertido para MusicXML, MIDI ou Lilypond e ainda exportado para aplicativos proprietários de notação musical como o Finale ou Sibelius. Ambos os aplicativos que realizam o contraponto dissonante manipulam um conjunto de dados de entrada fornecidos pelo compositor.

Um terceiro aplicativo foi criado (em C++) para a implementação desse sistema: um seletor de picos de frequências espectrais (SPFE), cuja função será posteriormente explicada em detalhes. Além desses três aplicativos (o SPFE e os dois geradores de contraponto dissonante), também são necessários os *softwares* de código livre Audacity[15], Lilypond e Fomus. O fluxograma da figura 7 ilustra o funcionamento geral da implementação desse sistema.

planejamento composicional. Nessa fase de planejamento são decididos aspectos particulares não contemplados pelo sistema composicional. No caso particular de *Espectros*, o aspecto textural não é previsto em nível sistêmico e, portanto, deve ser tratado em nível de planejamento composicional. Assim, um mesmo sistema composicional pode, a partir de planejamentos diferenciados, produzir obras inteiramente distintas, cujo parentesco existe apenas em nível mais profundo e sistêmico.

[13] Disponível em: <http://www.lilypond.org>. Acesso em: abr. 2014.

[14] Disponível em: <http://fomus.sourceforge.net>. Acesso em: abr. 2014.

[15] Disponível em: <http://audacity.sourceforge.net>. Acesso em: abr. 2014.

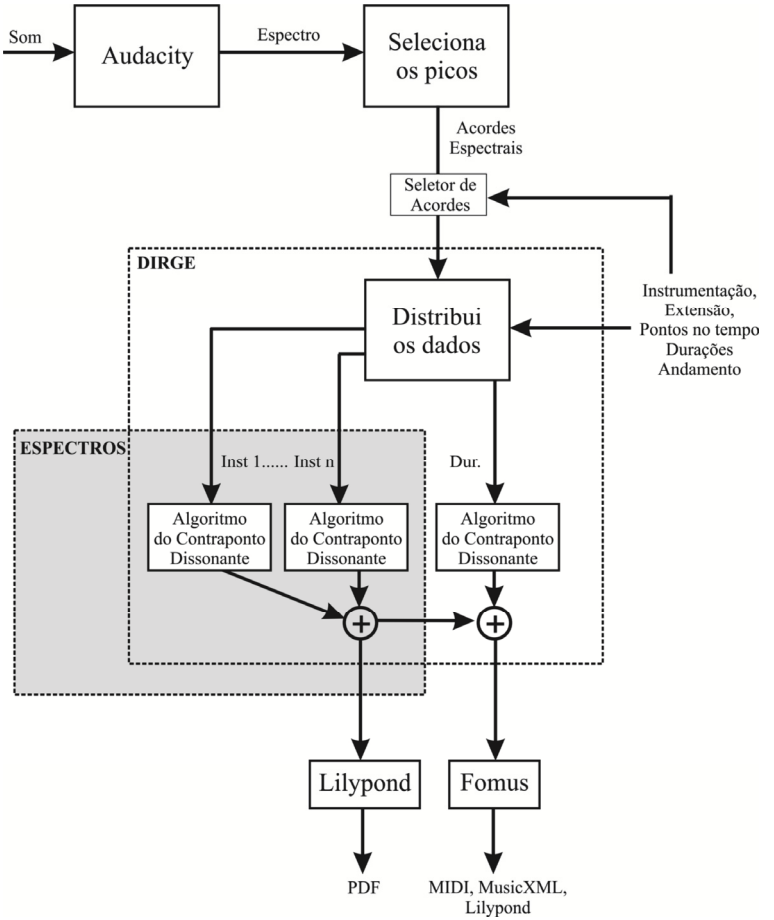


Fig. 7: Fluxograma do sistema composicional de *Espectros*, primeiro movimento de *Berimbau,* de Liduino Pitombeira, e de *Dirge,* de Raphael Santos.

O compositor deve, inicialmente, utilizar-se do *software* de código livre Audacity para realizar a análise espectral de um ou mais sons selecionados. Aqui, utiliza-se o primeiro aplicativo, desenvolvido nessa pesquisa, para a realização desse sistema: a partir do arquivo de texto proveniente da análise espectral realizada pelo Audacity, o SPFE seleciona as frequências que apresentam picos de amplitude e gera um novo arquivo com a lista de

alturas, no sistema temperado, que mais se aproximam, em frequência, a esses picos. Assim, o aplicativo fornece, em um arquivo de texto, um número de acordes espectrais equivalente ao número de arquivos de áudio fornecidos.

