

---

Alan Castilho da Roza

*Algoritmos Culturais Aplicados na Composição Musical  
Inteligente e Supervisionada*

---

Joinville  
2015

Alan Castilho da Roza

*Algoritmos Culturais Aplicados na Composição Musical  
Inteligente e Supervisionada*

Relatório Final de Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Graduação em Ciência da Computação, da Universidade do Estado de Santa Catarina (UDESC), como requisito da disciplina de Trabalho de Conclusão de Curso.

**Orientador: Profº Dr. Rafael Stubs Parpinelli**

Joinville

2015

Alan Castilho da Roza

*Algoritmos Culturais Aplicados na Composição Musical  
Inteligente e Supervisionada*

Relatório Final de Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Ciência da Computação da UDESC, como requisito para a obtenção do grau de BACHAREL em Ciência da Computação.

Aprovado em

**BANCA EXAMINADORA**

---

Profº Dr. Rafael Stubs Parpinelli

---

Profº Dr. Claudio Cesar de Sá

---

Profº Dr. Fernando Deeke Sasse

*"Para conquistarmos algo na vida não  
basta ter talento, não basta ter força, é  
preciso também viver um grande amor..."*  
*Wolfgang Amadeus Mozart*

## Agradecimentos

Primeiramente a Deus que permitiu que tudo isso acontecesse, ao longo de minha vida, e não somente nestes anos como universitário, mas que em todos os momentos é o maior mestre que alguém pode conhecer.

A Universidade do Estado de Santa Catarina, pela oportunidade de fazer o curso.

Ao professor Dr. Rafael Stubs Parpinelli, pela orientação, apoio e confiança.

Aos meus pais Luiz e Salete, pelo amor, incentivo e apoio incondicional.

Meus agradecimentos aos amigos Fazel, Gabriel, Jonathan e Anderson, companheiros de trabalhos e irmãos na amizade que fizeram parte da minha formação e que vão continuar presentes em minha vida com certeza.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

## Resumo

Este trabalho apresenta a proposta de um modelo de Algoritmo Cultural (AC) para realizar Composição Musical Inteligente (CMI) e supervisionada, gerando trechos musicais. O AC é uma meta-heurística evolutiva baseada no processo de evolução cultural da humanidade. A CMI é uma técnica que utiliza meta-heurísticas para composição de música automática. Ao final do trabalho busca-se com AC uma alternativa à Evolução Diferencial (ED) para CMI em determinados gêneros musicais fazendo uma análise comparativa dos resultados de ambos os métodos. Esses resultados são na forma de linhas melódicas que consigam possuir características de técnicas composicionais pré-definidas.

**Palavras-chave:** Algoritmos Culturais, Composição Musical Inteligente, Evolução Diferencial

# Abstract

This work proposes a model of Cultural Algorithm (CA) to perform Intelligent Music Composition (IMC) and supervised, generating music sections. The AC is a meta-heuristic based on evolutionary process of cultural evolution of mankind. The IMC is a technique using meta-heuristics for automatic music composition. At the end of the work seek to AC with an alternative to the Differential Evolution (DE) to IMC for certain genres doing a comparative analysis of the results of both methods. These results are in the form of melodic lines that can have characteristics of predefined compositional techniques.

**Keywords:** Cultural Algorithms, Intelligent Musical Composition, Differential Evolution

# Sumário

<b>Lista de Tabelas</b>	<b>8</b>
<b>Lista de Figuras</b>	<b>10</b>
<b>1 Introdução</b>	<b>11</b>
1.1 Composição Musical Inteligente Supervisionada . . . . .	12
1.2 Algoritmos Culturais . . . . .	13
1.3 Objetivos . . . . .	14
1.4 Estrutura do Trabalho . . . . .	16
<b>2 Fundamentos Teóricos - Algoritmos Evolutivos</b>	<b>17</b>
2.1 Algoritmos Genéticos . . . . .	17
2.2 Evolução Diferencial . . . . .	19
2.3 Algoritmos Culturais . . . . .	21
2.3.1 Espaço de Crenças . . . . .	22
2.3.1.1 Conhecimento Normativo . . . . .	22
2.3.1.2 Conhecimento Situacional . . . . .	22
2.3.1.3 Conhecimento Topográfico . . . . .	23
2.3.1.4 Domínio do Conhecimento . . . . .	23
2.3.1.5 Conhecimento Histórico . . . . .	23
2.3.2 Função de Influência . . . . .	24
2.3.3 Função de Aceitação . . . . .	24
2.3.4 Levantamento Bibliográfico . . . . .	24



<b>3</b>	<b>Fundamentos Teóricos - Composição Musical Inteligente</b>	<b>26</b>
3.1	Composição Musical Inteligente . . . . .	26
3.1.0.1	Notas Musicais . . . . .	26
3.1.0.2	Acidentes Musicais . . . . .	27
3.1.0.3	Durações Musicais . . . . .	27
3.1.0.4	Representações Musicais . . . . .	28
3.2	Composição Musical Inteligente Supervisionada . . . . .	28
3.2.0.5	Algoritmos Genéticos Aplicados na Composição Musical Inteligente . . . . .	28
3.2.0.6	Programação Genética Aplicada na Composição Musical Inteligente . . . . .	29
3.3	Síntese Sonora . . . . .	29
3.3.1	Ondas Sonoras . . . . .	30
3.3.1.1	Aparência das Ondas Sonoras . . . . .	31
3.3.1.2	Senoides, Frequência e Altura . . . . .	32
3.3.1.3	Construção de Senoides . . . . .	33
3.3.2	Ambientes de Desenvolvimento para Síntese Sonora . . . . .	34
3.3.2.1	Comparativo entre Ambientes de Desenvolvimento para Síntese Sonora . . . . .	35
<b>4</b>	<b>Modelo Desenvolvido</b>	<b>37</b>
4.1	Modelagem do Algoritmo Cultural com Evolução Diferencial . . . . .	37
4.1.1	Espaço de Crenças . . . . .	38
4.1.1.1	Conhecimento Situacional . . . . .	39
4.1.1.2	Conhecimento Normativo . . . . .	39
4.1.1.3	Conhecimento Topográfico . . . . .	40
4.1.1.4	Conhecimento Histórico . . . . .	41

4.1.2	Função de Aceitação . . . . .	43
4.1.3	Função de Influência Principal . . . . .	43
4.2	Modelagem da Composição Musical Inteligente . . . . .	43
4.2.1	Distância da Compressão Normalizada . . . . .	44
4.2.2	Função Objetivo . . . . .	45
4.2.3	Representação Genética . . . . .	45
4.2.3.1	Altura . . . . .	46
4.2.3.2	Oitava . . . . .	46
4.2.3.3	Duração . . . . .	46
4.2.3.4	Representação de Percussão . . . . .	47
<b>5</b>	<b>Experimentos, Resultados e Análises</b>	<b>50</b>
5.1	Evolução Diferencial . . . . .	50
5.2	Algoritmo Cultural Modelado . . . . .	51
5.3	Composição Musical Inteligente . . . . .	51
5.4	Estudo de Caso . . . . .	52
5.4.1	Melodias Codificadas . . . . .	53
5.5	Experimentos . . . . .	55
5.5.1	Estratégias da Evolução Diferencial . . . . .	56
5.5.2	Conhecimentos do Algoritmo Cultural Modelado . . . . .	58
5.6	Análise de Resultados . . . . .	62
<b>6</b>	<b>Conclusão</b>	<b>66</b>
	<b>Referências Bibliográficas</b>	<b>68</b>

## Lista de Tabelas

2.1	Conhecimentos Encontrados nos Artigos sobre Algoritmo Cultural . . . . .	25
3.1	Relação entre Tamanho de Onda, Frequência e Notas . . . . .	33
3.2	Relação entre Tamanho de Onda, Frequência e Notas . . . . .	35
4.1	Mapeamento da Altura das Notas . . . . .	47
4.2	Mapeamento da Duração das Notas . . . . .	48
4.3	Mapeamento das Peças de Bateria . . . . .	49
5.1	Comparativo de Resultados . . . . .	62

## Lista de Figuras

1.1	O <i>Framework</i> do Algoritmo Cultural. Fonte:(XUE; GUO, 2007) . . . . .	13
1.2	Processos, Entrada e Saídas Envolvidos. Fonte: Autoria Própria . . . . .	15
2.1	Fluxograma do Algoritmo Genético. Fonte: Autoria Própria . . . . .	18
2.2	Fluxograma da Evolução Diferencial. Fonte: Autoria Própria . . . . .	20
2.3	Fluxograma do Algoritmo Cultural. Fonte: Autoria Própria . . . . .	21
3.1	Acidentes Musicais. Fonte:(HEUSSENSTAMM, 1987) . . . . .	27
3.2	Durações Musicais. Fonte:(HEUSSENSTAMM, 1987) . . . . .	27
3.3	Sintetizador Roland Jupiter-80. Fonte:(ROLAND..., 2014) . . . . .	30
3.4	Onda Senoidal Simples. Fonte:(BALLORA, 2008) . . . . .	31
3.5	Onda Senoidal Complexa. Fonte:(BALLORA, 2008) . . . . .	32
3.6	Onda Senoidal ao Longo de 2 Segundos. Fonte:(BALLORA, 2008) . . . . .	32
3.7	Soma de Duas Ondas Senoidais. Fonte:(BALLORA, 2008) . . . . .	34
4.1	Fluxograma do Algoritmo Cultural com Evolução Diferencial. Fonte:(BECERRA; COELLO, 2004) . . . . .	38
4.2	Estrutura do Conhecimento Normativo . . . . .	39
4.3	Exemplo de Partição de dois Espaços Dimensionais por Árvore $k$ -d . . . . .	41
4.4	Estrutura do Conhecimento Histórico . . . . .	42
5.1	<i>Wim Mertens - Struggle for Pleasure</i> . . . . .	53
5.2	<i>Philip Glass - Truman Sleeps</i> . . . . .	54
5.3	Teste de Validação da Função Objetivo . . . . .	56
5.4	Gráfico de Convergência do ED/ <i>best</i> /1/ <i>bin</i> . . . . .	56

5.5	Gráfico de Convergência do ED/ <i>rand/1/bin</i> - 2 Guias . . . . .	57
5.6	Gráfico de Convergência do ED/ <i>rand – to – best/1/exp</i> . . . . .	58
5.7	Gráfico de Convergência do AC Situacional . . . . .	59
5.8	Gráfico de Convergência do AC Normativo . . . . .	59
5.9	Gráfico de Convergência do AC Histórico . . . . .	60
5.10	Gráfico de Convergência do AC Modelado - 2 Guias . . . . .	60
5.11	Gráfico de Convergência do ED/ <i>rand/1/bin</i> - 5 Guias . . . . .	61
5.12	Gráfico de Convergência do AC Modelado - 5 Guias . . . . .	62
5.13	Gráfico de Caixa para Comparação entre AC e ED - 2 Guias . . . . .	63
5.14	Gráfico de Caixa para Comparação entre AC e ED - 5 Guias . . . . .	63
5.15	Indivíduos Gerados pelo ED - 2 e 5 Guias . . . . .	64
5.16	Indivíduos Gerados pelo AC Modelado - 2 e 5 Guias . . . . .	65

# 1 Introdução

Composição Musical Inteligente (CMI) é um campo de investigação que envolvem pesquisadores e músicos. A implementação de sistemas que realizam esta tarefa, incorpora uma parte algorítmica que toma decisões sobre quais sons/notas serão ouvidas, quando serão ouvidas, por quanto tempo e quão alto (KALIAKATSOS-PAPAKOSTAS et al., 2013). Na literatura existem três subdivisões da composição musical inteligente:

- Composição inteligente não-supervisionada: Essa forma de composição musical inteligente é expressa por meio de regras simples que produzem saídas complexas, imprevisíveis, mas estruturadas, um comportamento que normalmente se assemelha a fenômenos naturais;
- Composição inteligente supervisionada: Algoritmos inteligentes são utilizados para modificar os parâmetros envolvidos no sistema de composição automática, de modo que atenda a critérios pré-definidos;
- Composição inteligente interativa: O sistema reconhece preferências humanas em tempo real e torna-se adaptado a ele, utilizando algoritmos inteligentes.

Na literatura existem vários exemplos de trabalhos relacionados com CMI supervisionada. (HORNER; GOLDBERG, 1991), aplicou um Algoritmo Genético (AG) para recriar composição musical, no que se tornou o primeiro trabalho explorando o uso de uma abordagem de Computação Evolucionária (CE) em uma tarefa relacionada com música. Desde então, um grande número de artigos sobre o assunto foram publicados (GRIF-FITH; TODD, 1999; PAPADOPOULOS; WIGGINS, 1999; MIRANDA; BILES, 2007). Hoje, a música utilizando CE compreende uma grande variedade de tarefas, incluindo a composição, harmonização, síntese de som, e improvisação (ÖZCAN; ERÇAL, 2008; DONNELLY; SHEPPARD, 2011; FORTIER; DYNE, 2011).

No presente trabalho, busca-se encontrar uma alternativa na CMI supervisionada. Realiza-se uma modelagem de um Algoritmo Cultural (AC), na tentativa de se obter melhores resultados em relação a CMI supervisionada encontrada na literatura, em que a maioria é modelada utilizando AGs. Para se ter como mensurar os resultados, ao final são realizadas

sínteses sonoras das composições.

Com o resultado desse trabalho, é possível de encontrar alternativas na composição musical tradicional. Técnicas evolutivas oferecem potencial para diminuir o esforço no trabalho do compositor ou engenheiro de som. Busca-se encontrar formas de compor automaticamente melodias em diversos gêneros musicais.

## 1.1 Composição Musical Inteligente Supervisionada

Os sistemas que pertencem a categoria de Composição Musical Inteligente (CMI) supervisionada utilizam algoritmos inteligentes, a fim de obter a capacidade de compor música sob algumas orientações que são chamadas de *features* (características) (MANARIS et al., 2007). Esses sistemas, por serem supervisionados possuem a capacidade de criar música com uma certa orientação estética e estilística, mas para isso eles incorporam os seguintes desafios:

- Criar uma interpretação de objetos matemáticos para a música (definição das funções de adaptação ou *fitness*);
- Aplicar um algoritmo inteligente para percorrer de forma ideal o espaço de busca dos objetos matemáticos;
- Selecionar um conjunto apropriado de *features* que descrevem a música desejada.

A seleção de *features* adequadas é de vital importância para o desempenho dos sistemas supervisionados (KALIAKATSOS-PAPAKOSTAS et al., 2013). Em um sentido abstrato, essas *features* devem fornecer marcos para o sistema compor músicas com as características desejadas, mas ao mesmo tempo permitir que seja possível a introdução de elementos novos para a música que está sendo composta. A seleção de *features* é, portanto, essencial para obter a base da música que será composta, mas não devem determinar exageradamente no resultado da composição.

O AC desenvolvido nesse trabalho é essencial para a seleção das *features* adequadas para cada gênero musical composto. Ele foi desenvolvido com base na melhor estratégia encontrada para o algoritmo evolutivo Evolução Diferencial, essa é uma definição importante. Conforme é visto durante o trabalho a codificação da música gerada envolve um certo nú-

mero de variáveis inteiras, dificultando assim a utilização de algoritmos tradicionais como os Algoritmos Genéticos (AGs), pois necessitam de uma codificação binária e extensa.

## 1.2 Algoritmos Culturais

Os Algoritmos Culturais (ACs) são algoritmos evolucionários baseados no processo de evolução cultural da humanidade (REYNOLDS, 1994; REYNOLDS; SALEEM, 2001; REYNOLDS; PENG, 2004; REYNOLDS; PENG, 2005). Os ACs foram propostos por Robert Reynolds em 1994 como um complemento a metáfora evolutiva utilizada na computação evolutiva, metáfora essa que se concentra nos aspectos genéticos da evolução e na teoria da seleção natural proposta por Darwin. Em contrapartida, os algoritmos culturais baseiam-se em teorias sociais e arqueológicas que modelam a evolução cultural dos povos (BECERRA; COELLO, 2004).

Conforme mostrado na Figura 1.1, os algoritmos culturais são compostos por dois componentes principais: o espaço populacional e o espaço de crenças.

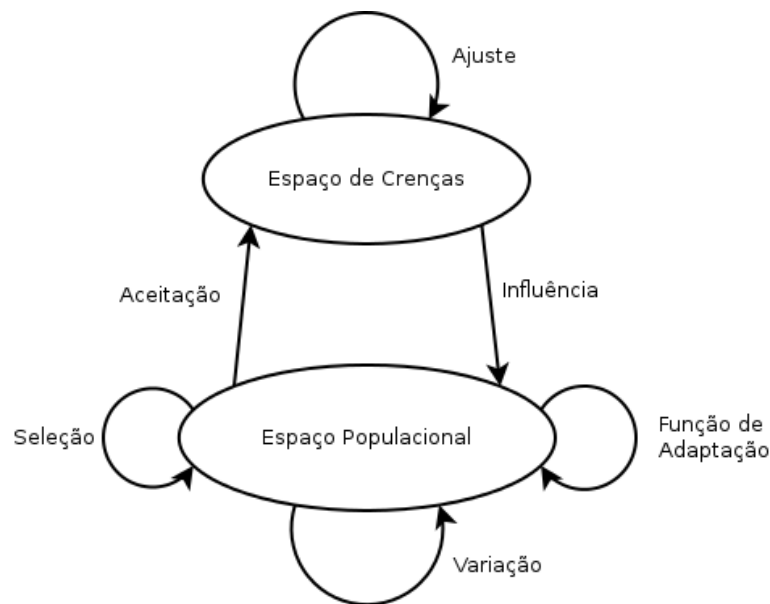


Figura 1.1: O *Framework* do Algoritmo Cultural. Fonte:(XUE; GUO, 2007)

No espaço populacional são representadas as características e comportamentos dos indivíduos (solução candidata). Essa representação pode ser feita por meio de qualquer técnica que faça uso de uma população de indivíduos, como é o caso dos algoritmos genéticos que são algoritmos estocásticos de busca inspirados no comportamento das espécies na natureza (COELLO; BECERRA, 2003).



