

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

NAILSON DOS SANTOS CUNHA

UM *FRAMEWORK* PARA A GERAÇÃO SEMIAUTOMÁTICA DE SOLOS DE
GUITARRA

JOÃO PESSOA

2016

NAILSON DOS SANTOS CUNHA

UM *FRAMEWORK* PARA A GERAÇÃO SEMIAUTOMÁTICA DE SOLOS DE
GUITARRA

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Informática da Universidade Federal da Paraíba, como requisito parcial para obtenção do Título de Mestre em Informática. Área de Concentração: Computação Distribuída.

Orientador: Prof. Dr. Anand Subramanian

João Pessoa

2016

9

1 1 1
framework 1 1 1 1 1 1
1 1 1 1 1 1 1 16
91
1 1 1 1 1 1 1
1 1 1 1 1 1 1 3
1 1 1 1 1 1 1 6

3

NAILSON DOS SANTOS CUNHA

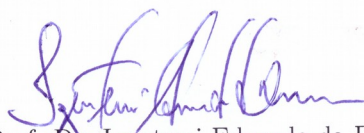
UM *FRAMEWORK* PARA A GERAÇÃO SEMIAUTOMÁTICA DE SOLOS DE
GUITARRA

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Informática da
Universidade Federal da Paraíba, como requisito parcial para obtenção do Título de Mestre
em Informática. Área de Concentração: Computação Distribuída.

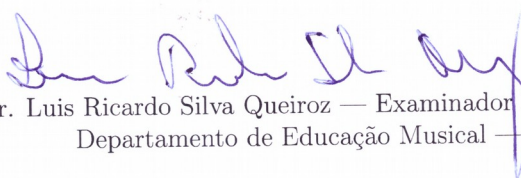
BANCA EXAMINADORA



Prof. Dr. Anand Subramanian — Orientador
Departamento de Engenharia de Produção — UFPB



Prof. Dr. Iguatemi Eduardo da Fonseca — Examinador Interno
Departamento de Sistemas de Computação — UFPB



Prof. Dr. Luis Ricardo Silva Queiroz — Examinador Externo ao Programa
Departamento de Educação Musical — UFPB



Prof. Dr. Igor Machado Coelho — Examinador Externo à Instituição
Instituto de Matemática e Estatística — UERJ

João Pessoa
2016

*When the storm is most violent
And the horizon seems to be far away
You'll find strength to go on
And conquer the misery that wants to bring
you down*

Mark Jansen (After Forever)

AGRADECIMENTOS

Primeiramente a Deus, por tornar possível e me dar condições de alcançar mais esta conquista.

Aos meus pais, a quem devo tudo. Sempre me incentivaram e fizeram o possível pra me dar condições de estudar e chegar aonde for possível.

A toda minha família, minha irmã Nadja, seu marido Emerson, meu sobrinho Brian e demais parentes.

A Anand, por ter acreditado em mim, me orientado durante este trabalho e se tornado um amigo. Espero poder participar de mais composições musicais suas e das suas sobrinhas.

A minha namorada, Eide, pela paciência e compreensão.

Aos meus amigos Peu, Igor, Alan e meu primo e amigo Michael “Tosco”. Vocês sempre acreditaram em mim, me deram força e incentivaram à prosseguir em todos os momentos.

Ao amigo Alysson “Buda”, pela grande ajuda e colaboração com os testes estatísticos.

Em especial ao meu grande amigo Thiago que, além de acreditar e incentivar, esteve presente em todos os momentos deste trabalho, contribuiu com ideias e participou ativamente de várias maneiras. Muito obrigado.

Aos meus companheiros de banda, Dan, Marconi, Bruno e Matheus, que me apoiaram e compreenderam os momentos de ausência.

A todo pessoal do LabMeQA, VNS, Yuree, Tuta, Rafa, Katicleide, Felipe, Fábio, Walton “Vinho no Cemitério” e, em especial, a Teobaldo, que me ajudou muito com o “polêmico” código de Marcos e também a Luciano que deu uma força com R.

A Dorien, que desde o começo apoiou, acreditou, abraçou a ideia e colaborou o tempo todo com esta pesquisa.

A Marcos Rosa e Léo Meira, grandes músicos e excelentes pessoas e com quem tivemos ótimas trocas de ideias sobre a parte mais musical deste trabalho.

A Zé Filho, que além de ter colaborado com ideias também aceitou fazer uma participação na defesa e fez uma incrível apresentação ao vivo, “duelando” com o computador.

Aos professores Iguatemi e Luiz Ricardo e Igor que aceitaram participar da banca e contribuíram com dicas preciosas para a evolução do trabalho.

E também a todas as pessoas, músicos, que disponibilizaram alguns minutos do seu dia e participaram da pesquisa de avaliação dos solos.

Muito Obrigado!

RESUMO

Este trabalho trata do desenvolvimento de um *framework* baseado em métodos computacionais e de otimização para a composição algorítmica, mais especificamente, para a geração de solos de guitarra. A abordagem proposta foi considerada semiautomática pois faz uso de pequenos fragmentos melódicos (*licks*) previamente criados a partir de modelos humanos. Os solos gerados possuem características do estilo musical *Blues* e são aplicados sobre um modelo de harmonia bastante conhecido denominado *12-Bar Blues*. Um banco de dados de *licks* foi criado, do qual são realizados sorteios de instâncias menores do conjunto, diversificando os possíveis candidatos a estarem no solo a ser gerado. De posse da instância, um problema de otimização, que consiste em sequenciar de forma otimizada um subconjunto de *licks*, é resolvido utilizando um modelo de programação linear inteira. Implementou-se um conjunto de regras para a criação de uma matriz que define o custo de transição entre os *licks*. As saídas geradas são armazenadas no formato MusicXML e podem ser lidas pela maioria dos aplicativos que possuam suporte a esse tipo de arquivo e disponibilizem visualização no formato de tablaturas. Os solos criados foram avaliados por uma amostra de 173 indivíduos, classificados como músicos iniciantes, intermediários e profissionais. Uma aplicação *web* foi desenvolvida para agilizar o processo de avaliação. Os resultados obtidos demonstram que os solos cujos *licks* foram sequenciados de forma otimizada foram estatisticamente mais bem avaliados que aqueles sequenciados aleatoriamente, indicando que a metodologia proposta foi capaz de produzir, em média, solos com percentual de aceitação favorável.

Palavras-chave: composição algorítmica, solos de guitarra, otimização, *Blues*.

ABSTRACT

This work deals with the development of a framework based on computational and optimization methods for algorithmic composition, more precisely, for the generation of guitar solos. The proposed approach was considered semiautomatic because it makes use of small melodic fragments (licks), previously created from human models. The solos generated are from the musical style Blues and they are applied over a well-known harmonic model called 12-Bar Blues. A licks database was created in which small instances containing a subset of them were randomly derived so as to diversify the possible candidates to be in the solo that will be generated. Once the instance is created, one solves an optimization problem that consists of determining the optimal sequence of a subset of licks by using a integer linear programming model. A set of rules was implemented for creating a matrix that defines the transition cost between the licks. The outputs generated were stored in the MusicXML format and they can be read by most applications that provide support for this type of file and are capable of displaying it using the tablatures format. The solos created were evaluated by a sample of 173 subjects, classified as beginners, intermediates and professional musicians. A web application was developed to streamline the evaluation process. The results obtained show that the solos whose licks were optimally sequenced were statistically much better evaluated than those randomly sequenced, which indicates that the proposed methodology was capable of producing, on average, solos with a favorable percentage of acceptance.

Keywords: algorithmic composition, guitar solos, optimization, Blues.

SUMÁRIO

1	INTRODUÇÃO	14
1.1	DEFINIÇÃO DO TEMA DA PESQUISA	14
1.2	JUSTIFICATIVA	16
1.3	OBJETIVOS	18
1.3.1	<i>Objetivo Geral</i>	18
1.3.2	<i>Objetivos Específicos</i>	18
1.4	ESTRUTURA DA DISSERTAÇÃO	18
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	PROGRAMAÇÃO MATEMÁTICA	19
2.1.1	<i>Programação Linear</i>	19
2.1.2	<i>Planos de Corte</i>	20
2.1.3	<i>Branch-and-bound</i>	21
2.1.4	<i>Branch-and-cut</i>	23
2.2	CONCEITOS BÁSICOS E TERMINOLOGIAS MUSICAIS	23
2.2.1	<i>Tablatura</i>	23
2.2.2	<i>Compasso</i>	24
2.2.3	<i>Quiáltera</i>	25
2.2.4	<i>Melodia e Harmonia</i>	26
2.2.5	<i>Altura Tonal (Pitch)</i>	26
2.2.6	<i>Blues</i>	26
2.2.7	<i>Características da Guitarra</i>	27
3	REVISÃO DA LITERATURA	29
4	ASPECTOS METODOLÓGICOS	37
4.1	METODOLOGIA PARA RESOLUÇÃO DO PROBLEMA	37
4.1.1	<i>Definição do Estilo Musical</i>	37
4.1.2	<i>Definição de Qualidade dos Solos</i>	38
4.1.3	<i>Regras para Criação dos Solos e Matriz de Transições</i>	39
4.1.4	<i>Banco de Dados de Licks</i>	41
4.1.5	<i>Licks de Repetição</i>	42
4.1.6	<i>Formulando o Problema de Geração de Solos Como Um Problema de Otimização Combinatória</i>	45
4.1.7	<i>Framework Proposto</i>	48
4.1.7.1	<i>Pré-processamento</i>	49
4.1.7.2	<i>Otimização</i>	50
4.1.7.3	<i>Pós-processamento</i>	50
4.2	ESTRATÉGIA PARA AVALIAÇÃO DOS SOLOS	52

5	RESULTADOS COMPUTACIONAIS	56
5.1	SISTEMA DESENVOLVIDO PARA A AVALIAÇÃO DOS SOLOS	56
5.2	RESULTADOS DAS AVALIAÇÕES	58
5.2.1	<i>Iniciantes</i>	61
5.2.2	<i>Intermediários</i>	63
5.2.3	<i>Profissionais</i>	65
5.2.4	<i>Sumário do Resultado das Avaliações</i>	66
6	CONSIDERAÇÕES FINAIS	68
	REFERÊNCIAS BIBLIOGRÁFICAS	72
	Apêndices	
A	- HARMONIA UTILIZADA NOS SOLOS	80
B	- TABLATURAS DOS SOLOS GERADOS ALEATORIAMENTE	81
C	- TABLATURAS DOS SOLOS OTIMIZADOS GERADOS	91

LISTA DE FIGURAS

2.1	Ilustração de planos de corte	21
2.2	Construção do espaço de soluções do B&B	22
2.3	Exemplo de representação por tablaturas	23
2.4	Exemplo de compassos	24
2.5	Exemplo de quiáltera	25
2.6	<i>12-Bar Blues</i> básico	27
4.1	<i>Licks</i> de repetição	43
4.2	Figuras rítmicas e suas respectivas durações	44
4.3	Visualização da geração de solos como variante do SPPRC	48
4.4	Fluxograma de funcionamento do sistema	49
4.5	Ilustração do pós-processamento. Caso 1	51
4.6	Ilustração do pós-processamento. Caso 2	52
5.1	Seleção de nível do músico	57
5.2	Página de avaliação dos solos	57
5.3	Opção de aceitação do utilização do solo avaliado	57
5.4	Visão geral. Solos aleatórios	58
5.5	Visão geral. Solos otimizados	59
5.6	<i>Box plot</i> do resultado do <i>Wilcoxon signed-rank test</i>	61
5.7	Resultado do <i>Wilcoxon test</i> para frequência de uso dos solos	61
5.8	Avaliação dos solos aleatórios pelos músicos iniciantes	62
5.9	Avaliação dos solos otimizados pelos músicos iniciantes	62
5.10	<i>Box plot</i> do resultado da comparação das médias para os músicos iniciantes	63
5.11	Avaliação dos solos aleatórios pelos músicos intermediários	64
5.12	Avaliação dos solos otimizados pelos músicos intermediários	64
5.13	<i>Box plot</i> do resultado da comparação das médias para os músicos intermediários	65
5.14	Avaliação dos solos aleatórios pelos músicos profissionais	66
5.15	Avaliação dos solos otimizados pelos músicos profissionais	66
5.16	<i>Box plot</i> do resultado da comparação das médias para os músicos profissionais	67
A.1	Harmonia <i>12-Bar Blues</i> utilizada nos solos	80
B.1	Tablatura do solo aleatório 1	81
B.2	Tablatura do solo aleatório 2	82
B.3	Tablatura do solo aleatório 3	82
B.4	Tablatura do solo aleatório 4	83
B.5	Tablatura do solo aleatório 5	84
B.6	Tablatura do solo aleatório 6	85
B.7	Tablatura do solo aleatório 7	85
B.8	Tablatura do solo aleatório 8	86

B.9	Tablatura do solo aleatório 9	87
B.10	Tablatura do solo aleatório 10	88
B.11	Tablatura do solo aleatório 11	89
B.12	Tablatura do solo aleatório 12	90
C.1	Tablatura do solo otimizado 1	91
C.2	Tablatura do solo otimizado 2	92
C.3	Tablatura do solo otimizado 3	93
C.4	Tablatura do solo otimizado 4	94
C.5	Tablatura do solo otimizado 5	95
C.6	Tablatura do solo otimizado 6	96
C.7	Tablatura do solo otimizado 7	97
C.8	Tablatura do solo otimizado 8	98
C.9	Tablatura do solo otimizado 9	99
C.10	Tablatura do solo otimizado 10	100
C.11	Tablatura do solo otimizado 11	101
C.12	Tablatura do solo otimizado 12	102
C.13	Tablatura do solo otimizado 13	103
C.14	Tablatura do solo otimizado 14	104
C.15	Tablatura do solo otimizado 15	105

LISTA DE TABELAS

3.1	Trabalhos relacionados em CA e CAC	35
4.1	Regras individuais implementadas	39
4.2	Regras de transição implementadas	40
4.3	Notas e significados para avaliação dos solos	54
5.1	Contabilização geral em “%” das avaliações dos solos	59
5.2	Resultado dos testes de normalidade	59
5.3	Resultado do <i>Wilcoxon signed-rank test</i>	60
5.4	Resultados da avaliação dos solos aleatórios	67
5.5	Resultados da avaliação dos solos otimizados	67

LISTA DE ABREVIACOES

ACO	<i>Ant Colony Optimization</i>
AG	Algoritmos Genéticos
AWS	<i>Amazon Web Services</i>
BPM	Batidas por Minuto
B&B	<i>Branch and Bound</i>
B&C	<i>Branch and Cut</i>
CA	Composio Algortmica
CAC	<i>Computer-aided Composition</i>
CE	Computao Evolutiva
HMM	<i>Hidden Markov Model</i>
IA	Inteligncia Artificial
MA	<i>Memetic Algorithm</i>
ME	Msica Evolutiva
PCV	Problema do Caixeiro Viajante
PO	Pesquisa Operacional
RN	Redes Neurais
SANE	<i>Symbiotic, Adaptive Neuro-Evolution</i>
SPPRC	<i>Shortest Path Problems with Resource Constraints</i>
SRN	<i>Simple Recurrent Network</i>
VNS	<i>Variable Neighborhood Search</i>
PL	Programao Linear
PLI	Programao Linear Inteira
RL	Relaxao Linear

1 INTRODUÇÃO

1.1 DEFINIÇÃO DO TEMA DA PESQUISA

Composição algorítmica (CA) é o processo que envolve o uso de algoritmos para a criação de composições musicais de maneira automática. Tais algoritmos são geralmente desenvolvidos através da identificação e estudo de padrões em métodos tradicionais de composição e suas aplicações na criação de novas músicas (Freitas, 2011). Alguns desses padrões podem ser implementados em forma de regras que o algoritmo deve seguir para o desenvolvimento de uma nova peça musical. Porém, apenas o uso de regras pode não ser suficiente para criar uma música de “qualidade”, resultando algumas vezes apenas em músicas que, apesar de respeitarem algumas regras básicas de teoria musical, não possuem dinâmica, variações e outros elementos que enriquecem a música.

Miranda e Biles (2007) afirmam que os algoritmos para CA são comumente estocásticos, baseados em regras e/ou em inteligência artificial (IA). Em uma visão mais ampla, de acordo com Papadopoulos e Wiggins (1999) e Gonçalves et al (2009), pode-se citar como principais métodos para a CA: modelos matemáticos, sistemas baseados em conhecimento, gramáticas, métodos evolucionários, sistemas baseados em aprendizado, sistemas distribuídos e sistemas híbridos.

O uso de algoritmos no processo de composição musical está longe de ser uma prática recente. O primeiro exemplo conhecido desta aplicação foi utilizado na peça *Musikalisches Würfelspiel* (Música de Dados), composta por Wolfgang Amadeus Mozart no século XVIII, de modo que os resultados obtidos jogando dados direcionavam a escolha de pequenas seções da música e as variações que seriam aplicadas e que juntas resultariam

na peça musical completa (Feijão, 2004).

Já a composição auxiliada por computador (CAC, do inglês *Computer-aided composition*) é uma área de pesquisa relativamente nova (Herremans e Sörensen, 2012). Os trabalhos de Hiller e Isaacson (1957, 1958) são considerados os primeiros a simular o processo de composição musical através de uma abordagem baseada em regras (Sandred et al, 2009).

Muitas pesquisas têm sido realizadas ultimamente nas áreas de CA e CAC. Grande parte envolve o uso de heurísticas e de meta-heurísticas aplicadas à criação de novas composições musicais, buscando, na maioria das vezes, alcançar uma realidade de composição próxima aos modelos usados pelos humanos, isto é, de tal forma que as composições geradas por algoritmos não careçam de dinamicidade e/ou criatividade. Várias técnicas podem ser aplicadas para se alcançar este objetivo. Langston (1989) descreve seis técnicas que foram utilizadas em um projeto onde era possível ouvir uma música gerada por computador através de uma ligação via o sistema de telefonia público.

Uma forma comum que músicos usam para aprender a tocar um determinado instrumento, como por exemplo a guitarra, é a imitação. Este tipo de aprendizado se resume basicamente em ouvir uma música e tentar reproduzir de maneira igual o que foi ouvido. Isso funciona bem para o aprendizado de improvisação e criação de solos. Para criar um solo, pode-se dizer que músicos possuem em mente um “banco de dados” de melodias que serve como base para o improviso. Esse método também é útil para que o músico possa utilizar tais melodias como ponto de partida para a criação das suas próprias. Um bom estudo sobre o aprendizado por imitação pode ser visto em Flor e Holder (1996).

Neste trabalho foi desenvolvido um *framework*, implementado por meio de uma abordagem computacional, com o objetivo de compor solos que tenham características do instrumento guitarra. O método desenvolvido modela a criação de solos de guitarra como um problema de otimização combinatória em conjunto com uma técnica que Langston (1989) chama de “*Riffology*”. Tal técnica tem como unidade básica de construção o “*riff*”, que é um fragmento melódico pequeno, baseado em um modelo humano, aplicado na construção de solos por meio do sequenciamento de uma determinada quantidade de *riffs*.

Também foi usado, em partes, a ideia do aprendizado por imitação, dado que existe um banco de dados de *licks* que serve com ponto de partida para a criação dos solos.

A abordagem utilizada foi considerada semiautomática devido ao fato de utilizar como unidade fundamental para a construção dos solos *licks* já disponíveis criados a partir de modelos humanos. Entretanto, a escolha preferencial por *licks* de tamanhos maiores que dois compassos, fazendo assim a subdivisão desses *licks* em vários outros menores, de apenas um compasso, tornou possível a criação de inúmeras melodias que diferem bastante das melodias originais dos *licks* completos, antes da sua subdivisão.

O termo “*riff*” neste trabalho possui o mesmo significado de “*lick*” ou “frase”, e estes termos serão usados de forma intercambiável. Outros termos específicos necessários à uma melhor compreensão desta pesquisa são explicados no Capítulo 2.

1.2 JUSTIFICATIVA

Com base em uma pesquisa na literatura sobre o tema, é possível perceber a existência de vários trabalhos na área de composição algorítmica, baseados, principalmente, em algoritmos genéticos, como pode ser encontrado em Freitas (2011), Miranda e Biles (2007), Biles (1994), Matic (2010) e Chen (2000); cadeias de markov, como pode ser visto em Ames (1989), Hori et al (2013), Cambouropoulos (1994), Yoshinaga et al (2012), Davismoon e Eccles (2010), Hall e Smith (1996) e Hori et al (2012); e alguns poucos que fazem uso de outras heurísticas e meta-heurísticas em sua abordagem como Herremans e Sörensen (2012) e Herremans e Sörensen (2013).

Neste contexto, o trabalho proposto traz como principais contribuições uma forma de utilizar algoritmos para compor solos de guitarra aplicando métodos de otimização, fundamentados em programação matemática, que se baseiam na geração de caminhos em subconjuntos de *licks* selecionados de forma prévia de um banco de dados, bem como o próprio banco de dados de *licks* e também o *framework* desenvolvido.

Até o presente momento, com base no que foi explorado durante o período de revisão da literatura, apenas o trabalho de Tanaka e Fujii (2015) foi encontrado como exemplo de pesquisa que explora a utilização de programação matemática para a CA, sendo que este

não apresenta a implementação da proposta e resultados práticos.

Garcia (2011b), em seu estudo sobre auto-aprendizagem em guitarra, destaca a importância da Internet como meio de colaboração na difusão e encurtamento dos caminhos entre os aprendizes de guitarra e seu aprendizado. Leme e Bellochio (2007) fizeram um estudo a respeito da utilização de tecnologias por professores de escolas de músicas, ressaltando, em alguns pontos, as vantagens e desvantagens de utilização de tecnologias neste contexto. Os autores consideraram importante o uso de recursos tecnológicos na educação como forma de potencializar os processos de ensinar e de aprender. Assim sendo, o presente trabalho também pode contribuir com uma forma de guiar a criação de uma ferramenta que pode vir a ser usada como um recurso tecnológico de apoio ao ensino e aprendizado musical.