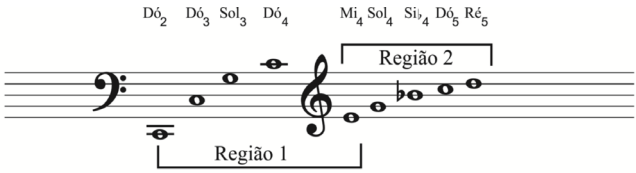
Após essa etapa inicial, o compositor deve determinar o conjunto de instrumentos da obra a ser composta, a tessitura a ser utilizada dentro do âmbito da extensão desses instrumentos, a duração total da obra, o andamento e o conjunto de valores rítmicos (somente no caso de *Dirge*), os conjuntos de componentes dos espectros que serão executados por cada instrumento e os pontos no tempo em que esses conjuntos passarão a ser utilizados.

Esses dados alimentam os aplicativos específicos para *Espectros* e *Dirge,* que realizam o algoritmo do contraponto dissonante separadamente para cada instrumento e utilizam como conjunto de elementos a serem sorteados os componentes do espectro em vigor no trecho de tempo em questão. Além disso, também é empregado um sistema de pesos independente ao procedimento do algoritmo e esse é utilizado para realizar uma interpolação entre os conjuntos de alturas, ou seja, uma transformação gradual entre esses conjuntos.

O funcionamento do algoritmo do contraponto dissonante associado ao sistema de pesos para gerar interpolação, pode ser exemplificado a partir do conjunto de alturas mostrado na figura 8, que foi dividido em duas regiões, as quais serão interpoladas ao longo de quatro sorteios.[16] Aqui dois processos estão em andamento simultaneamente.

No primeiro processo, em que se efetiva o algoritmo do contraponto dissonante, temos a função probabilística $f(x) = x^a$ onde $a = 3$ e x é a contagem de há quantas rodadas um elemento não é sorteado.

[16] Rigorosamente, do ponto de vista matemático, realizamos uma interpolação linear no fator β (figura 8), que afeta a probabilidade relativa dos elementos de ambos os conjuntos. Esse procedimento gera um efeito de interpolação conforme descrito em FINEBERG (2000:108).



		f(x)	β	β*f(x)*	S	*f(x)*	β	β*f(x)*	S	*f(x)*	β	β*f(x)*	S	*f(x)*	β	β*f(x)*	S	*f(x)*	β	β*f(x)*
Região 1	$Dó_2$	1^3	1	1	$Dó_2$	0^3	0,75	0		1^3	0,5	0,5		2^3	0,25	2		3^3	0	0
	$Dó_3$	1^3	1	1		2^3	0,75	6		3^3	0,5	13,5		4^3	0,25	16		5^3	0	0
	Sol_3	1^3	1	1		2^3	0,75	6		3^3	0,5	13,5		4^3	0,25	16		5^3	0	0
	$Dó_4$	1^3	1	1		2^3	0,75	6		3^3	0,5	13,5		4^3	0,25	16		5^3	0	0
	Mi_4	1^3	1	1		2^3	1	8	Mi_4	0^3	1	0		1^3	1	1		2^3	1	8
Região 2	Mi_4	1^3	1	1		2^3	1	8		0^3	1	0		1^3	1	1		2^3	1	8
	Sol_4	1^3	0	0		2^3	0,25	2		3^3	0,5	13,5		4^3	0,75	48		5^3	1	125
	$Si♭_4$	1^3	0	0		2^3	0,25	2		3^3	0,5	13,5	$Si♭_4$	0^3	0,75	0		1^3	1	1
	$Dó_5$	1^3	0	0		2^3	0,25	2		3^3	0,5	13,5		4^3	0,75	48		4^3	1	64
	$Ré_5$	1^3	0	0		2^3	0,25	2		3^3	0,5	13,5		4^3	0,75	48	$Ré_5$	0^3	1	0

Fig. 8: Simulação do algoritmo do contraponto dissonante associado ao sistema de pesos de interpolação, conforme utilizado em nosso sistema composicional.