O espaço de crenças é o repositório de símbolos que representam os conhecimentos adquiridos pelo espaço populacional ao longo do processo evolutivo. O espaço de crenças permite que os indivíduos sejam removidos da população sem que o conhecimento por eles adquiridos seja perdido. O espaço de crenças foi criado para guiar os indivíduos na direção das melhores regiões do espaço de busca (XUE; GUO, 2007).

Os protocolos de comunicação ditam as regras sobre quais indivíduos podem contribuir com conhecimentos para o espaço de crenças (função de aceitação) e como o espaço de crenças vai influenciar a geração de novos indivíduos (função de influência).

Na função de aceitação são selecionados indivíduos que irão influenciar o espaço de crenças atual. A função de influência estabelece como o conhecimento armazenado no espaço de crenças vai interferir nos operadores do espaço populacional. Geralmente é utilizada uma função de influência para cada tipo de conhecimento armazenado.

Com AC busca-se principalmente encontrar e modelar os conhecimentos necessários para criação de CMI supervisionada, o que pode ser um grande diferencial ao que é possível de encontrar na literatura.

## 1.3 Objetivos

O objetivo geral deste trabalho é encontrar uma solução viável para composição musical inteligente supervisionada utilizando algoritmo cultural. Para atingi-lo é preciso cumprir com alguns objetivos específicos:

- Compreender os fundamentos teóricos de algoritmos evolutivos;
- Compreender os fundamentos teóricos dos algoritmos culturais;
- Levantamento bibliográfico sobre as estratégias para composição musical inteligente e supervisionada utilizando algoritmos evolucionários;
- Levantamento bibliográfico sobre as estratégias atuais utilizadas para síntese sonora e escolher uma metodologia e ferramentas;
- Implementar o algoritmo cultural para realização da composição musical inteligente e supervisionada;
- Realizar testes utilizando evolução diferencial e o algoritmo cultural desenvolvido;

- Realizar uma análise comparativa entre o resultado de todos os testes.

O trabalho visa atribuir as tarefas que envolvem composição musical e síntese sonora conforme é mostrado na Figura 1.2.

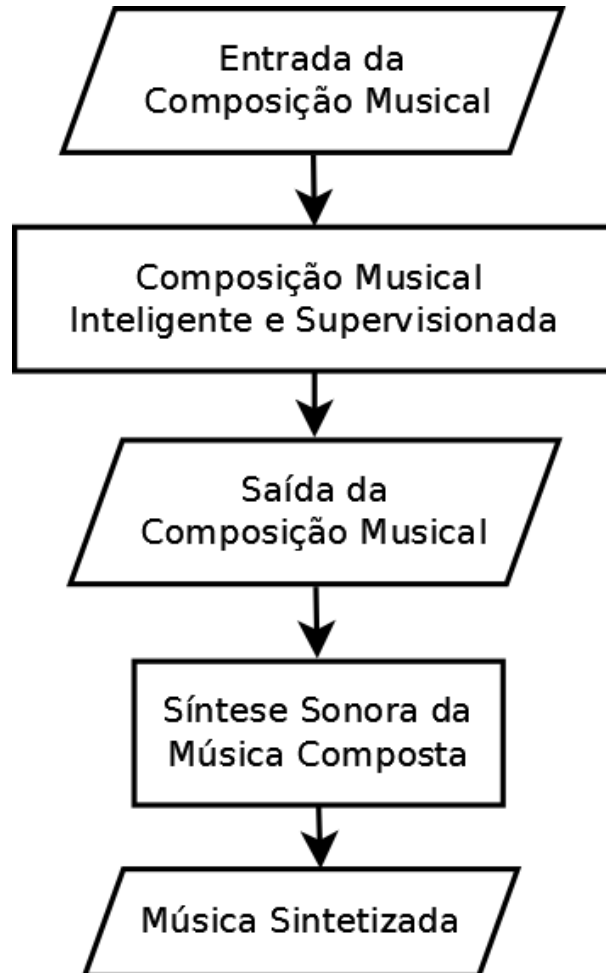


Figura 1.2: Processos, Entrada e Saídas Envolvidos. Fonte: Autoria Própria

As entradas da composição musical são *features* definidas no trabalho. Essas são na forma de sequência de notas.

Ex.: A(Lá),B(Si),C(Dó),F(Fá),G(Sol),C(Dó).

A CMI supervisionada é na forma do AC modelado no trabalho, que traduz as entradas e composições musicais.

A saída da composição musical é na forma de sequência de notas codificadas em inteiro para a entrada do sintetizador conforme o modelo definido.

Ex.: 10(Lá),12(Si),1(Dó),6(Fá),8(Sol),1(Dó).

A síntese sonora da composição é realizada por meio de uma ferramenta para síntese de som escolhida no trabalho que traduz a música codificada em som, sendo possível de escutá-la ao final do processo.

---

## 1.4 Estrutura do Trabalho

A base do trabalho a seguir foi obtida por meio de pesquisas bibliográficas citadas no texto. A construção do trabalho é definida pelos métodos positivista (COMTE, 1868) e empírico (LOCKE, 1700).

O trabalho é dividido nas seguintes seções: no segundo capítulo estão os fundamentos teóricos dos algoritmos evolutivos relacionados com o projeto, no terceiro capítulo estão os fundamentos teóricos da composição musical inteligente e síntese sonora, no quarto capítulo está o modelo proposto para o projeto, no quinto capítulo os experimentos, resultados e análises e no sexto capítulo está a conclusão.

## 2 Fundamentos Teóricos - Algoritmos Evolutivos

O presente capítulo apresenta os fundamentos teóricos de três algoritmos evolucionários, com objetivo de estudar, descrever e comparar. Os algoritmos estudados foram escolhidos conforme sua importância para o modelo proposto, seja a título de comparação ou para utilização na modelagem do problema. A escolha de descrever os algoritmos genéticos é a critério de comparação. A evolução diferencial e o algoritmo cultural foram modelados para o problema. A seguir são descritos os algoritmos: algoritmos genéticos, evolução diferencial, algoritmos culturais.

### 2.1 Algoritmos Genéticos

Os Algoritmos Genéticos (AGs) são métodos de busca estocástica que se baseiam na teoria da evolução de Charles Darwin (MITCHELL, 1996). Esses métodos operam com conjuntos de candidatos, chamados de população, que são constantemente modificados utilizando dois princípios básicos da evolução natural das espécies: seleção e variação. Estes princípios tentam representar a competição onde os indivíduos mais aptos se reproduzem e repassam seu material genético às gerações futuras, e os indivíduos menos aptos tendem a desaparecer da população (MITCHELL, 1996).

O termo Algoritmo Genético (AG) foi introduzido por John Holland em 1975. Os AGs possuem: populações de cromossomos, seleção de acordo com a aptidão, cruzamento para produzir novos descendentes e a mutação aleatória para maior variabilidade genética (MITCHELL, 1996).

Os cromossomos em AGs normalmente tomam a forma de sequências de bits. Cada locus (local onde fica localizado um gene) no cromossomo possui dois alelos (formas alternativas de um mesmo gene) possíveis: 0 e 1 (MITCHELL, 1996). Cada cromossomo pode ser pensado como um ponto no espaço de busca de soluções candidatas. O AG processa populações de cromossomos, substituindo sucessivamente uma população com a outra. O AG na maioria das vezes requer uma função de *fitness* que atribui uma pontuação (*fit-*

ness) para cada cromossomo na população atual. A aptidão de um cromossomo depende de quão bem o cromossomo resolve o problema.



Figura 2.1: Fluxograma do Algoritmo Genético. Fonte: Autoria Própria

Na Figura 2.1 é representado o fluxograma de um AG. Nas seções seguintes são descritas cada etapa do algoritmo.

- **Representação dos Cromossomos:** O algoritmo em sua forma mais simples representa cada cromossomo como uma cadeia de bits. Normalmente, os parâmetros numéricos podem ser representados por números inteiros, embora seja possível a utilização de representações contínuas (WHITLEY, 1994).
- **Inicialização da População:** Inicialmente muitas possíveis soluções são geradas aleatoriamente para formar a população inicial. O tamanho da população depende da natureza do problema. Tradicionalmente a população é gerada randomicamente, permitindo toda a gama de possíveis soluções. Ocasionalmente, por exemplo, em, as soluções podem ser estrategicamente colocadas em áreas onde as melhores soluções possam ser encontradas (WHITLEY, 1994).

- **Avaliação do Fitness:** A função *fitness* é definida sobre a representação genética e mede a qualidade da solução representada. A função de *fitness* varia conforme o problema. A adaptação de cada solução na população é avaliada por meio da função de *fitness* a cada geração do algoritmo, os melhores indivíduos são selecionados para a próxima geração (WHITLEY, 1994). A avaliação é o cálculo da função de *fitness*, que por sua vez é a minimização ou maximização da chamada função objetivo.
- **Seleção:** A partir da medição da função de *fitness*, nessa etapa são selecionados cromossomos na população para a reprodução. Quanto melhor o cromossomo, melhor são as chances dele ser selecionado para se reproduzir (WHITLEY, 1994).
- **Crossover:** Este operador escolhe aleatoriamente um locus entre dois cromossomos e troca os subseqüentes antes e depois daquele locus, entre dois cromossomos para criar dois filhos (WHITLEY, 1994). Por exemplo, nas cadeias de 10000100 e 11111111 se for escolhido o terceiro locus irá gerar após o *crossover* os seguintes filhos: 10011111 e 11100100. O operador de *crossover* imita a recombinação biológica entre dois organismos haploides (cromossomo único).
- **Mutação:** Este operador vira aleatoriamente alguns dos bits em um cromossomo (WHITLEY, 1994). Por exemplo, a cadeia 00000100 pode sofrer mutação na sua segunda posição, para se obter 01000100. A mutação pode ocorrer em cada posição de bit de uma cadeia com alguma probabilidade, geralmente muito pequeno (por exemplo, 0.001).

Os algoritmos genéticos são eficientes quando se manipula codificações inteiras transformando para binário. Porém para essa modelagem a codificação binária é inviável, necessitando assim encontrar uma alternativa aos algoritmos genéticos que utilize codificação com valores reais.

## 2.2 Evolução Diferencial

A Evolução Diferencial (ED) é um método que otimiza um problema de maneira iterativa tentando melhorar possíveis soluções em relação a uma determinada medida de qualidade (STORN; PRICE, 1995). No entanto, igualmente aos AGs, a ED não garante que uma solução ideal seja sempre encontrada. O principal diferencial da ED para os

AGs é a utilidade em funções multidimensionais com valores reais (ou contínuos), que se obtém um desempenho muito superior (VESTERSTROM; THOMSEN, 2004).

A ED é um método criado originalmente por Storn e Price em 1995 (STORN; PRICE, 1995). Esse algoritmo trabalha com candidatos a soluções chamados agentes. Esses agentes são vetores com variáveis reais que são movimentados no espaço de busca, utilizando fórmulas matemáticas para combinar as posições dos agentes existentes na população (QIN et al., 2009). Se a nova posição de um agente é uma melhoria aceita, ele passa a fazer parte da população, caso contrário, a nova posição é simplesmente descartada. O processo é repetido até que se encontre uma solução ao menos satisfatória.

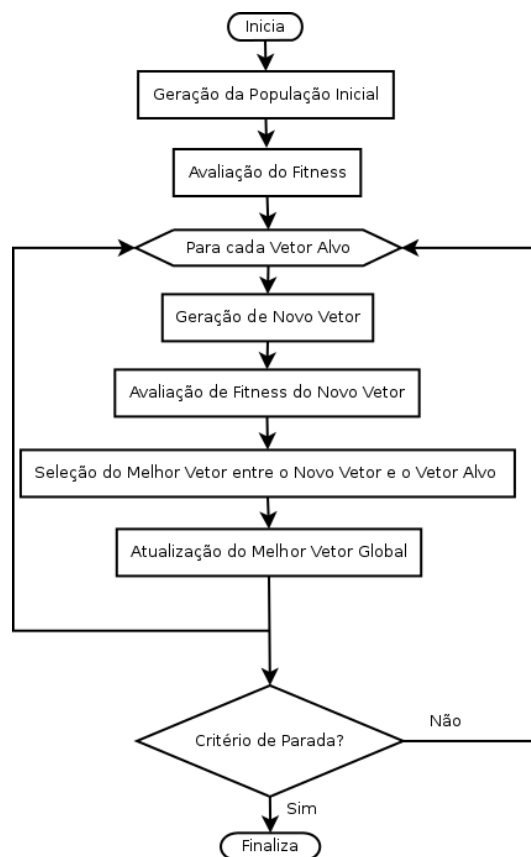


Figura 2.2: Fluxograma da Evolução Diferencial. Fonte: Autoria Própria

Na Figura 2.2 é representado o fluxograma de um ED. Ele se comporta de forma semelhante aos AGs. Porém existe o diferencial de que os indivíduos (ou agentes) são vetores, onde cada vetor é uma posição no espaço de busca. Essas posições são afetadas conforme os indivíduos são substituídos por melhores.

Por consequência do desempenho eficiente da ED para explorar o espaço de busca e também por utilizar variáveis reais ao invés da codificação binária, no desenvolvimento do trabalho foi utilizado ED como alternativa aos AGs tradicionais.

## 2.3 Algoritmos Culturais

Os Algoritmos Culturais (AC) consistem de uma população social e um espaço de crenças (REYNOLDS, 1994). Indivíduos selecionados de um espaço populacional contribuem com o conhecimento cultural por meio da função de aceitação. O conhecimento cultural está inserido no espaço de crenças onde é armazenado e atualizado baseado em experiências individuais e seus sucessos e falhas. Desta forma, o conhecimento cultural controla a evolução da população por meio da função de influência. Um AC assim estabelece um *framework* que acumula e comunica conhecimento para permitir adaptação própria quanto ao espaço populacional e ao espaço de crenças.

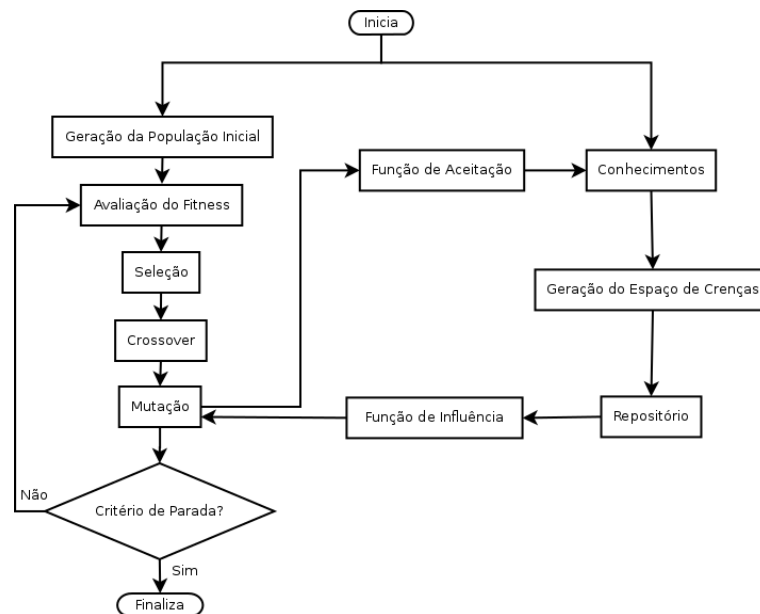


Figura 2.3: Fluxograma do Algoritmo Cultural. Fonte: Autoria Pr pria

Na Figura 2.3   demonstrado o fluxograma do AC tradicional. O AC tradicional   basicamente um AG com um espa o de cren as que influencia (fun o de influ ncia) na evolu o e   influenciado (fun o de aceita o) pela evolu o.

Para a cria o do espa o de cren as existem conhecimentos que s o pr -definidos ou obtidos durante a evolu o. Ap s a cria o das cren as elas s o armazenadas em um reposit rio de cren as que influenciam na evolu o a cada gera o.

Nas se es seguintes s o descritos todos os conhecimentos utilizados nos ACs.



### 2.3.1 Espaço de Crenças

Para os problemas de otimização, o espaço de crenças é modificado para conter diversos tipos de conhecimentos obtidos durante a evolução, que é usado em gerações sucessivas para criar melhores indivíduos. Existem pelo menos cinco categorias básicas do conhecimento cultural que são importantes no espaço de crenças de qualquer modelo de evolução cultural: normativo, situacional, topográfico, histórico ou temporal e domínio do conhecimento (SRINIVASAN; RAMAKRISHNAN, 2012b).

#### 2.3.1.1 Conhecimento Normativo

O Conhecimento Normativo (CN) contém os atributos (variáveis do problema) e os possíveis valores que esses atributos podem tomar. Estas informações são recolhidas a partir da função de aceitação ou conhecimentos pré-definidos. A fonte do conhecimento normativo é usada para armazenar valores máximos e mínimos para os atributos numéricos. Para cada atributo nominal ou discreto, uma lista separada armazena possíveis valores que os atributos podem tomar. O conhecimento normativo é atualizado a partir da função de aceitação e utilizado pelos agentes durante a mutação (SRINIVASAN; RAMAKRISHNAN, 2012b).

#### 2.3.1.2 Conhecimento Situacional

O Conhecimento Situacional (CS) consiste no melhor exemplar encontrado ao longo do processo evolutivo. Ele representa um líder para os outros indivíduos a seguir. Desta forma, os agentes usam o exemplo, em vez de um indivíduo escolhido aleatoriamente para a recombinação. Este conhecimento pode ser atualizado por meio do armazenamento dos melhores exemplos no fim de cada geração. Então são usados estes exemplos para a escolha de indivíduos para a reprodução. Além disso, o usuário pode especificar esquemas de condições para certos atributos que podem ser utilizados para a busca de indivíduos semelhantes ou diferentes que seja de interesse do usuário (SRINIVASAN; RAMAKRISHNAN, 2012b).