Uma ferramenta desse tipo pode servir como um sistema de auxílio à aprendizagem de criação de solos e improvisos de guitarra. Por exemplo, o estudante de música definiria algumas configurações de entrada e o sistema geraria um solo a partir do banco de dados de *licks* e o apresentaria em forma de tablatura, permitindo assim, que o estudante pudesse analisar e aprender com os solos gerados.

O sistema proposto também pode ser útil para auxiliar compositores no processo de criação de solos para suas músicas. Músicos que sentem alguma dificuldade em fazer improvisos e solos de guitarra podem utilizar um solo gerado semiautomaticamente como ponto de partida para o seu próprio solo ou até mesmo usar o solo por completo, se assim preferirem, também contando com a vantagem de estarem utilizando um sistema gratuito e de código aberto.

Existem no mercado softwares que possuem recursos de composição e criação de melodias e solos, como, por exemplo, o *Band-in-a-Box*¹. Entretanto, tais softwares tem como desvantagem o fato de, em sua maioria, ser necessário a aquisição de uma licença de utilização e de serem sistemas de código fechado, de modo que não se sabe com detalhes que tipo de critérios e técnicas são utilizados por eles.

¹<http://www.pgmusic.com/>

1.3 OBJETIVOS

1.3.1 *Objetivo Geral*

Desenvolver um *framework* capaz de criar automaticamente solos de guitarra através do uso de técnicas de otimização e de sequenciamento de *licks* existentes em um banco de dados.

1.3.2 *Objetivos Específicos*

- Analisar estratégias computacionais de criação de solos para guitarra.
- Identificar opções que favoreçam a fluência e musicalidade dos solos gerados.
- Compreensão das possibilidades de otimização dos materiais musicais criados.
- Elaborar a arquitetura de funcionamento do *framework*.
- Criar um banco de dados de *licks* de guitarra do estilo *Blues*.
- Desenvolver e implementar um modelo matemático para o sequenciamento dos *licks*.
- Elaborar regras de bonificação/penalização para a transição entre os *licks*.
- Testar as abordagens propostas utilizando o banco de dados criado.
- Submeter os resultados obtidos à uma avaliação humana.
- Comparar os resultados obtidos entre as diferentes abordagens implementadas.

1.4 ESTRUTURA DA DISSERTAÇÃO

Os demais capítulos deste trabalho estão organizados da seguinte forma. O Capítulo 2 apresenta uma breve fundamentação teórica a respeito dos conceitos de otimização e de notação musical necessários à uma melhor compreensão deste trabalho. O Capítulo 3 descreve os principais trabalhos relacionados a esta pesquisa. O Capítulo 4 explica a metodologia adotada para o desenvolvimento do trabalho. O Capítulo 5 mostra os resultados obtidos, bem como a estratégia de avaliação e análise do experimento. Por fim, o Capítulo 6 traz as considerações finais e propostas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta uma breve fundamentação teórica a respeito dos conceitos de programação matemática importantes para a compreensão das técnicas que foram aplicadas para a criação dos solos, bem como os principais conceitos musicais necessários ao entendimento da terminologia apresentada neste trabalho.

2.1 PROGRAMAÇÃO MATEMÁTICA

Programação matemática é um ramo da pesquisa operacional (PO) que compreende a utilização de modelos matemáticos para a resolução de problemas em que a solução pode ser de natureza combinatória ou contínua. Possui aplicação em várias áreas como, por exemplo, indústria, transportes, saúde, educação, agricultura, finanças, economia, administração pública, entre outras (Maculan e Fampa, 2006).

Para uma melhor compreensão do presente trabalho, é importante ter uma noção básica dos conceitos principais de programação linear (PL), planos de corte, *branch-and-bound* e *branch-and-cut* (B&C), assuntos abordados no decorrer desta seção.

2.1.1 Programação Linear

A PL faz parte das disciplinas que compõem a programação matemática. Muitos problemas clássicos de otimização podem ser modelados como problemas de PL (Maculan e Fampa, 2006). Por sua natureza ser buscar a maximização ou minimização de resultados, recursos ou custos, a PL é também uma ferramenta de auxílio ao processo de tomada de

decisões (Ceciliano, 2007).

Os problemas de PL são problemas de otimização nos quais a função objetivo e as restrições são todas lineares. Um modelo de PL apresenta uma estrutura padronizada e repetitiva que permite resolver os mais variados problemas. Seus principais elementos são: variáveis de decisão, função objetivo e restrições (Lachtermacher, 2004).

Nesse trabalho foi empregada uma abordagem de programação matemática em que o objetivo principal, a criação de solos de guitarra *Blues*, foi modelado como um problema de programação linear inteira (PLI), que pode ser entendida como um caso específico da programação linear em que as variáveis apenas podem assumir valores inteiros, no caso de programação inteira, ou apenas algumas variáveis podem assumir valores inteiros, no caso de programação inteira mista.

Para a resolução de problemas de PLI comumente são utilizados programas conhecidos como *solvers*. Como exemplos de tais programas pode-se citar o CPLEX, Xpress, Gurobi e GPLK.

2.1.2 Planos de Corte

Algoritmos de planos de corte resolvem problemas de programação inteira modificando o espaço de soluções de uma relaxação linear (RL) de um problema de programação linear até que uma solução inteira seja obtida. Seu funcionamento, de uma maneira simples, baseia-se em particionar a região de soluções viáveis em subdivisões, removendo soluções fracionárias sem remover soluções inteiras viáveis, através da adição de novas restrições ou desigualdades válidas. As novas restrições adicionadas vão reduzindo sucessivamente o espaço de soluções possíveis até que a solução inteira ótima seja encontrada (Chen et al, 2010; Avriel, 2003).

Na Figura 2.1, é possível verificar o funcionamento de um modelo básico de planos de corte. Na etapa (a), a relaxação linear foi resolvida, porém, a solução encontrada possui variáveis com valor fracionário. Logo, deve-se adicionar uma nova restrição (corte) e resolver novamente a relaxação, conforme descrito na etapa (b) da figura. Por fim, na etapa (c), a solução encontrada foi um valor inteiro, chegando assim à solução do problema.

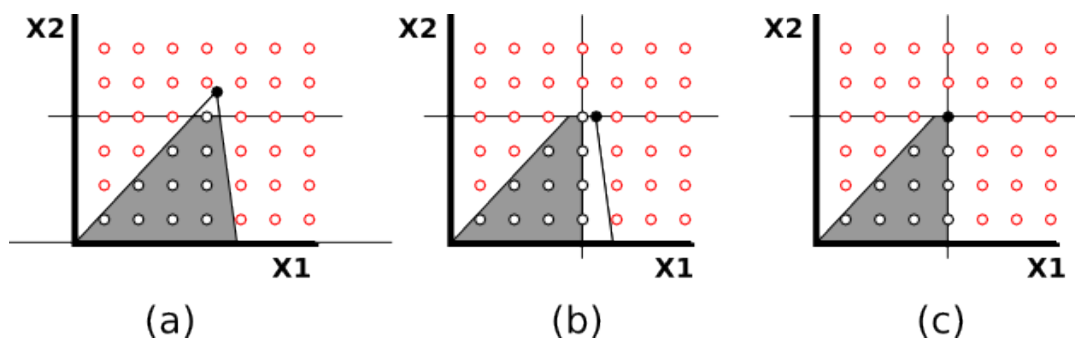


Figura 2.1: Ilustração de planos de corte

Há classes de cortes genéricas, como por exemplo os cortes de Gomory, que geralmente são válidas para qualquer modelo de PLI. Uma descrição detalhada acerca dos procedimentos utilizados para separar tais cortes pode ser encontrada em Chen et al (2010). Por outro lado, também é possível obter classes de desigualdades válidas voltadas para problemas específicos. Porém, nesse último caso, faz-se necessário desenvolver procedimentos especializados para geração de cortes, principalmente quando a classe em questão possuir uma quantidade exponencial de restrições.

2.1.3 *Branch-and-bound*

O *branch-and-bound* é um arquétipo para a construção de algoritmos eficientes com o intuito de resolver problemas de otimização combinatória. Seu funcionamento baseia-se na exploração do espaço de soluções, utilizando uma abordagem de dividir e conquistar, que consiste em dividir um problema em subproblemas menores e mais fáceis de resolver, e aplicando limites (*bounds*) à função que se deseja otimizar, em busca da melhor solução (Chen et al, 2010).

A maneira mais comum de representação de um B&B é através de uma árvore dinamicamente gerada, denominada árvore de *branch-and-bound*, onde o conjunto completo está na raiz da árvore e os subproblemas gerados são representados como os nós da árvore, processados a cada iteração (Chen et al, 2010; Clausen, 1999).

Cada iteração do B&B é dividida em três etapas: processamento do nó, cálculo dos limites e *branching*. O algoritmo de B&B explora os nós da árvore e, antes da enumeração das soluções candidatas, é realizada uma comparação entre os limites superiores e inferiores

estimados. Caso não seja possível encontrar uma solução melhor que uma previamente encontrada, o nó é descartado, isto é, a etapa de *branching* não é realizada sobre tal nó (Clausen, 1999).

A Figura 2.2 ilustra o funcionamento básico do B&B e como é trabalhado seu espaço de soluções. Na etapa (a), têm-se o conjunto completo que será a raiz da árvore. Após a seleção do nó a ser investigado é realizada a operação de *branching*, isto é, subdivisão do nó em subespaços a serem investigados nas iterações subseqüentes, como pode ser visto na etapa (b). Cada subespaço gerado é testado para conferir se ele é formado por uma única solução que será comparada à outra solução encontrada anteriormente. Se a função de *bounding* confirmar que o nó não pode possuir a solução ótima, o mesmo é descartado, caso contrario ele é mantido como ativo, conforme ilustrado na etapa (c).

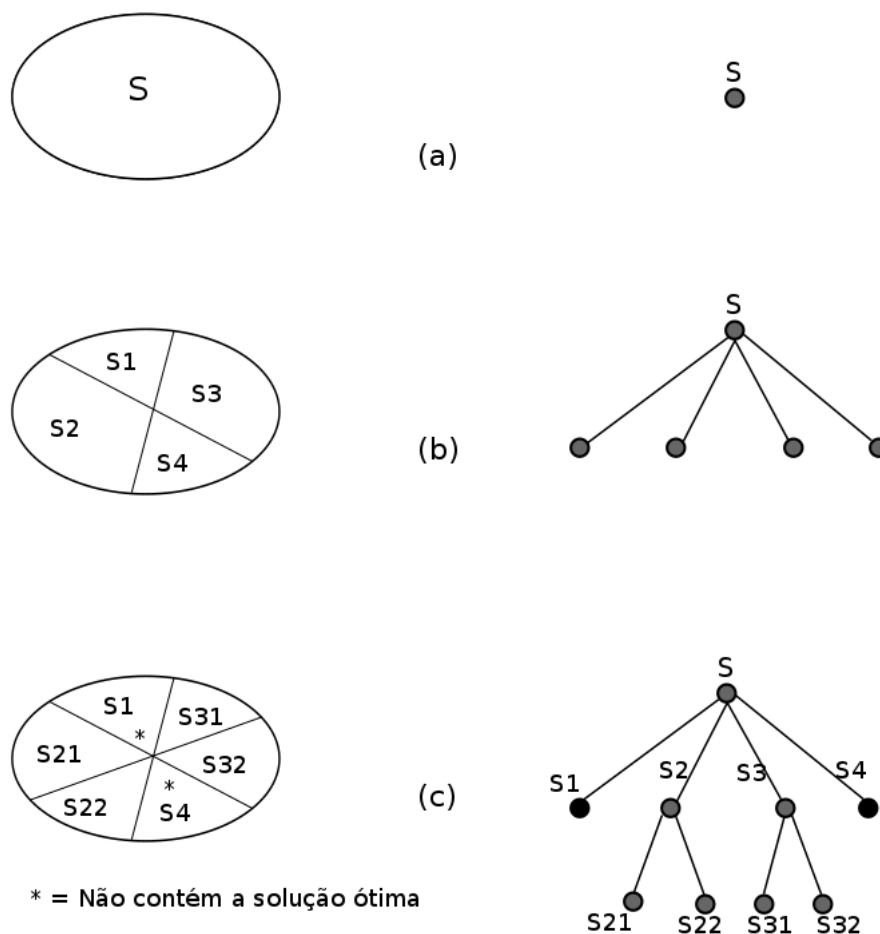


Figura 2.2: Construção do espaço de soluções do B&B

2.1.4 *Branch-and-cut*

Algoritmos de B&C combinam o método de B&B com planos de corte (Wolsey, 1998). Mais especificamente, os nós da árvore de B&B são resolvidos utilizando algoritmos de planos de corte. Vale ressaltar que a maioria dos *solvers* existentes para resolução de problemas de PLI, utilizam essa abordagem na prática.

2.2 CONCEITOS BÁSICOS E TERMINOLOGIAS MUSICAIS

Nesta seção serão expostos os principais conceitos musicais necessários para uma melhor compreensão desse trabalho. Para mais detalhes sobre o assunto consulte Lacerda (1967); Sadie e Tyrrell (2001); Randel (2003); Kennedy (2003).

2.2.1 *Tablatura*

É uma forma de notação musical muito popular para instrumentos de corda como guitarra, cavaquinho, contrabaixo, ukulele, etc. Neste tipo de diagrama existem símbolos que representam a posição, no braço do instrumento, em que as cordas devem ser pressionadas. Também é possível representar na tablatura as técnicas utilizadas nesses instrumentos (Kennedy, 2003).

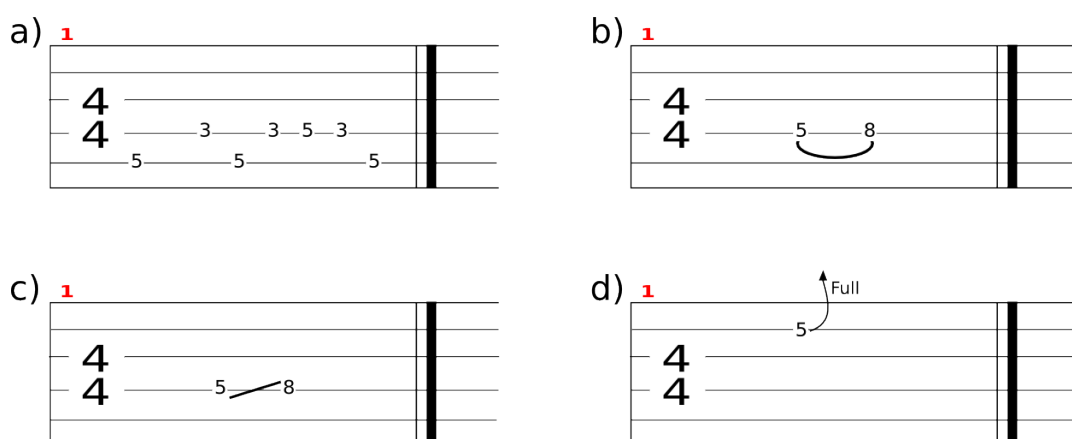


Figura 2.3: Exemplo de representação por tablaturas

A Figura 2.3 mostra exemplos de representação musical através do uso de tablaturas e também como algumas técnicas de guitarra são representadas neste modelo. Na Figura

2.3a, têm-se um exemplo simples de tablatura, mostrando qual corda e em que parte do braço do instrumento ela deve ser pressionada. Na Figura 2.3b, têm-se o exemplo de representação da técnica *hammer on*. Já na Figura 2.3c, têm-se o exemplo de representação da técnica *slide*. Finalmente, na Figura 2.3d, têm-se o exemplo de representação da técnica *bend*.

2.2.2 Compasso

O compasso é a forma de divisão da música em pequenas partes que não precisam necessariamente ter a mesma duração. A duração do compasso é definida pela fórmula do compasso, que são números que determinam a unidade de tempo e a quantidade de tempos do compasso (Sadie e Tyrrell, 2001).

É comum a fórmula do compasso aparecer no início das tablaturas indicando o tempo principal da música, também podendo aparecer em outros compassos da tablatura, indicando uma mudança no tempo da música a partir daquele momento.

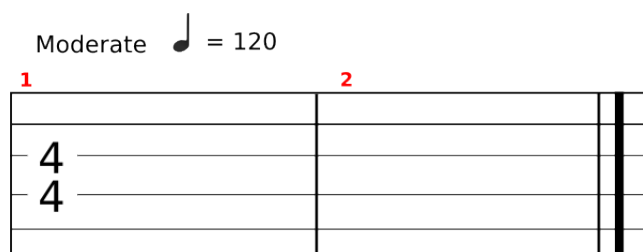


Figura 2.4: Exemplo de compassos

A Figura 2.4 mostra um exemplo de como é representado o compasso nas tablaturas. Nela existem dois compassos vazios onde a barra vertical demarca a separação entre eles. O compasso dois, por ser o último, é finalizado por duas barras sendo a segunda mais espessa que primeira. O indicador 4/4 denota a fórmula do compasso sendo que número 4 superior define que o compasso possui quatro tempos e o número 4 inferior indica a unidade de tempo, o que significa a nota que equivale a um tempo, neste caso a semínima.

2.2.3 Quiáltera

Quiáltera é o aumento ou diminuição da quantidade de notas subdividindo um batimento de tempo a partir do definido como padrão para o compasso. É geralmente identificada através da utilização de um numeral que indica a quantidade de notas dentro da batida (Sadie e Tyrrell, 2001).

As quiálteras são bastante utilizadas no *Blues* e, conseqüentemente, foram muito frequentes nos *licks* utilizados para compor o banco de *licks*, o que ocasionou uma diferença a ser considerada ao se calcular os tempos para identificar *licks* de repetição, conforme detalhado no Capítulo 4

A Figura 2.5 exemplifica a representação de quiálteras na tablatura. No primeiro compasso todas as notas estão no seu andamento normal, sendo a figura rítmica delas a colcheia, ocasionando assim duas notas por tempo. O segundo compasso, mesmo utilizando a mesma figura rítmica das notas do primeiro compasso, está subdividido em grupos de três notas, comumente chamadas de tercinas. Isso ocorre por conta da marcação da quiáltera indicada logo abaixo da figura musical que indica o tempo da nota.

Moderate ♩ = 120

1 2

4
4

3-3-3-3 3-3-3 3-3-3 3-3-3

3 3 3 3

Figura 2.5: Exemplo de quiáltera

Quiálteras não são necessariamente apenas tercinas, existem outros valores possíveis para as mesmas, como seis, sete, nove, entre outros. Porém, devido a grande quantidade de variações que se pode ter com o uso das quiálteras, neste trabalho, foram consideradas apenas as tercinas para os cálculos e identificação dos *licks* de repetição.

2.2.4 *Melodia e Harmonia*

Melodia é uma sucessão de sons de alturas tonais e valores diferentes que obedece a um sentido lógico musical. Ou seja, é uma sucessão coerente de sons que se desenvolvem em uma sequência linear, dando sentido e identidade a uma composição musical encontrando apoio na harmonia e no ritmo (Kennedy, 2003).

Harmonia trata da relação entre os sons considerando como estes soam quando tocados simultaneamente e a maneira como essa relação é organizada no tempo (Randel, 2003). Diferente da melodia, a harmonia se desenvolve verticalmente e, também pode ser considerado como sendo a combinação de notas simultâneas para a formação de acordes e a maneira de concatená-los (Sadie e Tyrrell, 2001).

2.2.5 *Altura Tonal (Pitch)*

De uma maneira objetiva, *pitch* é a propriedade do som de ser mais grave ou mais agudo, mais especificamente, é a localização do som na escala tonal, sendo determinado pela frequência da nota. Assim sendo, têm-se que as altas frequências produzem os sons mais agudos e as baixas produzem os sons mais graves (Randel, 2003).

2.2.6 *Blues*

O *Blues* é um estilo musical nascido no sul dos EUA sem uma data de origem precisa, mas consolidado mais fortemente durante as décadas de 1920 e 1930, período de atuação de grandes intérpretes como Charley Patton, Son House, Tommy Johnson e Robert Johnson (Carrijo, 2014). Seu som, em sua origem, é bastante associado a um estado de depressão e melancolia. Segundo Lewis et al (2005), *Blues* é um gênero musical que possui uma grande probabilidade de produzir uma emoção específica às pessoas familiarizadas com a cultura ocidental.

O *12-Bar Blues* (*Blues* doze compassos), em sua forma mais básica, é um modelo de *Blues* que utiliza em sua progressão três acordes principais, que são os acordes I, IV e V da escala referente a tonalidade da música. Sua progressão de acordes preenche ciclos de doze compassos, daí a origem do nome (Krenz, 2009). Um exemplo básico, em formato

de tablatura, na tonalidade de dó, de um *12-Bar Blues* é mostrado na Figura 2.6.

	I	II	III	IV	V	VI	VII	VIII
	DÓ	RE	MI	FÁ	SOL	LÁ	SI	DÓ

	I										
	C	C	C	C	C	C	C	C			
1	0	0	2	0	0	3	0	0	4	0	0
	1	1	1	1	1	1	1	1	1	1	1
4	0	0	0	0	0	0	0	0	0	0	0
4	2	2	2	2	2	2	2	2	2	2	2
	3	3	3	3	3	3	3	3	3	3	3

	IV				I						
	F	F	F	F	C	C	C	C			
5	1	1	6	1	1	7	0	0	8	0	0
	1	1	1	1	1	1	1	1	1	1	1
4	2	2	2	2	2	0	0	0	0	0	0
4	3	3	3	3	2	2	2	2	2	2	2
	3	3	3	3	3	3	3	3	3	3	3
	1	1	1	1	1	1	1	1	1	1	1

	V		IV		I						
	G	G	F	F	C	C	C	C			
9	3	3	10	1	1	11	0	0	12	0	0
	3	3	1	1	1	1	1	1	1	1	1
4	4	4	2	2	2	0	0	0	0	0	0
4	5	5	3	3	2	2	2	2	2	2	2
	5	5	3	3	3	3	3	3	3	3	3
	3	3	1	1	3	3	3	3	3	3	3

Figura 2.6: *12-Bar Blues* básico

Para este trabalho, com a finalidade de reduzir o escopo no contexto musical, foi escolhido como estilo o *Blues*, no modelo *12-Bar Blues*, devido a suas características marcantes como gênero musical.

