

# **Redução de Dimensionalidade em Problemas com Muitos Objetivos: Uma Aplicação em Composição Algorítmica**

Alan Robert Resende de Freitas  
Universidade Federal de Minas Gerais

Tese submetida ao  
Programa de Pós-Graduação em Engenharia Elétrica  
da Universidade Federal de Minas Gerais  
para obtenção do título de doutor em Engenharia Elétrica



*Dedico este trabalho a meus pais, exemplos de coragem, determinação e bondade para  
minha vida.*



# **Redução de Dimensionalidade em Problemas com Muitos Objetivos: Uma Aplicação em Composição Algorítmica**

## **Resumo**

Em aplicações de métodos de inteligência computacional, é usualmente necessária a consideração de vários objetivos em um problema de otimização. Este trabalho descreve uma abordagem baseada em Árvores de Agregação para o tratamento de problemas de otimização multiobjetivo com um número alto de objetivos. Este tratamento se dá usualmente através da procura de soluções que sejam melhores em relações aos objetivos mas neste trabalho focamos no caso em que temos muito objetivos e se torna também importante explicar a relação entre estes objetivos. A técnica de Árvores de Agregação é então aplicada a um problema de composição algorítmica para música tonal, ou seja, geração de composições musicais através de algoritmos.

A técnica de Árvores de Agregação se baseia em uma representação visual dos conflitos existentes entre soluções para um problema. Algumas funções objetivo podem ser otimizadas bem conjuntamente enquanto outras podem apresentar conflito. Dois objetivos harmônicos são aqueles que podem ser otimizados juntos sem grandes problemas. Baseando-se tanto na quantidade de harmonia e conflito entre objetivos quanto na posição destas características, é possível inferir o potencial de reducibilidade dos objetivos e visualizar com as árvores a relação entre os objetivos para facilitar o processo de tomada de decisão entre várias soluções para um problema.

A partir de uma análise de medidas que podem ser utilizadas para a avaliação de melodias, uma revisão da literatura é feita e são criados vários objetivos a serem con-

siderados simultaneamente para um problema de geração de melodias. Neste caso, são apresentadas medidas de função objetivo que consideram alturas tonais, tonalidade, ritmo e padrões. Algumas medidas são baseadas em psicologia da expectativa enquanto outras são baseadas em regras musicais.

Com o processo de transformação destas medidas em objetivos concretos, as Árvores de Agregação são utilizadas para explicar a relação entre os objetivos com soluções práticas.

# **Dimensionality Reduction in Many-Objective Problems: An Application in Algorithmic Composition**

## **Abstract**

In a real-world environment, it is usually necessary to consider many objectives in an optimization problem. This work describes an approach based on Aggregation Trees for the treatment of a many-objective problem. This treatment is usually through the search of better solutions regarding all the objectives but we focus on the case of many-objective problems in this work. In this context it is also important to explain the relation between the objectives. The technique of Aggregation Trees is then applied to a algorithmic composition problem for tonal music.

The technique of Aggregation Trees is based on a visual representation of conflict existent between solutions for a problem. Some objective functions may be optimized well in group while other objectives may be conflicting. Two harmonious objectives are the ones that can be optimized together without much difficulty. With the amount of harmony and conflict between objectives as well as the position of those features, it is possible to infer the potential of reducibility for the objectives, visualize the relation between those objectives, and make the decision making process easier.

From an analysis of metrics which can be employed for the evaluation of melodies, we survey the literature to create many objectives to be considered simultaneously for a problem of generating melodies. In this case, we present metrics that consider pitch, tonality, rhythm, and patterns. Some metrics are based on the psychology of expectation while others are based on musical rules.

By transforming those metrics into concrete objectives, Aggregation Trees are used

to explain the relation between the objectives with practical solutions.

# Declaração

Esta tese é resultado de meu próprio trabalho, exceto onde referência explícita é feita ao trabalho de outros, e não foi submetida para obtenção de título nesta nem em outra universidade.

Alan Robert Resende de Freitas



## Agradecimentos

Agradeço a *meus pais*, pela bondade, consideração e paciência. Agradeço a *minha família*, pelo apoio e aconchego.

Agradeço aos *amigos*, pelos conselhos e companhia.

Agradeço ao *Frederico*, meu orientador, pelos conselhos acadêmicos e tempo gasto. Agradeço ao *Rogério*, meu co-orientador, pelo direcionamento do trabalho e paciência.

Agradeço aos *professores*, pela preocupação e dedicação. Agradeço à *UFMG, PPGEE e Belo Horizonte*, pela oportunidade de uma nova experiência. Agradeço à *UFOP, DECOM e Ouro Preto*, pelos ensinamentos que formaram uma base sólida. Agradeço à *UFSJ, DCOMP e São João del-Rei*, pelo acolhimento e também pelas lições. Agradeço à *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)* e ao *Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)*.

Agradeço aos *esquecidos*, não mencionados explicitamente aqui, mas que me ajudam direta ou indiretamente neste trabalho.

Muito Obrigado.



# Prefácio

Esta tese apresenta aspectos da utilização de inteligência artificial para composição algorítmica. Como em toda pesquisa interdisciplinar, nem todos os leitores podem estar habituados com as terminologias específicas. Para mais detalhes relativos à terminologia musical, Kennedy & Bourne (2004) podem prover uma boa referência.