No segundo processo, referente à interpolação, a probabilidade de um elemento ser selecionado é maior conforme sua proximidade do ponto no tempo em que o conjunto de alturas que o contém foi fixado. No exemplo da figura 8, consideramos que todas as durações são iguais. Assim uma região temporal é dividida proporcionalmente pelo número de sorteios, gerando o fator ß. Esse fator ß pode ser calculado para qualquer ponto no tempo, através da fórmula mostrada na equação III, onde a é a altura, t é o instante no tempo em que ocorrerá o sorteio, R é a região (1 ou 2) e d é o espaço de tempo em que ocorre a interpolação. Nesse caso aqui apresentado, d = 4.

$$\beta = g(t) = \begin{cases} 1, se\ a \in R_1 \cap R_2 \\ t/d, se\ a\ \in R_2 - R_1 \\ 1 - t/d, se\ a \in R_1 - R_2 \end{cases} \quad \text{(III)}$$

Nesse exemplo, iniciamos na Região 1, fixada no instante zero e caminhamos em direção à Região 2, fixada no instante 4. No instante 0, o valor de ß para os componentes da Região 1 é 1, por estarem o mais próximo possível do ponto em que o conjunto foi fixado. Já no instante 4, o ß dos componentes desta mesma região têm o valor 0, por estarem o mais distante possível do ponto em que o conjunto foi fixado. O inverso ocorre com os componentes da Região 2. Para o cálculo do peso final de um elemento ser selecionado os resultados dos dois processos são multiplicados (ß.*f(x)*). Assim, por exemplo, a altura $Dó_2$, que pertence somente à região 1, tem seu peso inicial igual a 1, que resulta da multiplicação do fator oriundo do algoritmo do contraponto dissonante (1^3) pelo valor de ß, que é igual a 1, porque t=0, já que, segundo a equação III, 1-0/4 = 0. Uma vez que a altura $Dó_2$ é sorteada, o fator oriundo do algoritmo do contraponto dissonante torna-se 0 e o valor de ß torna-se 0,75, já que, como essa altura pertence somente à região 1, segundo a equação III, o valor de ß é 1-1/4, ou seja, 3/4, porque t=1. Devemos observar que, para todos os elementos que pertencem unicamente à região 1, como o $Dó_2$, à medida que o tempo avança em direção a 4, o fator ß se aproxima de 0. Por outro lado, para um elemento que pertence somente à região 2, por exemplo, Sib4, o fator ß faz o caminho inverso, ou seja, inicia em 0 e progride em direção a 1. Por sua vez, o fator ß de um elemento que pertence às duas regiões permanece constante, como é o caso do Mi_4. Desta forma, obtém-se a variabilidade proporcionada pelo algoritmo do contraponto dissonante e a transição contínua obtida pelo processo de interpolação.

Para as durações de *Dirge*, ao contrário das alturas, um único algoritmo do contraponto dissonante é utilizado para todos os instrumentos e o conjunto de valores rítmicos determinado pelo compositor é utilizado como conjunto de elementos a serem sorteados.

O aplicativo gera um trecho que tem a duração especificada pelo compositor e, ao fim, produz um arquivo do Fomus, no caso de *Dirge,* ou diretamente no formato Lilypond, no caso de *Espectros.* No caso da saída gerada em Fomus, esse arquivo pode ser transformado em formatos como MusicXML, MIDI ou Lilypond.

Planejamento composicional de *Espectros*, de Liduino Pitombeira

O planejamento de *Espectros* se inicia com a escolha dos timbres a terem os seus espectros analisados. Escolhemos seis fragmentos de um som de berimbau e, com o auxílio do Audacity, geramos seis arquivos de texto contendo informações de amplitude e frequência dos componentes espectrais, desconsiderado informações de fase. Aplicamos

cada um desses arquivos ao SPFE, limitando o valor mínimo de intensidade em -50 dB, isto é, valores abaixo de -50 dB foram desconsiderados. A tabela 8 mostra a saída do aplicativo para um dos fragmentos (berimbau_1.wav), cujo espectro original é mostrado na figura 9. A coluna da esquerda indica o valor MIDI[17] referente à frequência de cada componente espectral e a coluna da direita indica a intensidade (em dB) para cada um desses valores MIDI. Esses valores representados em notação musical são mostrados na figura 10.

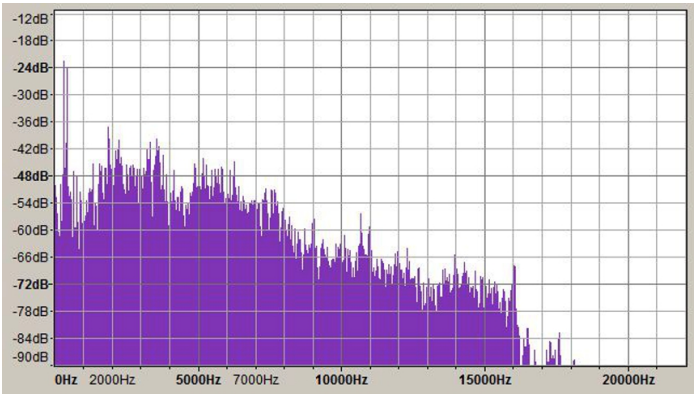


Fig. 9: Análise espectral do arquivo sonoro berimbau_1.wav, utilizado em *Espectros*, primeiro movimento de *Berimbau,* de Liduino Pitombeira.