2.2.7 Características da Guitarra

A guitarra elétrica, de corpo sólido, que é conhecida atualmente, é um instrumento já consolidado, presente em vários estilos musicais, como *Jazz*, *Blues*, *Country*, *Rock*, *Pop*, *Funk*, *Fusion*, entre outros. Sua importância e versatilidade é comprovada por sua utilização contemporânea, que vai desde a elaboração de arranjos para guitarra solo, passando por complemento instrumental na formação base de grupos musicais, chegando até a orquestras com composições e solos elaborados especificamente para o instrumento (Garcia, 2011a).

De acordo com Rocha (2005), a guitarra é um instrumento que oferece uma interface bastante direta com o músico e é capaz de criar música em várias texturas diferentes, podendo ser utilizada para desempenhar tanto papéis harmônicos como melódicos. É um instrumento que apresenta grande flexibilidade estilística e idiomática, possuindo forte expressividade devido a sua ampla gama de possibilidades de manipulação sonora e efeitos de performance.

Pode-se afirmar que, na atualidade, a guitarra elétrica, além de um instrumento musical, tornou-se um ícone da cultura *pop* em geral. Ela adquiriu grande popularidade em virtude da sua utilização por músicos de todo o mundo, sua aplicação na construção sonora de variados gêneros musicais e suas aparições em variadas propagandas de televisão, revistas, *outdoors* e catálogos de produtos distintos (Garcia, 2011a).

Dentro do contexto do *Blues*, a guitarra, inicialmente acústica, foi um dos instrumentos considerados básicos para o estilo. Por ser, à época, barato, fácil de transportar e possuir estrutura propícia a sustentação de notas, é um excelente instrumento para acompanhamentos e desenvolvimento de técnicas, como a do uso de *bottleneck*, atualmente conhecido como *slide*, técnica que é muito característica e bastante utilizada no *Blues* (da Silva, 2001).

Uma das características do *Blues* é a utilização de variações microtonais, algo que não se pode ser obtido em alguns instrumentos, como, por exemplo, o piano, e que na guitarra pode ser feito através do ato de “suspender as cordas”, técnica que é mais conhecida como *bend*. A guitarra também permite a utilização de várias afinações diferentes. Uma outra prática, bastante comum nos primórdios do *Blues*, era a utilização de afinações que formavam acordes abertos¹ (da Silva, 2001).

Dado o exposto, optou-se por trabalhar com a criação de solos de guitarra em virtude das características peculiares deste instrumento e sua importância dentro do *Blues*, através de suas aplicáveis técnicas e expressividade. Outro fator determinante para esta escolha foi o fato da pouca quantidade de trabalhos na área de CA e CAC específicos para esse instrumento.

¹Por exemplo, ao se afinar a guitarra no padrão sol aberto (D-G-D-G-B-D), ao tocar simultaneamente as cordas sem pressioná-las no braço do instrumento, caracteriza-se o acorde de sol maior.

3 REVISÃO DA LITERATURA

Este capítulo apresenta uma revisão da literatura abordando os principais trabalhos encontrados relacionados à CA e CAC.

Grande parte dos trabalhos realizados em CA faz uso de métodos baseados em computação evolutiva (CE), que são estratégias computacionais que aplicam os conceitos de evolução na criação de inteligência artificial, principalmente através do uso de algoritmos genéticos (AG). Miranda e Biles (2007) afirmam que essa área ficou conhecida como música evolutiva (ME). Uma aplicação interessante de AG na criação de melodias é o GenJam (*Genetic Jammer*), criado por Biles (1994). O GenJam é um AG desenvolvido para gerar solos de *Jazz* que, segundo seu criador, segue um modelo baseado na forma em que músicos de *Jazz* iniciantes aprendem a improvisar. Para criar a improvisação, o sistema utiliza alguns dados pré-definidos que são: tonalidade, tempo, ritmo, número de compassos e a progressão de acordes. Existe um mapeamento acorde/escala que determina as notas seguras de se usar para criar o improviso sobre determinado acorde. A avaliação do *fitness*¹ dos indivíduos gerados é feita de forma interativa, ou seja, através intervenção humana avaliando o solo criado, se este foi bom será utilizado para a criação de novos indivíduos, se foi ruim será descartado. Isso é o que o autor denomina de gargalo de *fitness* (*fitness bottleneck*).

Uma tentativa de remover o *fitness bottleneck* foi feita por Biles et al (1996) através do desenvolvimento de uma rede neural (RN) treinada para avaliar o *fitness* dos indivíduos

¹*Fitness*, ou função de aptidão, são valores usados em um processo de seleção natural para escolher quais possíveis soluções ou indivíduos vão continuar para a próxima geração e quais serão descartados (Goldberg, 1989).

gerados. Essa tentativa não foi bem sucedida, com a conclusão de que é complexa e sutil a forma que os humanos ouvem música e isso não é fácil de ser capturado por modelos estatísticos.

Biles (2001) aperfeiçoou seu algoritmo para torná-lo *fitness-free* eliminando o *fitness bottleneck*. Isso foi alcançado criando um banco de dados de *licks* de *Jazz* derivados de performance humana e evoluindo essa população usando operadores genéticos cuidadosamente selecionados. O cruzamento dessas populações gera novas melodias originais. O autor afirma que, apesar de o *fitness* não ser mais necessário, o AutoGenJam continua sendo um AG.

Também foi desenvolvida uma versão *real time* do GenJam², na qual o algoritmo interage com o músico em tempo real em um modo conhecido como *trading fours*, onde cada músico improvisa por quatro compassos alternadamente. Através do uso de uma *interface pitch-to-midi*, que converte as notas tocadas pelo músico em sinal *midi*, o sistema tornou-se capaz de, em tempo real, ler e codificar o improviso do músico e criar mutações usando o solo como base, gerando assim novos indivíduos com características próximas ao que está sendo tocado (Biles, 2002).

Outra pesquisa que envolve o uso de AG e RN para composição musical foi realizada por Chen e Miikkulainen (2001). O objetivo foi desenvolver uma RN para gerar melodias e usar um conjunto de regras de composição baseadas no estilo de um determinado compositor para avaliá-las e assim conseguir novas melodias com estilo próximo ao desse compositor, com algumas variações criativas. Para tanto, foi criada uma RN fundamentada em um modelo chamado *Simple Recurrent Network* (SRN) com um esquema de entrada e saída. A criação da melodia foi feita um compasso por vez de modo que o compasso atual era usado como entrada para a criação do próximo compasso. Para evoluir a rede criada foi utilizado o SANE (*Symbiotic, Adaptive Neuro-Evolution*) que é um AG desenvolvido por Moriarty e Miikkulainen (1996). Foram criados dois conjuntos de regras, o primeiro conjunto baseado no estilo de um determinado compositor e o segundo conjunto baseado em regras gerais de teoria musical. A cada regra foi associado um peso e cada melodia recebia uma porcentagem desse peso. Essas informações serviam como base para

²<https://www.youtube.com/watch?v=rFBhwQUZGxg>

que o AG fizesse a avaliação e evolução dos indivíduos e assim alcançando bons resultados próximos ao que eram esperados pelos autores.

Freitas (2011) também utilizou técnicas de CE como possibilidade para a CA. Em seu trabalho ele propõe um gerador melódico e um gerador harmônico utilizando uma abordagem baseada em IA através do uso de AG. Para o gerador melódico foi utilizada uma população de compassos com uma avaliação de *fitness* implícita mínima, também destacando que o processo de composição é o próprio processo evolucionário e todos os indivíduos de todas as gerações são candidatos a estarem na melodia final. Foi implementada o que o autor chamou de matriz *takeover*³ para auxiliar na avaliação dos indivíduos. O gerador harmônico foi criado para gerar de uma determinada melodia recebida como entrada, usando uma abordagem multiobjetivo com a definição das regras de harmonização divididas entre duas funções objetivo.

Bäckman (2010) se baseou em métodos de *swarm intelligence*, que é uma disciplina de IA interessada na concepção de sistemas multi-agentes inteligentes, inspirando-se no comportamento coletivo dos insetos e animais sociais (Blum e Li, 2008), juntamente com uma variação do problema do caixeiro viajante (PCV), um problema clássico de otimização bastante conhecido e estudado na literatura, para criar solos de *Jazz* improvisados. O autor considerou as notas como os pontos a serem visitados durante o *tour* do caixeiro, sendo permitido visitar uma mesma nota mais de uma vez, algumas notas podendo não ser visitadas e também sem a necessidade de retornar à primeira nota. Este solo criado através do processo descrito anteriormente serve como entrada para um algoritmo evolucionário, baseado na meta-heurística colônia de formigas (ACO, do inglês *Ant Colony Optimization*), que faz seu aprimoramento. Um detalhe desta implementação foi o uso de uma lista tabu para permitir a repetição de algumas notas de acordo com regras definidas pelo autor. Esse trabalho é parte de um projeto maior, intitulado *Evolutionary Jazz Improvisation* que foca no uso de algoritmos evolucionários com avaliação automática de *fitness* aplicados à criação de improvisos de harmonias e melodias de *Jazz* (Bäckman, 2009, 2012, 2013).

Também usando ACO como método principal, Geis e Middendorf (2007) criaram

³Em algoritmos evolucionários, a palavra *takeover* se refere ao fenômeno no qual a população se torna cópias de um ou poucos indivíduos da população (Eiben e Smith, 2003)

um algoritmo para composição de melodias e harmonias de acordo com regras do estilo barroco. O algoritmo proposto pelos autores possui dois estágios, um para criação da melodia e outro para a criação da harmonia. O que diferencia cada estágio é o grafo, função objetivo e restrições utilizadas. Para a criação das melodias os vértices do grafo representavam o conjunto de *itches* possíveis para a melodia. Transições entre vértices são possíveis em ambas as direções e, para possibilitar a repetição, cada vértice possui uma ligação unidirecional a si mesmo. Para a harmonização, cada vértice representa uma tupla ordenada de quatro notas, referente as quatro vozes (baixo, tenor, alto e soprano) que são tocadas simultaneamente.

Herremans e Sørensen (2012) desenvolveram um algoritmo baseado na meta-heurística *variable neighbourhood search* (VNS) para a composição automática de contraponto, que é a combinação de melodias ocorrendo simultaneamente de acordo com um conjunto de regras (Sadie e Tyrrell, 2001), de primeira espécie⁴. A escolha do tipo de composição foi devido ao fato de existirem regras formais, escritas, que os autores consideraram não muito difíceis de quantificar e também por ser o contraponto de primeira espécie o mais restritivo das cinco espécies. Em um trabalho seguinte Herremans e Sørensen (2013) conseguiram aprimorar seu algoritmo para gerar até a quinta espécie⁵ de contraponto. Para uma maior contextualização sobre contrapontos consulte Fux (1965).

Através do uso de *constraint programming*, um paradigma de programação que introduz técnicas para resolver problemas de satisfação de restrições (Apt, 2009), Anders (2007) desenvolveu um sistema genérico de composição musical baseado em restrições, o Strasheela. A ideia do autor foi possibilitar a implementação de um modelo formal de teoria musical como um conjunto de restrições sobre um conjunto de variáveis, criando assim o que ele chamou de *musical constraint satisfaction problem*. O sistema foi implementado em um modelo que o tornou genérico, possibilitando a criação de uma composição concentrando-se apenas na definição formal do resultado final desejado, não precisando o usuário compreender os detalhes de como isso é alcançado. Em uma pesquisa posterior, Anders e Miranda (2008) estenderam o Strasheela adicionando capacidade de composição

⁴Conhecido como nota contra nota. Consiste em uma linha melódica na qual para cada nota da melodia é realizada uma nota de contraponto.

⁵Aplicação em conjunto da primeira, segunda, terceira e quarta espécie de contraponto.

em tempo real.

Hori et al (2013) criaram uma abordagem que utiliza cadeias ocultas de markov (HMM, do inglês *Hidden Markov Model*) aplicado a criação automática de arranjos para guitarra. O arranjo é definido como o processo de encontrar um *fingerings* viável para uma determinada partitura que não é possível de tocar na guitarra por conta de limitações como *pitch range* e quantidade de notas tocadas simultaneamente. O modelo HMM foi utilizado para resolver o que é conhecido na literatura como *fingerings decision*, que é o processo de decidir qual dedo deve ser colocado sobre qual corda e casa do braço da guitarra, para cada nota, dado uma partitura de guitarra sem tablatura, sendo ele tratado como um subproblema do arranjo.

AG, juntamente com o uso de RN, também foram aplicados por Tuohy e Potter (2005, 2006) e Tuohy et al (2006) na criação de arranjos para guitarra. Os autores desenvolveram um sistema para criação de tablaturas que se preocupa em sempre mantê-las possíveis de se tocar e que também é capaz de converter uma peça musical em um arranjo para guitarra apropriado.

Hall e Smith (1996) propuseram um algoritmo para compor melodias de *Blues* para uma determinada sequência de acordes. Para isso foram usadas técnicas baseadas em cadeias de markov. Eles definiram dois estágios de trabalho. O primeiro, o estágio de análise, utiliza tabelas de transição de primeira e segunda ordem. Com o objetivo de captar a relação entre harmonia e melodia é produzido um conjunto de tabelas de transição para cada acorde da música analisada. O segundo estágio, chamado síntese, utiliza sempre que possível as tabelas de transição de segunda ordem geradas pela análise. Os resultados foram submetidos a um experimento de audição que mostrou que os ouvintes não foram capazes de distinguir com segurança as melodias geradas por computador das feitas por humanos.

Recentemente, McVicar et al (2014a,b) desenvolveram dois trabalhos focados em CAC para guitarra. O primeiro foi o AutoRhythmGuitar, que é um sistema com o objetivo de gerar o ritmo para uma dada sequência de acordes baseando-se na análise do estilo de um guitarrista específico. O segundo, o AutoLeadGuitar, tem o objetivo de criar os solos para

uma sequência de acordes e tonalidade pré-definidos, também levando em consideração o estilo de determinado guitarrista. Ambos sistemas foram “treinados” através da análise de tablaturas das músicas dos guitarristas escolhidos por meio de um *framework* probabilístico criado. O resultado é dado no formato MusicXML e pode ser visto como tablatura em diversos programas musicais existentes. Os autores fizeram uma análise qualitativa dos resultados gerados e confirmaram que as tablaturas representavam composições musicais realistas e, na maior parte, possíveis de tocar.

Uma abordagem que modela a composição como um problema de otimização e para resolvê-lo utiliza algoritmos meméticos (MA, do inglês *memetic algorithm*), que é uma técnica baseada em algoritmos evolucionários e que leva em consideração a evolução cultural, além da evolução genética, foi utilizada por Munoz et al (2014). A sua pesquisa explorou a aplicação de algoritmos meméticos em um modelo de composição conhecido como *unfigured bass*. Tal modelo tem como objetivo construir uma improvisação de acordes sobre uma linha de baixo criada previamente. O ponto principal da abordagem é a escolha dos acordes possivelmente mais “agradáveis” para a linha de baixo introduzida. Para tal, os autores atribuíram pesos a pares de acordes, para cada par de nota harmonizável da melodia do baixo, levando em consideração regras harmônicas conhecidas que poderiam indicar quais acordes seriam mais adequados e poderiam soar melhor. Essa etapa foi resolvida por meio de programação dinâmica, gerando assim a saída do algoritmo e, conseqüentemente, o arranjo para o *unfigured bass*.

Tanaka e Fujii (2015) apresentaram uma proposta com o intuito de criar composições musicais através do uso de programação inteira. Em seu estudo, eles consideram uma música como sendo sequências de padrões musicais, tais como padrões de acordes, padrões rítmicos e padrões melódicos. Como objetivo principal, foi abordado um modelo que pode dar especificações de características das estruturas musicais globais no contexto de geração de musical, podendo assim ajudar usuários a criar novas peças musicais e/ou obter as estruturas musicais desejadas. Com base nesse objetivo, eles realizaram um estudo sobre a peça musical Op. 101 No.74 de Ferdinand Beyer, relacionando como os padrões de composição nela encontrados poderiam ser representados como restrições de programação inteira. No mesmo trabalho também foi realizada a análise de padrões rítmicos sobre a

famosa melodia *Ode to Joy* de Beethoven. Entretanto, não houve uma implementação prática dos conceitos apresentados.

A Tabela 3.1 apresenta uma visão geral a respeito dos trabalhos relacionados, seus objetivos e suas respectivas abordagens.

Tabela 3.1: Trabalhos relacionados em CA e CAC

Referência	Abordagem	Tipo de composição
Biles (1994)	Algoritmos genéticos	Solos de <i>Jazz</i>
Hall e Smith (1996)	Cadeias de markov	Composição de melodias de <i>Blues</i>
Chen e Miikkulainen (2001)	Algoritmos genéticos e redes neurais	Melodias baseadas no estilo de um determinado compositor
Biles (2002)	Algoritmos genéticos	Improvisos de <i>Jazz</i> em tempo real
Tuohy e Potter (2005)	Algoritmos genéticos e redes neurais	Arranjos para guitarra
Anders (2007)	<i>Constraint programming</i>	Composição musical completa, harmonia e melodia
Geis e Middendorf (2007)	Colônia de formigas	Melodias e harmonias no estilo barroco
Bäckman (2010)	Colônia de formigas	Solos de <i>Jazz</i>
Freitas (2011)	Algoritmos genéticos	Melodias e harmonias
Herremans e Sörensen (2012)	<i>Variable neighbourhood search</i>	Composição de contraponto
Hori et al (2013)	Cadeias de markov	Arranjos para guitarra
Herremans e Sörensen (2013)	<i>Variable neighbourhood search</i>	Composição de contraponto
McVicar et al (2014b)	Baseado em um <i>framework</i> probabilístico	Guitarra rítmica baseada no estilo de um guitarrista
McVicar et al (2014a)	Baseado em um <i>framework</i> probabilístico	Guitarra solo baseada no estilo de um guitarrista
Munoz et al (2014)	Algoritmos meméticos	Modelo de composição <i>unfigured bass</i>
Tanaka e Fujii (2015)	Programação inteira	Identificação de padrões musicais globais e sua representação como restrições de programação inteira

Com base nos trabalhos pesquisados, percebe-se que existem vários autores que exploram a utilização de métodos baseados em CE, principalmente através do uso de AG, para CA e CAC. Isto porque, de acordo com Freitas (2011), o processo evolucionário compartilha similaridades com o processo criativo que ocorre em composição e a estrutura de um AG, por se basear em experiências passadas para desenvolver novas criações, se

adapta bem ao processo de composição musical. Entretanto, tais métodos são caracterizados pela necessidade de cálculo da aptidão, ou *fitness*, dos indivíduos gerados e isso muitas vezes requer intervenção humana, podendo assim criar um gargalo. Remover esse gargalo nem sempre é simples, em alguns pode ser necessário a aplicação de outros métodos em conjunto com os AG. Chen e Miikkulainen (2001), por exemplo, utilizaram RN com esse objetivo.

Contudo, outras abordagens, que não se baseiam em métodos evolucionários, foram aplicadas com sucesso na área de CAC. É o caso da composição de contrapontos, por meio da meta-heurística VNS, explorada por Herremans e Sörensen (2012, 2013), ou ainda a utilização de processos estocásticos, como a pesquisa realizada por Hall e Smith (1996) e Hori et al (2013). Estas pesquisas, juntamente com outros trabalhos recentes como os de McVicar et al (2014a,b) e Tanaka e Fujii (2015), se mostram como boas alternativas à utilização de CE para a CA.

Apesar de Papadopoulos e Wiggins (1999) e Gonçalves et al (2009) citarem modelos matemáticos entre os principais métodos para a CA, esta é a estratégia aplicada com menor frequência, de acordo com o que foi encontrado na literatura, durante a pesquisa por trabalhos relacionados. Os autores que fazem uso desta abordagem geralmente baseiam-se em métodos estatísticos como principal fator para criação das composições. Nesse contexto, existe um caminho pouco explorado para a CAC, que é utilização de técnicas de otimização baseadas em PLI, de modo que foi encontrado apenas a pesquisa de Tanaka e Fujii (2015) como referência. Assim sendo, essa é a abordagem utilizada nesta pesquisa, procurando gerar solos de guitarra *Blues* via PLI.

4 ASPECTOS METODOLÓGICOS

Neste capítulo será apresentada a metodologia adotada para a resolução do problema, abordando todos os procedimentos e atividades que foram realizadas durante a pesquisa para obtenção dos resultados.

4.1 METODOLOGIA PARA RESOLUÇÃO DO PROBLEMA

A metodologia aqui empregada consiste em considerar a criação de solos como um problema de otimização combinatória. No caso, tal problema pode ser resolvido por meio de um método exato, tendo como base os conceitos de programação linear inteira apresentados no Capítulo 2.

Neste trabalho o objetivo é gerar soluções, nesse caso os solos, que possam ser considerados bons, do ponto de vista musical. Sendo assim, se faz necessário especificar o que é esperado dos solos gerados. Toda a definição do que é considerado como “qualidade” para este trabalho é contextualizada na Subseção 4.1.2.

4.1.1 *Definição do Estilo Musical*

Vários trabalhos na área de CA e CAC, com o objetivo de reduzir o escopo, selecionam um estilo musical e concentram os esforços de desenvolvimento dos algoritmos de composição nas características do estilo escolhido.

Biles (1994), Feijão (2004) e Bäckman (2010), por exemplo, escolheram o *Jazz* como estilo para a criação das composições. Herremans e Sørensen (2013, 2012), Geis

e Middendorf (2007) e Menezes (2009) escolheram trabalhar regras de composições mais clássicas, como fugas e contrapontos. Desta forma, com os mesmos fins de redução de escopo, para este trabalho foi escolhido criar solos no estilo *Blues* utilizando uma harmonia de um modelo de *12-Bar Blues* como base.