### 2.3.1.3 Conhecimento Topográfico

O Conhecimento Topográfico (CT) é uma estratégia para manter a diversidade, principalmente em sistemas evolutivos multiobjetivos. O CT é utilizado para manter as soluções distribuídas uniformemente no conjunto ótimo de Pareto (quando não é possível de melhorar a situação), em vez de reunir soluções em apenas uma pequena região. Acasalamento restrito, em que o acasalamento é permitido somente quando a distância entre os dois pais é grande o suficiente (SRINIVASAN; RAMAKRISHNAN, 2012c).

### 2.3.1.4 Domínio do Conhecimento

O domínio do conhecimento (DC) contém o vetor de valores de métricas para cada conhecimento. Indivíduos produzidos são avaliados no final de cada geração e o vetor de *fitness* calculado. O DC é atualizado com estes vetores de *fitness*. Os vetores de *fitness* em DC são comparados uns com os outros usando estratégia de otimização de Pareto para escolher os indivíduos de elite no fim de cada geração. Os indivíduos de elite, assim escolhidos são armazenados no conhecimento histórico (SRINIVASAN; RAMAKRISHNAN, 2012c).

### 2.3.1.5 Conhecimento Histórico

O Conhecimento Histórico (CH) registra em uma lista os melhores indivíduos, e são atualizados no final de cada geração. Algoritmos evolutivos são conhecidos por utilizar menos memória, uma vez que não retêm memória das gerações anteriores. No entanto tentativas foram feitas para manter os indivíduos de elite em cada geração como uma população elite separada, tendo-se o máximo de memória possível para os algoritmos evolucionários. O algoritmo cultural utiliza a memória para a estratégia evolutiva de uma forma sistemática, utilizando as diferentes fontes de conhecimento. CH pode ser usado para armazenar os indivíduos de elite de cada geração, mantendo assim a memória por meio das gerações (SRINIVASAN; RAMAKRISHNAN, 2012c).

### 2.3.2 Função de Influência

A Função de Influência (FI) decide quais fontes de conhecimento vão influenciar os indivíduos (SRINIVASAN; RAMAKRISHNAN, 2012b). A seleção é feita, por exemplo, pela roleta com base no desempenho das fontes de conhecimento nas gerações anteriormente usadas.

### 2.3.3 Função de Aceitação

A Função de Aceitação (FA) determina o que os indivíduos e seus comportamentos podem afetar no espaço de crenças (SRINIVASAN; RAMAKRISHNAN, 2012c). Com base em parâmetros selecionados, por exemplo, uma porcentagem dos melhores desempenhos são aceitos.

### 2.3.4 Levantamento Bibliográfico

Foram levantados alguns trabalhos sobre ACs. Não foi possível encontrar na literatura nenhum AC aplicado na geração de composição musical inteligente. Porém foram selecionados 14 que tinham informações pertinentes para auxílio em futuras decisões. No caso, foi realizada uma comparação entre os conhecimentos utilizados em cada um dos trabalhos. Na tabela a seguir são descritos os conhecimentos na seguinte ordem: Conhecimento Normativo (CN), Conhecimento Situacional (CS), Conhecimento Topográfico (CT), Domínio do Conhecimento (DC) e Conhecimento Histórico (CH).

Conforme verificado na Tabela 2.1 é possível perceber uma tendência em se utilizar apenas o CN e o CS, isso ocorre provavelmente por serem os principais conhecimentos dentro do espaço de crenças de um AC. E dentre todos os trabalhos levantados apenas o trabalho de Becerra e Coello (2004) possuía uma pesquisa aprofundada na parte dos conhecimentos, tendo comprovada e demonstrada toda a modelagem matemática dos cinco conhecimentos. Este por sua vez é estudado mais a fundo e aproveitado para guiar nas decisões deste trabalho.

Tabela 2.1: Conhecimentos Encontrados nos Artigos sobre Algoritmo Cultural

Artigos	CN	CS	CT	DC	CH
(BECERRA; COELLO, 2004)	X	X	X	X	X
(HO; TAY, 2004)	-	-	-	X	-
(BECERRA; COELLO, 2005)	-	X	-	X	-
(XUE; GUO, 2007)	X	X	X	X	X
(RODRIGUES, 2007)	X	X	-	-	-
(COELHO et al., 2009)	X	X	-	-	-
(GUO; LIU, 2011)	X	X	-	-	-
(MATHIYALAGAN et al., 2011)	X	X	-	-	-
(HE; XU, 2011)	X	X	-	-	-
(BHATTACHARYA et al., 2012)	X	X	-	-	-
(SRINIVASAN; RAMAKRISHNAN, 2012a)	X	X	X	X	X
(SRINIVASAN; RAMAKRISHNAN, 2012b)	X	X	X	X	X
(HOCHREITER; WALDHAUSER, 2014)	X	X	-	-	-
(KHAN et al., 2014)	X	X	-	-	-

## 3 Fundamentos Teóricos - Composição Musical Inteligente

O presente capítulo apresenta a fundamentação teórica sobre a Composição Musical Inteligente (CMI).

### 3.1 Composição Musical Inteligente

Para introduzir o assunto da CMI, se fazem necessários conhecimentos básicos de notação musical e de teoria musical. As seções seguintes descrevem brevemente esses fundamentos musicais.

Melodia, ritmo e harmonia são considerados na literatura como os três elementos fundamentais da música (WHITE, 1976).

Melodia é uma sucessão de sons e silêncios coerentes, que se desenvolve em uma sequência linear (RANDEL, 2003). Pode ser considerada também o primeiro plano de uma música. Ritmo é um movimento marcado pela sucessão regulada de elementos fortes ou fracos (RANDEL, 2003). Pode ser considerado na música como a batida ou marcação do tempo musical.

Harmonia é o uso de notas, ou acordes (combinação de três ou mais notas) simultâneos. O estudo da harmonia envolve a construção de acordes, progressões de acordes e seus princípios (RANDEL, 2003). Diz-se que harmonia é o aspecto vertical da música, enquanto a melodia é o aspecto horizontal (RANDEL, 2003).

A pesquisa visa realizar composição musical de melodia, deixando ritmo e harmonia para trabalhos futuros.

#### 3.1.0.1 Notas Musicais

Na música ocidental as notas musicais são nomeadas da seguinte forma: Dó (C), Ré (D), Mi (E), Fá (F), Sol (G), Lá (A) e Si (B). Cada nota representa uma faixa de frequências, sendo que as frequências baixas realizam sons graves e as frequências altas

realizam sons agudos.

### 3.1.0.2 Acidentes Musicais

Além das notas musicais básicas, podem ser criadas notas intermediárias que são necessárias para se completar todo o conjunto de notas. Para a notação dessas notas intermediárias são utilizados os acidentes musicais (descrito na Figura 3.1). A distância padrão entre duas notas é chamada de tom e a menor distância entre duas notas é chamado de semitom, logo, dois semitons fazem um tom. Um tempo sem som é chamado de pausa. Notas e pausas tem um tamanho (uma duração no tempo).

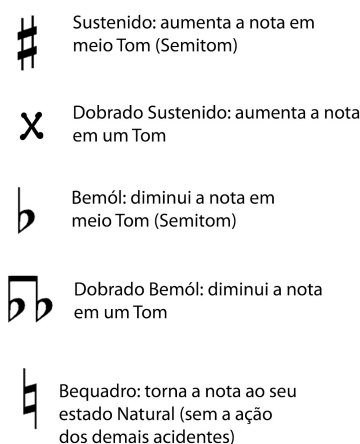


Figura 3.1: Acidentes Musicais. Fonte:(HEUSSENSTAMM, 1987)

### 3.1.0.3 Durações Musicais

Conforme descrito na Figura 3.2, existem sete durações de tempo diferentes (de 1, que corresponde ao maior tempo, até 1/64), cada duração tem o dobro do tempo da seguinte. Seus nomes: semibreve, mínima, semínima, colcheia, semicolcheia, fusa, semifusa.



Figura 3.2: Durações Musicais. Fonte:(HEUSSENSTAMM, 1987)

#### 3.1.0.4 Representações Musicais

Uma peça de música pode ser representada de várias formas: por partituras, por tablaturas, por cifras (A-G), por números (1-88) correspondendo as notas de um piano, por frequências das notas. Além de sistemas de codificação para se gravar ou reproduzir uma música em um computador, tais como, MIDI, WAVE, MP3, entre outros.

## 3.2 Composição Musical Inteligente Supervisionada

Composição Musical Inteligente (CMI) supervisionada é realizada por meio de um modelo que é construído por um programador compositor (KALIAKATSOS-PAPAKOSTAS et al., 2013). Esse modelo pode incorporar um conjunto de parâmetros que definem o estilo da composição. Nesse modelo é importante que a combinação correta desses parâmetros seja definida para a criação automática da música, então poderá exibir certas características (*features*) e valores estéticos. Conforme a pesquisa realizada em CMI supervisionada, pôde-se dividir os trabalhos relacionados em dois grupos: Algoritmos Genéticos (AG) e Programação Genética (PG). Pois foram encontrados apenas esses dois tipos de algoritmos para CMI supervisionada.

#### 3.2.0.5 Algoritmos Genéticos Aplicados na Composição Musical Inteligente

A utilização de AGs é uma das formas de providenciar valores adequados para os parâmetros musicais, dando uma medida qualitativa do que se busca na música produzida. Assim, os problemas do programador compositor estão relacionados não somente com a formulação de um modelo paramétrico adequado, mas também para a formalização de medidas que descrevem com precisão o estilo de música alvo.

O AG por ser inspirado na evolução natural e atuar de forma iterativa é classificado por inicialmente tentar adivinhar um conjunto de soluções aleatórias para posteriormente encontrar uma solução adequada após várias gerações. A noção fundamental no AG é a descrição precisa e informativa do que é uma solução ótima. No caso das CMI supervisionadas é preciso indicar corretamente quais atributos a música composta deve abranger. Entre os primeiros trabalhos para CMI supervisionada utilizando critérios musicais objetivos para atribuição das avaliações de aptidão está o trabalho de Papadopoulos e Wiggins

(1999). Nesse trabalho, foi desenvolvido um sistema que compunha solos de jazz durante uma determinada progressão de acordes. As soluções para o problema eram as melodias em si, especificamente pares de altura/duração, e após uma inicialização aleatória, no AG foram aplicados alguns operadores genéticos com sentidos musicais, sendo possível assim novas gerações com melodias soluções (PAPADOPOULOS; WIGGINS, 1999). O processo de avaliação da solução de cada candidato foi baseado em oito indicadores de avaliação, emprestado da teoria musical. Após este primeiro trabalho, tiveram diversos outros projetos que se utilizaram AGs para criação de CMI supervisionada (GRIFFITH; TODD, 1999; ALFONSECA et al., 2006; MIRANDA; BILES, 2007; ÖZCAN; ERÇAL, 2008; DONNELLY; SHEPPARD, 2011; FORTIER; DYNE, 2011).

#### **3.2.0.6 Programação Genética Aplicada na Composição Musical Inteligente**

A Programação Genética (PG) funciona sob o mesmo princípio evolutivo do AG, tendo inicialmente soluções aleatória evoluindo até chegar na solução ótima. A diferença entre PG e AG está na formulação do problema. No AG, as metas estão no processo de otimização de parâmetros do modelo, enquanto que no PG é possível de otimizar o modelo em si, uma vez que as populações de possíveis soluções incorporam programas inteiros que realmente formam o modelo. Ou seja, enquanto nos AGs são otimizadas as variáveis para o problema nas PGs são otimizados os algoritmos que resolvem o problema.

Embora o PG ofereça um aspecto totalmente novo em relação à formulação do problema, a avaliação da aptidão desses programas continua a ser uma questão importante. Em Spector e Alpern (1994), um esquema de PG que atuou em uma melodia inicial, realizando várias operações musicais, como retrógrado e de transposição, produzindo novas melodias (SPECTOR; ALPERN, 1994). Após este primeiro trabalho, tiveram alguns outros projetos que utilizaram PGs para criação de CMI supervisionada (SPECTOR; ALPERN, 1995; PHON-AMNUAISUK et al., 2007; MANARIS et al., 2007)

### **3.3 Síntese Sonora**

Síntese sonora é a técnica de geração de som usando equipamentos eletrônicos ou de software, a partir do zero (ROADS, 1996). O uso mais comum de síntese é musical, quando os instrumentos eletrônicos chamados sintetizadores são usados na execução e



gravação de música. Síntese de som tem muitas aplicações, tanto acadêmicas ou artísticas, e geralmente são utilizados sintetizadores e métodos de síntese para:

- Gerar timbres interessantes incapazes de serem produzidos acusticamente;
- Recriar ou modelar sons de instrumentos acústicos ou sons do mundo real;
- Facilitar a automação de sistemas e processos (exemplo, *softwares text-to-speech*).

Com o objetivo de facilitar a síntese sonora, foram desenvolvidos hardwares específicos para este fim, chamados sintetizadores (ROADS, 1996). Na Figura 3.3 tem um exemplo de um hardware totalmente dedicado a síntese de som.



Figura 3.3: Sintetizador Roland Jupiter-80. Fonte:(ROLAND..., 2014)

### 3.3.1 Ondas Sonoras

A definição de métrica (acústica) do som é a variação em ondas de pressão e densidade causada pela propagação das ondas através de um meio (BALLORA, 2008). Entre cerca de 25 Hz e 18 kHz, os sistemas auditivos humanos sentem essas ondas, pois fazem o tímpano se mover. Este movimento mecânico é traduzido em sinais eletroquímicos na cóclea (parte auditiva do ouvido interno) como impulsos nervosos, e enviado para a região auditiva do cérebro para análise. As ondas sonoras, sendo a variação na pressão de ar ao longo de um tempo, pode ser representada como uma voltagem variável ou um fluxo de dados ao longo de um tempo. Esta é uma representação de 'tempo/amplitude' do som, também conhecida como a linha de tempo da amplitude. A amplitude representa o deslocamento molecular causada pelas mudanças na pressão do ar. No domínio digital, a amplitude é tipicamente representado por um valor entre 1 e -1 em que 1 e -1 representam amplitudes máximas positivas e negativas do sinal, e 0 representa a amplitude zero.

### 3.3.1.1 Aparência das Ondas Sonoras

A forma de onda na Figura 3.4 é chamada de onda senoidal. Ondas senoidais podem ser consideradas blocos de construção fundamentais do som (BALLORA, 2008). A figura demonstra que a amplitude varia ao longo do tempo, mas que o padrão de variação se repete periodicamente. A forma de onda na Figura 3.5 é mais complexa do que a senoidal.

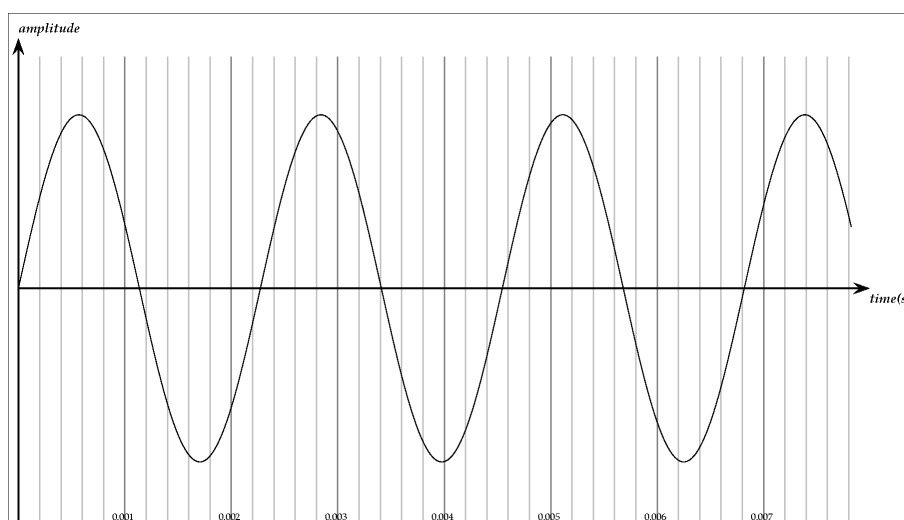


Figura 3.4: Onda Senoidal Simples. Fonte:(BALLORA, 2008)

Há altos e baixos de diferentes amplitudes, e, embora o padrão se repete ao longo do tempo é mais difícil de detectar (BALLORA, 2008). Da mesma maneira que uma onda senoidal se comporta de uma maneira simples e soa simples, essa comporta-se de maneira que o som possui uma maior complexidade. Por esta razão, sons complexos, detalhados, que mudam ao longo do tempo, muitas vezes não têm características discerníveis quando se avalia, não há nenhum padrão de repetição ou comportamento que se pode usar para nos dizer algo sobre o som. Na Figura 3.6 é dada uma visão de um som ao longo de cerca de 2 segundos. A partir dessa perspectiva, pode-se perceber a forma geral as mudanças de amplitude de som ao longo do tempo; em particular, as partes com alta amplitude podem facilmente ser percebidos, por exemplo, como batidas de tambor, pois aparecem de repente e caem em amplitude muito rapidamente (BALLORA, 2008). Pode ser muito difícil dizer qual tipo de instrumento está sendo utilizado, se esse som foi visto ao longo do intervalo de poucos milissegundos. A partir disso, pode-se concluir que a perspectivas do intervalo curto ou longo mostram diferentes tipos de informações e que a seleção da perspectiva correta pode ser crucial para atender as necessidades corretamente.