Midi value	dB
-7.23	-46.80
34.28	-48.73
57.06	-39.55
62.14	-42.12
62.61	-42.43
64.08	-12.68
67.63	-46.29
68.30	-36.47
68.95	-11.36
69.47	-48.00
72.75	-48.34
76.01	-31.36
78.68	-38.50
81.00	-40.39

[17] O valor da altura MIDI para o Dó central do piano é 60.

83.10	-42.04
84.92	-37.86
86.49	-40.19
86.61	-36.81
88.01	-45.66
88.15	-25.73
89.56	-42.58
90.68	-47.18
90.89	-36.81
91.85	-45.19
91.91	-42.46
92.08	-40.46
93.00	-40.05
93.19	-48.49
93.27	-41.34
93.91	-48.70

Tab. 8: Valores MIDI da onda berimbau_1.wav e suas respectivas intensidades.

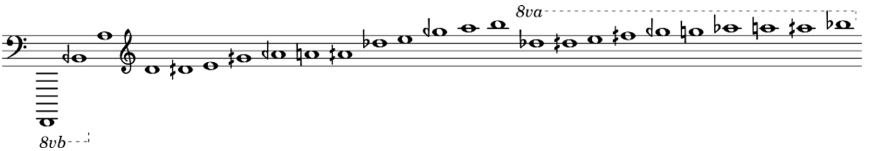


Fig. 10: Valores da tabela 8 representados em notação musical.

Em seguida, alimentamos cada um dos acordes no aplicativo do contraponto dissonante específico para a obra *Espectros*. Esse aplicativo consiste de dois módulos (escritos em Python 3). No módulo intitulado dcconfig.py são especificados os acordes, os instrumentos, o valor da constante *a* (alpha), necessária para o cálculo do contraponto dissonante (veja equação II e o texto correspondente) e a constante max_count, que indica o valor máximo de contagem, para o sorteio efetivado pelo algoritmo do contraponto dissonante. Consideramos as seguintes tessituras (em valores MIDI) para cada instrumento: Requinta = [55—91], Clarinete = [52—91] e Clarone = [36—65]. Os valores superiores para a requinta e para o clarone estão um pouco abaixo das extensões indicadas em manuais tradicionais de orquestração, tais como Adler (2002). A tabela 9 mostra as linhas de código do módulo dcconfig.py e os valores de entrada. Esses valores serão enviados ao segundo módulo do aplicativo (disscounter.py), que produzirá a saída no formato Lilypond. Observe-se que os valores MIDI foram arredondados (62,14=62 e 62,61=63, por exemplo)

e que os valores além da tessitura foram eliminados.

```
alpha = 2

max_count = 5

chords = {
'Requinta' : [
      (0, [62,63,64,68,69,73,76,79,81,83,85,86,88,90,91]),
      (60, [58,62,64,65,66,67,68,69,70,71,72,73,74,76]),
      (120, [57,58,64,65,66,67,68,69,70,71,72,73,76,77])
   ],
'ClarineteBbI' : [
      (0, [58,62,64,65,66,67,68,70,71,74,83,87,88,90]),
      (90, [60,62,64,65,66,67,68,69,70,71,73,74,75,77,78,80,81,85,86,87]),
      (120, [56,57,58,59,61,62,64,65,66,67,68,69,70])
   ],
'ClarineteBbII' : [
      (0, [57,62,63,64,68,69,73,76,79,81,83,85,86,88,90,91]),
      (60, [58,62,64,65,66,67,68,69,70,71,72,73,74,76]),
      (120, [57,58,64,65,66,67,68,69,70,71,72,73,76,77])
   ],
'Clarone' : [
      (0, [58,62,64,65]),
      (90, [60,62,64,65]),
      (120, [45,47,56,57,58,59,61,62,64,65])
   ]
}

score = [
   ('Requinta','treble'),
   ('ClarineteBbI','treble'),
   ('ClarineteBbII','treble'),
   ('Clarone', 'bass')
]
```

Tab. 9: Código do módulo dcconfig.py.