O resultado esperado é que os solos criados remetam a um estilo mais clássico do *Blues*, onde pode-se citar como referências guitarristas como Eric Clapton, B. B. King, Johnny Winter, Muddy Waters, entre outros. Para tanto, foram selecionados para o banco apenas os *licks* com as características do *Blues* desse estilo, bem como podem eventualmente aparecer nos solos gerados *licks* de músicas dos guitarristas citados.

4.1.2 Definição de Qualidade dos Solos

Definir a qualidade de uma música possui um certo grau de subjetividade, geralmente influenciado por diversos fatores, algumas deles pessoais, como preferências e experiências musicais de quem está ouvindo. Desta forma, uma etapa essencial da pesquisa foi definir os critérios que caracterizam os solos criados como sendo boas soluções dentro do escopo tratado.

Dada a principal abordagem adotada para a criação dos solos, que é através do sequenciamento de subconjuntos de *licks*, as boas soluções são aquelas que mais se mantêm fiéis às regras definidas como fatores que ditam o que é uma boa transição entre dois *licks*. Estas regras foram inferidas empiricamente, após a audição e análise de uma grande quantidade de solos de *Blues* em busca de características em comum e que pudessem ser representadas na forma de regra.

Esta fase de audição e análise de solos foi dividida em duas etapas. No primeiro momento foram ouvidos solos de canções e repertórios de músicos referências do *Blues*, como os citados na Seção 4.1.1. Também foram absorvidos conceitos de livros, vídeos e vídeo-aulas a respeito de improvisação e outros contextos relacionados a música e, principalmente, ao *Blues*.

A segunda etapa consistiu em, já utilizando uma pequena amostragem de *licks* que compuseram o banco de dados, criar solos usando um sequenciamento aleatório, ouví-

los e verificar a existência das características observadas durante a primeira etapa. Em alguns casos foi realizada a reorganização manual da ordem dos *licks* para verificar se isso realmente implicava em diferenças significativas.

4.1.3 Regras para Criação dos Solos e Matriz de Transições

Para a definição das regras de criação dos solos, foi adotada uma abordagem minimalista com a programação de poucas regras, porém que cobrissem o que foi catalogado durante as fases de audição e análise de solos. A ideia é evitar que a utilização de muitas regras faça com que os solos criados deixem de pertencer a um conjunto mais generalizado e passem a ser de gosto mais específico ou mais pessoal.

O conjunto de regras implementadas pode ser dividido em dois grupos: individuais e de transição. As regras individuais dizem respeito àquelas que consideram os *licks* individualmente e isso possibilitou que elas fossem programadas diretamente como restrições no modelo matemático, explicado com detalhes na Subseção 4.1.6. As regras individuais implementadas foram as mostradas na Tabela 4.1.

Tabela 4.1: Regras individuais implementadas

Regra	Descrição	Quantidade
1	Utilização de <i>licks</i> do tipo <i>turnaround</i>	= 1
2	Utilização de <i>licks</i> de repetição	= 1
3	Utilização de <i>licks</i> iniciados e/ou terminados em pausa	≤ 3

Os valores referentes as quantidades, nas regras individuais 2 e 3, podem ser alterados através da passagem de parâmetros na inicialização do algoritmo. Essa funcionalidade visa possibilitar uma flexibilização para definir as preferências do usuário do *framework*.

As regras de transição são aquelas que consideram sempre pares de *licks*, bonificando ou penalizando a ocorrência do *lick* B após o *lick* A, levando em conta características importantes de ambos. Tais regras foram aplicadas na criação, em tempo de execução, da matriz de penalidades de um subconjunto *licks* sorteado do banco de dados. Para a criação da matriz de transições, bonificar significa diminuir o custo de transição e penalizar significa aumentar o custo de transição. As regras de transição implementadas, com seus

respectivos bônus e penalizações, são mostradas na Tabela 4.2:

Tabela 4.2: Regras de transição implementadas

Regra	Transição	Tipo	Valor
1	De qualquer <i>lick</i> para um <i>lick</i> de repetição	bônus	-50
2	De <i>lick</i> de repetição para qualquer outro <i>lick</i>	bônus	-50
3	De <i>lick</i> não terminado em pausa para qualquer outro iniciado em pausa \leq a 1 tempo	bônus	-15
4	De <i>lick</i> terminado em pausa \leq a 1 tempo para qualquer outro não iniciado em pausa	bônus	-15
5	De qualquer <i>lick</i> A para qualquer <i>lick</i> B onde a primeira nota de B seja a imediata anterior ou posterior a última nota de A na escala pentatônica (maior ou menor)	bônus	-15
6	De qualquer <i>lick</i> A para qualquer <i>lick</i> B onde a primeira nota de B seja igual a última nota de A	bônus	-15
7	De qualquer <i>lick</i> que termine em pausa $>$ 1 tempo para outro que comece com pausa $>$ 1 tempo	penalidade	+25
8	De qualquer <i>lick</i> que termine em pausa $>$ 2 tempos para qualquer outro	penalidade	+25
9	De qualquer <i>lick</i> para outro que iniciado em pausa $>$ 2 tempos	penalidade	+25
10	De qualquer <i>lick</i> para qualquer outro do tipo <i>turnaround</i>	penalidade	+100000

Todas as transições da matriz são inicializadas com o valor 100. Em seguida, é realizada uma varredura completa dos *licks*, em pares, e verificado se a transição entre eles está de acordo com alguma das regras, caso sim, é subtraída sua bonificação ou adicionada sua penalidade conforme os valores listados na Tabela 4.2. Estes valores foram definidos após vários testes com diferentes custos e audições dos solos gerados, de modo que a seleção dos *licks* não fique tendenciosa, existindo assim um equilíbrio entre os custos das transições.

As regras 1 e 2 tratam de bonificar a ocorrência de *licks* de repetição, pois de acordo com o que está explicado na Seção 4.1.5, eles sempre aparecem como uma boa alternativa em, praticamente, qualquer momento do solo.

As regras 3 e 4 têm a função de bonificar pausas curtas entre as transições. A pausa se mostrou uma característica bastante presente em solos e improvisos de *Blues* e sua utilização causa, geralmente, uma boa impressão de mudança de contexto no decorrer da melodia.

As regras 5 e 6 são mais simples e básicas de improvisação, principalmente para

iniciantes. Fritts (2012) afirma que essa prática é sempre uma boa alternativa para se começar a fazer seus improvisos ou solos, dentro deste contexto de junção de *licks*.

As regras 7, 8 e 9 têm o objetivo de evitar a ocorrência de pausas longas no decorrer do solo. Como dito anteriormente, pausas curtas podem causar uma boa sensação de mudança de contexto na melodia, porém, se as pausas acontecerem de forma prolongada podem causar uma sensação de frustração ou expectativa que não se mostram tão interessantes na maioria dos casos.

Em relação a regra 10, fez-se necessário colocar um valor alto para a transição de *licks turnaround* para qualquer *lick* de modo a evitar que os mesmos fossem selecionados no início ou meio dos solos, visto que eles devem obrigatoriamente aparecer apenas compondo os dois últimos compassos de cada ciclo de 12 compassos.

4.1.4 Banco de Dados de Licks

O uso de *licks*, baseado na ideia de *riffology*, explicada no Capítulo 1, consistiu, principalmente, em tirar proveito do fato de, por representar um pequeno bloco melódico, nele já estarem encapsuladas algumas das principais técnicas aplicadas à guitarra como, por exemplo, *bend*, *slide*, *hammer on* e *pull off*.

Pensando desta maneira, em um bloco melódico completo, a manipulação dos arquivos dos *licks* torna-se mais simplificada. Tais arquivos estão no formato MusicXML¹, uma especificação baseada em XML² para a marcação de notação musical de forma a possibilitar a leitura dos arquivos por diferentes aplicações.

Os *licks* de guitarra que popularam o banco de dados foram obtidos da Internet, de livros (sendo Marshall (1999) e Fritts (2012) as principais referências) e vídeos e, na maioria das vezes, tiveram que ser transcritos e/ou convertidos para MusicXML manualmente através da utilização de softwares específicos de terceiros.

Vale ressaltar que na maior parte das vezes os *licks* encontrados possuíam entre 3 e 5 compassos. Dado que a abordagem utilizada nesta pesquisa trabalha com a otimização

¹<http://www.musicxml.com/>

²É um conjunto de regras para codificação de documentos. Muito utilizado para armazenamento e transferência de dados.

de ciclos de 12 compassos, não seria interessante a utilização de frases de guitarra maiores que 1 compasso. Sendo assim, estes *licks* grandes foram divididos em vários *licks* criando, portanto, uma maior possibilidade de variação dos resultados.

Com um grande número de *licks* no banco de dados e com métodos de sorteio de subconjuntos de *licks* pode-se reduzir a possibilidade da geração de solos repetidos, diminuindo o grau de determinismo do *framework* proposto, descrito na Seção 4.1.7.

No banco de dados criado, os *licks* estão organizados de acordo com a velocidade do *Blues* ao qual eles são mais adequados. Até o presente momento, compõem o banco de dados 215 *licks* de *moderate Blues*, 50 *licks* de *slow Blues*, 65 *licks* de *fast Blues* e 12 *licks turnaround* que se aplicam a qualquer um dos tipos citados.

4.1.5 *Licks de Repetição*

Uma das convenções mais originais na música é a rápida repetição de figuras melódicas encontradas, principalmente, no *Blues* e em outros estilos influenciados por ele, como algumas variações do *Rock*, por exemplo. Essas figuras repetidas são uma maneira bastante efetiva de construir e manter momentos importantes e/ou de grande intensidade na música (Fritts, 2012). Aqui, neste trabalho, esses *licks*, que possuem tal característica, serão chamados de *licks* de repetição.

Foi percebido que os *licks* de repetição podem ser aplicados como uma espécie de “coringa”, o qual sua ocorrência em praticamente qualquer momento do solo é sempre uma boa alternativa.

São considerados como *licks* de repetição aqueles em que, dividindo o *lick* em 4 partes, A, B, C e D, têm-se que:

- Caso 1 — A, B, C e D são iguais.
- Caso 2 — A, B, e C são iguais.
- Caso 3 — B, C e D são iguais.
- Caso 4 — A é igual C e B é igual a D.

A Figura 4.1 exemplifica os quatro casos de *licks* de repetição considerados neste

trabalho. As linhas tracejadas subdividem cada quarto dos *licks*. A Figura 4.1a demonstra um *lick* em que todos os quartos são iguais. Já na Figura 4.1b o *lick* se encaixa no Caso 2 onde apenas as partes A, B e C são iguais. A Figura 4.1c é um exemplo do Caso 3, em que apenas as partes B, C e D são iguais. Por último, a Figura 4.1d mostra um *lick* de repetição que se encaixa no Caso 4, onde a parte A é igual C e a parte B é igual a parte D.

Figure 4.1 shows four guitar licks (a, b, c, d) on a four-string staff, each divided into four quarters (A, B, C, D) by dashed lines. Each lick starts with a red '1' and has a '4' on the second and third lines. Lick a) has identical patterns in all quarters: 11-8-8. Lick b) has identical patterns in quarters A, B, and C (11-8-8), but quarter D is 8-11-10. Lick c) has identical patterns in quarters B, C, and D (11-8-8), but quarter A is 8-11-10. Lick d) has identical patterns in quarters A and C (8-10-12), and identical patterns in quarters B and D (8-10-8).

Figura 4.1: *Licks* de repetição

O primeiro passo no processo de identificação de um *lick* de repetição é verificar se o mesmo pode ser dividido em quatro quartos de tempo. Para essa identificação foi primeiramente necessário o mapeamento do valor da duração para cada nota ou figura rítmica, isto é mostrado na Figura 4.2.

Tomando como exemplo um compasso 4/4, têm-se que uma semibreve preenche

os quatro tempos, ou quatro batidas, do compasso. Já a mínima preenche dois tempos, sendo assim, são necessárias duas mínimas para o preenchimento do compasso. No caso da semínima seriam necessárias quatro, uma vez que sua duração equivale a uma batida de tempo. O mesmo raciocínio é seguido para as demais figuras.








						
Semibreve	Mínima	Semínima	Colcheia	Semicolcheia	Fusa	Semifusa
1	2	4	8	16	32	64

Figura 4.2: Figuras rítmicas e suas respectivas durações

Com base nos valores mostrados na Figura 4.2 chega-se ao quarto de compasso quando a soma dos valores das notas é igual a 1, sendo o valor de cada nota dentro da batida pode ser calculado por $x = 4(\frac{1}{valordanota})$, isto implica em dizer que o compasso está completo quando a soma dos valores dos quartos do compasso é igual a 4.

Entretanto, se no *lick* existirem quiálteras, o cálculo do valor da nota dentro da batida de tempo muda, como explicado no Capítulo 2. Ressaltando, novamente, que foram consideradas apenas as tercinas por conta da quantidade de possibilidades que as outras variações das quiálteras podem ocasionar.

Dessa forma, o valor de uma nota dentro de uma batida alterada por uma quiáltera é igual a quatro terços do seu valor normal, $qNota = valordanota(\frac{4}{3})$ e, com isso, o cálculo do valor da nota dentro da batida é dado por $x = 4(\frac{1}{qNota})$. A soma dos valores da nota dentro da batida tem que ser igual 1,125 para se chegar ao quarto de compasso.

O Algoritmo 1 mostra o pseudocódigo da função que reconhece se o *lick* é ou não de repetição. A função *lickDivisivel(lick)* verifica se o *lick* pode ser subdividido em quatro partes. Para tal, é necessário identificar se nele existe uma nota com duração maior que uma semínima. Por exemplo, se no *lick* contêm uma mínima, esta nota têm duração de duas batidas de tempo num compasso 4/4, portanto, não seria possível subdividir o *lick* em quatro partes. A função *testaQuialtera(nota)* confere se a nota corrente está marcada como quiáltera, visto que, como explicado anteriormente, essa marcação

resulta em uma mudança no cálculo do tempo das notas do compasso. Por fim, a função *calculaTempoQuarto(quarter)* retorna a duração do quarto de tempo para saber se o mesmo está preenchido e pode ser adicionado a estrutura que armazena os quartos do compasso.

Algoritmo 1 Identificação de *Licks* de Repetição

```

1: função LICKDEREPETICAO(lick)
2:   quarter ← ∅ ▷ Estrutura utilizada para armazenar 1/4 de um lick
3:   quarterlist ← ∅ ▷ Estrutura utilizada para armazenar os itens do tipo quarter
4:   se LICKDIVISIVEL(lick) então ▷ Testa se o lick pode ser dividido em 4
5:     para cada nota in lick faça
6:       quarter ← nota
7:       se TESTAQUIALTERA(nota) então ▷ Verifica se existe de quiáltera na nota
8:         se CALCULATempoQUARTO(quarter) = 1,125 então
9:           quarterlist ← quarter
10:          quarter ← ∅ ▷ Esvazio o quarter
11:        fim se
12:      senão se CALCULATempoQUARTO(quarter) = 1,0 então
13:        quarterlist ← quarter
14:        quarter ← ∅ ▷ Esvazio o quarter
15:      fim se
16:    fim para
17:    se (quarterlist[0] = quarterlist[1] = quarterlist[2]) ou (quarterlist[1] = quarterlist[2] =
18:      quarterlist[3]) então
19:      retorna verdadeiro
20:    senão se (quarterlist[0] = quarterlist[2]) e (quarterlist[1] = quarterlist[3]) então
21:      retorna verdadeiro
22:    senão
23:      retorna falso
24:    fim se
25:  retorna falso
26: fim função

```

4.1.6 Formulando o Problema de Geração de Solos Como Um Problema de Otimização Combinatória

Como já explicado anteriormente neste capítulo, a criação de solos de guitarra foi tratada e modelada como um problema de otimização combinatória o qual foi desenvolvido um modelo matemático para sua resolução. Um dos benefícios da utilização de um modelo matemático é a possibilidade de programação em forma de restrições de algumas regras referentes aos *licks*. Estas foram as chamadas regras individuais, explicadas na Seção 4.1.3. Isto torna mais leve e, conseqüentemente, mais rápida a fase de pré-processamento do *framework* desenvolvido, pois isto implica em menos regras à serem analisadas em tempo de execução do algoritmo, durante a fase que é criada a matriz de transição.

O problema pode ser definido sobre um grafo $G = (L, A)$, em que L representa o conjunto de *licks* e A o conjunto de arcos conectando dois *licks*, tal que $A = \{(i, j) \mid i \neq j\}$. Seja p_{ij} a penalidade de transição entre o *lick* i e o *lick* j , c_i a quantidade de compassos do *lick* i , NC uma constante que define a quantidade de compassos disponíveis para o solo, LR uma constante que define a quantidade de *licks* de repetição que podem ser selecionados para o solo, LP uma constante que define a quantidade de *licks* que iniciam e/ou terminam em pausa que também podem ser selecionados para o solo, R o conjunto de *licks* de repetição, T o conjunto de *licks turnarounds*, P o conjunto de *licks* que se iniciam e/ou terminam em pausa. Defina x_{ij} como sendo uma variável binária que assume valor 1 se o arco $(i, j) \in A$ for utilizado e 0, caso contrário e y_i como sendo uma variável auxiliar que assume o valor 1 caso o *lick* $i \in L$ tenha sido utilizado e 0, caso contrário.

A formulação matemática proposta para o problema é a seguinte:

$$\min \sum_{i \in L} \sum_{j \in L} p_{ij} x_{ij} \quad (4.1)$$

$$\text{s.a.} \quad \sum_{j \in L} x_{ij} = y_i \quad \forall i \in L \quad (4.2)$$

$$\sum_{i \in L} x_{ij} = y_j \quad \forall j \in L \quad (4.3)$$

$$\sum_{i \in L} c_i y_i = NC \quad (4.4)$$

$$\sum_{i \in R} y_i \leq LR \quad (4.5)$$

$$\sum_{i \in P} y_i \leq LP \quad (4.6)$$

$$\sum_{i \in T} x_{i0} = 1 \quad (4.7)$$

$$x_{ij} + x_{ji} \leq 1 \quad \forall i \in L, \forall j \in L, i < j \quad (4.8)$$

$$\sum_{i \in L} \sum_{j \in \bar{L}} x_{ij} \geq 1 \quad \forall L \subseteq V', |S| > 2 \quad (4.9)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in L, \forall j \in L \quad (4.10)$$

$$y_i \in \{0, 1\} \quad \forall i \in L \quad (4.11)$$

A função objetivo (4.1) minimiza o custo de transição entre os *licks*. As Restrições (4.2) e (4.3) estabelecem, respectivamente, que caso o *lick* tenha sido utilizado, nele chega apenas um arco e dele sai apenas um arco. A Restrição (4.4) define que o total da soma dos compassos dos *licks* utilizados tem que ser igual a quantidade de compassos disponíveis para o solo. A Restrição (4.5) assegura que o solo conterà, no máximo, LR *licks* de repetição. A restrição (4.6) garante que no existirão máximo, LP *licks* que iniciam e/ou finalizam em pausa. A Restrição (4.7) estabelece a obrigatoriedade de utilização de um *lick* do tipo *turnaround*. As Restrições (4.8) e (4.9) eliminam a ocorrência de *subtours*. Como existe um número exponencial de Restrições (4.9), elas são adicionadas sob demanda através de um procedimento de geração de cortes. Por fim, (4.10) e (4.11) definem a natureza das variáveis.

Vale ressaltar que o problema aqui definido pode ser visto como sendo um Problema de Caminho Mínimo com Restrição de Recursos (SPPRC, do inglês *Shortest Path Problem with Resource Constraints*) (Irnich et al, 2005), considerado fortemente \mathcal{NP} -difícil.

A visualização da geração de solos como um SPPRC está ilustrada na Figura 4.3. Neste caso, têm-se que os nós representam os *licks* e os compassos referem-se a janela de tempo na qual os *licks* podem ser adicionados a solução. O *lick* “d” equivale a um *lick dummy*, que é utilizado como ponto de partida e de finalização do solo. Os nós com preenchimento listrado são representações de *licks* de repetição, que, assim como os *licks* comuns, podem ser selecionados em qualquer momento do solo. Os nós de cor cinza representam os *licks turnaround*, sendo que eles preenchem de 2 compassos e só podem ocorrer no final de cada ciclo de 12 compassos, ocupando os dois últimos compassos dos solos. O caminho encontrado como solução é ilustrado pelos nós preenchidos em preto, conectados pelos arcos que contêm os respectivos custos da penalidade de transição entre os *licks*.

A formulação matemática, bem como o procedimento de geração de cortes de *subtour*, foram implementados na linguagem programação C++ utilizando para sua resolução a biblioteca *concert* do *solver* CPLEX versão 12.4, que por sua vez faz uso de um algoritmo de B&C para resolução de problemas de programação linear inteira.