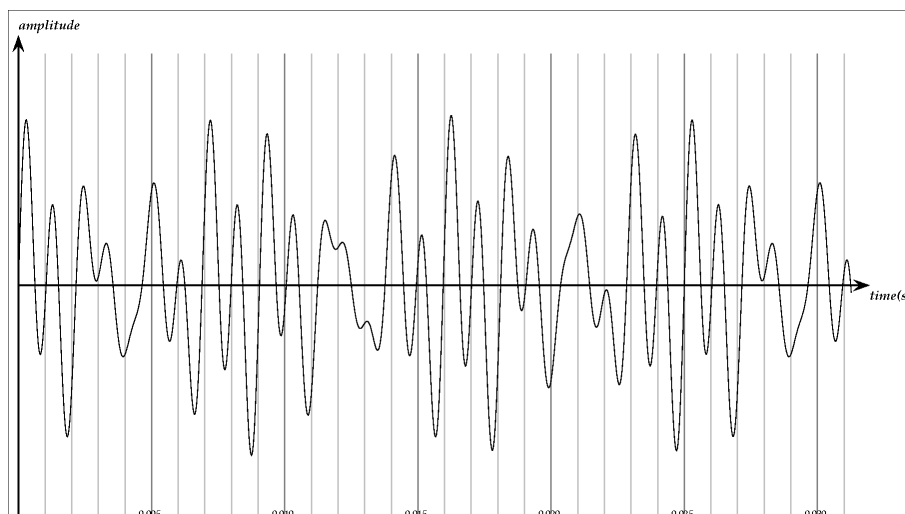


Figura 3.5: Onda Senoidal Complexa. Fonte:(BALLORA, 2008)

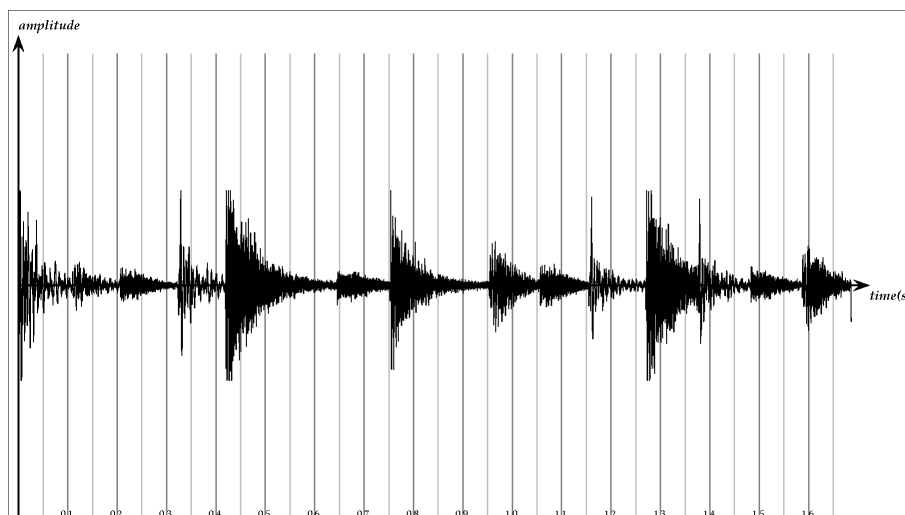


Figura 3.6: Onda Senoidal ao Longo de 2 Segundos. Fonte:(BALLORA, 2008)

### 3.3.1.2 Senoides, Frequência e Altura

A onda senoidal tem uma forma periódica que se repete a cada  $T$  segundos, o qual é conhecido como período, ciclo (BALLORA, 2008). A onda também tem uma amplitude positiva máxima,  $A$ , e uma amplitude máxima negativa,  $-A$ . A frequência,  $f$ , de uma onda senoidal é o número de ciclos por segundo e é medida em *Hertz* (Hz). Podemos obter a frequência do comprimento de onda a partir da seguinte equação:

$$f = \frac{1}{T}. \quad (3.1)$$

Além disso, pode-se expressar uma onda senoidal com a seguinte forma matemática (com ângulos em radianos):

$$p(t) = A \sin\left(\frac{2\pi t}{T}\right) = A \sin(2\pi ft). \quad (3.2)$$

Sendo psicométrico, frequências mais elevadas (por exemplo, acima de 1,5 kHz) são frequentemente associados com palavras como 'brilho', enquanto que as frequências mais baixas (por exemplo, abaixo de 200 Hz) são frequentemente associados com "profundidade" ou "baixo". A faixa intermédia pode ser associado com o termo "calor" (BALLORA, 2008). Por exemplo, um instrumento como uma guitarra elétrica tocada limpa pode ser chamado de "claro" ou "agudo" enquanto um contrabaixo acústico pode ser referido como "escuro" ou "grave". Termos como estes não são quantidades objetivas que podemos medir com precisão, mas são muitas vezes utilizados para descrever o timbre, ou cor de tom de um som particular. As várias amplitudes de frequências presentes em um som, e sua evolução ao longo do tempo são os principais fatores associados ao timbre, e existem infinitos tons de timbre que pode ser alcançado por meio de combinações de diferentes frequências que compõem um som. Na Tabela 3.1 tem uma relação entre tamanho de onda, frequência e notas na escala padrão ocidental:

Tabela 3.1: Relação entre Tamanho de Onda, Frequência e Notas

Tamanho de Onda(t)	Frequência (Hz)	Nota
156.82 cm	220.0	A3
139.71 cm	246.94	B3
131.87 cm	261.63	C4
117.48 cm	293.66	D4
104.66 cm	329.63	E4
98.79 cm	349.23	F4
88.01 cm	392.0	G4
78.41 cm	440.0	A4

### 3.3.1.3 Construção de Senoides

Utilizando a análise de Fourier, ondas senoidais podem ser consideradas as componentes fundamentais do som, uma vez que uma única onda senoidal é uma única frequência (BALLORA, 2008). Na análise Fourier, combinando senoides de diferentes frequência, ap-

tidão e fase, pode-se recriar o espectro de frequência de qualquer som. Da mesma forma, os sons complexos podem ser analisados em termos de frequência, amplitude e fase.

A Figura 3.7 demonstra a soma de duas ondas senoidais. As características de ambas as ondas são combinadas na forma de onda resultante. Da mesma forma que o som é construído, é possível filtrar as duas frequências dos componentes do conjunto. Isso geralmente é feito por meio da análise da forma de onda no domínio da frequência.

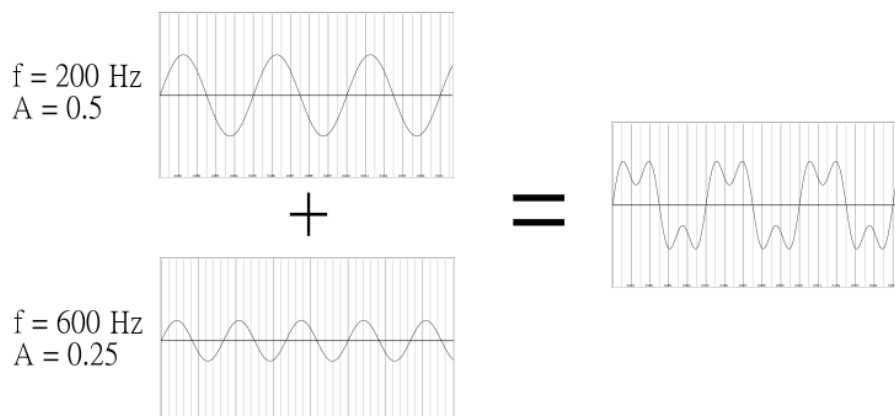


Figura 3.7: Soma de Duas Ondas Senoidais. Fonte:(BALLORA, 2008)

Um algoritmo comumente utilizado para a finalidade da síntese sonora, mais especificamente na técnica de síntese aditiva (onde são adicionadas ondas sonoras para geração de timbre) é a chamada FFT (*Fast Fourier Transform* ou Transformada Rápida de Fourier) (LOAN, 1992). Esse algoritmo consiste em calcular a Transformada Discreta de Fourier (DFT) e o seu inverso. A análise Fourier transforma o tempo (ou espaço) em frequência, uma FFT calcula rapidamente tais transformações fatorando a matriz DFT em uma matriz esparsa.

### 3.3.2 Ambientes de Desenvolvimento para Síntese Sonora

A síntese sonora é essencial para se ter um *feedback* dos resultados obtidos por meio da CMI. Para que seja possível essa síntese de uma forma prática é necessário um ambiente de desenvolvimento com certas características.

Ambientes de desenvolvimento para síntese sonora consistem de uma linguagem de programação de áudio e um ambiente de usuário para projetar e executar a linguagem. Embora muitos desses ambientes são comparáveis em suas habilidades de produzir áudio, as suas

diferenças e especialidades são o que pode ou não atrair um usuário.

### 3.3.2.1 Comparativo entre Ambientes de Desenvolvimento para Síntese Sonora

Realiza-se uma breve análise comparativa entre os principais ambientes de desenvolvimento livres para síntese sonora.

Tabela 3.2: Relação entre Tamanho de Onda, Frequência e Notas

Ambientes de Desenvolvimento em Síntese Sonora	Desenvolvimento do Sistema	Linguagens da API	Tipo do Sistema
ChucK	Instável (2004)	-	Estático
Csound	Estável (1986)	C, C++	Estático
Nsound	Instável (2003)	C, C++	Dinâmico
Pure Data	Estável (1996)	C, C++	Estático
SuperCollider	Estável (1996)	C, C++	Dinâmico
Byond	Beta (2013)	-	Estático

Conforme verificado na Tabela 3.2 é possível perceber uma pequena vantagem no ambiente de desenvolvimento *SuperCollider*.

O *SuperCollider* é um dos sistemas livres mais estáveis na área de síntese sonora, foi desenvolvido por James McCartney em 1996 (SUPERCOLLIDER..., 2014). Em 2002 foi atualizado sob a licença GNU *General Public License* (GPL). Possui aplicações na síntese em tempo real, *live coding*, composição algorítmica, pesquisa acústica, entre outras utilidades.

A partir da versão 3, o ambiente *SuperCollider* foi dividido em dois componentes: um servidor, *scsynth*; e um cliente, *sclang*. Esses componentes se comunicam usando OSC (protocolo para sintetizadores de som, computadores e outros dispositivos multimídia em rede) (SUPERCOLLIDER..., 2014).

A linguagem SC combina estrutura orientada a objetos de *Smalltalk* e características de programação funcional, com uma sintaxe familiar a C (SUPERCOLLIDER..., 2014). A aplicação do servidor SC suporta simples *plug-in* API em C, tornando mais fácil a escrita

de algoritmos eficientes de som. Além de que, devido a todo o controle externo no servidor acontecer via OSC, é possível de usá-lo com outras linguagens ou aplicações.

O ambiente de desenvolvimento *SuperCollider* é a escolha para realização da implementação do modelo a seguir. Nele é possível de realizar o passo da síntese sonora da saída da CMI supervisionada.

## 4 Modelo Desenvolvido

O presente capítulo apresenta a modelagem encontrada para a resolução do problema. O modelo consiste em descrever um Algoritmo Cultural (AC) que é aplicado no desenvolvimento da CMI supervisionada. E consiste também da modelagem musical: função *fitness* e representação dos indivíduos (possíveis soluções).

### 4.1 Modelagem do Algoritmo Cultural com Evolução Diferencial

O AC tradicional utiliza Algoritmos Genéticos (AGs) como base, porém nessa abordagem é utilizado Evolução Diferencial (ED). Conforme o descrito no Capítulo 2 utilizar o ED como base para o modelo será mais eficaz do que com AG.

As motivações para utilização de ED no lugar de AG são diversas. Porém em relação a codificação do indivíduo do problema, apesar de ele possuir uma codificação inteira faz mais sentido utilizar essa estratégia, pois possibilita a utilização de indivíduos de tamanho maiores do que seria possível com AGs binários. Teria também a opção de se utilizar AGs para problemas contínuos, porém já se provou que o ED é um algoritmo superior aos AGs para problemas contínuos, além de ser mais eficiente ao percorrer o espaço de busca de um problema complexo (VESTERSTROM; THOMSEN, 2004).

A Figura 4.1 demonstra como funciona o fluxograma do algoritmo.

Nos passos iniciais do algoritmo, uma população de indivíduos (*pop*) é criada (no formato que é representado na seção 4.2.3), assim como um espaço de crenças (variáveis utilizadas nos conhecimentos que são modelados na seção 4.1.1). Em seguida, para a geração dos filhos, o operador de variação do ED é influenciado pelo espaço de crenças. Então a cada geração esses filhos gerados são comparados com os já existentes. O algoritmo então segue o ciclo de comparações até alcançar o critério de parada. Para se determinar se um filho é melhor do que seu pai, e pode substituí-lo, usam-se as seguintes regras:

- Um indivíduo viável (quando cumpre todas as regras) é sempre melhor do que um



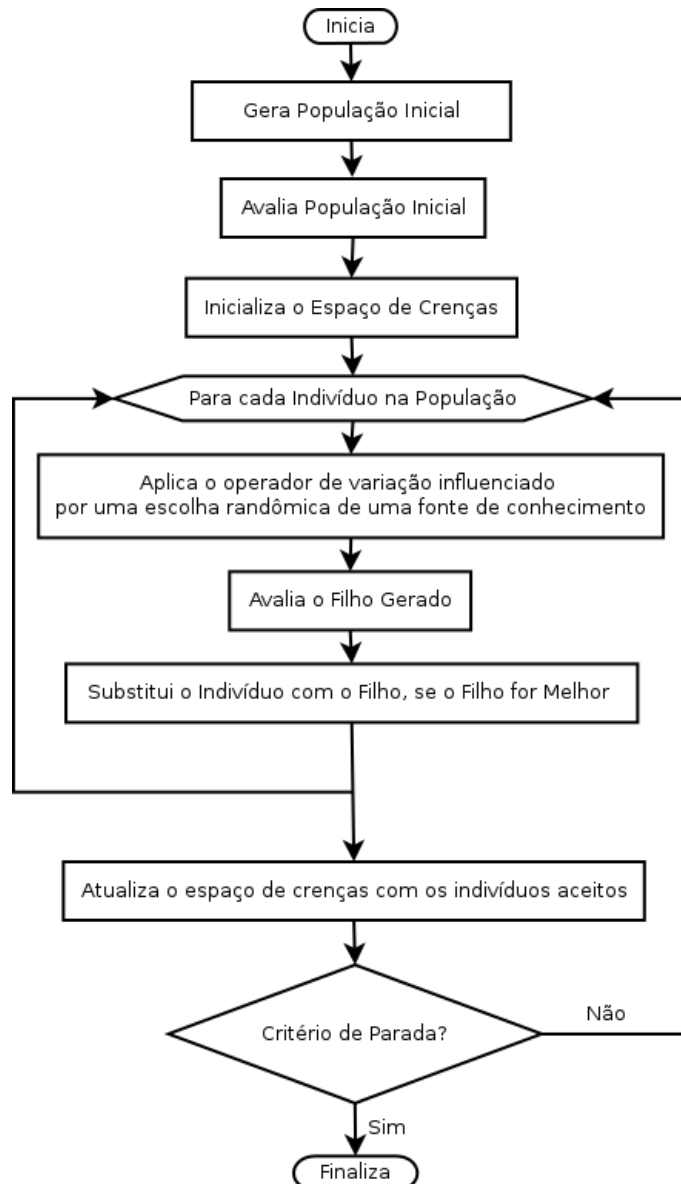


Figura 4.1: Fluxograma do Algoritmo Cultural com Evolução Diferencial.  
 Fonte:(BECERRA; COELLO, 2004)

inviável;

- Se os dois são viáveis, o indivíduo com o melhor valor de função objetivo é melhor;
- Se ambos forem inviáveis, o indivíduo com menor quantidade de violações de regras é melhor.

#### 4.1.1 Espaço de Crenças

No modelo proposto busca-se o maior número possível de conhecimentos para influenciar os indivíduos, com o objetivo de buscar saber quais os conhecimentos que são mais

ou menos importantes em um espaço de crenças.

Nessa abordagem o espaço de crenças é dividido em quatro fontes de conhecimento (BECERRA; COELLO, 2004): Conhecimento Situacional (CS), Conhecimento Normativo (CN), Conhecimento Topográfico (CT) e Conhecimento Histórico (CH).

#### 4.1.1.1 Conhecimento Situacional

O CS consiste em encontrar o melhor exemplar ao longo do processo evolutivo. Ele representa um líder para os outros indivíduos (BECERRA; COELLO, 2004). Os operadores de variação de evolução diferencial são influenciados da seguinte forma (BECERRA; COELLO, 2004)

$$x'_{i,j} = E_i + F \times (x_{i,r1} - x_{i,r2}), \quad (4.1)$$

onde  $x'_{i,j}$  é o operador de variação da ED (indivíduo gerado),  $E_i$  é o  $i$ -ésimo componente do indivíduo armazenado no conhecimento situacional,  $F$  é o peso diferencial (um valor definido pelo usuário entre 0 e 2),  $x_{i,r1}$  e  $x_{i,r2}$  são dois indivíduos escolhidos aleatoriamente. Desta forma, usa-se o líder, em vez de um indivíduo escolhido aleatoriamente para a recombinação, recebendo os filhos para mais perto do melhor ponto encontrado. A atualização do conhecimento situacional é realizada por meio da substituição do indivíduo armazenado,  $E$ , pelo melhor indivíduo encontrado na população atual,  $x_{melhor}$ , apenas se  $x_{melhor}$  for melhor do que  $E$ .