O arquivo Lilypond gerado pelo aplicativo produz um arquivo PDF com 120 acordes separados em 30 blocos de 4 acordes. A figura 11 mostra o primeiro sistema desse repositório de acordes. As indicações rítmicas são meramente organizacionais, uma vez que o algoritmo do contraponto dissonante não manipula o parâmetro rítmico (duração ou ponto de ataque) em *Espectros*. Observa-se também que o algoritmo não fornece informações sobre dinâmica, articulação ou diversificação textural: as dinâmicas e

articulações serão atribuídas livremente durante a composição e a distribuição textural será planejada de tal forma a atuar como um filtro adicional, ocultando partes dos acordes. Escolhemos para a textura a mesma distribuição textural utilizada por Milton Babbitt (1916-2011) em sua obra *Composition for Four Instruments*. Essa distribuição, mostrada na figura 12, se baseia nas possibilidades de subconjuntos de um conjunto com quatro elementos. Como se sabe, "um conjunto com *n* elementos tem 2^n subconjuntos, incluindo-se o próprio conjunto e o conjunto vazio" (WEISSTEIN, 1999: 1756). Assim, ao excluirmos o conjunto vazio (silêncio), verificamos que os quatro clarinetes podem aparecer na textura musical em 15 distribuições diferentes (2^4 -*1*), da mesma forma que foi utilizada na obra de Babbitt[18]. Os retângulos em linha pontilhada na figura 11 indicam a seleção dos instrumentos ativos e, consequentemente dos trechos revelados dos acordes a serem utilizados para cada compasso. Quando a mesma altura estiver presente em mais de um instrumento, esta será executada por apenas um desses instrumentos[19]. Como temos 120 acordes distribuídos segmentados em 30 blocos de 4 acordes, decidimos utilizar esse mesmo plano de distribuição duas vezes, porém de forma retrógrada da segunda vez, conferindo um caráter palindrômico a essa distribuição. A figura 13 apresenta os sete compassos iniciais de *Espectros.*

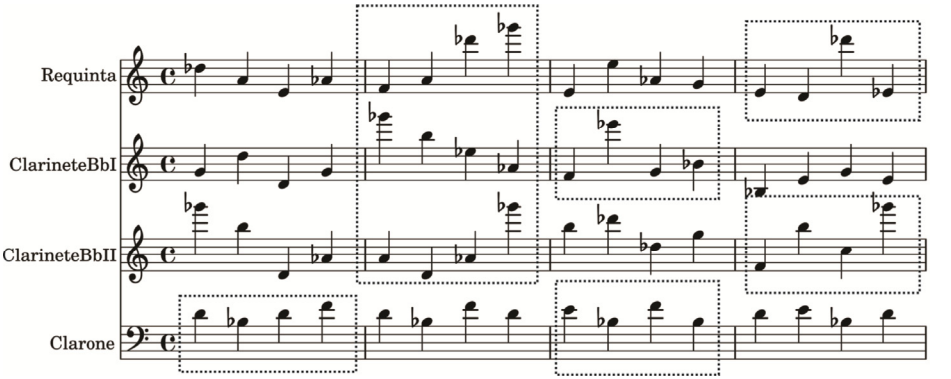


Fig. 11: Dezesseis acordes iniciais do repositório gerado pelo algoritmo do contraponto dissonante para *Espectros*, primeiro movimento de *Berimbau,* de Liduino Pitombeira.

[18] Babbitt utilizou essa distribuição textural por região. No caso de *Espectros,* utilizamos por compasso.
[19] Como é o caso do Solb$_6$ (Fá#$_6$), no segundo bloco de acordes, que só é executado pelo primeiro clarinete, embora esteja presente também no terceiro (neste trabalho estamos considerando que o Dó central do piano é o Dó$_4$).

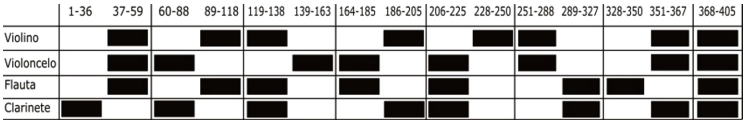


Fig. 12: Distribuição instrumental em *Composition for Four Instruments,* de Babbitt.

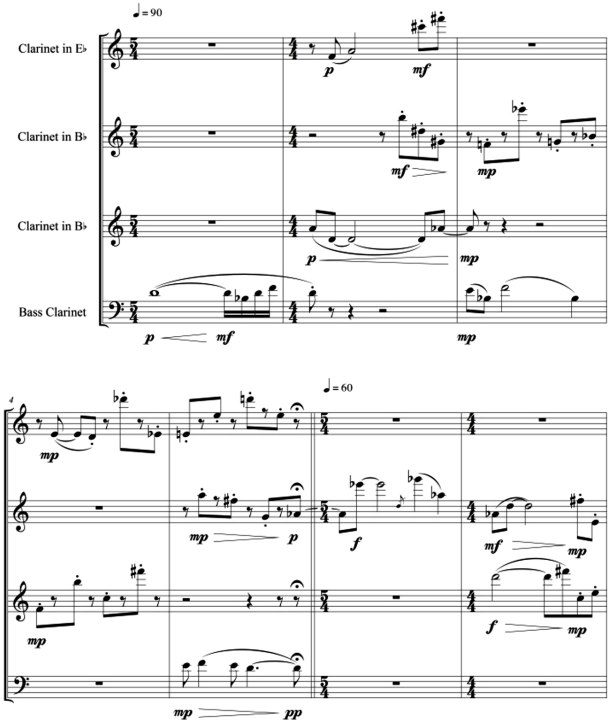


Fig. 13: *Espectros*, primeiro movimento de *Berimbau,* de Liduino Pitombeira (comp. 1-7).