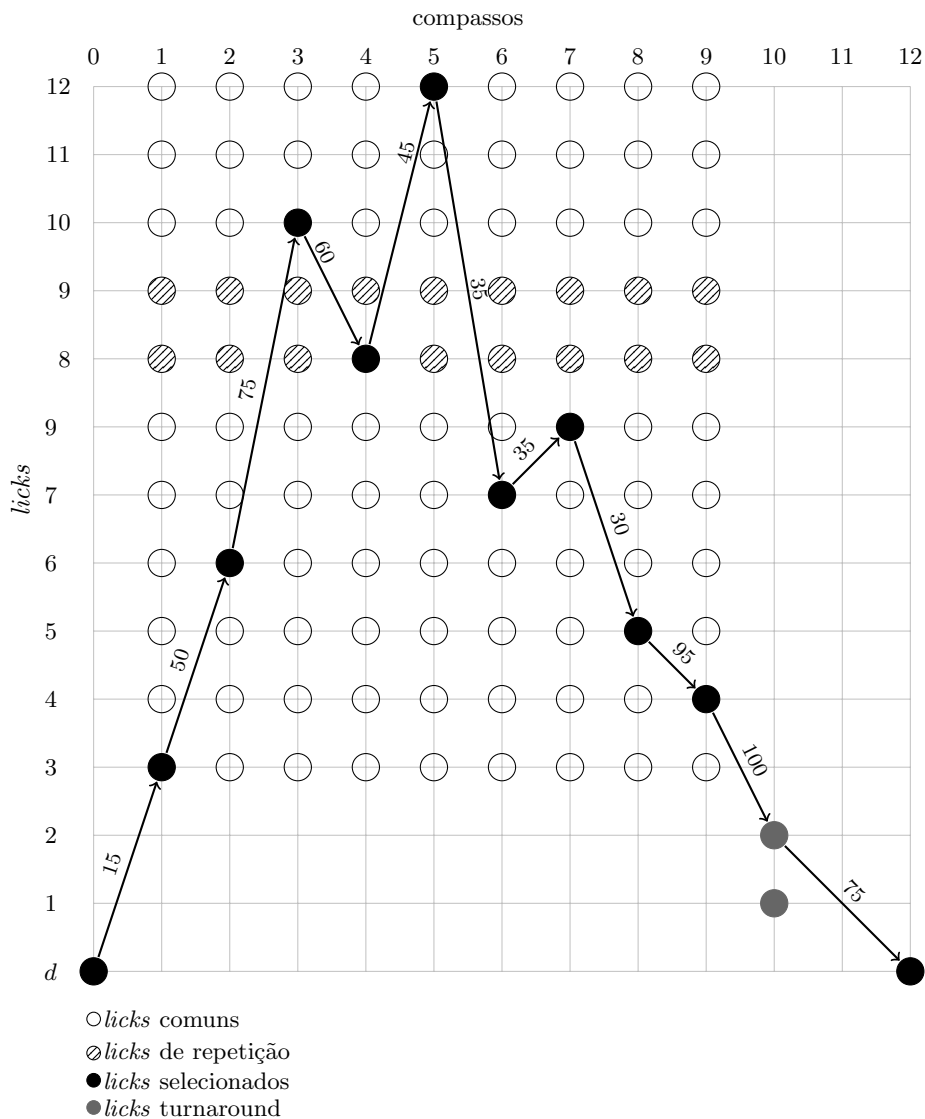


Figura 4.3: Visualização da geração de solos como variante do SPPRC

4.1.7 Framework Proposto

O *framework* proposto para a manipulação dos arquivos MusicXML divide em três etapas o processo necessário para a criação de solos de guitarra, a saber: pré-processamento, otimização e pós-processamento. A visão completa da arquitetura do *framework* é exibida na Figura 4.4.

Na Etapa 1 são recebidas como entrada informações referentes ao tamanho da instância que será sorteada, velocidade da música e a quantidade de compassos disponíveis para o solo. Na Etapa 2 ocorre o sorteio do subconjunto de *licks*, de acordo com o parâmetro definido na inicialização. O sistema coleta aleatoriamente do banco de dados

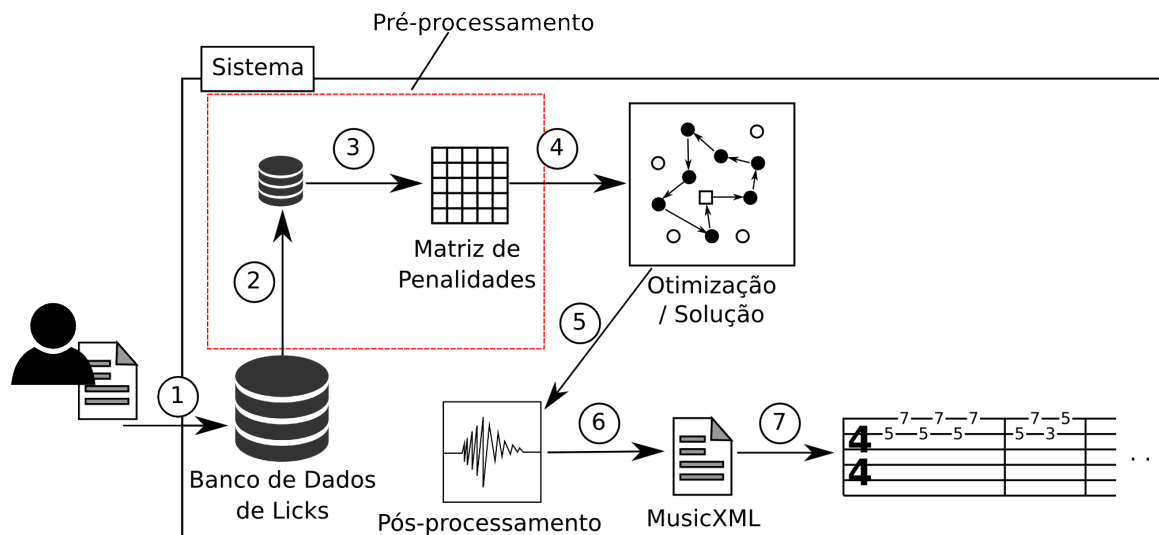


Figura 4.4: Fluxograma de funcionamento do sistema

a quantidade de *licks* que serão candidatos a estarem na solução. Na Etapa 3 é gerada a matriz de penalidades referente ao subconjunto de *licks* sorteados. A Etapa 4 é onde o otimizador, fazendo uso da matriz de penalidade e das informações recebidas no primeiro passo, gera a solução encontrada para aquela instância dos *licks* escolhidos na Etapa 2. Na Etapa 5, já com os valores da solução, opcionalmente pode ser feito o pós-processamento e refinamento do solo, ajustando e removendo as tensões indesejadas. Na Etapa 6, é exportada a saída da solução no formato MusicXML e, finalmente, a Etapa 7 a ilustra no formato de tablatura.

4.1.7.1 Pré-processamento

A fase de pré-processamento acontece da seguinte forma. O algoritmo é iniciado e recebe como parâmetros de entrada o tamanho do subconjunto de *licks* que será sorteado do banco de dados, a quantidade de compassos disponíveis para o solo que será gerado como saída, o tipo de harmonia com base na velocidade, podendo ser *slow Blues* (60 BPM), *moderate Blues* (100 BPM) ou *fast Blues* (150 BPM) e a opção de usar ou não o refinamento que acontece na fase de pós-processamento, explicado na Seção 4.1.7.3.

Com o valor da quantidade de *licks* que serão sorteados a partir do banco de dados completo, define-se de forma randômica o subconjunto que será a instância utilizada para a criação da matriz de penalidades de transição que posteriormente é passada para a

formulação matemática.

A matriz de penalidades de transição é construída através da comparação entre pares de *licks*, realizando a leitura das suas características e aplicando as bonificações e penalidades de acordo com as regras de transição explicadas na Seção 4.1.3. Em seguida, acontece a chamada ao otimizador e têm-se início a fase de otimização.

4.1.7.2 Otimização

É na etapa de otimização onde acontece, de fato, a resolução do problema. Nesse momento o algoritmo faz o carregamento dos parâmetros adicionais, onde estão presentes as informações referentes aos *licks* de repetição, aos *licks* com pausa no início e/ou no final e aos *licks turnaround*. Todos esses dados, juntamente com a matriz de transição, são utilizados para resolução do modelo matemático por meio de um *solver* que usa um algoritmo B&C. Esse, por sua vez, retorna o melhor caminho encontrado com base nas penalidades e restrições definidas. Os dados retornados são armazenados e trabalhados na fase de pós-processamento e criação do arquivo do solo.

4.1.7.3 Pós-processamento

Uma vez que uma solução para a instância selecionada é encontrada, pode ser realizado um pós-processamento do solo, com o objetivo de evitar o acontecimento de grandes dissonâncias ou tensões que possam causar a impressão de que o solo está numa tonalidade incorreta ou fazendo uso de notas inadequadas em algum determinado momento.

Nesta fase já é possível saber o acorde da harmonia em que o *lick* está ocorrendo. Entretanto, a harmonia aplicada se baseia em acordes dominantes, que são aqueles em que existe um trítono³ entre as notas terça maior e a sétima menor (Sadie e Tyrrell, 2001). A utilização desse tipo de acorde geralmente proporciona uma liberdade maior de improvisação, pois já existe neles uma tensão característica, mas, ainda assim, pode acontecer de algumas notas causarem uma sensação de tensão ainda maior, dependendo de sua duração dentro do contexto do solo.

³É o intervalo de três tons inteiros entre duas notas.

Diante disso, o algoritmo de pós-processamento foi adicionado justamente para tentar reduzir a possibilidade a ocorrência dessas notas e, conseqüentemente, reduzir os momentos de tensão não desejados no solo. Este algoritmo foi adicionado como uma função opcional que pode ser ativada através de parâmetros na inicialização do *framework*.

Apenas duas regras foram implementadas para o refinamento. A primeira implica em verificar no *lick* a existência da nota que corresponde a sétima maior do acorde da harmonia. Caso exista, se a duração da nota for igual ou maior que uma batida de tempo, é acrescentado a ela meio tom, transformando-a assim, na tônica. O mesma ideia é aplicada a nota que corresponde a segunda menor, com a diferença que é nela é feito o decréscimo de um semitom para que ela também seja transposta para a tônica do acorde.

A Figura 4.5 ilustra o primeiro caso citado. Nela, o *lick* ocorre sob o acorde de C7 (dó com sétima). No primeiro *lick* a última nota, a que está circulada, é um B (si), o que equivale a sétima maior da escala de dó maior. A sua duração equivale a uma batida de tempo completa, pois sua figura rítmica é uma semínima e, conforme explicado anteriormente, este tipo de tensão está sendo evitado. Desta forma, a solução encontrada no algoritmo de pós-processamento foi a de fazer a transposição da nota, aumentando um semitom transformando-a assim na nota equivalente a tônica, neste caso um C (dó). O *lick* que resulta da transposição é o que está posicionado na parte de baixo da figura.

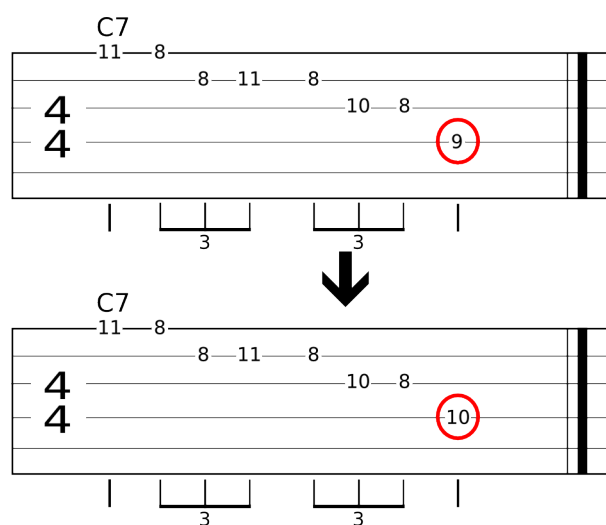


Figura 4.5: Ilustração do pós-processamento. Caso 1

Já na Figura 4.6, observa-se a ilustração do segundo caso citado. Nela, o *lick*

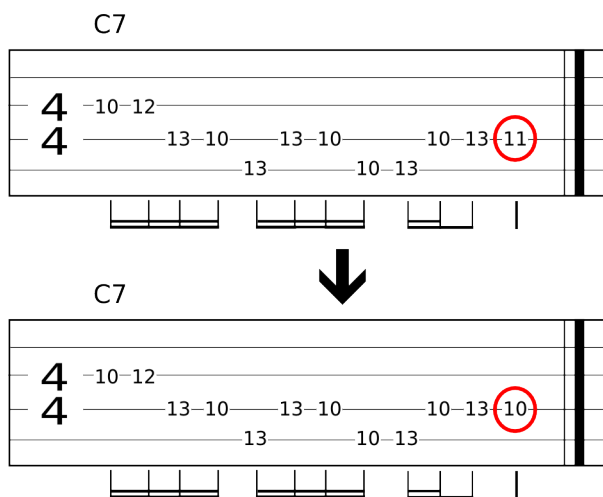


Figura 4.6: Ilustração do pós-processamento. Caso 2

também ocorre sob o acorde de C7 (dó com sétima). No primeiro *lick* a última nota, que está circulada, é um C# (dó sustenido), o que equivale a segunda menor na escala de dó maior. A sua figura rítmica é a semínima, o que implica em dizer que sua duração equivale a uma batida de tempo completa e, conforme explicado anteriormente, esse é outro tipo de tensão que também deseja-se evitar. Desta forma, a solução encontrada no algoritmo de pós-processamento é fazer a transposição da nota, reduzindo um semitom transformando-a assim na nota equivalente a tônica, neste caso um C (dó). O *lick* que resulta da transposição é o que está posicionado na parte de baixo da figura.

4.2 ESTRATÉGIA PARA AVALIAÇÃO DOS SOLOS

Percebe-se, com base na literatura, que as abordagens adotadas para a avaliação dos resultados nas pesquisas na área de CAC são bastante distintas, dependendo principalmente do objetivo do trabalho. Hall e Smith (1996), que desenvolveram uma abordagem para composição de melodias de *Blues*, submeteram seus resultados a um teste de audições em que, no fim, os ouvintes não conseguiam diferenciar entre uma composição gerada pelo sistema e uma criada por humanos. Bäckman (2009) submeteu os resultados de seu trabalho, que consistiu em uma abordagem evolutiva para criação de harmonias de *Jazz*, a um teste de *performance*, em que uma banda de *Jazz* tentava tocar as harmonias criadas.

Como principal objetivo da avaliação dos resultados deste trabalho, pretende-se

mostrar que os solos gerados através da otimização da sequência de *licks* soam mais interessantes que uma sequência aleatória e também avaliar a possível aceitação de uso dos solos, seja na criação de novas ideias e improvisos ou no seu uso em composições musicais.

Desta forma, a estratégia adotada para a avaliação dos solos foi submetê-los a um teste de audições, em que o ouvinte classifica o quanto gostou, ou acha que o solo está bom, usando o modelo de escala de Likert. Esse é um modelo de escala muito utilizado por gerentes e pesquisadores, principalmente da área de *marketing*, com o objetivo de medir o grau de discordância ou concordância em relação a um determinado contexto (Albaum, 1997).

A abordagem completa, para possibilitar a avaliação dos resultados, aconteceu da seguinte forma: foi desenvolvido um sistema em que músicos e/ou entusiastas pudessem fazer a audição dos solos via Internet; foram inseridos no sistema solos criados aleatoriamente e solos criados pelo *framework*; cada usuário cadastrado para votação recebeu seis solos para avaliar, sendo três aleatórios e três não-aleatórios, porém o ouvinte não era informado sobre isso; após um solo ser avaliado o próximo é selecionado randomicamente, podendo ser escolhido de qualquer um dos dois tipos, contanto que ainda não tenha recebido nenhuma classificação; por fim, o usuário responde sim ou não a uma questão com o intuito de saber se ele concorda que o solo poderia ser aplicado em uma música ou apresentação ao vivo.

Os usuários avaliadores foram divididos em três grupos com base no nível musical em que eles se enquadram:

- Iniciantes — indivíduos que não tocam nenhum instrumento ou que estão começando ou começaram há pouco tempo em alguma área relacionada a música, seja ela, aprendizado, produção, edição, entre outros.
- Intermediários — indivíduos que já tem alguma experiência em alguma área musical, mas não tem na música seu principal trabalho, ou atividade complementar, podendo ser instrumentista, cantor, etc.
- Profissionais — indivíduos que têm muita experiência na área musical e também têm na música sua profissão principal ou complementar como, por exemplo, professores

de música, músicos de bandas profissionais, produtores, entre outros.

As notas possíveis para a avaliação dos solos, seguindo uma das estratégias da escala de Likert, variam de 1 a 7, sendo que, nesse caso, 1 significa que o avaliador acha o solo completamente ruim e 7 que ele acredita que o solo está excelente. Os valores utilizados na escala e seus respectivos significados estão descritos na Tabela 4.3

Tabela 4.3: Notas e significados para avaliação dos solos

1	2	3	4	5	6	7
Terrível	Muito ruim	Ruim	Moderado	Bom	Muito Bom	Excelente

Foi explicado na Seção 4.1.7 que a primeira etapa após a inicialização do *framework* é o sorteio de um subconjunto de *licks* do banco de dados, capturando assim uma instância menor para fazer o processo de otimização e criação do solo. Isso implica que cada solo gerado parte de uma instância diferente. No entanto, com a finalidade de se ter uma avaliação mais justa, evitando comparar solos gerados a partir de subconjuntos diferentes, todos os solos foram gerados a partir de uma mesma instância de um subconjunto do banco de dados de *licks*.

Ao se utilizar a mesma instância para criar todos os solos, o otimizador tende a gerar sempre a mesma solução, a que ele encontrou inicialmente como a melhor. Para evitar isso, e somente para gerar os solos que foram submetidos a avaliação, foi adotada uma abordagem que consiste em, durante uma rodada de criação de solos, remover da instância sorteada um *lick* que já foi utilizado no solo criado na rodada anterior e adicioná-lo a um conjunto de *licks* que não podem mais ser utilizados. Com esse objetivo foi adicionada à formulação matemática uma restrição temporária. Seja F o conjunto de *licks* proibidos de serem utilizados:

$$\sum_{i \in F} y_i = 0 \quad (4.12)$$

A restrição (4.12) define que os *licks* que pertencem ao conjunto F não podem mais ser utilizados. Em termos práticos, ao se proibir a utilização de um determinado

lick, o otimizador tende a procurar um caminho diferente para alcançar a solução, devido, principalmente, à característica da instância de existirem várias soluções de mesmo custo.

Também foi definido que os solos submetidos à avaliação utilizassem a mesma harmonia e finalizassem sempre com o mesmo *lick turnaround*. Essa medida busca evitar que a base da música ou o último *lick* escutado influenciem de maneira indevida a avaliação de um ouvinte, tentando assim manter no solo o foco da avaliação. A tablatura da harmonia utilizada está disponível no Apêndice A, bem como as tablaturas dos solos aleatórios estão no Apêndice B e a dos solos gerados pelo *framework* no Apêndice C.

5 RESULTADOS COMPUTACIONAIS

Este capítulo mostra os resultados obtidos pela abordagem proposta. Inicialmente é explicado o sistema desenvolvido para a avaliação dos solos. Em seguida, são exibidos gráficos e dados com base em uma análise estatística obtida como resultado da avaliação.

O *framework* proposto foi programado utilizando a linguagem *Python 2.7* e a biblioteca *lxml* Versão 3.4 para a manipulação de textos e arquivos no formato XML. É importante frisar que durante a fase de otimização o tempo médio de execução do *solver* para resolver as instâncias utilizadas na geração dos solos submetidos a avaliação, contendo cerca de 160 *licks*, é de aproximadamente 4,3 segundos.

5.1 SISTEMA DESENVOLVIDO PARA A AVALIAÇÃO DOS SOLOS

Para computar as avaliações dos solos e com o intuito de possibilitar a participação de ouvintes e colaboradores de diversos lugares, foi desenvolvido um sistema utilizando o *Django*¹, que é um *framework web* escrito em *Python*, na Versão 1.8. O sistema foi hospedado na plataforma *Amazon Web Services*² (AWS), em uma máquina virtual configurada com o Linux Debian 8, *Python 2.7* e servidor *web* NGINX³.

Após cadastrados no sistema, os usuários avaliadores receberam as credenciais de acesso para possibilitar seu *login* no sistema e dar início ao processo de avaliação dos solos. No primeiro acesso o usuário é perguntado a respeito de sua classificação quanto ao seu nível musical. Isso é mostrado na Figura 5.1.

¹<https://www.djangoproject.com/>

²<https://aws.amazon.com/>

³<https://www.nginx.com/>

Figura 5.1: Seleção de nível do músico

Após a seleção de seu nível o(a) usuário(a) é redirecionado para uma tela com uma breve explicação do objetivo e funcionamento do sistema de avaliação e então, no próximo passo, é levado à tela de avaliação. Nessa são exibidos os solos que serão avaliados, um por vez, e as possíveis notas para classificá-los, conforme exibido na Figura 5.2.

Figura 5.2: Página de avaliação dos solos

Por fim, antes de salvar seu voto, o avaliador precisa selecionar a opção sim ou não, para indicar se ele(a) acredita que o solo poderia ser aplicado em uma composição musical ou em uma apresentação ao vivo, conforme ilustrado na Figura 5.3.

Figura 5.3: Opção de aceitação do utilização do solo avaliado

5.2 RESULTADOS DAS AVALIAÇÕES

A primeira forma de verificação dos resultados das avaliações dos solos foi através da contabilização total de cada opção da escala de Likert, dividindo os solos em dois grupos, aleatórios e otimizados, sendo que o termo “otimizados” é apenas para explicitar que os solos foram gerados pelo *framework* desenvolvido. Nesta visualização, não há diferenciação entre os níveis dos avaliadores.

É possível perceber, através da análise dos gráficos mostrados nas Figuras 5.4 e 5.5, que os solos otimizados tiveram uma classificação melhor que os solos aleatórios. Na Figura 5.4 as parcelas que correspondem as avaliações que vão de terrível até moderado correspondem a 51,26% do gráfico, enquanto que estas mesmas parcelas, para os otimizados, correspondem a apenas 36,43%, restando assim, aproximadamente, 63,57% para as avaliações bom, muito bom e excelente.