#### 4.1.1.2 Conhecimento Normativo

O CN contém os intervalos para as variáveis de decisão onde as boas soluções têm sido encontradas, a fim de alcançar novas soluções em relação a esses intervalos (BECERRA; COELLO, 2004). Assim, o conhecimento normativo tem a estrutura mostrada na Figura 4.2. Na Figura 4.2,  $l_i$  e  $u_i$  são os limites inferior e superior, respectivamente, para a

$l_1$	$u_1$	$l_2$	$u_2$	$\dots$	$l_n$	$u_n$	$dm_1 dm_2 \dots dm_n$
$L_1$	$U_1$	$L_2$	$U_2$	$\dots$	$L_n$	$U_n$	

Figura 4.2: Estrutura do Conhecimento Normativo

$i$ -ésima variável de decisão, e  $L_i$  e  $U_i$  são os valores da função de *fitness* associada com aquele intervalo. Além disso, o conhecimento normativo inclui um fator de escala,  $dm_i$ ,

para influenciar o operador de mutação adotado na evolução diferencial. A expressão a seguir mostra a influência do conhecimento normativo sobre os operadores de variação (BECERRA; COELLO, 2004):

$$x'_{i,j} = \begin{cases} x_{i,r3} + F \times |x_{i,r1} - x_{i,r2}| & \text{se } x_{i,r3} < l_i \\ x_{i,r3} - F \times |x_{i,r1} - x_{i,r2}| & \text{se } x_{i,r3} > u_i \\ x_{i,r3} + \frac{u_i - l_i}{dm_i} \times F \times (x_{i,r1} - x_{i,r2}) & \text{senão} \end{cases}$$

onde  $x'_{i,j}$  é o operador de variação da ED (indivíduo gerado),  $F$  é o peso diferencial (um valor definido pelo usuário entre 0 e 2),  $x_{i,r1}$ ,  $x_{i,r2}$  e  $x_{i,r3}$  são três indivíduos escolhidos aleatoriamente. A atualização do conhecimento normativo pode reduzir ou ampliar os intervalos armazenados nele. Uma expansão ocorre quando os indivíduos aceitos não encaixam no intervalo atual, enquanto que a redução ocorre quando todos os indivíduos aceitos estão dentro do intervalo atual e os valores extremos têm uma aptidão melhor e são viáveis. Os valores  $dm_i$  são atualizados com a maior diferença  $|x_{i,r1} - x_{i,r2}|$  encontrada durante a aplicação dos operadores de variação da geração inferior.

#### 4.1.1.3 Conhecimento Topográfico

A utilidade do CT é criar um mapa da paisagem do *fitness* do problema durante o processo evolutivo (BECERRA; COELLO, 2004). O CT é composto por um conjunto de células, e o melhor indivíduo encontrado em cada célula. O conhecimento topográfico, também, tem uma lista ordenada das melhores  $b$  células, com base no valor da aptidão do melhor indivíduo em cada uma delas. Por causa de um gerenciamento de memória mais eficiente, na presença de alta dimensionalidade (ou seja, muitas variáveis de decisão), usa-se uma estrutura espacial de dados, chamada árvore  $k$ -d, ou árvore binária  $k$ -dimensional. Em árvores  $k$ -d, cada nó pode ter apenas dois filhos (ou nenhum, se é um nó folha), e representa uma divisão ao meio de qualquer uma das dimensões (ver Figura 4.3). A função de influência tenta mover os filhos para qualquer das  $b$  células na lista (BECERRA; COELLO, 2004):

$$x'_{i,j} = \begin{cases} x_{i,r3} + F \times |x_{i,r1} - x_{i,r2}| & \text{se } x_{i,r3} < l_{i,c} \\ x_{i,r3} - F \times |x_{i,r1} - x_{i,r2}| & \text{se } x_{i,r3} > u_{i,c} \\ x_{i,r3} + F \times (x_{i,r1} - x_{i,r2}) & \text{senão} \end{cases}$$

onde  $l_{i,c}$  e  $u_{i,c}$  são os limites inferior e superior da célula, escolhidos aleatoriamente a partir da lista das melhores células,  $x'_{i,j}$  é o operador de variação da ED (indivíduo gerado),  $F$  é o peso diferencial (um valor definido pelo usuário entre 0 e 2),  $x_{i,r1}$ ,  $x_{i,r2}$  e  $x_{i,r3}$  são três indivíduos escolhidos aleatoriamente. A função de atualização divide um nó se não for encontrada uma solução melhor na célula, e se a árvore não atingiu a sua profundidade máxima. A dimensão em que a divisão é feita, é aquela que tem uma maior diferença entre a solução armazenada e a nova solução de referência (isto é, a nova solução considerada como a "melhor" encontrada até agora).

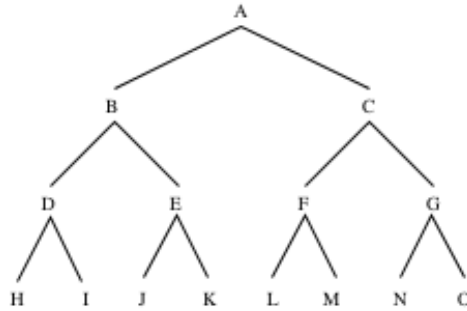


Figura 4.3: Exemplo de Partição de dois Espaços Dimensionais por Árvore  $k$ -d

#### 4.1.1.4 Conhecimento Histórico

Essa fonte de conhecimento foi originalmente proposta para funções objetivo dinâmicas (BECERRA; COELLO, 2004). O CH armazena o conhecimento em uma lista, a localização do melhor indivíduo encontrado antes de cada mudança. Essa lista tem um tamanho máximo. A estrutura do CH é mostrada na Figura 4.4, onde  $e_i$  é o melhor indivíduo encontrado antes da  $i$ -ésima geração mudar,  $ds_i$  é a distância média das alterações de parâmetros,  $i$ , e  $dr_i$  é a direção média se houver mudanças de parâmetros,  $i$ . Se uma solução permanecer como a melhor durante as últimas gerações, sabe-se que está preso

em um ótimo local.

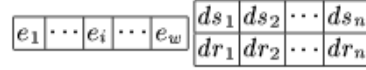


Figura 4.4: Estrutura do Conhecimento Histórico

A expressão da função de influência do CH é a seguinte (BECERRA; COELLO, 2004):

$$x'_{i,j} = \begin{cases} x_{i,e_w} + dr_i \times F \times |x_{i,r1} - x_{i,r2}| & \text{se } rand(0,1) < \alpha \\ x_{i,e_w} + \frac{1.5 \times ds_i}{dm_i} \times (x_{i,r1} - x_{i,r2}) & \text{se } rand(0,1) < \beta \\ rand(lb_i, ub_i) & \text{senão} \end{cases}$$

onde  $x'_{i,j}$  é o operador de variação da ED (indivíduo gerado),  $F$  é o peso diferencial (um valor definido pelo usuário entre 0 e 2),  $x_{i,r1}$  e  $x_{i,r2}$  são dois indivíduos escolhidos aleatoriamente,  $x_{i,e}$  é a  $i$ -ésima variável de decisão dos melhores  $e_w$  anteriores armazenados na lista do conhecimento histórico,  $dm_i$  é a diferença máxima da  $i$ -ésima variável, armazenada no CN,  $lb_i$  e  $ub_i$  são os limites inferior e superior da variável  $x_i$ , dada como entrada para o problema,  $rand(a,b)$  é um número aleatório entre  $a$  e  $b$ .  $\alpha$  e  $\beta$  são valores entre 0 e 1 definido pelo usuário.

Para atualizar o CH, pode-se adicionar à lista quaisquer ótimos locais encontrados durante o processo evolutivo. Se a lista alcançou o seu comprimento máximo  $w$ , o elemento mais antigo é descartado. As distâncias médias e direções de variação são calculadas por (BECERRA; COELLO, 2004)

$$ds_i = \frac{\sum_{k=1}^{w-1} |x_{i,e_{k+1}} - x_{i,e_k}|}{w-1}, \quad (4.2)$$

$$dr_i = \text{sgn} \left( \sum_{k=1}^{w-1} \text{sgn}(x_{i,e_{k+1}} - x_{i,e_k}) \right), \quad (4.3)$$

onde  $ds_i$  é a distância média e  $dr_i$  é a direção de variação.

### 4.1.2 Função de Aceitação

O número de indivíduos aceitos para a atualização do espaço de crença é calculado de acordo com o desenho de uma função dinâmica. O número de indivíduos aceitos diminui, enquanto aumenta o número da geração (BECERRA; COELLO, 2004). Na função de aceitação é preciso redefinir o número de indivíduos aceitos quando a melhor não mudou nas últimas  $p$  gerações. Obtém-se o número de indivíduos aceitos,  $n_{aceito}$ , com a seguinte expressão (BECERRA; COELLO, 2004)

$$n_{aceitos} = pc \times pop + \frac{(1 - pc) \times pop}{g}, \quad (4.4)$$

onde  $pc$  é um parâmetro dado pelo usuário, entre (0,1). Recomenda-se usar 0,2.  $pop$  é o tamanho da população e  $g$  é o contador de geração, mas é redefinida para 1 quando a melhor solução não mudou nas últimas  $p$  gerações.

### 4.1.3 Função de Influência Principal

A função de influência principal é responsável por escolher a fonte de conhecimento para ser aplicada ao operador de variação do ED (BECERRA; COELLO, 2004). No início, todas as fontes de conhecimento tem a mesma probabilidade de de ser aplicada,  $pc_{ks} = \frac{1}{4}$ , pois são 4 fontes de conhecimento. Mas durante o processo evolucionário, a probabilidade da fonte de conhecimento  $ks$  é aplicada como (BECERRA; COELLO, 2004)

$$pc_{ks} = 0.1 + 0.6 \frac{v_{ks}}{v}, \quad (4.5)$$

onde  $v_{ks}$  são as vezes que um indivíduo gerado pela fonte de conhecimento  $ks$  substitui seu pai na geração atual, e  $v$  são as vezes que um indivíduo gerado (por uma fonte de conhecimento) substitui seu pai na geração atual. O limite inferior do valor  $pc$  é de 0.1, para garantir que qualquer fonte de conhecimento tenha sempre probabilidade  $> 0$  em ser aplicada. Se  $v = 0$  durante uma geração,  $pc_{ks} = \frac{1}{4}$ , como no início.

## 4.2 Modelagem da Composição Musical Inteligente

A modelagem da CMI supervisionada é realizada da seguinte forma:

- Definir a base matemática utilizada para a função de objetivo (adaptação);
- Definir a função de objetivo e todos os atributos que a envolvem;
- Definir a representação genética dos indivíduos.

Essas definições são realizadas a seguir.

### 4.2.1 Distância da Compressão Normalizada

A procura por uma métrica universal tem sido realizada a muito tempo, um dos objetivos principais das teorias que envolvem *clustering* (ALFONSECA et al., 2006). Com a avaliabilidade de tal métrica seria possível de se aplicar o mesmo algoritmo em diversos problemas relacionados com *clustering*, tais como, classificação de música, textos, sequência de genes, entre outros.

Em particular, algoritmos evolucionários precisam definir funções de *fitness* para comparar indivíduos diferentes, para simular a evolução, e classificar eles de acordo com seu grau de adaptação no ambiente (ALFONSECA et al., 2007).

Em diversos casos, as funções *fitness* computam a distância entre cada indivíduo ao objetivo desejado (CILIBRASI et al., 2003). Supondo que se quer gerar uma composição que se assemelhe a uma sinfonia de *Mozart*; nessa situação, pode-se medir um *fitness* natural: um indivíduo tem um alto *fitness* se compartilha muitas características com um (ou mais) das sinfonias de *Mozart*. O problema é como selecionar essas características e suas respectivas métricas.

Existe uma métrica de semelhança universal que resume todas as possíveis características: distância da compressão normalizada. Ela é universal no sentido que, quando alguma métrica mede uma pequena distância entre dois objetos dados, a distância da compressão normalizada também mede uma pequena distância entre os mesmos dois objetos; dessa forma, essa se torna uma métrica tão boa quanto qualquer outra métrica computável (DALHOUM et al., 2008). A distância da informação normalizada é matematicamente definida como

$$d(x, y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}, \quad (4.6)$$

onde  $K(x|y)$  é a condicional da complexidade de Kolmogorov da *string*  $x$  dada a *string*  $y$ , cujo valor seria o comprimento do programa mais curto, no qual quando rodasse como entrada a *string*  $y$ , tivesse como saída a *string*  $x$ .  $K(x)$  é o caso degenerado  $K(x|\lambda)$ ,

onde  $\lambda$  é a *string* vazia. Infelizmente, ambas as complexidades condicionais e incondicionais são funções incomputáveis. Uma estimativa computável da distância da informação normalizada é apresentada como

$$\hat{d}(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}, \quad (4.7)$$

onde  $C(x)$  é o tamanho obtido por  $x$  comprimido pelo compressor  $C$ , e  $xy$  é a concatenação de  $x$  e  $y$ . Sendo que o compressor  $C$  foi implementado utilizando o algoritmo de compressão LZ77.

### 4.2.2 Função Objetivo

O esquema do algoritmo proposto inclui um passo de pré-processamento feito da seguinte forma (ALFONSECA et al., 2006):

- Uma ou mais peças musicais são selecionadas como alvos ou guias para a geração musical:  $\Omega = \{\omega_i\}_1^N$ ;
- Os indivíduos na população são codificados da mesma forma que o conjunto guia;
- A seguinte função objetivo é usada

$$f(x) = \frac{\sum_{i=1}^N \hat{d}(x, \omega_i)}{N}, \quad (4.8)$$

onde  $\hat{d}(x, y)$  foi definido na equação (4.7). Maximizando  $f(x)$  (minimizando a soma das distâncias), é esperado que maximize o número de características compartilhadas pelo indivíduos envolvidos com o conjunto guia. Por exemplo, se  $\Omega$  fosse um conjunto de sinfonias de *Mozart*, um indivíduo com o menor valor de função objetivo deve se assemelhar com uma sinfonia de Mozart.

### 4.2.3 Representação Genética

Uma melodia pode ser vista como uma sequência de notas individuais e pausas. Essas notas individuais têm propriedades que definem como uma determinada nota deve ser tocada. As seguintes propriedades são usadas nessa representação:

- Altura: A altura determina quais notas serão tocadas. Possíveis valores são C, D, E, F, G, A, B e todas as possíveis variantes usando sustenidos ( $\sharp$ ) e bemóis ( $\flat$ );



- Oitava: A oitava determina em qual oitava certa nota é tocada. Um piano tem sete oitavas;
- Duração: A duração determina por quanto tempo uma certa nota é tocada. Possíveis valores são nota inteira,  $\frac{1}{2}$  de nota,  $\frac{1}{4}$  de,  $\frac{1}{8}$  de nota,  $\frac{1}{16}$  de nota,  $\frac{1}{32}$  de nota ou  $\frac{1}{64}$  de nota.

As notas possuem outras propriedades além dessas. Velocidade, por exemplo, indica quão forte a nota é tocada. Porém, essas propriedades não serão consideradas.

Uma solução no espaço de busca para geração de melodia consiste de notas individuais. Uma solução é representada por um cromossomo com número fixo de genes. Os genes no cromossomo representam as notas individuais e pausas, cada um com suas próprias características. Uma nota pode ser representada por um gene composto de um ou três inteiros. Quando for representado por apenas um inteiro, este inteiro corresponde a altura da nota, quando for representado por três inteiros eles correspondem a altura, oitava e duração. Cada gene é descrito em maiores detalhes.

#### 4.2.3.1 Altura

A altura pode ser representada como um número de 1 ao 12. Como existem doze semitons em uma oitava, cada semitom pode ser representado por um número entre 1 e 12. Na Tabela 4.1 são mapeadas as notas.

#### 4.2.3.2 Oitava

Uma oitava pode ser representada por um número no qual uma certa nota é tocada. A oitava é indicada do número 1 ao 7, onde 1 representa a oitava mais baixa no teclado do piano e 7 a mais alta.

#### 4.2.3.3 Duração

A duração pode ser representada por um número entre 1 e 7. Na Tabela 4.2 são mapeadas as durações.

Tabela 4.1: Mapeamento da Altura das Notas

Valor	Altura da Nota
1	C (Dó)
2	C# ou Db
3	D (Ré)
4	D# ou Eb
5	E (Mi)
6	F (Fá)
7	F# ou Gb
8	G (Sol)
9	G# ou Ab
10	A (Lá)
11	A# ou Bb
12	B (Si)
0	pausa

#### 4.2.3.4 Representação de Percussão

Da mesma forma que a melodia, a percussão é composta por notas individuais e pausas, porém essa notação é feita de um modo diferenciado. Utilizando a notação da partitura de bateria, a percussão possui apenas duas propriedades: duração (cuja notação é a mesma que na melodia) e peça da bateria (consiste em qual peça será tocada). Na Tabela 4.3 são mapeadas as peças de bateria, sendo elas do número 1 ao 13:

Nessa modelagem conforme será visto no Capítulo 5, foi realizado testes apenas com a codificação da altura das notas, deixando a questão de adicionar os outros elementos para trabalhos futuros.

Tabela 4.2: Mapeamento da Duração das Notas

Valor	Duração da Nota
1	Nota inteira
2	$\frac{1}{2}$ da nota
3	$\frac{1}{4}$ da nota
4	$\frac{1}{8}$ da nota
5	$\frac{1}{16}$ da nota
6	$\frac{1}{32}$ da nota
7	$\frac{1}{64}$ da nota

Tabela 4.3: Mapeamento das Peças de Bateria

Valor	Peça de Bateria
1	Caixa
2	Bumbo
3	Chimbal com baqueta
4	Prato de condução
5	Prato de ataque
6	Tom-tom médio
7	Tom-tom menor
8	Surdo
9	Segundo bumbo
10	Aro da caixa
11	Campana
12	Chimbal com pedal
13	Prato chinês

## 5 Experimentos, Resultados e Análises

Nesta seção são realizados diversos experimentos em algumas condições pré-definidas, com objetivo de validar o modelo desenvolvido. Para cada teste existe duas formas de análise: pelo gráfico de convergência e pelo melhor indivíduo gerado. O gráfico de convergência que consiste dos valores dos melhores e da média de cada geração, é obtido após o término de um determinado número de execuções do algoritmo, ao final de cada execução é gerado um indivíduo melhor.