Planejamento composicional de *Dirge*, de Raphael Santos

Para a composição de *Dirge*, para quinteto de metais (2 trompetes, trompa, trombone e tuba), foram utilizados cinco espectros: 3 notas graves de um piano, um tantã e um bumbo de bateria. O andamento foi fixado em 52 semínimas por minuto, a duração

total foi fixada em 4 minutos e 30 segundos e os únicos valores rítmicos utilizados foram os apresentados na tabela 10. Nessa tabela, a linha superior contém os valores rítmicos e na linha inferior são indicados os valores numéricos correspondentes tomando-se a colcheia como unidade.

1	2	3	4	5	6	7	8	10	12

Tab. 10: Figuras rítmicas utilizadas em *Dirge,* de Raphael Santos

A distribuição rítmica é realizada a partir de uma única instância do algoritmo do contraponto dissonante cujos valores sorteados são distribuídos entre todos os instrumentos. Na sequência de sorteios realizada pelo algoritmo, o instrumento a receber a duração sorteada é aquele cuja soma das durações acumuladas até o momento apresentar o menor valor. Em caso de empate, o instrumento que vier antes, de cima para baixo, na partitura, recebe a duração.

As tabelas 11 e 12 demonstram o processo passo a passo de distribuição dos 20 primeiros valores rítmicos sorteados pelo algoritmo do contraponto dissonante. A tabela 11 mostra as durações atribuídas a cada instrumento a cada passo do sorteio. A tabela 12 mostra o valor acumulado de cada instrumento até aquele ponto.

Inicialmente, todos os instrumentos estão empatados sem nenhuma duração. Assim, a primeira duração sorteada (5) é atribuída ao primeiro trompete. Com duração acumulada 5, isto o coloca a frente dos demais, ainda empatados sem nenhuma duração. Este processo se repete por mais quatro sorteios até que todos os instrumentos possuam alguma duração. No sexto sorteio, o instrumento que possui menor duração acumulada (1) é a trompa e, por isso, ela recebe a próxima duração (3). Com duração acumulada 4, a trompa ainda permanece como o instrumento a receber a próxima duração (5). Isto faz com que sua duração acumulada torne-se 9. Assim, o primeiro trompete, por ter duração acumulada 5, recebe o resultado do próximo sorteio (2) e sua duração acumulada torna-se 7. Isto gera um empate de menor duração acumulada entre os dois trompetes. Desta forma, o primeiro trompete recebe a próxima duração (12) e o segundo trompete recebe a seguinte (1). Este processo de sorteio e atribuição se repete até que toda a duração pré-determinada da peça seja preenchida.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Tr1	5							2	12											
Tr2		7								1	8								8	
Tp			1			3	5					4				5				10
Tn				12											6					
Tb					8								3	2			1	4		

Tab. 11: Vinte primeiras durações sorteadas em *Dirge,* de Raphael Santos.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Tr1	5							7	19											
Tr2		7								8	16								24	
Tp			1			4	9					13				18				28
Tn				12											18					
Tb					8								11	13			14	18		

Tab. 12: Valores acumulados das vinte primeiras durações sorteadas em *Dirge,* de Raphael Santos.

Tendo a estrutura rítmica definida, o algoritmo do contraponto dissonante é realizado independentemente para cada instrumento, atribuindo uma altura a cada duração. Para isto, os espectros foram distribuídos igualmente ao longo da duração da obra. A distribuição desses espectros foi realizada de forma a gerar variação de harmonicidade[20]. Assim, os espectros de um tantã e de um bumbo de bateria, menos harmônicos, intercalam os espectros oriundos de um piano, mais harmônicos. Além disto, para cada instrumento, foram definidas regiões de sua extensão que agem como filtros para as alturas contidas no espectro. Desta forma, somente as alturas do espectro atuante no trecho da peça e contidas dentro da região especificada são utilizadas pelo algoritmo. O procedimento de interpolação, no entanto, cuida da gradual transição entre regiões e, consequentemente, entre espectros. Essas regiões foram distribuídas temporalmente, como demonstrado na figura 14. Esta etapa do planejamento permitiu algum controle sobre a textura resultante e sua evolução, mesmo em um procedimento algorítmico envolvendo sorteios, através da possibilidade de determinar a utilização e ocupação dos registros. A figura 15 apresenta a primeira página da obra *Dirge*.