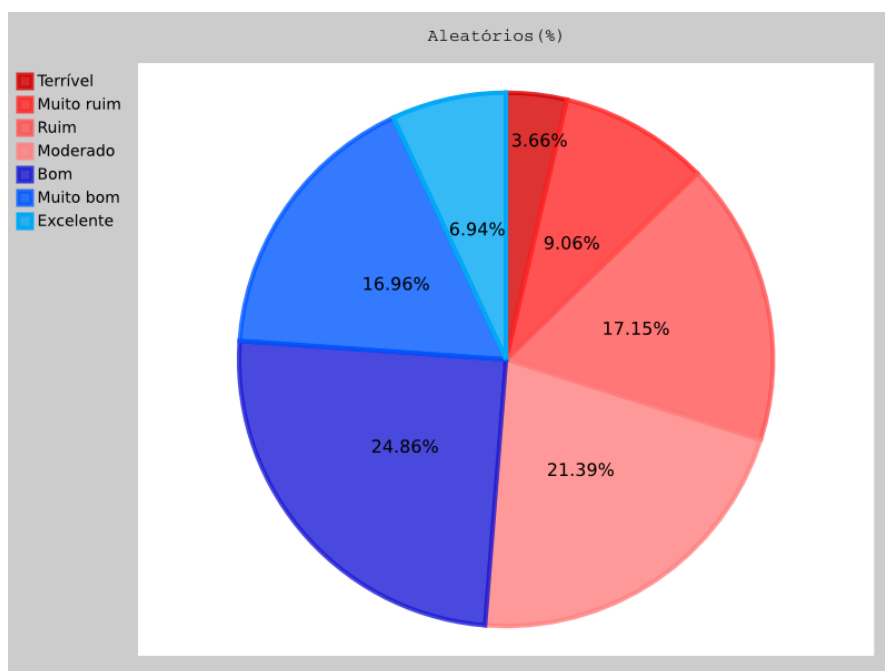


Figura 5.4: Visão geral. Solos aleatórios

A Tabela 5.1 mostra os valores da contabilização referentes aos solos gerados aleatoriamente em comparação aos solos otimizados, gerados pelo *framework* e que foram submetidos a avaliação dos músicos.

Outra parte da análise dos resultados do experimento foi utilizando o pacote es-

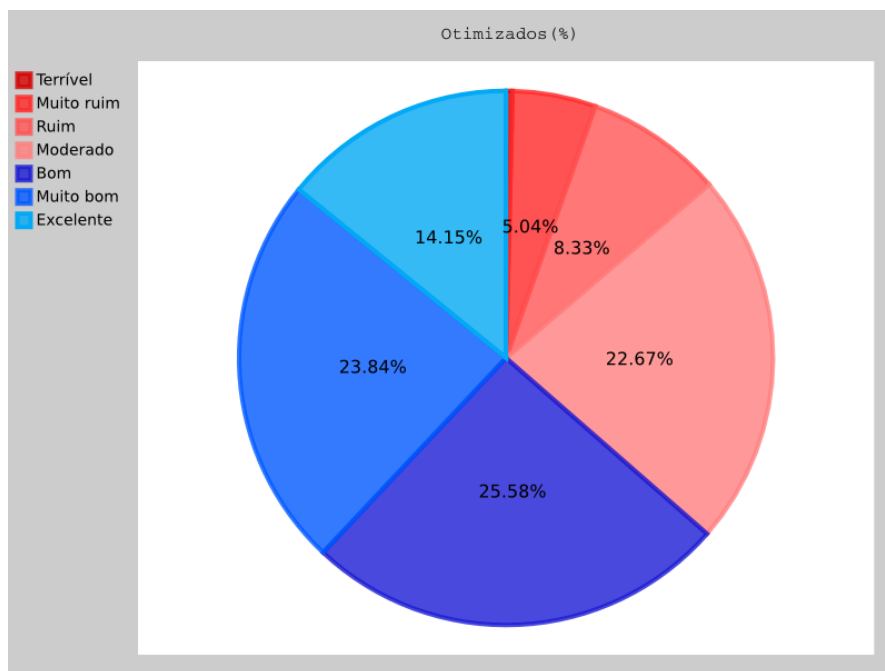


Figura 5.5: Visão geral. Solos otimizados

Tabela 5.1: Contabilização geral em “%” das avaliações dos solos

	Terrível	Muito ruim	Ruim	Moderado	Bom	Muito Bom	Excelente
Aleatórios	3,66	9,06	17,15	21,39	24,86	16,96	6,94
Otimizados	0,39	5,04	8,33	22,67	25,58	23,84	14,15

tatístico R (Team, 2013), uma ferramenta para análise estatística e geração de gráficos. Nesse caso foram calculadas e comparadas as médias dos solos aleatórios e otimizados e as diferenças entre elas foram submetidas ao testes de normalidade de Shapiro e Wilk (1965), Anderson e Darling (1954) e Lilliefors (1967) para verificar se distribuição de probabilidade associada poderia ser aproximada pela distribuição normal. O resultado dos testes, conforme exibidos na Tabela 5.2, com nível de significância de 5%, mostra que os dados não seguem a distribuição normal.

Tabela 5.2: Resultado dos testes de normalidade

Teste	p-Valor
Shapiro-Wilk	0,03186
Anderson-Darling	0,002275
Kolmogorov-Smirnov	0,00005665

Uma vez que os dados não estão normalizados torna-se mais adequado a aplicação

de um teste não-paramétrico, como é o caso do *Wilcoxon signed-rank test*, que é utilizado, geralmente, para a comparação de amostras relacionadas, combinadas, com base nas diferenças (Woolson, 2008). O resultado do teste, mostrado na Tabela 5.3, com nível de significância de 1%, mostrou que existe evidência de diferença estatística entre as amostras aleatórias e otimizadas para os dados analisados.

Tabela 5.3: Resultado do *Wilcoxon signed-rank test*

	p-Valor
Iniciantes	0,00005985
Intermediários	0,0001161
Profissionais	0,0000654

Os resultados dos testes confirmaram que, perceptivelmente, os solos otimizados foram considerados mais “agradáveis” do que os aleatórios, segundo as avaliações dos ouvintes participantes. De acordo com a avaliação das pontuações médias de cada músico, o *p-value* resultante, com nível de significância de 1%, indica que os solos otimizados tendem a ter, significativamente, uma melhor classificação em relação aos solos aleatórios.

A Figura 5.6 reforça o resultado do teste através da visualização, em forma de diagrama de caixa (do inglês, *Box plot* ou *Boxplot*), da avaliação média para ambos os solos, aleatórios (“rand”), e otimizados (“opt”).

Para cada solo de guitarra posto para a avaliação também foi computada a frequência com que a opção “Este solo poderia ser usado em uma música ou em uma *performance* ao vivo” foi marcada como “sim”. O *Wilcoxon test*, obteve o resultado de $p = 0,0027$ mostrando novamente que existe diferença e relevância estatística entre os dados da amostra. Sendo assim têm-se que a média da distribuição da frequência de uso dos solos, otimizados e aleatórios, exibida na Figura 5.7, é significativamente diferente, de tal maneira que os solos otimizados são selecionados como possíveis de utilizar em uma média de 81% contra 68% para os solos aleatórios.

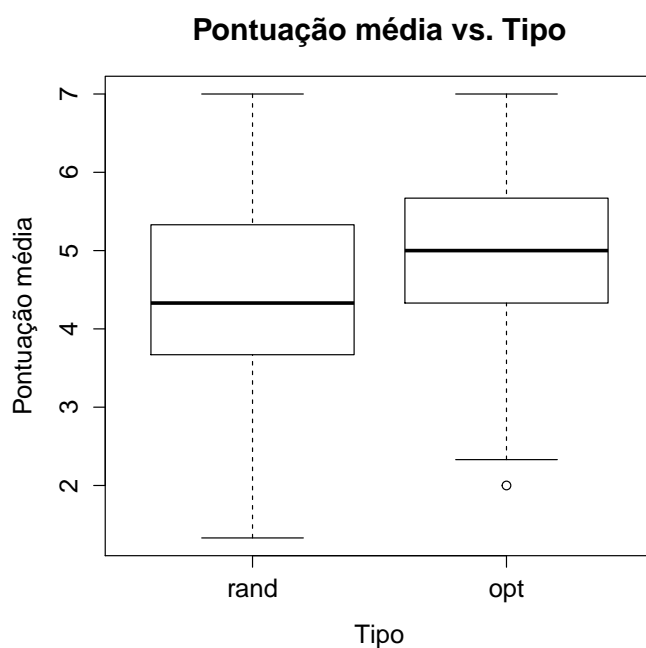


Figura 5.6: *Box plot* do resultado do *Wilcoxon signed-rank test*

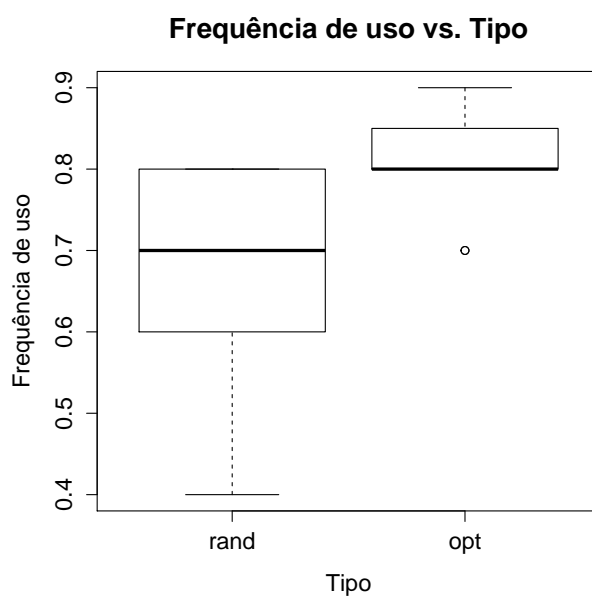


Figura 5.7: Resultado do *Wilcoxon test* para frequência de uso dos solos

5.2.1 Iniciantes

Nas Figuras 5.8 e 5.9 é possível verificar que nas avaliações dos solos, feitas pelos músicos iniciantes, que os otimizados têm uma vantagem sobre os aleatórios quanto a melhor pontuação. É possível perceber que para estes avaliadores, aproximadamente

65,77% do gráfico dos solos otimizados corresponde as classificações de bom, muito bom e excelente e estes não tiveram nenhum solo classificado como terrível. Nos solos aleatórios esta mesma parcela corresponde a 54,66%. O total de participantes classificados como músicos iniciantes foi de 75.

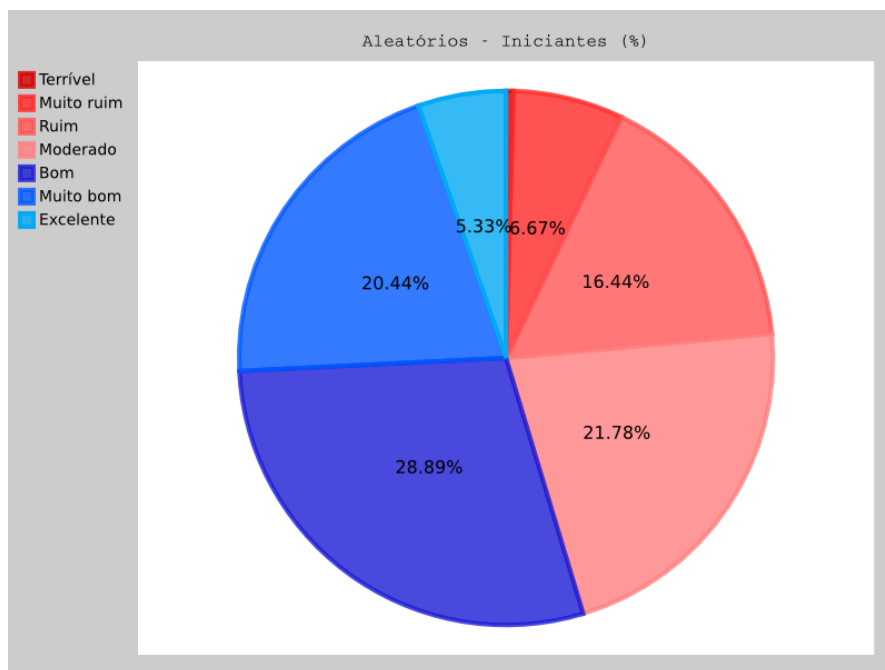


Figura 5.8: Avaliação dos solos aleatórios pelos músicos iniciantes

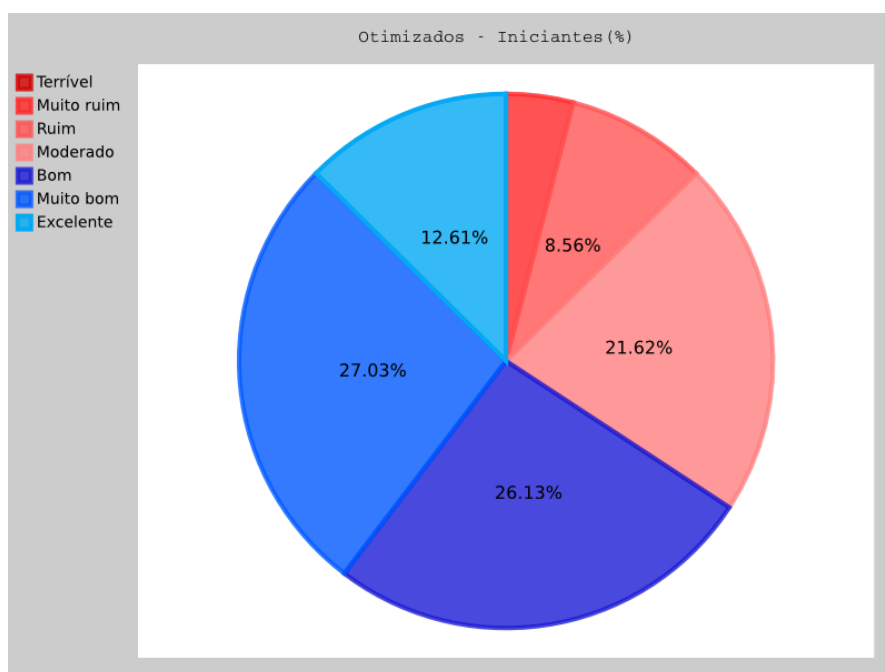


Figura 5.9: Avaliação dos solos otimizados pelos músicos iniciantes

A avaliação da comparação das médias obtidas também confirma a tendência da melhor pontuação dos solos otimizados, o que pode ser visualizado no diagrama de caixa exibido na Figura 5.10 que mostra a comparação da pontuação entre as duas médias. Nessa Figura, o *box* “rand” equivale aos solos aleatórios e o *box* “opt” refere-se aos solos otimizados.

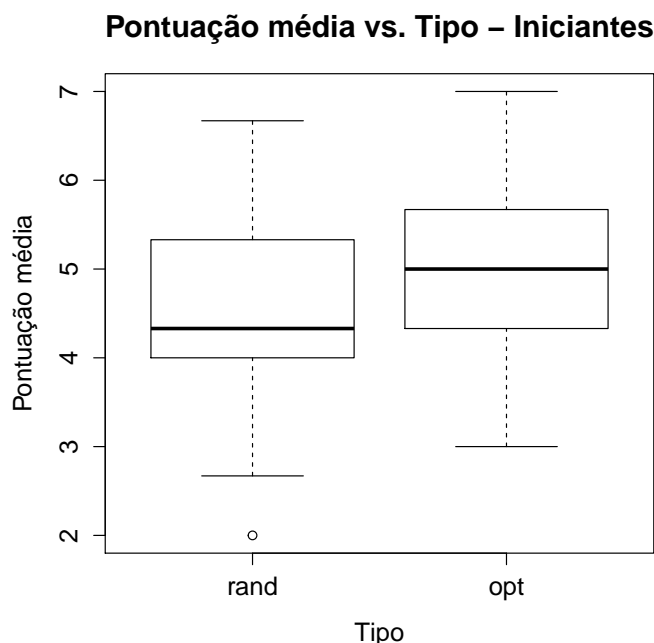


Figura 5.10: *Box plot* do resultado da comparação das médias para os músicos iniciantes

5.2.2 Intermediários

Para os avaliadores de nível intermediário a diferença de classificação entre os solos aleatórios e otimizados ficou bem maior do que em relação aos músicos de nível iniciante. Na Figura 5.11 têm-se 54, 91% do gráfico está para as parcelas que classificam os solos de terrível à moderado.

Entre os solos otimizados, Figura 5.12, esta mesma parcela, de terrível à moderado, corresponde a aproximadamente 37, 26%, restando assim, aproximadamente, 62, 74% para as classificações que consideram os solos com bom, muito bom e excelente. Participaram desta avaliação um total de 51 músicos de nível intermediário.

O *boxplot* da comparação entre as médias obtidas das avaliações dos músicos, visualizado na Figura 5.13, mostra que os solos otimizados possuem uma maior pontuação

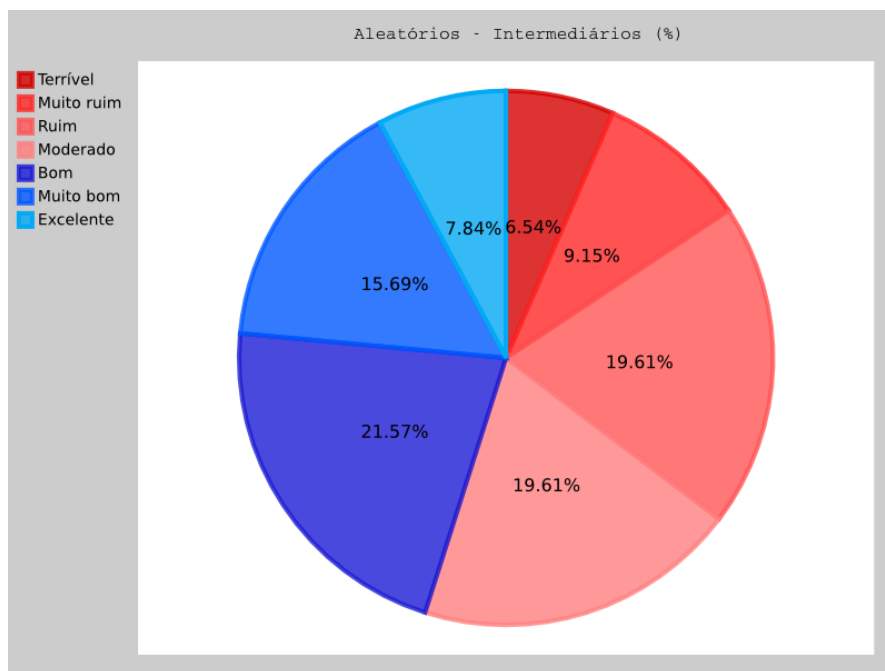


Figura 5.11: Avaliação dos solos aleatórios pelos músicos intermediários

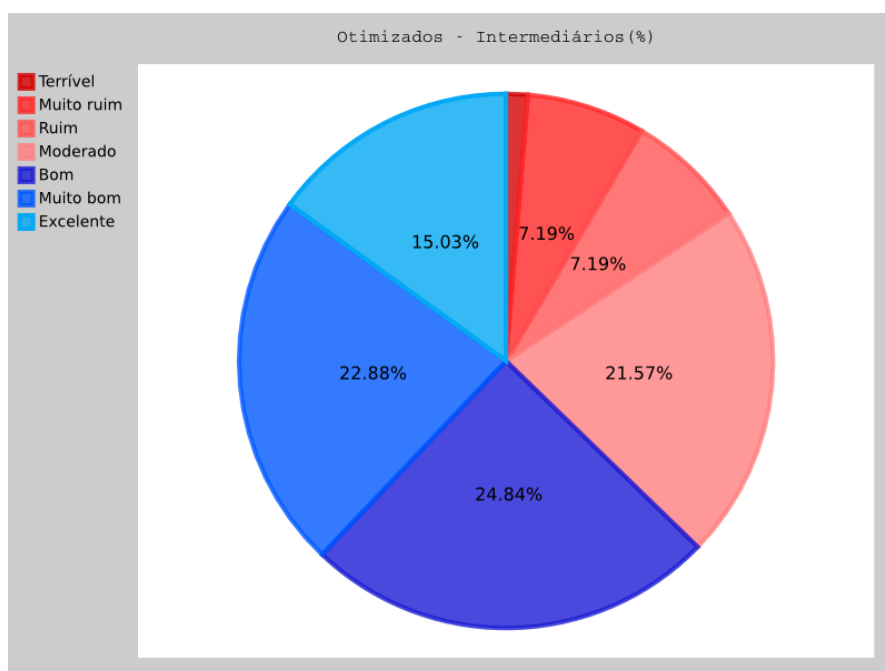


Figura 5.12: Avaliação dos solos otimizados pelos músicos intermediários

média em comparação aos solos aleatórios. Nessa figura, o *box* “rand” equivale aos solos aleatórios e o *box* “opt” refere-se aos solos otimizados.

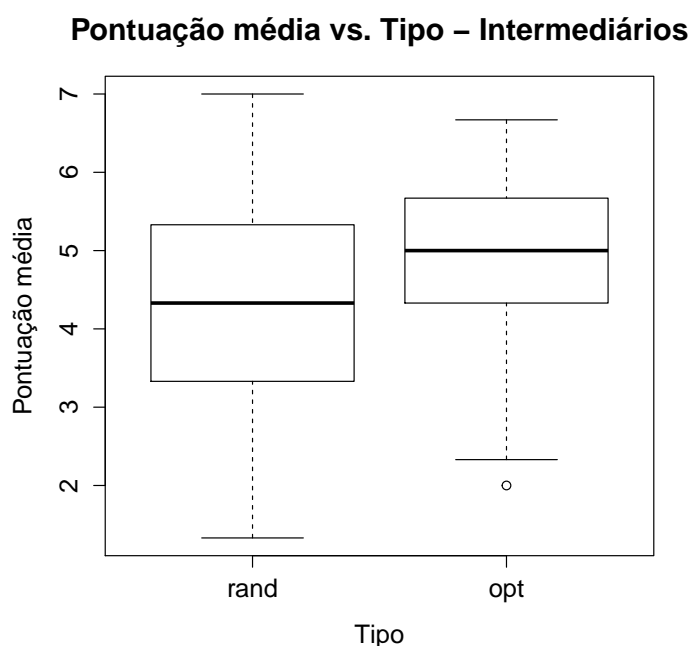


Figura 5.13: *Box plot* do resultado da comparação das médias para os músicos intermediários

5.2.3 *Profissionais*

As avaliações dos músicos profissionais, mostradas nas Figuras 5.14 e 5.15, também confirmam a tendência de uma melhor classificação dos solos considerados otimizados. Com um total de 47 participantes, têm-se que, aproximadamente, 60,99% das classificações dos solos otimizados foram entre bom, muito bom e excelente. Já para os solos aleatórios esta mesma parcela corresponde a apenas 43,27%, ficando assim, aproximadamente, 56,73% para as classificações terrível, muito ruim, ruim e moderado.