Todos os experimentos foram realizados em um computador com processador Intel Core 2 Duo 2.00 GHz e 4 GB de memória RAM, rodando no sistema operacional Microsoft Windows 7. A aplicação foi desenvolvida utilizando a linguagem de programação C. Os algoritmos utilizados nos testes são: Evolução Diferencial (ED) e o Algoritmo Cultural (AC) modelado.

### 5.1 Evolução Diferencial

A ED conforme descrita no Capítulo 2 é um algoritmo que cria novos indivíduos a partir dos já existentes utilizando de diversas estratégias matemáticas. Existe uma convenção para nomeação de suas estratégias dado o seguinte modelo: ED/ $x/y/z$ . Onde  $x$  indica uma cadeia de caracteres de um vetor (indivíduo) que será alterado. O  $y$  é o número de novos vetores que serão formados a partir de um indivíduo da geração anterior. E o  $z$  é o método de *crossover*, se ele será exponencial (exp) ou binomial (bin).

Os parâmetros de entrada para o ED são  $F$  e  $CR$ . O  $F$  é uma constante utilizada para a criação do novo indivíduo, nela é atribuída normalmente um valor entre 0,5 e 1, nos testes foram utilizados o valor de 0,75 conforme recomendado na literatura. O segundo parâmetro é a fator de *crossover* ( $CR$ ), esse parâmetro influencia diretamente no número de novos indivíduos que serão gerados em uma geração, são atribuídos valores entre 0 e 1, nos testes foram utilizados o valor 0,5 conforme recomendado na literatura. Foram realizados testes com as duas estratégias de ED mais comuns na literatura, a *rand/1/bin* e a *best/1/bin*, e uma terceira estratégia, a *rand – to – best/1/exp*, a qual foi possível obter o melhor resultado entre os EDs (STORN; PRICE, 1995).

## 5.2 Algoritmo Cultural Modelado

O AC desenvolvido conforme detalhado no Capítulo 4, utiliza o ED como base para seu desenvolvimento, logo ele utiliza os mesmos valores para  $F$  e  $CR$ . Ele é implementado em cima da estratégia de ED que obteve melhor desempenho nos testes (conforme será mostrado a seguir), a  $ED/rand - to - best/1/exp$ . No entanto o AC possui espaços de crenças que irão ajudar a guiar a evolução melhorando assim a convergência e o desempenho do ED.

Os parâmetros de entrada adicionais do AC estão diretamente ligados ao espaço de crenças e as funções de aceitação e influência descritos no capítulo 4. O primeiro parâmetro é o fator de aceitação ( $pc$ ), é o atributo que está diretamente ligado ao número de indivíduos que serão aceitos a cada geração do algoritmo, esse parâmetro tem valor 0,2. O segundo parâmetro é o contador de geração para zerar a contagem de geração ( $g$ ) que está inclusa na função de aceitação, foi definido através de testes um valor de 5 para esse parâmetro. O terceiro parâmetro é o tamanho da lista do conhecimento histórico, definido através de testes como 10. Os quarto e quinto parâmetros são os valores  $\alpha$  e  $\beta$  são utilizados para para escolhas aleatórias do o conhecimento histórico, definidos na literatura como 0,45 (BECERRA; COELLO, 2004).

Durante a implementação do AC foi verificado que apenas três dos quatro conhecimentos descritos na modelagem faziam sentido para a evolução do problema. O conhecimento topográfico sugere um conhecimento a mais sobre o espaço de busca quando não se tem ele bem definido. Porém no problema de Composição Musical Inteligente (CMI) esse espaço de busca está bem definido e não sofre alterações. Com isso os testes se resumem aos conhecimentos situacional, normativo e histórico.

## 5.3 Composição Musical Inteligente

Nos testes são realizados cinco execuções dos algoritmos com aproximadamente um milhão de avaliações da função em cada execução. Para se chegar a esse número de avaliações foi necessário um número de dez mil gerações utilizando o padrão da literatura de cem indivíduos em cada geração. Os dois últimos parâmetros do algoritmo são o número de variáveis e as quantidades de guias musicais incluídas na função objetivo para a evolução dos indivíduos. Foram realizados testes com dois e cinco guias.

A escolha de realizar testes com dois guias foi de exatamente conseguir encontrar de forma simples os padrões dos indivíduos que foram gerados ao final do algoritmo. Com dois guias o tamanho deles e dos indivíduos são de 50 variáveis, sendo que cada variável corresponde a altura de uma nota musical conforme a codificação definida na modelagem no Capítulo 4. A escolha de cinco guias é exatamente para testar como a função objetivo se comporta ao se ampliar a complexidade de comparação dos indivíduos com os guias.

Se a codificação for apenas da altura da nota, para que seja possível que duas ou mais peças sejam guias na evolução é necessário que ao menos elas sejam da mesma escala musical, isso para que as notas geradas ao final dos algoritmo não sejam dissonantes. Se a codificação for a tripla de altura, oitava e duração além das peças serem da mesma escala, devem estar também na mesma fórmula de compasso para que as durações (ritmo) façam sentido ao final da evolução.

## 5.4 Estudo de Caso

Este estudo de caso aborda conceitos estilísticos e de técnicas composicionais da música erudita minimalista (CERVO, 1999). O Minimalismo musical surgiu na década de 60 nos Estados Unidos por meio de seus quatro pais fundadores: *La Monte Young* (1935-), *Terry Riley* (1935-), *Steve Reich* (1936-) e *Philip Glass* (1937-), é um dos movimentos estéticos mais significativos dos últimos quarenta anos. Esse estilo musical influenciou grande parte da música pós-moderna e da música pop recente como por exemplo o *new age* e o *world music*, principalmente por causa de sua simplicidade em comparação com outros estilos eruditos. O minimalismo possui as seguintes características marcantes:

- Estrutura formal contínua: nas obras minimalistas não existem mudanças abruptas melódicas ou rítmicas. As linhas melódicas se sofrem mudanças, são mudanças graduais praticamente imperceptíveis, criando um sentido de continuidade ou de ciclo.
- Repetições de padrões rítmicos e melódicos: devido a estrutura formal contínua, não existem linhas melódicas que criam sentido de início e fim. Tanto a melodia quando o ritmo sugerem repetições de padrões.

Como guia para a evolução, são utilizadas duas peças de natureza minimalista: *Struggle for Pleasure* do compositor *Wim Mertens* e *Truman Sleeps* do compositor *Philip Glass*. Nesse estudo ambas as peças se encontram na escala de Fá menor (Fm) com a fórmula de compasso 4/4. Foram retiradas melodias com cinquenta notas em ambas as peças, a decisão de utilizar esse número de notas foi de forma comparativa com outros trabalhos relacionados. Na peça *Struggle for Pleasure* foi retirada a melodia presente do compasso 1 ao 12, na peça *Truman Sleeps* foi retirada a melodia presente do compasso 1 ao 7. Os trechos melódicos de ambas as músicas sugerem as duas características tanto no sentido de continuidade, quanto a repetição de padrões. Ao final da evolução, buscaram-se indivíduos que se enquadrem nessas mesmas características.

#### 5.4.1 Melodias Codificadas

- Guia 1 - Partitura:

**Struggle for pleasure**

♩ = 140 Wim Mertens

The musical score for 'Struggle for pleasure' by Wim Mertens is presented in three systems. The first system contains measures 1 through 3, the second system contains measures 4 through 7, and the third system contains measures 9 through 12. The music is written for piano in 4/4 time, F minor (three flats). The melody is primarily in the right hand, with some accompaniment in the left hand in the final measure of the third system.

Figura 5.1: *Wim Mertens - Struggle for Pleasure*



- Guia 1 - Codificação Musical:

C,F,Ab,F,F,Db,F,Ab,C,F,Ab,Bb,Eb,G,Eb,Eb,Eb,Ab,Db,F,C,C,  
F,Ab,F,F,Db,F,Ab,C,F,Ab,Bb,Eb,G,Eb,Eb,Eb,Bb,F,Ab,F,F,F,Bb,Eb,G,Eb,Eb,Eb

- Guia 1 - Codificação Inteira:

1,6,9,6,6,2,6,9,1,6,9,11,4,8,4,4,4,9,2,6,1,1,6,9,6,6,  
2,6,9,1,6,9,11,4,8,4,4,4,11,6,9,6,6,6,11,4,8,4,4,4

- Guia 2 - Partitura:

## Truman Sleeps

Philip Glass

= 56♩ =

**Moderato, poco rubato**

Piano

*mp*

Figura 5.2: *Philip Glass - Truman Sleeps*

- Guia 2 - Codificação Musical:

F,Ab,C,F,Ab,C,F,Ab,F,Ab,Db,F,Ab,Db,F,Ab,Eb,Ab,C,Eb,Ab,C,  
Eb,Ab,E,G,C,E,G,C,E,G,F,Ab,C,F,Ab,C,F,Ab,F,Ab,Db,F,Ab,Db,F,Ab,Eb,Ab

- Guia 2 - Codificação Inteira:

6,9,1,6,9,1,6,9,6,9,2,6,9,2,6,9,4,9,1,4,9,1,4,9,5,8,1,  
5,8,1,5,8,6,9,1,6,9,1,6,9,6,9,2,6,9,2,6,9,4,9

Para os testes com 5 guias foi codificado mais três melodias de cinquenta notas. Duas melodias da *Struggle for Pleasure* que estão localizadas do compasso 27 ao 30. E o último guia é um melodia da música *Truman Sleeps* que se encontra do compasso 9 ao 34.

- Guia 3 - Codificação Inteira:

9,1,9,1,9,1,9,1,9,1,9,1,9,1,9,2,9,2,9,2,9,2,9,1,9,  
1,9,1,9,1,8,1,8,1,8,1,8,1,8,1,8,1,8,1,8,1,6,1

- Guia 4 - Codificação Inteira:

6,1,6,1,6,1,6,1,6,1,6,1,6,1,6,2,6,2,6,2,6,2,6,1,6,  
1,6,1,6,1,4,11,4,11,4,11,4,11,4,11,4,11,4,11,4,11,2,9

- Guia 5 - Codificação Inteira:

6,8,9,11,1,2,1,1,6,8,9,11,1,2,4,5,9,8,6,6,4,5,9,8,6,6,1,  
1,6,8,9,11,1,2,1,1,6,8,9,11,1,2,4,5,6,6,6,6,5,5

## 5.5 Experimentos

As análises dos experimentos são realizadas com base nos gráficos de convergência, em que é verificado o comportamento de cada algoritmo ou estratégia através do melhor indivíduo e da média de indivíduos a cada geração, quanto menor o valor da função objetivo melhor o indivíduo.

Primeiramente, antes de realizar qualquer análise dos algoritmos foi realizado um teste de validação da função objetivo, utilizando o AC modelado. Utilizando apenas um guia de 50 variáveis com apenas uma nota C(1), deveria ser possível se chegar ao final da execução com o melhor indivíduo igual ao guia. Na Figura 5.3 está o gráfico de convergência desse teste.

Algo que é possível de perceber claramente com esse gráfico é que não é necessário que a função objetivo chegue ao 0 para que encontre o valor ótimo da função. Nesse caso o valor ótimo foi de 0,176471. Gerando então o indivíduo igual ao guia com 50 variáveis de valor 1, validando assim a função objetivo.

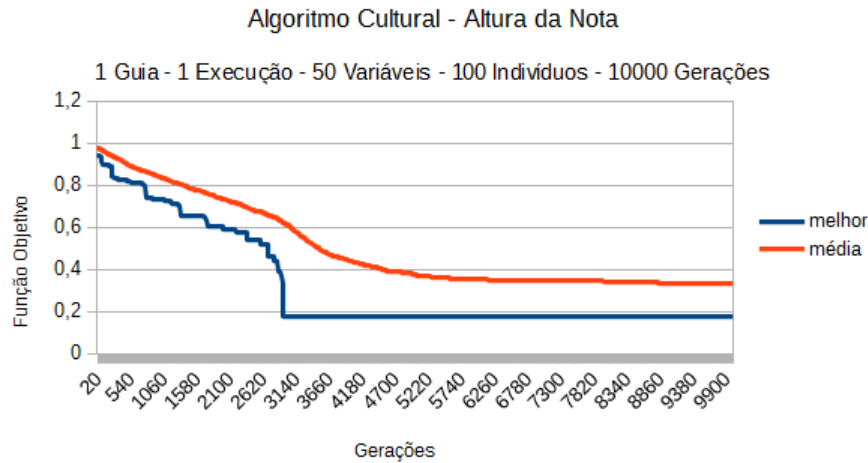


Figura 5.3: Teste de Validação da Função Objetivo

### 5.5.1 Estratégias da Evolução Diferencial

Foi realizado comparações entre três estratégias de EDs para se encontrar a melhor estratégia que será usada como parâmetro de comparação com o AC desenvolvido neste trabalho. A seguir na Figura 5.4 é mostrado o gráfico de convergência de uma execução da estratégia ED/*best/1/bin*.

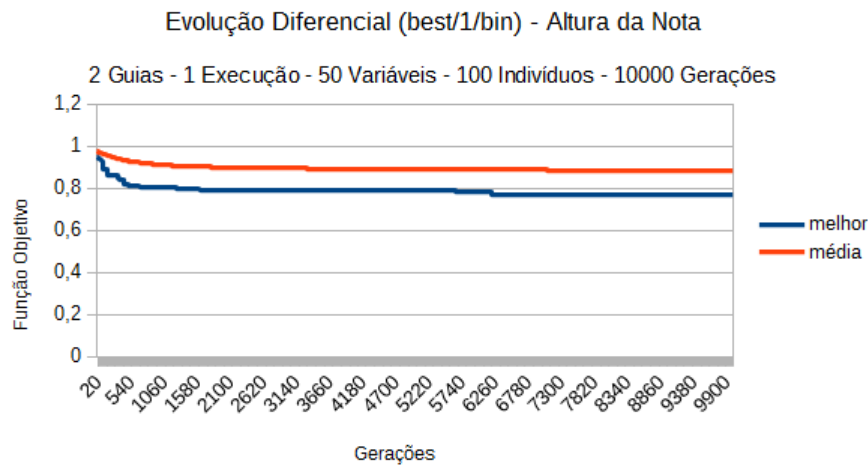


Figura 5.4: Gráfico de Convergência do ED/*best/1/bin*

Claramente pode-se perceber que nessa estratégia o algoritmo teve dificuldade para evoluir, estagnando-se no mesmo valor de função objetivo do melhor indivíduo de 0,770833 a partir da geração 6213. Essa é uma função objetivo muito aquém do que se busca como resultado para o problema. Ao final da evolução, foi gerado o seguinte indivíduo em co-

dificação inteira:

9;6;4;8;10;2;11;8;4;4;4;9;1;6;10;6;4;8;10;2;2;9;5;

8;1;5;8;1;2;3;8;6;9;4;9;1;4;9;1;6;9;6;6;2;1;1;4;2;3;8

Pode-se perceber que nesse indivíduo que foi gerado, existem notas que não fazem parte das guias e que estão fora da escala de Fá menor (Fm), por exemplo, A(10), D(3). Além de que não é possível perceber qualquer característica estilística da música minimalista nessa melodia.

Um segundo teste foi realizado com a estratégia ED/*rand*/1/*bin*. Na Figura 5.5 é mostrado o seu gráfico de convergência para uma execução.

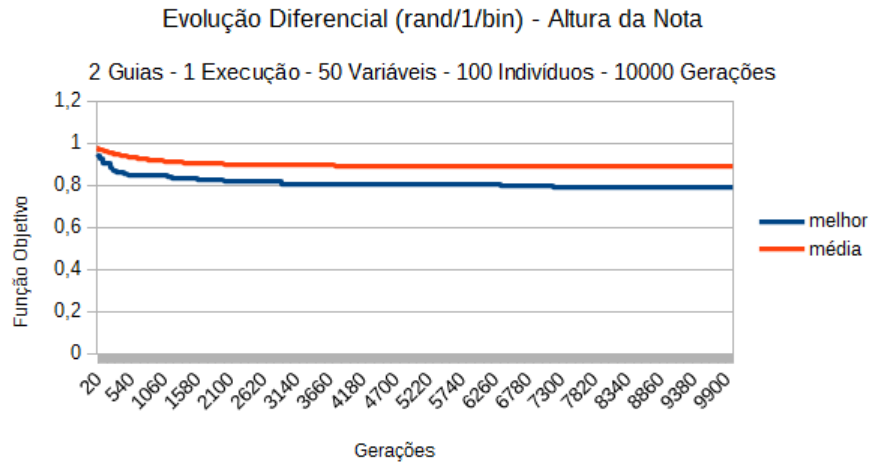


Figura 5.5: Gráfico de Convergência do ED/*rand*/1/*bin* - 2 Guias

Com essa estratégia se obteve uma evolução de forma mais contínua do que a anterior, porém o resultado foi pior do que a anterior. O melhor indivíduo encontrado alcançou a função objetivo de 0,789474, gerando a seguinte melodia:

4;1;11;4;8;11;3;1;11;3;0;0;8;6;0;6;9;1;6;9;6;9;2;

10;0;11;4;8;4;4;1;11;4;8;11;3;1;11;3;0;0;8;6;9;2;6;0;6;9;1

Esse indivíduo gerado tem características interessantes, pois ele possui repetições conforme característica da música minimalista. Mas existem as mesmas duas notas que não fazem parte da escala de Fm, D(3) e A(10).

Então foi realizado o teste com uma terceira estratégia, ED/*rand* – *to* – *best*/1/*exp*. A seguir a Figura 5.6 está gráfico de convergência para cinco execuções dessa estratégia.