[20] O grau de harmonicidade cresce à medida que um determinado espectro se aproxima de um espectro harmônico, cujos componentes apresentam frequências derivadas de uma única série harmônica. Uma série harmônica apresenta uma relação inteira entre sua fundamental e os demais parciais.

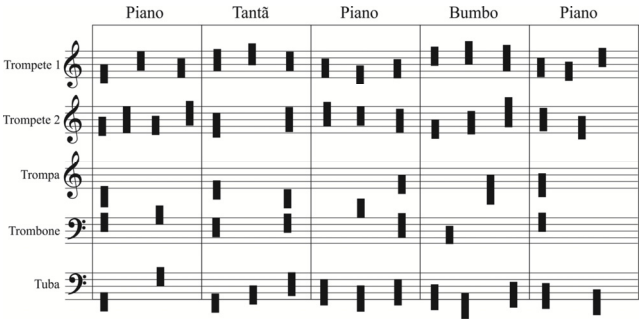


Fig. 14: Planejamento da distribuição temporal da tessitura instrumental em *Dirge*, de Raphael Santos.

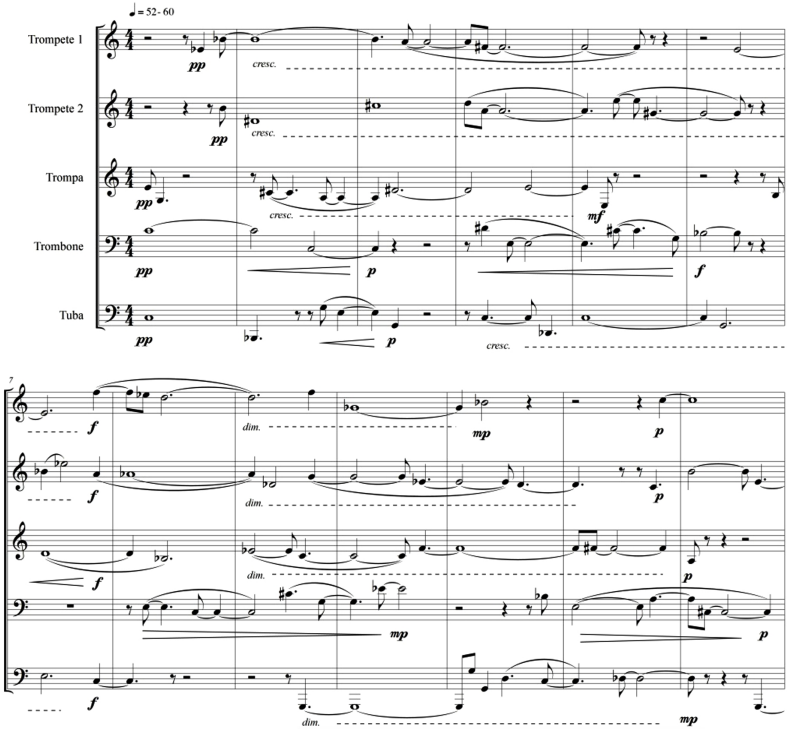


Fig. 15: Página inicial de *Dirge*, de Raphael Santos.

Considerações finais

O contraponto dissonante, em sua forma original teorizada por Cowell e Seeger, tem o potencial de estabelecer relações entre alturas que se distanciam esteticamente das relações praticadas no sistema tonal. O algoritmo do contraponto dissonante de Tenney, que surgiu a partir da observação da produção composicional da *American Atonal School,* produz sequências de elementos com polarização reduzida, ou seja, sequências com baixíssimo nível hierárquico. Apesar de sua aplicação inicial no campo das alturas, tais elementos podem ser oriundos de outros campos paramétricos (durações, intensidades etc.). Generalização semelhante ocorreu na prática serial que também teve início no campo das alturas (dodecafonismo) e posteriormente foi aplicada a outros parâmetros (serialismo integral).

Este artigo examinou as possibilidades de aplicar o algoritmo do contraponto dissonante tanto na determinação de parciais de espectros de sons concretos como na estruturação rítmica. Essas técnicas foram utilizadas no planejamento composicional de duas obras musicais – um quarteto de clarinetes e um quinteto de metais – e, para isso, foram implementados três aplicativos computacionais. Na primeira obra, o algoritmo atuou somente na escolha e interpolação das alturas extraídas de sons de berimbau. Todos os demais parâmetros foram manipulados livremente pelo compositor. Na segunda obra, o algoritmo atuou nas decisões referentes aos parâmetros altura e ritmo (duração e ponto de ataque), sendo as articulações e as dinâmicas decididas posteriormente pelo compositor.