Assim como ocorreu com os avaliadores iniciantes, não foi observado nos solos otimizados nenhum solo classificado como terrível. Também é possível verificar que foi por parte dos avaliadores profissionais que os solos otimizados tiveram a melhor classificação entre os excelentes, com 15,60%, contra 15,03% por parte dos avaliadores de nível intermediário e 12,61% por parte dos avaliadores de nível iniciante.

Visualizando o gráfico exibido na Figura 5.16 é possível perceber a maior diferença média entre os solos otimizados e aleatórios ao se comparar com as pontuações obtidas pelos avaliadores de nível intermediário e iniciante. Assim como nos casos anteriores o *box*

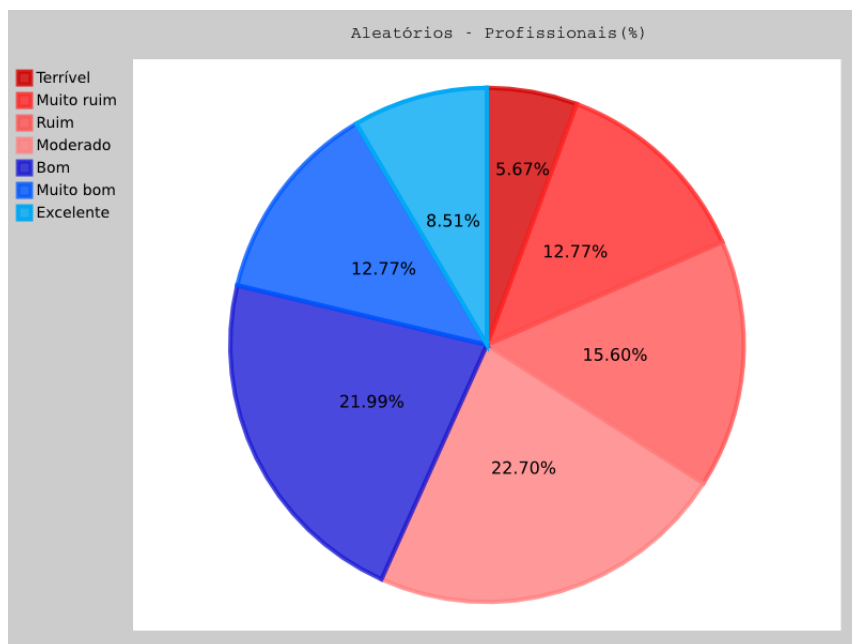


Figura 5.14: Avaliação dos solos aleatórios pelos músicos profissionais

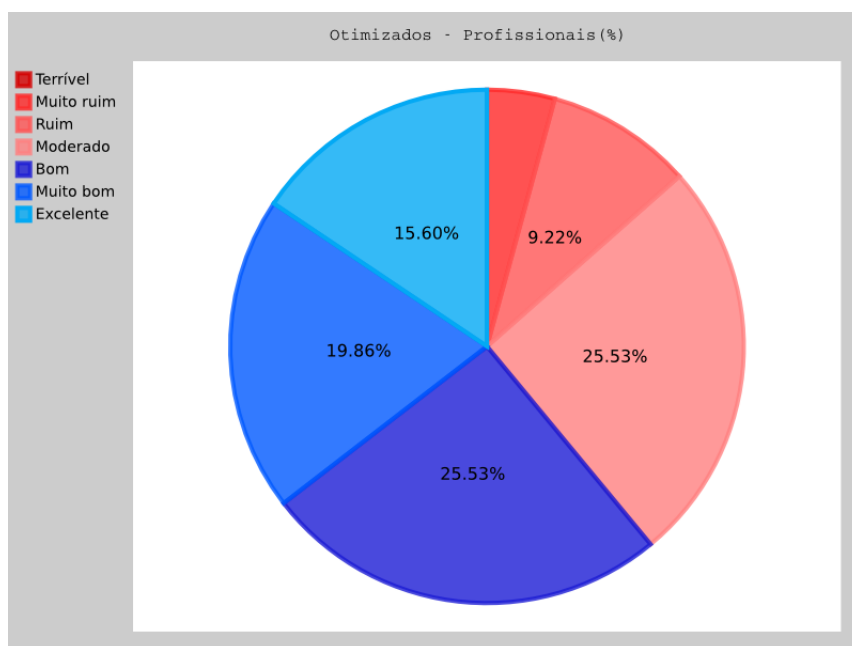


Figura 5.15: Avaliação dos solos otimizados pelos músicos profissionais

“rand” equivale aos solos aleatórios e o *box* “opt” refere-se aos solos otimizados.

5.2.4 Sumário do Resultado das Avaliações

As Tabelas 5.4 e 5.5 sintetizam, respectivamente, os valores das avaliações referentes aos solos aleatórios e solos otimizados, detalhando o resultado alcançado por cada nível

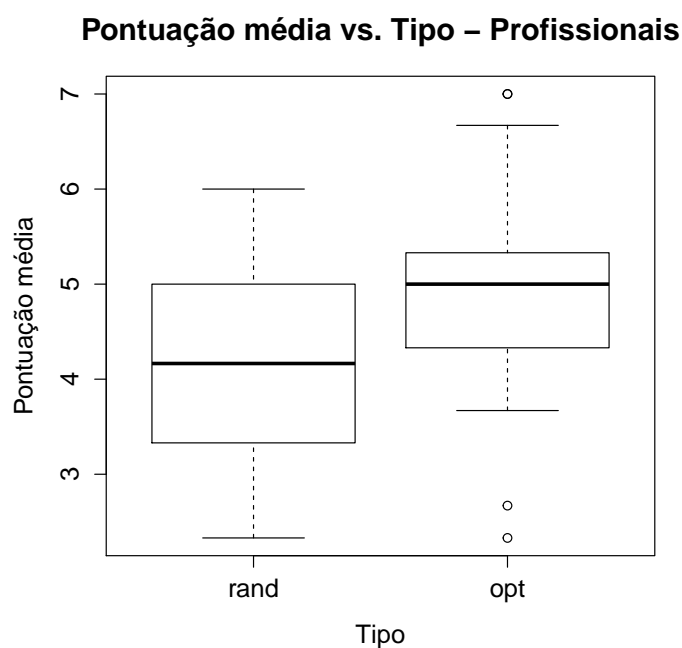


Figura 5.16: *Box plot* do resultado da comparação das médias para os músicos profissionais dos avaliadores.

Tabela 5.4: Resultados da avaliação dos solos aleatórios

	Iniciantes	Intermediários	Profissionais
Terrível	0,44%	6,54%	5,67%
Muito ruim	6,67%	9,15%	12,77%
Ruim	16,44%	19,61%	15,60%
Moderado	21,78%	19,61%	22,70%
Bom	28,89%	21,57%	21,99%
Muito Bom	20,44%	15,69%	12,77%
Excelente	5,33%	7,84%	8,51%

Tabela 5.5: Resultados da avaliação dos solos otimizados

	Iniciantes	Intermediários	Profissionais
Terrível	0,0%	1,31%	0,0%
Muito ruim	4,05%	7,19%	4,26%
Ruim	8,56%	7,19%	9,22%
Moderado	21,62%	21,57%	25,53%
Bom	26,13%	24,84%	25,53%
Muito Bom	27,03%	22,88%	19,86%
Excelente	12,61%	15,03%	15,60%

6 CONSIDERAÇÕES FINAIS

Modelar a criação de solos de guitarra como um problema de otimização combinatória e resolvê-lo através da aplicação de programação matemática e técnicas de otimização mostrou-se um caminho bastante promissor para a CAC. As melodias geradas através da utilização dessa abordagem, foram avaliadas de forma favorável, de acordo com as classificações dadas pelos músicos que participaram da avaliação. O resultado do experimento mostrou uma melhor aceitabilidade dos solos otimizados do que os solos criados de maneira aleatória, mesmo ambos tendo sido criados a partir da mesma instância de um subconjunto do banco de dados de *licks*. Tal fato mostra como o foco na maneira como os *licks* são concatenados têm uma grande importância na criação dos solos, não bastando apenas que eles estejam na mesma tonalidade ou simplesmente sigam a mesma escala.

O *framework* proposto, juntamente com as técnicas utilizadas, podem encaminhar o desenvolvimento de aplicações de apoio à composição ou de apoio à educação, no contexto de ensino e aprendizado musical. Até mesmo outras utilizações seriam possíveis, como, por exemplo, criação de melodias para vídeos e/ou jogos eletrônicos. Todas essas possibilidades ficam dependentes apenas dos *licks* que compõem o banco de dados e de pequenas alterações nas regras de transição que se deseja aplicar.

O banco de dados de *licks* de *Blues* criado, pode ser uma grande contribuição para a continuidade deste ou para o desenvolvimento de novos trabalhos de pesquisa na área. Uma vez que todos os *licks* estão no formato MusicXML, e a maioria dos trabalhos na área utilizam este formato para o armazenamento e intercâmbio de informações, ter vários arquivos referentes a *riffs*, frases e pequenos fragmentos de solos de guitarra concentrados

e acessíveis em um único lugar, torna este banco de dados um agente facilitador. Esta foi uma das dificuldades encontradas neste trabalho, onde, por sua vez, todos os *licks* que agora populam o banco de dados tiveram que ser transcritos manualmente, um a um, utilizando softwares específicos para edição de tablaturas.

O utilização de um modelo matemático, onde a função principal tem por objetivo minimizar a penalização total de transição entre os *licks*, flexibilizou o processo de definições de regras de transição. Dessa forma, as funções de penalização de transição, uma vez definidas, adicionam ou subtraem a penalidade da ocorrência de um determinado *lick* após outro. Assim sendo, tudo que se aplica à transição entre *licks* está concentrado em uma matriz, que foi chamada de matriz de transição ou matriz de penalidades. Diferentes formas de construção da matriz de penalidades vão sempre resultar em solos com características totalmente distintas, sempre tendo como base as regras utilizadas para construir a matriz.

Outra maneira que se mostrou eficiente para diminuir o grau de determinismo do *framework* foi trabalhar com instâncias menores do banco de dados de *licks*. Para tanto, foram realizados sorteios de subconjuntos, onde eram escolhidos de forma randômica uma menor quantidade em relação ao conjunto completo do banco de dados, fazendo assim a criação de solos variados usando sempre conjuntos distintos. Entretanto, a forma adotada para criar os solos que foram submetidos a avaliação, que foi criar vários a partir de mesma instância, adicionando a proibição de utilização de um *lick* por rodada, também foi uma maneira eficiente de criar solos distintos e interessantes. Esta abordagem, inclusive, pode ser adotada para uma futura ferramenta, aplicativo ou sistema de geração de solos que possa ser desenvolvido.

As abordagens de construção aleatória e de construção através do *framework* com a utilização de regras de transição mostraram que existe uma diferença que pode ser percebida quando se usa algumas regras adicionais para gerar uma melodia, mesmo que ambas estratégias utilizem um mesmo conjunto para extrair os *licks* aplicados. Isso é confirmado pelas classificações recebidas pelos solos otimizados em relação aos solos aleatórios. Portanto, uma maior flexibilização da definição das regras pode ajudar a criar os solos gerados de acordo com preferências mais específicas de cada usuário.

Vale ressaltar que a influência exercida pelas regras de transição nos solos resultantes pode sofrer algum impacto por parte das características dos *licks* que populam o banco de dados. Por exemplo, neste trabalho foi definido um pequeno conjunto de regras de transição, dentre elas a utilização de pequenas pausas e notas próximas entre a passagem de um *lick* para outro. Portanto, se os *licks* que compõem o banco de dados, em sua natureza, possuem essas características, torna-se difícil perceber a diferença entre um solo com sequência aleatória e outro sequenciado de forma otimizada.

Partindo da abordagem e *framework* que foram propostos, algumas possibilidades podem ser sugeridas como novas pesquisas ou continuação deste trabalho. O desenvolvimento, de fato, de um aplicativo ou programa de computador com a implementação das funcionalidades aqui apresentadas é uma delas.

Conforme explicado no Capítulo 1, a abordagem utilizada foi considerada semiautomática por assumir que os *licks*, unidade principal de construção dos solos, já estavam disponíveis e foram copiados a partir de modelos humanos. Diante disso, outro caminho a ser trabalhado posteriormente, é a transformação da abordagem de semiautomática para automática. Para tal, se faz necessário a implementação de uma metodologia de criação de *licks* para tornar, assim, o sistema autossuficiente com relação ao seu banco de dados.

Duas novas propostas também podem ser adicionadas em novas pesquisas. A primeira diz respeito a mudança de tonalidade do *lick* dentro da melodia gerada com base no acorde da harmonia em que o *lick* é tocado. Tomando por base a harmonia utilizada para os solos deste trabalho, os *licks* estão na tonalidade de dó maior e dó menor. Mesmo durante a mudança de acordes para o F7 e G7 os *licks* permanecem na mesma tonalidade. Sendo assim, a ideia seria alterar os *licks* para as tonalidades de fá maior e fá menor, e também sol maior e sol menor no momento em que eles são identificados sob esses acordes. A segunda ideia é mudar não só as tonalidades dos *licks* como também trabalhar a harmonia em diferentes tonalidades mantendo as mesmas abordagens quanto aos tons dos *licks*, gerando alguns solos com eles no tom principal da música, e também variando-os de acordo com o acorde tocado.

Também, como possível trabalho futuro, pode-se adicionar uma funcionalidade que

sirva de *interface* do usuário com o sistema, de modo que seja possível definir, com um nível maior de detalhes, preferências mais específicas do músico, gerando assim solos mais personalizados. Por exemplo, o usuário forneceria como entrada ao sistema quais suas técnicas preferidas e/ou quais as que não são de sua preferência. O sistema então faria o processamento dessa informação e as técnicas que foram definidas como preferidas recebem bonificação na matriz de transição e as técnicas não preferenciais poderiam receber penalizações.

Assim sendo, as possibilidades supracitadas podem se tornar complementos promissores, enriquecendo ainda mais as possibilidades alcançadas pelo *framework* desenvolvido neste trabalho, ficam como propostas de trabalhos futuros.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALBAUM, G. The likert scale revisited: an alternate version. *Journal of the Market Research Society*, v. 39, n. 2, p. 331–348, 1997.
- AMES, C. The markov process as a compositional model: A survey and tutorial. *Leonardo*, v. 22, n. 2, p. pp. 175–187. ISSN 0024094X. URL <http://www.jstor.org/stable/1575226>, 1989.
- ANDERS, T. *Composing Music by Composing Rules: Design and Usage of a Generic Music Constraint System*. PhD thesis, Queen’s University Belfast, 2007.
- ANDERS, T.; MIRANDA, E. R. Constraint-based composition in realtime, 2008.
- ANDERSON, T. W.; DARLING, D. A. A test of goodness of fit. *Journal of the American statistical association*, v. 49, n. 268, p. 765–769, 1954.
- APT, K. *Principles of Constraint Programming*. Cambridge University Press, United Kingdom, 0 edição. ISBN 0521125499, 2009.
- AVRIEL, M. *Nonlinear Programming: Analysis and Methods*. Dover Publications, Inc., Mineola, N.Y. ISBN 0-486-43227-0, 2003.
- BÄCKMAN, K. Automatic jazz harmony evolution. *SMC: Proceedings of the 6th Sound and Music Computing Conference 23-25 July 2009 Casa da Música, Porto-Portugal*, p. 349–354, 2009.
- BÄCKMAN, K. Automatic fitness in generative jazz solo improvisation. *EIMAS 2012 Conference Proceedings*, p. 1–10, 2012.

- BÄCKMAN, K. Evolutionary jazz improvisation-jazz harmony and solo improvisation created by means of evolutionary algorithms with automatic fitness. *Sport and Art*, v. 1, n. 3, 2013.
- BILES, J. A. Genjam: A genetic algorithm for generating jazz solos. *International Computer Music Conference*, 1994.
- BILES, J. A. Autonomous genjam: eliminating the fitness bottleneck by eliminating fitness. *Proceedings of the GECCO-2001 Workshop on Non-routine Design with Evolutionary Systems*, 2001.
- BILES, J. A. Genjam: Evolutionary computation gets a gig. *Proceedings of the 2002 Conference for Information Technology Curriculum*, 2002.
- BILES, J. A.; ANDERSON, P. G.; LOGGI, L. W. Neural network fitness functions for a musical iga. *Proceedings of the International ICSC Symposium on Intelligent Industrial Automation (IIA'96) and Soft Computing (SOCO'96)*, 1996.
- BLUM, C.; LI, X. *Swarm intelligence in optimization*. Springer, 2008.
- BÄCKMAN, K. Ant colony optimization and evolutionary algorithms applied to jazz solo improvisation. *Encontro Internacional de Música e Artes Sonoras (EIMAS)*, 2010.
- CAMBOUROPOULOS, E. Markov chains as an aid to computer assisted composition. *Musical Praxis*, v. 1, n. 1, p. 41–52, 1994.
- CARRIJO, D. D. P. Cinema & blues: Representações audiovisuais do gênero no século xxi. Dissertação de Mestrado, Universidade Federal do Paraná, Curitiba, 2014.
- CECILIANO, W. R. A. Aplicação de um método de simulação-otimização na cadeia produtiva de minérios de ferro. Dissertação de Mestrado, Universidade de São Paulo, São Paulo, 2007.
- CHEN, C.-C. Automatic music composition using genetic algorithm and neural networks: A constrained evolution approach. Undergraduate Honors Thesis HR-00-02, Department

- of Computer Sciences, The University of Texas at Austin. URL <http://nn.cs.utexas.edu/?chen:ugthesis00>, 2000.
- CHEN, C.-C. J.; MIIKKULAINEN, R. Creating melodies with evolving recurrent neural networks. *Proceedings of the INNS-IEEE International Joint Conference on Neural Networks*, p. 2241–2246, Piscataway, NJ. IEEE. URL <http://nn.cs.utexas.edu/?chen:ijcnn01>, 2001.
- CHEN, D.-S.; BATSON, R. G.; DANG, Y. *Applied integer programming: modeling and solution*. John Wiley & Sons, 2010.
- CLAUSEN, J. *Branch and bound algorithms - principles and examples*, 1999.
- DA SILVA, J. *Toquei Algo Como Blues, Sem Me Dar Conta Disso...* Thesaurus Editora. ISBN 9788570622952. URL <https://books.google.com.br/books?id=yI05GAce54cC>, 2001.
- DAVISMUN, S.; ECCLES, J. Combining musical constraints with markov transition probabilities to improve the generation of creative musical structures. *Applications of Evolutionary Computation*, p. 361–370. Springer, 2010.
- EIBEN, A. E.; SMITH, J. E. *Introduction to Evolutionary Computing*. Springer-Verlag, 2003.
- FEIJÃO, P. C. Um algoritmo de criação de improvisos com harmonia de jazz. Dissertação de Mestrado, Universidade Estadual de Campinas, Campinas, 2004.
- FLOR, N. V.; HOLDER, B. Hearing with the eyes: A distributed cognition perspective on guitar song imitation. COTTRELL, G. W. (Ed.), *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*, p. 18–148. Lawrence Erlbaum, 1996.
- FREITAS, A. R. R. Música evolutiva: uma abordagem computacional para composição algorítmica. Dissertação de Mestrado, Universidade Federal de Ouro Preto, Ouro Preto, 2011.
- FRITTS, L. *2000 Blues: Licks that rock!* CENTERSTREAM Publishing, 2012.

- FUX, J. J. *The study of counterpoint from Johann Joseph Fux's Gradus ad Parnassum*. Number 277. WW Norton & Company, 1965.
- GARCIA, M. D. R. Ensino e aprendizagem de guitarra em espaços músico-educacionais diversos de João Pessoa. Dissertação de Mestrado, Universidade Federal da Paraíba, João Pessoa - PB, 2011a.
- GARCIA, M. D. R. Processos de autoaprendizagem em guitarra e as aulas particulares de ensino do instrumento. *Revista da ABEM*, v. 19, n. 25, 2011b.
- GEIS, M.; MIDDENDORF, M. An ant colony optimizer for melody creation with baroque harmony. *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, p. 461–468. doi: 10.1109/CEC.2007.4424507, 2007.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edição. ISBN 0201157675, 1989.
- GONÇALVES, A.; LOPES, E.; PAIVA, A. The music of paintings: a rhythmic perspective. *12th Generative Art Conference GA2009*, 2009.
- HALL, M. A.; SMITH, L. A computer model of blues music and its evaluation. *The Journal of the Acoustical Society of America*, v. 100, n. 2, p. 1163–1167, 1996.
- HERREMANS, D.; SÖRENSEN, K. Composing fifth species counterpoint music with a variable neighborhood search algorithm. *Expert Systems with Applications*, v. 40, n. 16, p. 6427 – 6437. ISSN 0957-4174. doi: <http://dx.doi.org/10.1016/j.eswa.2013.05.071>. URL <http://www.sciencedirect.com/science/article/pii/S0957417413003692>, 2013.
- HERREMANS, D.; SÖRENSEN, K. Composing first species counterpoint with a variable neighbourhood search algorithm. *Journal of Mathematics and the Arts*, v. 6, n. 4, p. 169–189. doi: 10.1080/17513472.2012.738554. URL <http://dx.doi.org/10.1080/17513472.2012.738554>, 2012.
- HILLER, L.; ISAACSON, L. *Illiac Suite*. Score, Theodore Presser Co, New York, USA, 1957.