Conforme verificado no gráfico, com essa estratégia a evolução foi muito interessante, pode-se perceber uma evolução contínua até praticamente o final das gerações. Nessas

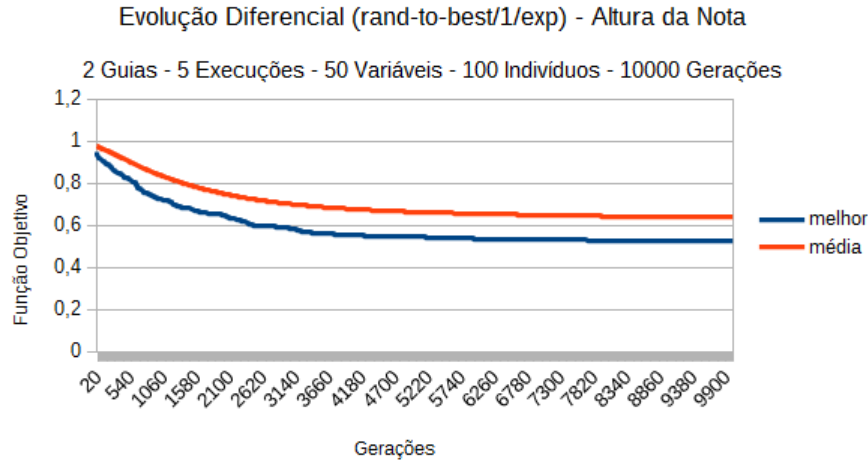


Figura 5.6: Gráfico de Convergência do ED/*rand – to – best/1/exp*

cinco execuções, o melhor indivíduo encontrado possui a função objetivo de 0,511905, sendo que a média dos melhores indivíduos gerados foi de 0,5240894 e o desvio padrão de 0,0079342033. Por esse ótimo desempenho que essa foi a estratégia escolhida para ser o ED base para o AC desenvolvido. O melhor indivíduo gerado após as cinco execuções possui a seguinte característica:

11;6;9;6;6;6;9;4;9;1;4;11;4;8;4;4;4;9;2;6;9;1;6;9;  
6;6;2;6;9;1;6;9;6;9;2;6;9;2;6;9;4;9;1;4;9;5;8;1;5;8

Esse indivíduo gerado possui padrões de repetições curtas típicas da música minimalista, mesmo que com notas diferentes. Por exemplo o padrão melódico da nota 2 até a 6 é o mesmo que o da 13 até a 17, além de que todas as notas pertencem a escala de Fm ou estão nos guias.

### 5.5.2 Conhecimentos do Algoritmo Cultural Modelado

Foram realizados quatro testes para o AC desenvolvido. O primeiro deles é apenas utilizando o conhecimento Situacional. A Figura 5.7 mostra o gráfico de convergência desse teste.

Somente com o conhecimento Situacional, foi possível de obter um resultado bom, porém ainda inferior ao melhor ED, o melhor indivíduo gerado possui a função objetivo de 0,5125, com média dos melhores indivíduos de 0,5330502 e um desvio padrão de 0,0309965324. O melhor indivíduo gerado foi:

9;2;6;9;1;6;9;6;6;6;11;4;8;4;4;4;9;2;6;9;4;9;1;4;9;

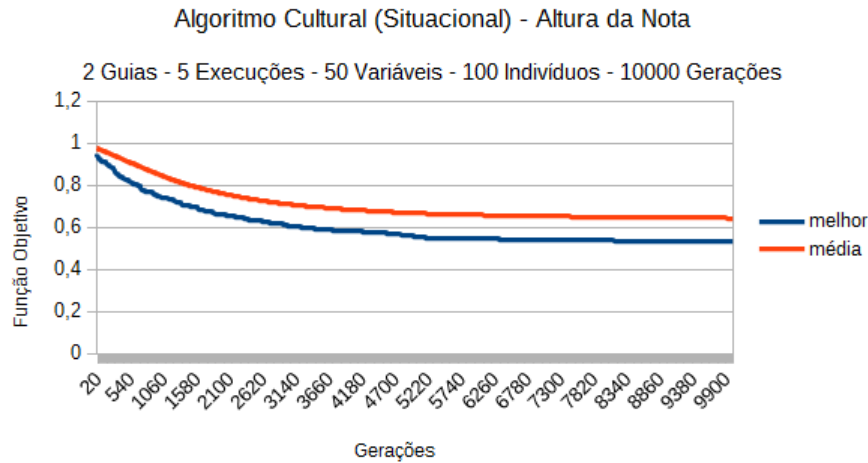


Figura 5.7: Gráfico de Convergência do AC Situacional

2;6;9;2;6;9;4;9;1;4;1;5;8;6;9;4;9;1;4;9;5;8;1;5;8

O segundo teste foi apenas utilizando o conhecimento Normativo. A Figura 5.8 mostra o gráfico de convergência desse teste.

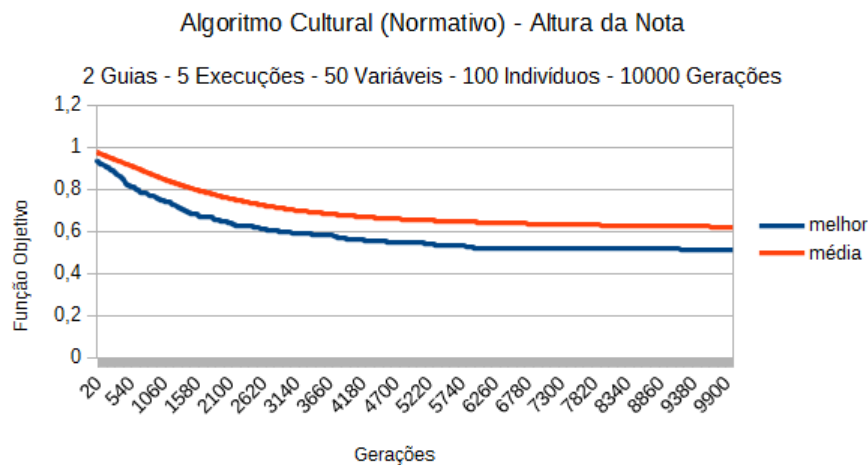


Figura 5.8: Gráfico de Convergência do AC Normativo

Somente com o conhecimento Normativo, foi possível de obter um resultado muito bom, e superior ao melhor ED, o melhor indivíduo gerado possui a função objetivo de 0,486842, com média dos melhores indivíduos de 0,5148256 e um desvio padrão de 0,0225761724. O melhor indivíduo gerado foi:

6;11;4;8;4;4;4;9;2;6;9;1;6;9;6;9;1;6;9;1;6;9;6;9;

1;6;9;1;4;9;5;8;1;5;8;1;5;8;4;4;4;9;2;6;9;4;9;1;6;9

O terceiro teste foi apenas utilizando o conhecimento Histórico. A Figura 5.9 mostra o gráfico de convergência desse teste.

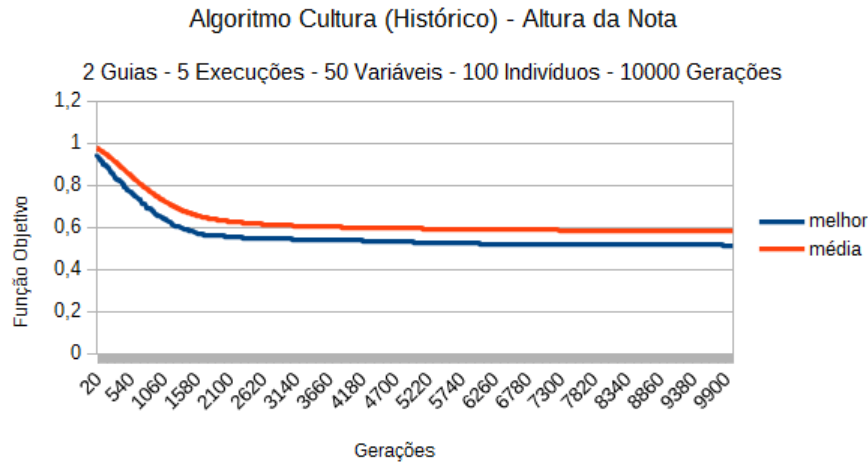


Figura 5.9: Gráfico de Convergência do AC Histórico

Somente com o conhecimento Histórico, foi possível de obter um resultado ligeiramente superior ao melhor ED, o melhor indivíduo gerado possui a função objetivo de 0,5, com média dos melhores indivíduos de 0,5151436 e um desvio padrão de 0,0110832144. O melhor indivíduo gerado foi:

4;8;4;4;4;9;2;6;9;4;9;1;4;9;5;8;1;5;8;1;5;8;6;11;

4;8;4;4;4;9;2;6;9;1;6;9;6;9;1;6;6;9;1;6;9;6;9;2;6;9

E o teste final do AC desenvolvido, com os três conhecimentos implementados. A Figura 5.10 mostra o gráfico de convergência desse teste.

Com o AC desenvolvido foi possível de obter os melhores indivíduos e a melhor média,

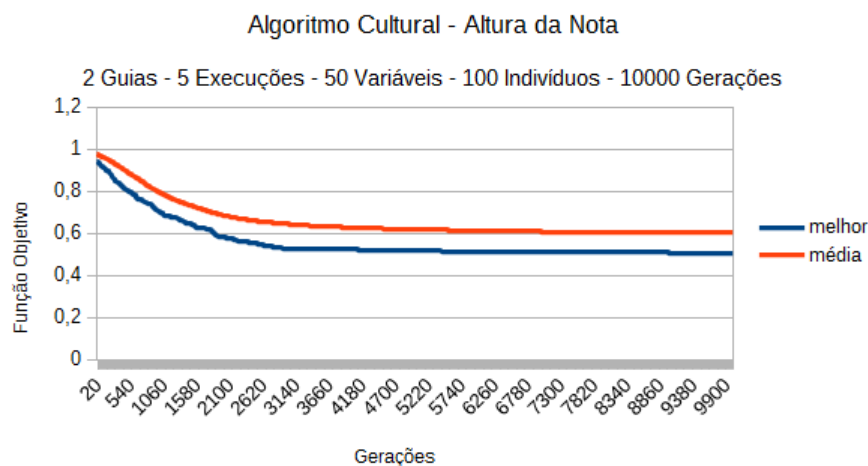


Figura 5.10: Gráfico de Convergência do AC Modelado - 2 Guias

apesar de que o desvio padrão dele ainda é inferior a melhor estratégia do ED. O melhor indivíduo encontrado foi de valor 0,472973, com média de 0,5068178 e desvio padrão de

0,0305867758. O melhor indivíduo encontrado, possui a seguinte melodia:

9;2;6;9;1;6;9;1;6;9;1;6;9;6;6;6;11;4;8;4;4;4;9;2;6;

9;1;6;9;1;6;9;1;6;9;6;9;2;6;9;4;9;5;8;1;5;8;1;5;8

É possível de perceber claramente nesse indivíduo as características de ambos os guias, repetições melódicas curtas e bem definidas formando um estilo de composição minimalista totalmente dentro da escala definida de Fá menor (Fm).

Foram realizados também dois testes com 5 guias. O primeiro utilizando o a melhor estratégia do ED, conforme mostra a Figura 5.11.

O melhor indivíduo encontrado foi de valor 0,768421, com média de 0,783166 e desvio

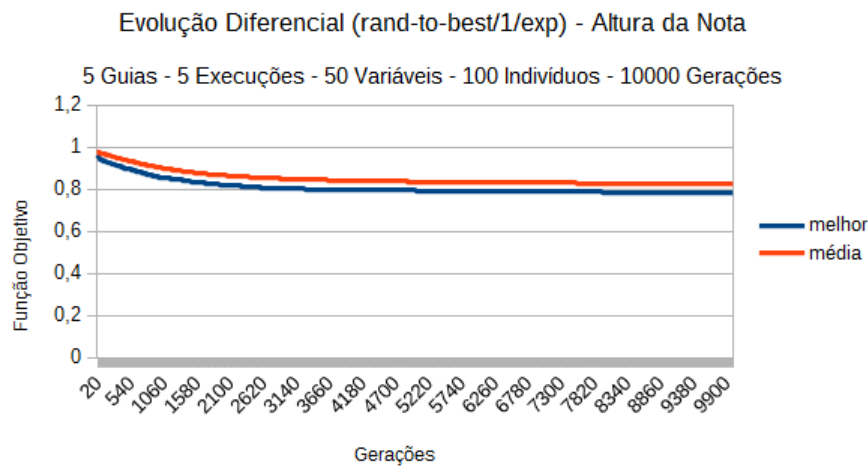


Figura 5.11: Gráfico de Convergência do ED/*rand/1/bin* - 5 Guias

padrão de 0,0111958602. O melhor indivíduo encontrado, possui a seguinte melodia:

1;6;2;6;9;4;9;1;4;9;5;8;1;1;6;9;6;6;6;9;1;6;9;

6;6;6;11;4;8;4;4;4;9;2;6;9;1;6;9;6;6;6;11;4;8;4;4;8;1;1

E o segundo teste utilizando o AC desenvolvido nesse trabalho. A gráfico de convergência está na Figura 5.12.

Com o AC desenvolvido foi possível de obter os melhores resultados do que com o ED, o desvio padrão encontrado . O melhor indivíduo encontrado foi de valor 0,761111, com média de 0,7663956 e desvio padrão de 0,0052408163, foi o melhor desvio padrão entre todos os testes realizados. O melhor indivíduo encontrado, possui a seguinte melodia:

6;9;8;1;1;6;1;6;2;6;9;1;6;9;8;1;1;6;9;6;6;2;6;9;

1;6;9;8;1;1;6;1;6;2;6;9;4;9;1;4;9;1;4;9;5;8;1;1;

Os testes realizados com cinco guias demonstram que quanto maior o número de guias, mais difícil é de se encontrar um indivíduo ótimo, devido a complexidade em encontrar



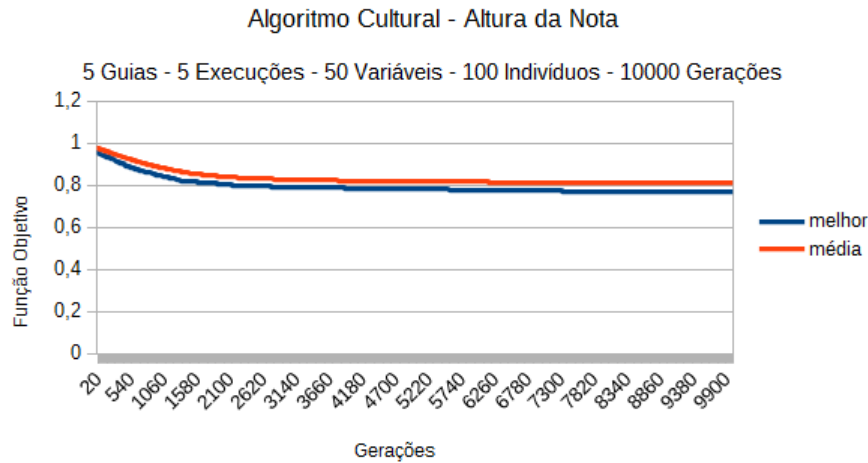


Figura 5.12: Gráfico de Convergência do AC Modelado - 5 Guias

um padrão comum a todos os guias.

## 5.6 Análise de Resultados

Na Tabela 5.1 são estão sumarizados todos os resultados dos testes que forma relevantes:

Tabela 5.1: Comparativo de Resultados

Algoritmo	Melhor Indivíduo	Média	Desvio Padrão
ED/ <i>rand – to – best</i> /1/ <i>exp</i> - 2 Guias	0,511905	0,5240894	0,0079342033
AC (Situacional) - 2 Guias	0,5125	0,5330502	0,0309965324
AC (Normativo) - 2 Guias	0,486842	0,5148256	0,0225761724
AC (Histórico) - 2 Guias	0,5	0,5151436	0,0110832144
AC Modelado - 2 Guias	0,472973	0,5068178	0,0305867758
ED/ <i>rand – to – best</i> /1/ <i>exp</i> - 5 Guias	0,768421	0,783166	0,0111958602
AC Modelado - 5 Guias	0,761111	0,7663956	0,0052408163

Com base na tabela comparativa é possível perceber claramente que o AC desenvolvido é melhor do que o melhor resultado de ED encontrado para esse problema de CMI.

Pois tanto com 2 ou 5 guias é possível de obter um melhor indivíduo e uma melhor média de indivíduos.

A seguir nas Figuras 5.13 e 5.14 estão dois gráficos de caixa comparando os melhores resultados de AC e ED.

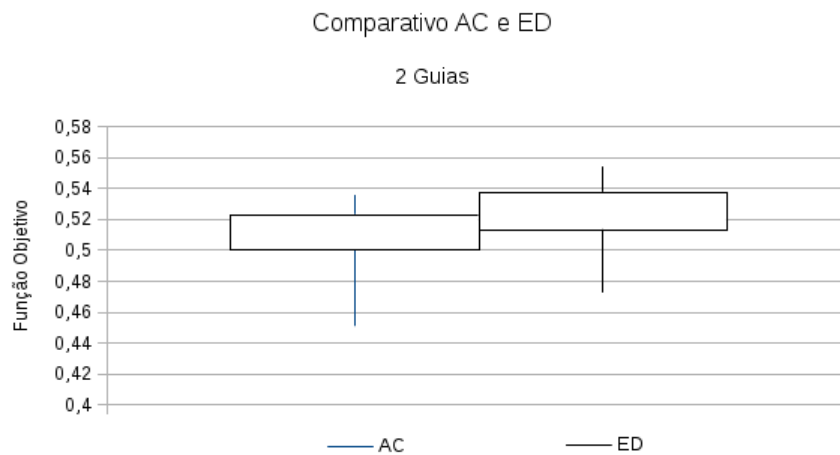


Figura 5.13: Gráfico de Caixa para Comparação entre AC e ED - 2 Guias

Para a configuração de 2 guias o AC obteve resultados ligeiramente melhores do que o ED, em média e a maioria dos indivíduos gerados no AC conseguem ser melhores do que o ED.