Vemos, pelos resultados composicionais obtidos (figuras 11 e 13), que as tendências estéticas individuais de cada obra não são mecanicamente gerenciadas pelo algoritmo, ou seja, esse algoritmo atua apenas como um modelo computacional que pode ser aplicado em diversos contextos.

Pesquisas futuras podem apontar para uma maior generalização do algoritmo, buscando sua aplicação em outros parâmetros, incluindo a capacidade de sugerir aspectos estruturais. Outra possibilidade futura é a conjugação desse algoritmo com outro aplicativo produzido em nosso grupo de pesquisa (OLIVEIRA, 2014), o qual segmenta estruturas gestalticamente, aliás, outra linha de pesquisa também associada a Tenney.

Referências

ADLER, Samuel. *The Study of Orchestration*. New York: W.W. Norton, 2002.

AMES, Charles. Tutorial on Automated Composition. In: PROCEEDINGS OF THE ICMC, INTERNATIONAL COMPUTER MUSIC ASSOCIATION, 1987, Urbana, Illinois: p. 1-8.

COWELL, Henry. *New Musical Resources*. Cambridge: Cambridge Music Press, 1996.

FINEBERG, Joshua. Guide to the Basic Concepts and Techniques of Spectral Music. *Contemporary Music Review*, v. 19, parte 2, p. 81-113, 2000.

LIMA, Flávio. *Desenvolvimento de sistemas composicionais a partir de intertextualidade.* Dissertação (Mestrado em Música). Universidade Federal da Paraíba, João Pessoa, 2011.

OLIVEIRA, Helder. *Aplicação de princípios gestálticos no planejamento de estruturas composicionais utilizadas na peça Segmentos*. Dissertação (Mestrado em Música). Universidade Federal da Paraíba, João Pessoa, 2014.

POLANSKY, Larry; BARNETT, Alex; WINTER, Michael. A Few More Words About James Tenney: Dissonant Counterpoint and Statistical Feedback. *Journal of Mathematics and Music*, v. 5, n. 2, p. 63-82, 2011.

RUGGLES, Carl. *Evocations*. New York: American Music Edition, 1956. Partitura.

SEEGER, Charles. *Manual of dissonant counterpoint, Studies in Musicology II: 1929-1979* (Ann Pescatello, ed.), Berkeley: University of California Press, 1994.

SPILKER, John D. *"Substituting a New Order": Dissonant Counterpoint, Henry Cowell, and the Network of Ultra-Modern Composers*. Tese (Doutorado). Florida State University, 2010.

TENNEY, J. The Chronological Development of Carl Ruggles' Melodic Style. *Perspectives of New Music*, v. 16, n. 1, p. 36-69, 1977.

WEISSTEIN, Eric W. *CRC* Concise Encyclopedia of Mathematics. London: CRC Press, 1999.

Raphael Santos é bacharelando em composição pela Universidade Federal de Campina Grande. Foi aluno de composição de Liduino Pitombeira e, em cursos e festivais, teve aulas com Caspar Johannes Walter, Stefano Gervasoni, Stephen David Beck e Tristan Murail. Tem desenvolvido trabalhos na área de computação musical, composição algorítmica, música estocástica, música espectral, música eletroacústica mista e na criação de novos instrumentos com recursos eletrônicos. Iniciou seus estudos musicais com o violino e, mais tarde, foi aluno de piano de Maria Di Cavalcanti. contact@raphaelss.com

Liduino Pitombeira é professor de composição da Escola de Música da Universidade Federal do Rio de Janeiro (UFRJ). É doutor em composição pela Louisiana State University (EUA). Suas obras têm sido executadas pelo Quinteto de Sopros da Filarmônica de Berlim, Louisiana Sinfonietta, Orquestra Filarmônica de Poznan (Polônia) e Orquestra Sinfônica do Estado de São Paulo. Suas composições são publicadas pela Peters, Bella Musica, Cantus Quercus, Conners, Alry, Criadores do Brasil (OSESP), RioArte e Irmãos Vitale. Tem recebido diversas premiações em concursos de composição no Brasil e no exterior, incluindo o 1° prêmio no Concurso Camargo Guarnieri de 1998 e o prêmio 2003 MTNA-Shepherd Distinguished Composer of the Year. pitombeira@yahoo.com