- HILLER, L.; ISAACSON, L. Musical composition with a high-speed digital computer. *Journal of Audio Engineering Society* 6, 1958.
- HORI, G.; KAMEOKA, H.; SAGAYAMA, S. Input-output hmm applied to automatic arrangement for guitars. *Journal of Information Processing*, v. 21, n. 2, p. 264–271. doi: 10.2197/ipsjjip.21.264, 2013.
- HORI, G.; YOSHINAGA, Y.; FUKAYAMA, S.; SAGAYAMA, S. Automatic arrangement for guitars using hidden markov model. *Proceedings of the 9th Sound and Music Computing Conference*, p. 450–456, 2012.
- IRNICH, S.; DESAULNIERS, G. ET AL. Shortest path problems with resource constraints. *Column generation*, v. 6730, p. 33–65, 2005.
- KENNEDY, M. *The Concise Oxford Dictionary of Music*. Oxford University Press, Oxford, 2003.
- KRENZ, S. *Blues Guitar*. Legacy Learning Systems, 2009.
- LACERDA, O. *Compêndio de Teoria Elementar da Música*. Ricordi Brasileira, São Paulo, 1967.
- LACHTERMACHER, G. *Pesquisa operacional na tomada de decisões: modelagem em Excel*. Elsevier, 2004.
- LANGSTON, P. Six techniques for algorithmic music composition. *15th International Computer Music Conference*. Citeseer, 1989.
- LEME, G. R.; BELLOCHIO, C. R. Professores de escolas de música: um estudo sobre a utilização de tecnologias. *Revista da ABEM*, v. 15, n. 17, 2007.
- LEWIS, R.; WIECZORKOWSKA, A.; RAS, Z. W. Pentatonic harmonics in fourier transforms: Why the blues are blue, 2005.
- LILLIEFORS, H. W. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, v. 62, n. 318, p. 399–402, 1967.

- MACULAN, N.; FAMPA, M. H. C. *Otimização linear*. EdUnB. ISBN 9788523009274. URL <https://books.google.com.br/books?id=s02VPgAACAAJ>, 2006.
- MARSHALL, W. *101 Must-Know Blues Licks*. Hal Leonard, 1999.
- MATIĆ, D. A genetic algorithm for composing music. *Yugoslav Journal of Operations Research ISSN: 0354-0243 EISSN: 2334-6043*, v. 20, n. 1, 2010.
- MCVICAR, M.; FUKAYAMA, S.; GOTO, M. Autoleadguitar: Automatic generation of guitar solo phrases in the tablature space. p. 201–213, 2014a.
- MCVICAR, M.; FUKAYAMA, S.; GOTO, M. Autorhythmuitar: Computer-aided composition for rhythm guitar in the tab space. *40th International Computer Music Conference (ICMC) joint with the 11th Sound and Music Computing conference (SMC), Music Technology meets Philosophy: From Digital Echos to Virtual Ethos*, 2014b.
- MENEZES, J. M. A. *Composição algorítmica de fugas ao estilo de j.s. bach*, 2009.
- MIRANDA, E. R.; BILES, J. A. *Evolutionary computer music*. Springer-Verlag. ISBN 1-84628-599-2, 2007.
- MORIARTY, D. E.; MIIKKULAINEN, R. Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, , n. AI94-224, p. 11–32. URL <http://nn.cs.utexas.edu/?moriarty:mlj96>, 1996.
- MUNOZ, E.; CADENAS, J.; ONG, Y. S.; ACAMPORA, G. Memetic music composition, 2014.
- PAPADOPOULOS, G.; WIGGINS, G. AI methods for algorithmic composition: A survey, a critical view and future prospects. *In AISB Symposium on Musical Creativity*, 1999.
- RANDEL, D. M. *The Harvard Dictionary of Music*. Harvard University Press, 4 edição. ISBN 0674011635, 2003.
- ROCHA, M. E. L. *Elaboração de arranjo para guitarra solo*. Dissertação de Mestrado, Universidade Estadual de Campinas, Campinas, 2005.

- SADIE, S.; TYRRELL, J. *The New Grove Dictionary of Music and Musicians*. Oxford University Press, Oxford. ISBN 0195170679, 978-0195170672, 2001.
- SANDRED, O.; LAURSON, M.; KUUSKANKARE, M. Revisiting the illiac suite - a rule based approach to stochastic processes. *Sonic Ideas/Ideas Sonicas*, p. 42–46, 2009.
- SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). *Biometrika*, v. 52, n. 3/4, p. 591–611, 1965.
- TANAKA, T.; FUJII, K. Describing global musical structures by integer programming on musical patterns. *Mathematics and Computation in Music*, p. 52–63. Springer, 2015.
- TEAM, R. C. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>, 2013.
- TUOHY, D. R.; POTTER, W.; CENTER, A. I. An evolved neural network/hc hybrid for tablature creation in ga-based guitar arranging. *Proceedings of the International Computer Music Conference (ICMC'06)*, 2006.
- TUOHY, D.; POTTER, W. A genetic algorithm for the automatic generation of playable guitar tablature. *Proc. International Computer Music Conference*, 2005.
- TUOHY, D.; POTTER, W. Ga-based music arranging for guitar. *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, p. 1065–1070. doi: 10.1109/CEC.2006.1688427, 2006.
- WOLSEY, L. A. *Integer programming*. Wiley, 1998.
- WOOLSON, R. Wilcoxon signed-rank test. *Wiley Encyclopedia of Clinical Trials*, 2008.
- YOSHINAGA, Y.; HORI, G.; FUKAYAMA, S.; SAGAYAMA, S. Fingering determination and automatic arrangement for guitars using hidden markov model. *Proc. of ASJ Spring Meeting*, p. 1011–1014, 2012.

Apêndices

B - TABLATURAS DOS SOLOS GERADOS ALEATORIAMENTE

As Figuras deste apêndice representam a visualização, em modo tablatura, dos solos gerados aleatoriamente que foram disponibilizados para avaliação dos músicos através da ferramenta web desenvolvida neste trabalho.

rand-20151209135839

♩ = 110 (♩=♩♩)

1

6

full

1/2

Figura B.1: Tablatura do solo aleatório 1

rand-20151209135846

$\text{♩} = 110$ ($\text{♩} = \text{♩}^{\text{-3}}$)

The image shows three systems of guitar tablature for a solo. The first system (measures 1-4) starts with a treble clef and a 4/4 time signature. It features a sequence of notes: 8-9-10, 8, 8-6-5-4-5, and 8-9-10-8. The second system (measures 5-8) includes a 'full' vibrato mark over the first measure, followed by notes 8-9-10-8, 8-9-10, 13-10-12-10, and 8-10-8. The third system (measures 9-12) starts with a treble clef and includes notes 8-9-10-8, 10-5-5-8-10-1, 8-8-11-8, and 8-10-8. Various techniques like triplets and vibrato are indicated throughout.

Figura B.2: Tablatura do solo aleatório 2

rand-20151209135847

$\text{♩} = 110$ ($\text{♩} = \text{♩}^{\text{-3}}$)

The image shows three systems of guitar tablature for a solo. The first system (measures 1-4) starts with a treble clef and a 4/4 time signature. It features a sequence of notes: 8-10-8-8, 11-11-11-11, 10-(10)-8, and 11-11-11-11. The second system (measures 5-8) includes notes 8-10-8-8, 8-10-8, 10-8, 10-(10)-8-10-8, and 7-7-8. The third system (measures 9-12) starts with a treble clef and includes notes 13-10-12-13-12, 8-8-11-8, and 8-10-8. Various techniques like vibrato and half-note bends are indicated throughout.

Figura B.3: Tablatura do solo aleatório 3

rand-20151209135850

$\text{♩} = 110$ ($\text{♩} = \text{♩} = \text{♩}$)

1

11 8 11 8 8 11 8 10 11 10 8 10 8 6 5 3

1/2

8 10 10 8 10 8 9 8 10 10 8 10 8 11 11 8 11 8 11 11 10 8 10 8

1/4 1/4

8 8 11 8 10 10 8 10 8 10 10 8 10 10 8 10 10

full 1/2

Figura B.5: Tablatura do solo aleatório 5

rand-20151209135909

$\text{♩} = 110$ ($\text{♩} = \text{♩}^{-3}$)

The image shows three systems of guitar tablature for a solo. Each system consists of a six-line staff with fret numbers and rhythmic markings. The first system starts with a measure containing a whole note chord (7-7-8) and a quarter rest, followed by a measure with a quarter note (10) and a quarter rest, and a final measure with a quarter note (11) and a quarter rest. The second system begins with a measure of a quarter note (6) and a quarter rest, followed by a measure with a quarter note (8) and a quarter rest, and a final measure with a quarter note (8) and a quarter rest. The third system starts with a measure of a quarter note (10) and a quarter rest, followed by a measure with a quarter note (10) and a quarter rest, and a final measure with a quarter note (10) and a quarter rest. Various annotations include 'full', '1 1/2', and '1/4' with arrows pointing to specific notes or rests. Wavy lines above notes indicate vibrato. Fingering numbers (1-3) are shown above some notes. A 3-measure rest is indicated in the first system.

Figura B.6: Tablatura do solo aleatório 6

rand-20151209135910

$\text{♩} = 110$ ($\text{♩} = \text{♩}^{-3}$)

The image shows three systems of guitar tablature for a solo. Each system consists of a six-line staff with fret numbers and rhythmic markings. The first system starts with a measure containing a whole note chord (14-13) and a quarter rest, followed by a measure with a quarter note (11) and a quarter rest, and a final measure with a quarter note (13) and a quarter rest. The second system begins with a measure of a quarter note (6) and a quarter rest, followed by a measure with a quarter note (11) and a quarter rest, and a final measure with a quarter note (11) and a quarter rest. The third system starts with a measure of a quarter note (8) and a quarter rest, followed by a measure with a quarter note (8) and a quarter rest, and a final measure with a quarter note (8) and a quarter rest. Various annotations include 'full', '1/4', and '1/2' with arrows pointing to specific notes or rests. Wavy lines above notes indicate vibrato. Fingering numbers (1-3) are shown above some notes. A 3-measure rest is indicated in the first system.

Figura B.7: Tablatura do solo aleatório 7

rand-20151209135911

$\text{♩} = 110$ ($\text{♩} = \text{♩}^{\text{-3}}$)

1

8

9 10

10

11 10 8

11 10 8

10

18 18 15 15

17 17 15

full

6

8 9

15 13

8 9

10 8

10

5 5 8 10 1

3 3 6 8 1

13 10 13 12

full

11

8 10 8

10 8

8 8

11 8

10 10 8

10

8 10

10

full

1/2

Figura B.8: Tablatura do solo aleatório 8

rand-20151209135929

♩ = 110 (♩♩=♩♩♩)

The image displays a guitar solo in 4/4 time, consisting of 13 measures. The tempo is marked as ♩ = 110, with a note value of ♩♩ = ♩♩♩. The piece is titled 'rand-20151209135929'. The notation includes standard musical symbols such as stems, beams, and slurs, along with fret numbers (e.g., 5, 8, 10, 11, 12, 13, 14, 16) and specific techniques like triplets and bends. The word 'full' is written above the staff with arrows pointing to specific notes in measures 4 and 8. Wavy lines above the staff indicate vibrato or tremolo effects. The tablature is presented in three systems: measures 1-5, 6-8, and 9-11, with a final system for measures 12-13. The bottom system shows a double bar line at the end of measure 13.

Figura B.9: Tablatura do solo aleatório 9

rand-20151209143528

$\text{♩} = 110$ ($\text{♩} = \text{♩}^{\text{-3}}$)

The image displays a guitar solo in standard notation and tablature. The tempo is marked as 110 BPM, with a note value of a quarter note equal to a triplet eighth note. The piece is in 4/4 time. The solo is divided into four systems of measures:

- System 1 (Measures 1-8):** Starts with a triplet of 10s, followed by 10-8, 10, and a wavy line. The second measure contains a triplet of 8s, followed by 8-10-12, 8-10, 8-10-12, and 8-10-12. The third measure contains a triplet of 9s, followed by 7-10-7, 9-7, 7-10, 7-9, and 9.
- System 2 (Measures 9-12):** Measure 9 has a triplet of 8s. Measure 10 has a triplet of 5s, followed by 3, 3, and 8. Measure 11 has a triplet of 10s, followed by 8, 10-8, 8-10, and 8-10. Measure 12 has a triplet of 8s, followed by 10-10, 8-10, 8-10, and 9.
- System 3 (Measures 13-16):** Measure 13 has a triplet of 1s, followed by 1, 4-1-4, 1-1, and 1. Measure 14 has a triplet of 10s, followed by 9-10, 10, and 8-7. Measure 15 has a triplet of 10s, followed by 8, 10, 8, 10, and 8. Measure 16 has a triplet of 8s, followed by 8, 11-8, 10, 10-8, and 10.
- System 4 (Measure 17):** Measure 17 has a triplet of 8s, followed by 10, and 10.

Dynamic markings include 'full' (indicated by an upward arrow) and articulation markings such as '1/4', '1/2', and '1/4'. Rhythmic patterns are indicated by brackets with numbers 3, 2, and 4.

Figura B.10: Tablatura do solo aleatório 10

rand-20151209143529

$\text{♩} = 110$ (♩=♩♩)

The image displays a guitar solo in 4/4 time, starting at a tempo of 110 BPM. The notation is presented in four systems, each consisting of a six-line staff and a corresponding fretboard diagram below it. The solo begins at measure 1 and concludes at measure 13. The first system (measures 1-4) features a sequence of notes on the 8th and 10th frets, with a triplet of eighth notes on the 8th fret in the first measure. The second system (measures 5-8) includes a triplet of eighth notes on the 8th fret in the fifth measure and a 'full' dynamic marking above the 9th fret in the sixth measure. The third system (measures 9-12) contains a triplet of eighth notes on the 10th fret in the tenth measure and another 'full' dynamic marking above the 10th fret in the twelfth measure. The fourth system (measure 13) shows a final triplet of eighth notes on the 8th fret. The fretboard diagrams use vertical lines to represent fret positions and horizontal lines for the strings, with some lines shaded to indicate fingerings or specific techniques.

Figura B.11: Tablatura do solo aleatório 11

rand-20151209143530

$\text{♩} = 110$ ($\text{♩} = \text{♩} = \text{♩}$)

The image displays three staves of guitar tablature for a solo piece. The first staff begins at measure 1 and concludes at measure 10. The second staff starts at measure 6 and ends at measure 10. The third staff starts at measure 1 and ends at measure 10. The tablature includes fret numbers, accidentals, and performance instructions such as 'full', '1/2', and '2'. A tempo marking of 110 BPM is shown at the top left.

Figura B.12: Tablatura do solo aleatório 12

C - TABLATURAS DOS SOLOS OTIMIZADOS GERADOS

As Figuras deste apêndice representam a visualização, em modo tablatura, dos solos otimizados, isto é, que foram gerados pelo *framework* criado, e que foram disponibilizados para avaliação dos músicos através da ferramenta *web* desenvolvida neste trabalho.

opt-20151209132845

$\text{♩} = 110$ (♩=♩♩)

The tablature consists of four systems, each with two staves (treble and bass clef). The first system contains four measures of music, featuring fret numbers 8, 10, 12 and triplets. The second system contains four measures, including vibrato, a 1/4 note bend, and a 'full' bend. The third system contains four measures, with fret numbers 10, 11, 13 and triplets. The fourth system contains three measures, including a 1/2 note bend and a 'full' bend. The piece concludes with a double bar line.

Figura C.1: Tablatura do solo otimizado 1

opt-20151123225030

$\text{♩} = 110$ ($\text{♩} = \text{♩}^{\text{3}}$)

The image displays a guitar solo in 4/4 time, starting at a tempo of 110 BPM. The notation is presented in five systems, each consisting of a six-line staff with fret numbers and rhythmic markings. The first system begins with a first-measure repeat sign and contains two measures of music. The second system also has two measures. The third system contains three measures. The fourth system contains four measures, including a wavy line indicating a vibrato effect. The fifth system is a short concluding phrase of two measures. Rhythmic values are indicated by flags, beams, and slurs. Specific techniques are marked with 'full' (full bends) and '1/2' (half bends). Fingering is indicated by numbers 1-4. The piece concludes with a double bar line.

Figura C.2: Tablatura do solo otimizado 2

opt-20151209132850

$\text{♩} = 110$ ($\text{♩} = \text{♩} = \text{♩}$)

The image displays a guitar solo in standard notation with a corresponding tablature below. The tempo is marked as 110 BPM. The notation includes various techniques such as triplets, bends (labeled 'full', '1/4', '1 1/2'), and vibrato (indicated by wavy lines). The tablature uses fret numbers (e.g., 13, 11, 12, 13, 6, 4, 6, 4, 5, 8, 4, 6) and includes rhythmic indicators like vertical bars and stems. The solo is divided into four systems of notation and tablature.

Figura C.3: Tablatura do solo otimizado 3

opt-20151209132910

$\text{♩} = 110$ ($\text{♩} = \text{♩}^3$)

The image displays a guitar solo in 4/4 time, consisting of four systems of notation. Each system includes a standard musical staff with a treble clef and a 4/4 time signature, and a corresponding guitar tablature below it. The tablature uses numbers 8, 9, 10, 11, and 12 to indicate fret positions. The solo begins with a first-measure rest (marked '1') and features various techniques such as triplets (indicated by a '3' under a bracket), slurs, and accents labeled 'full'. Specific rhythmic values are noted above the staff: a quarter note (1/4) and a half note (1/2). The piece concludes with a double bar line.

Figura C.6: Tablatura do solo otimizado 6

opt-20151209132914

♩ = 110 (♩=♩♩)

The image displays a guitar solo in 4/4 time, consisting of four systems of notation. Each system includes a six-line staff with fret numbers, rhythmic markings, and a corresponding fretboard diagram below. The first system starts at measure 1 and ends at measure 8. The second system starts at measure 9 and ends at measure 12. The third system starts at measure 13 and ends at measure 16. The fourth system is a short fragment starting at measure 17. The notation includes various techniques such as triplets (indicated by a '3' under a bracket), slurs, and accents. Specific markings include '1/2' and '1/4' above notes, and 'full' above a note in the third system. The fretboard diagrams use vertical lines to represent strings and horizontal bars to represent fretted notes.

Figura C.7: Tablatura do solo otimizado 7

opt-20151209132926

$\text{♩} = 110$ ($\text{♩} = \text{♩}^3$)

The image displays a guitar solo in 4/4 time, starting at a tempo of 110 BPM. The notation is presented in four systems, each consisting of a six-line staff with fret numbers and a corresponding rhythmic diagram below. The first system begins with a first-measure repeat sign and contains a sequence of triplets and sixteenth-note runs. The second system features a quarter-note triplet and a half-note triplet. The third system includes three 'full' (f) accents and a half-note triplet. The fourth system concludes with a quarter-note triplet and a half-note triplet. The piece ends with a wavy line indicating a fade-out.

Figura C.9: Tablatura do solo otimizado 9

opt-20151209132932

♩ = 110 (♩♩♩♩⁻³⁻)

The image displays a guitar solo in 4/4 time, starting at measure 1. The tempo is marked as ♩ = 110. The notation includes fret numbers (8, 10, 11, 13, 15, 17, 18) and various techniques such as triplets, bends (marked 'full' and '1/2'), and vibrato (wavy lines). The solo is divided into three systems of measures, with measure numbers 1, 9, and 13 indicated. The first system (measures 1-8) features a sequence of eighth notes and triplets. The second system (measures 9-12) includes a triplet of eighth notes and a full bend. The third system (measures 13-16) features a sequence of eighth notes and a full bend. A final measure (17) shows a sequence of eighth notes and a full bend. The tablature is accompanied by a rhythmic diagram below each system of measures.

Figura C.10: Tablatura do solo otimizado 10

opt-20151209132938

$\text{♩} = 110$ ($\text{♩} = \text{♩}^{\text{-3}}$)

The image displays three systems of guitar tablature for an optimized solo. Each system consists of a six-line staff with fret numbers and rhythmic markings, and a corresponding bass line below it.

- System 1 (Measures 1-5):** Starts with a first position (1) and a 4/4 time signature. The first measure contains a triplet of eighth notes on the 8th fret. The second measure features a half note on the 10th fret. The third measure has a quarter note on the 10th fret, marked with a $1/2$ dynamic. The fourth measure contains a quarter note on the 7th fret. The fifth measure has a quarter note on the 10th fret. Wavy lines above the staff indicate vibrato or tremolo effects.
- System 2 (Measures 6-10):** Starts with a sixth position (6). The first measure has a quarter note on the 13th fret, marked with a *full* dynamic. The second measure has a quarter note on the 10th fret. The third measure has a quarter note on the 8th fret. The fourth measure has a quarter note on the 11th fret. The fifth measure has a quarter note on the 11th fret. Wavy lines above the staff indicate vibrato or tremolo effects.
- System 3 (Measures 11-15):** Starts with a first position (1). The first measure has a quarter note on the 11th fret. The second measure has a quarter note on the 8th fret. The third measure has a quarter note on the 11th fret. The fourth measure has a quarter note on the 8th fret. The fifth measure has a quarter note on the 11th fret. The sixth measure has a quarter note on the 10th fret, marked with a $1/4$ dynamic. The seventh measure has a quarter note on the 8th fret. The eighth measure has a quarter note on the 11th fret, marked with a *full* dynamic. The ninth measure has a quarter note on the 10th fret. The tenth measure has a quarter note on the 8th fret. The eleventh measure has a quarter note on the 10th fret, marked with a $1/2$ dynamic. Wavy lines above the staff indicate vibrato or tremolo effects.

Figura C.11: Tablatura do solo otimizado 11

opt-20151209132941

♩ = 110 (♩=♩♩)

1/4

1

The image displays a guitar solo in 4/4 time, starting at measure 1. The tempo is 110 BPM. The notation includes fret numbers (e.g., 11, 13, 8, 10, 7, 10, 8, 11, 10, 8, 10, 11, 10, 8, 11, 8, 10, 8, 10, 8) and various techniques such as triplets (indicated by a '3' under a bracket), 'full' notes (indicated by an upward arrow), and vibrato (indicated by a wavy line). The solo is divided into four systems of measures. The first system contains measures 1-4, the second contains measures 5-8, the third contains measures 9-12, and the fourth contains measure 13. Measure 13 is marked with a '1/2' time signature change. The tablature is accompanied by a rhythmic diagram below each system of measures, showing the timing of notes and rests.

9

13

1/2

Figura C.12: Tablatura do solo otimizado 12