Para a configuração de 5 guias o AC obteve melhores resultados do que o ED em pra-

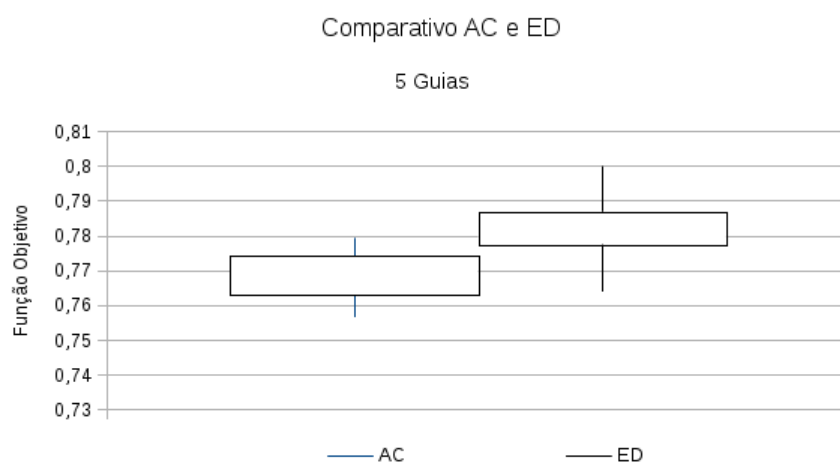


Figura 5.14: Gráfico de Caixa para Comparação entre AC e ED - 5 Guias

ticamente todos os testes realizados, mostrando assim que quanto mais a complexidade maior é a diferença de desempenho entre os algoritmos.

Nas Figuras 5.15 e 5.16 estão na notação de partitura as melodias dos melhores indivíduos dos dois algoritmos com 2 guias (do compasso 1 ao 7) e com 5 guias (do compasso 8 ao 14). Esses melodias foram escritas dentro de uma mesma oitava e todas as notas com a mesma duração, para que fosse possível analisar melhor a melodia.

## Evolução Diferencial

2 Guias



6

5 Guias



11



Figura 5.15: Indivíduos Gerados pelo ED - 2 e 5 Guias

Pode-se perceber em ambos os trechos melódicos as características da música minimalista, principalmente nas melodias geradas por 2 guias. A estrutura formal contínua e melodias repetidas é percebida claramente nos 5 primeiros compassos do melodia gerado pelo AC desenvolvido. A sequência da linha melódica sofre pequenas alterações no compasso, a melodia presente no compasso 1 e 2 são repetidos da mesma forma nos compassos 3, 4 e 5. Porém se for analisar as melodias presentes no compasso 2 e 3, percebe-se a mesma lógica de sequência de notas, porém com notas diferentes.

### Algoritmo Cultural

2 Guias



6

5 Guias



11



The figure displays three musical staves, each representing a different number of guides (Guias) generated by the Cultural Algorithm. The first staff, labeled '2 Guias', shows a melody in the treble clef and a bass line in the bass clef, both in 4/4 time. The second staff, labeled '5 Guias', shows a more complex melody in the treble clef and a bass line in the bass clef, both in 4/4 time. The third staff, labeled '11', shows a melody in the treble clef and a bass line in the bass clef, both in 4/4 time. The notation includes various musical symbols such as notes, rests, and bar lines.

Figura 5.16: Indivíduos Gerados pelo AC Modelado - 2 e 5 Guias

## 6 Conclusão

Na pesquisa relacionada com Algoritmos Culturais (ACs), pôde-se verificar uma tendência em se utilizar apenas os Conhecimentos Normativos (CN) e Situacionais (CS). A justificativa para isso ocorrer é de que são considerados pela maior parte dos pesquisadores os conhecimentos mais importantes no espaço de crenças. Na pesquisa sobre Composição Musical Inteligente (CMI) supervisionada, pôde-se perceber uma dificuldade em encontrar modelos bem embasados matematicamente, a maioria dos trabalhos tentam modelar as equações com base na teoria musical, o que demonstra não ser tão eficaz quando se quer definir um modelo geral de composição inteligente.

Na modelagem, foi encontrado um AC integrado com Evolução Diferencial (ED) com o objetivo de buscar novas alternativas ao AC tradicional, que utiliza Algoritmo Genético. Na parte musical, foi modelado uma função objetivo com base na equação da distância da compressão normalizada, sendo possível encontrar métricas genéricas para a composição inteligente. Sendo possível a compor estilos melódicos bem definidos através das guias da evolução.

Esse trabalho foi apenas um primeiro passo na utilização de algoritmos culturais para geração de música inteligente e supervisionada. Pôde-se validar nos testes que ele é um algoritmo evolutivo interessante para essa aplicação. Sendo que o principal diferencial é o conhecimento normativo e o conhecimento histórico.

No normativo, com o ajuste dos limites laterais, diminui a perda de recursos com variáveis que não nunca serão um bom resultado, olhando para a questão musical isso é de grande importância, dado que certas notas jamais serão tocadas em uma determinada escala musical. Conseguindo encontrar resultados melhores de forma mais rápida.

Com o histórico de índices dos melhores indivíduos da geração, a evolução segue um ritmo visionário, o algoritmo consegue prever onde é possível encontrar um bom indivíduo, fazendo com que quando o algoritmo produz um novo indivíduo, faça sempre as melhores escolhas para *crossover*.

A escolha da evolução diferencial também foi importante nesse trabalho, pois se tratando de indivíduos que são trechos melódicos uma alteração ou mudança de ordem das notas pode estar criando um indivíduo melhor ou pior do que o anterior. Percebe-se que o ED é

uma implementação típica para problemas contínuos e o problema descrito nesse trabalho é facilmente codificado de forma discreta, porém a forma de codificação do indivíduo é totalmente problemática em algoritmos evolutivos discretos, pois os indivíduos possuem uma quantidade considerável de variáveis e transformar essas variáveis para binário é inviável comprometendo a aplicação. Logo a escolha do ED foi a melhor opção obtida.

Os resultados obtidos pelo AC proposto foram plausíveis para a evolução de melodias separadamente. O algoritmo conseguiu encontrar os padrões das guias gerando um melodia totalmente nova e com o mesmo estilo das guias. Trabalhos futuros naturais a esse é de tentar evoluir juntamente com a melodia, o ritmo e a harmonia das músicas. A inclusão desses elementos envolve um estudo aprofundado de teoria e harmonia musical. Outros trabalhos que poderiam ser feitos é de testar novos algoritmos evolutivos para a mesma função objetivo e compará-los com a modelagem realizada nesse trabalho.

## Referências Bibliográficas

ALFONSECA, M.; CEBRIAN, M.; ORTEGA, A. A fitness function for computer-generated music using genetic algorithms. *WSEAS Transactions on Information Science and Applications*, v. 3, n. 3, p. 518–525, 2006.

ALFONSECA, M.; CEBRIÁN, M.; ORTEGA, A. A simple genetic algorithm for music generation by means of algorithmic information theory. In: SRINIVASAN, D.; WANG, L. (Ed.). *2007 IEEE Congress on Evolutionary Computation*. Singapore: IEEE Press, 2007. p. 3035–3042. ISBN 1-4244-1340-0.

BALLORA, M. Gareth Iyler: Musimathics: The mathematical foundations of music, volume 2. *Computer Music Journal*, v. 32, n. 3, p. 103–105, 2008. Disponível em: <<http://dx.doi.org/10.1162/comj.2008.32.3.103>>.

BECERRA, R. L.; COELLO, C. A. C. A cultural algorithm with differential evolution to solve constrained optimization problems. In: LEMAÎTRE, C.; GARCÍA, C. A. R.; GONZÁLEZ, J. A. (Ed.). *Advances in Artificial Intelligence - IBERAMIA 2004, 9th Ibero-American Conference on AI, Puebla, México, November 22-26, 2004, Proceedings*. Springer, 2004. (Lecture Notes in Computer Science, v. 3315), p. 881–890. ISBN 3-540-23806-9. Disponível em: <[http://dx.doi.org/10.1007/978-3-540-30498-2\\_88](http://dx.doi.org/10.1007/978-3-540-30498-2_88)>.

BECERRA, R. L.; COELLO, C. A. C. A cultural algorithm for solving the job shop scheduling problem. In: *Knowledge Incorporation in Evolutionary Computation*. [S.l.]: Springer, 2005. p. 37–55.

BHATTACHARYA, B.; MANDAL, K. K.; CHAKRAVORTY, N. Cultural algorithm based constrained optimization for economic load dispatch of units considering different effects. *International Journal of Soft Computing & Engineering*, 2012. ISSN 22312307. Disponível em: <<http://www.ijjsce.org/attachments/File/v2i2/B0505032212.pdf>; <http://www.doaj.org/doi?func=openurl&genre=article&issn=22312307&date=2012&volume=2&issue=2&spage>>.

CERVO, D. Post-minimalism: A valid terminology? *Ictus*, v. 1, p. 37–52, 1999.

- CILIBRASI, R.; VITÁNYI, P. M. B.; WOLF, R. de. Algorithmic clustering of music. *CoRR*, cs.SD/0303025, 2003. Disponível em: <<http://arxiv.org/abs/cs.SD/0303025>>.
- COELHO, L. dos S.; SOUZA, R. C. T.; MARIANI, V. C. Improved differential evolution approach based on cultural algorithm and diversity measure applied to solve economic load dispatch problems. *Mathematics and Computers in Simulation*, v. 79, n. 10, p. 3136–3147, 2009. Disponível em: <<http://dx.doi.org/10.1016/j.matcom.2009.03.005>>.
- COELLO, C. A. C.; BECERRA, R. L. Evolutionary multiobjective optimization using a cultural algorithm. In: IEEE. *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. [S.l.], 2003. p. 6–13.
- COMTE, A. *The positive philosophy of Auguste Comte*. [S.l.]: W. Gowans, 1868.
- DALHOUM, A. A.; ALFONSECA, M.; CEBRIAN, M.; SANCHEZ-ALFONSO, R.; ORTEGA, A. Computer-generated music using grammatical evolution. In: *Proceedings of the Middle-East Simulation Multiconference*. [S.l.: s.n.], 2008. p. 55–60.
- DONNELLY, P.; SHEPPARD, J. Evolving four-part harmony using genetic algorithms. In: *Applications of Evolutionary Computation*. [S.l.]: Springer, 2011. p. 273–282.
- FORTIER, N.; DYNE, M. V. A genetic algorithm approach to improve automated music composition. *International Journal of Computers*, v. 5, n. 4, p. 525–532, 2011.
- GRIFFITH, N.; TODD, P. M. *Musical networks: Parallel distributed perception and performance*. [S.l.]: MIT Press, 1999.
- GUO, Y.-N.; LIU, D. Multi-population cooperative particle swarm cultural algorithms. In: DING, Y.; WANG, H.; XIONG, N.; HAO, K.; WANG, L. (Ed.). *Seventh International Conference on Natural Computation, ICNC 2011, Shanghai, China, 26-28 July, 2011*. IEEE, 2011. p. 1351–1355. ISBN 978-1-4244-9950-2. Disponível em: <<http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=6021899>>.
- HE, J.; XU, F. Chaotic-search-based cultural algorithm for solving unconstrained optimization problem. *Modelling and Simulation in Engineering*, Hindawi Publishing Corp., v. 2011, p. 13, 2011.
- HEUSSENSTAMM, G. *The Norton manual of music notation*. [S.l.]: WW Norton New York, 1987.



- HO, N. B.; TAY, J. C. GENACE: An efficient cultural algorithm to solve the flexible job-shop problem. In: *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*. Portland, Oregon: IEEE Press, 2004. p. 1759–1766. ISBN 0-7803-8515-2.
- HOCHREITER, R.; WALDHAUSER, C. Zombie politics: Evolutionary algorithms to counteract the spread of negative opinions. *CoRR*, abs/1401.6420, 2014. Disponível em: <<http://arxiv.org/abs/1401.6420>>.
- HORNER, A.; GOLDBERG, D. E. Genetic algorithms and computer-assisted music composition. *Urbana*, v. 51, p. 61801, 1991.
- KALIAKATSOS-PAPAKOSTAS, M. A.; FLOROS, A.; VRAHATIS, M. N. 10 intelligent music composition. 2013.
- KHAN, S. et al. Correction of faulty sensors in phased array radars using symmetrical sensor failure technique and cultural algorithm with differential evolution. *The Scientific World Journal*, Hindawi Publishing Corporation, v. 2014, 2014.
- LOAN, C. V. *Computational frameworks for the fast Fourier transform*. [S.l.]: Siam, 1992.
- LOCKE, J. *An essay concerning human understanding*. [S.l.]: Eliz. Holt, 1700.
- MANARIS, B. et al. A corpus-based hybrid approach to music analysis and composition. In: MENLO PARK, CA; CAMBRIDGE, MA; LONDON; AAAI PRESS; MIT PRESS; 1999. *Proceedings of the National Conference on Artificial Intelligence*. [S.l.], 2007. v. 22, n. 1, p. 839.
- MATHIYALAGAN, P.; SURIYA, S.; SIVANANDAM, S. N. Hybrid enhanced ant colony algorithm and enhanced bee colony algorithm for grid scheduling. *Int. J. of Grid and Utility Computing*, Inderscience Publishers, v. 2, p. 45–58, maio 09 2011. ISSN 1741-8488. Disponível em: <<http://www.inderscience.com/link.php?id=39980>>.
- MIRANDA, E. R.; BILES, J. A. *Evolutionary computer music*. [S.l.]: Springer, 2007.
- MITCHELL, M. *An Introduction to Genetic Algorithms*. [S.l.]: The MIT Press, 1996.
- ÖZCAN, E.; ERÇAL, T. A genetic algorithm for generating improvised music. In: SPRINGER. *Artificial Evolution*. [S.l.], 2008. p. 266–277.

- PAPADOPOULOS, G.; WIGGINS, G. Ai methods for algorithmic composition: A survey, a critical view and future prospects. In: EDINBURGH, UK. *AISB Symposium on Musical Creativity*. [S.l.], 1999. p. 110–117.
- PHON-AMNUAISUK, S.; LAW, E. H. H.; KUAN, H. C. Evolving music generation with som-fitness genetic programming. In: *Applications of Evolutionary Computing*. [S.l.]: Springer, 2007. p. 557–566.
- QIN, A. K.; HUANG, V. L.; SUGANTHAN, P. N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evolutionary Computation*, v. 13, n. 2, p. 398–417, 2009. Disponível em: <<http://dx.doi.org/10.1109/TEVC.2008.927706>>.
- RANDEL, D. M. *The Harvard dictionary of music*. [S.l.]: Harvard University Press, 2003.
- REYNOLDS, R. G. An introduction to cultural algorithms. In: SINGAPORE. *Proceedings of the third annual conference on evolutionary programming*. [S.l.], 1994. p. 131–139.
- REYNOLDS, R. G.; PENG, B. Cultural algorithms: Modeling of how cultures learn to solve problems. In: *ICTAI*. IEEE Computer Society, 2004. p. 166–172. ISBN 0-7695-2236-X. Disponível em: <<http://doi.ieeecomputersociety.org/10.1109/ICTAI.2004.45>>.
- REYNOLDS, R. G.; PENG, B. Knowledge learning and social swarms in cultural systems. *Journal of Mathematical Sociology*, Taylor & Francis, v. 29, n. 2, p. 115–132, 2005.
- REYNOLDS, R. G.; SALEEM, S. Cultural and social evolution in dynamic environments. *Thursday July 5, 2001*, p. 23, 2001.
- ROADS, C. *The Computer Music Tutorial*. [S.l.]: MIT Press, 1996. 904 p.
- RODRIGUES, N. M. *Um algoritmo cultural para problemas de despacho de energia elétrica*. Tese (Doutorado) — PhD thesis, Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Maringá, 2007.
- ROLAND.UK Jupiter-80 Version 2: Synthesizer. 2014. <http://www.roland.co.uk/products/productdetails.aspx?p=1165>. Acessado em: 01-06-2014.

SPECTOR, L.; ALPERN, A. Criticism, culture, and the automatic generation of artworks. In: *AAAI*. [S.l.: s.n.], 1994. p. 3–8.

SPECTOR, L.; ALPERN, A. Induction and recapitulation of deep musical structure. In: *Proceedings of International Joint Conference on Artificial Intelligence, IJCAI*. [S.l.: s.n.], 1995. v. 95, p. 20–25.

SRINIVASAN, S.; RAMAKRISHNAN, S. Cultural algorithm toolkit for interactive knowledge discovery. Academy & Industry Research Collaboration Center (AIRCC), 2012. ISSN 2231007X. Disponível em: <<http://airccse.org/journal/ijdkp/papers/2512ijdkp05.pdf>; <http://www.doaj.org/doi/func=openurl&genre=article&issn=2231007X&date=2012&volume=2&issue=5&spage>>.

SRINIVASAN, S.; RAMAKRISHNAN, S. Cultural algorithm toolkit for multi-objective rule mining. *CoRR*, abs/1209.2948, 2012. Disponível em: <<http://arxiv.org/abs/1209.2948>>.

SRINIVASAN, S.; RAMAKRISHNAN, S. Nugget discovery with a multi-objective cultural algorithm. *CoRR*, abs/1207.2630, 2012. Disponível em: <<http://arxiv.org/abs/1207.2630>>.

STORN, R.; PRICE, K. *Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces*. Berkeley, CA, mar. 1995.

SUPERCOLLIDER Sourceforge. 2014. <http://supercollider.sourceforge.net/>. Acessado em: 15-04-2014.

VESTERSTROM, J.; THOMSEN, R. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: IEEE. *Evolutionary Computation, 2004. CEC2004. Congress on*. [S.l.], 2004. v. 2, p. 1980–1987.

WHITE, J. D. *The analysis of music*. [S.l.]: Prentice-Hall Englewood Cliffs, NJ, 1976.

WHITLEY, D. A genetic algorithm tutorial. *Statistics and Computing*, v. 4, p. 65–85, 1994.

XUE, Z.; GUO, Y. Improved cultural algorithm based on genetic algorithm. In: IEEE. *Integration Technology, 2007. ICIT'07. IEEE International Conference on*. [S.l.], 2007. p. 117–122.