Inicialmente, é definida a proposta do trabalho, com sua devida justificativa e objetivos. Uma revisão bibliográfica descreve abordagens utilizadas anteriormente e suas diferentes implicações. Métodos mais antigos de composição algorítmica eram normalmente fundamentados em algoritmos baseados em regras enquanto abordagens mais novas tendem a ser mais baseadas em inteligência artificial. Dado este ponto de vista, problemas da utilização dos diferentes métodos para criação musical são brevemente analisados, descrevendo soluções já propostas para solucioná-los.

Em trabalhos anteriores, foi proposto um gerador de melodias onde medidas mínimas de aptidão são utilizadas para retornar um grupo de bons indivíduos que representam compassos de uma melodia. Este gerador inclui detalhes da implementação e operadores genéticos específicos, assim como uma Matriz de *Takeover* para medir as relações entre diferentes gerações, levando em conta um valor de compromisso entre originalidade e diversidade. Através desta matriz de *Takeover*, o próprio processo evolutivo pode ser usado como um critério de decisão em vez de se usar apenas simples medidas individuais de aptidão.

Já para a geração de harmonias representadas como cadências de acordes, ainda em um trabalho anterior, foi proposta uma abordagem multiobjetivo para a geração de harmonias. Combinações de objetivos podem ser formadas com regras de teoria musical ou preferências do compositor. Por isso, a abordagem multiobjetivo pode ser útil para permitir que o processo evolucionário procure um conjunto de soluções que represente o compromisso entre funções objetivo distintas.

Nesta tese, apresenta-se agora um método para o tratamento de problemas com muitos objetivos. Este método depende da relação de conflito e harmonia entre os objetivos para deduzir fatores como reducibilidade de objetivos e posição de conflito. Este método é então utilizado para a análise de melodias geradas com cadeias de Markov e avaliadas por parâmetros multiobjetivo.

Para tal, são descritas neste trabalho medidas de avaliação automática de melodias. Estranhamente, a maior parte dos trabalhos com algoritmos fortemente baseados na avaliação dos resultados parciais não dá atenção integral às características a serem analisadas e seu reflexo na medida de qualidade final das melodias. Estas medidas são úteis para algoritmos de composição onde métodos de aprendizado de máquina não são utilizados e não se pode então evitar o entendimento do problema de avaliação. Esta necessidade ocorre principalmente em algoritmos onde o foco é a geração de melodias em um determinado estilo.

# Sumário

<b>Lista de Figuras</b>	<b>xix</b>
<b>Lista de Tabelas</b>	<b>xxiii</b>
<b>Nomenclatura</b>	<b>1</b>
<b>1 Introdução</b>	<b>3</b>
1.1 Motivação . . . . .	4
1.2 Objetivos do Trabalho . . . . .	6
1.3 Organização do Texto . . . . .	8
1.4 Contribuições . . . . .	9
<b>2 Revisão da Literatura</b>	<b>11</b>
2.1 Introdução . . . . .	11
2.2 Métodos Históricos . . . . .	11
2.3 Primeiras Composições Assistidas por Computador . . . . .	12
2.4 Modelos de Markov em Composição Algorítmica . . . . .	13
2.4.1 Modelos Ocultos de Markov em Composição Algorítmica . . . . .	14
2.4.2 Modelos de Markov para Geração de Estruturas . . . . .	15
2.5 Gramáticas Gerativas em Composição Algorítmica . . . . .	16

2.5.1	Gramáticas Gerativas para Análise Musical . . . . .	20
2.6	EMI - Experimentos em Inteligência Musical - e Redes de Transição . . . . .	21
2.7	Sistemas de Lindenmayer em Composição Algorítmica . . . . .	24
2.8	Caos e Autossimilaridade em Composição Algorítmica . . . . .	25
2.9	Autômatos Celulares em Composição Algorítmica . . . . .	26
2.10	Redes Neurais Artificiais em Composição Algorítmica . . . . .	27
2.11	Algoritmos Genéticos em Composição Algorítmica . . . . .	30
2.12	Conclusão . . . . .	34
<b>3</b>	<b>Otimização Multiobjetivo</b>	<b>35</b>
3.1	Introdução . . . . .	35
3.1.1	Métodos de Otimização Multiobjetivo . . . . .	36
3.1.2	Visualização da Qualidade das Soluções em Problemas Multiobjetivo	38
3.2	Otimização de Problemas com Muitos Objetivos . . . . .	38
3.2.1	Visualização dos Resultados para Problemas com Muitos Objetivos	42
3.2.2	Problema de escalabilidade . . . . .	44
3.3	Conclusão . . . . .	47
<b>4</b>	<b>Árvores de Agregação</b>	<b>49</b>
4.1	Introdução . . . . .	49
4.2	Medidas de Conflito e Harmonia . . . . .	57
4.2.1	Conflito Direto . . . . .	59
4.2.2	Conflito Maxmin . . . . .	60
4.2.3	Conflito Exponencial ou Logarítmico . . . . .	62
4.2.4	Conflito Não-Paramétrico de Ranqueamento . . . . .	64
4.3	Algoritmo de Redução de Objetivos . . . . .	66

4.4	Localidade de Conflito . . . . .	68
4.5	Construindo uma Árvore de Agregação . . . . .	72
4.6	Conclusão . . . . .	74
<b>5</b>	<b>Avaliação de Melodias</b>	<b>75</b>
5.1	Introdução . . . . .	75
5.2	Avaliação Melódica . . . . .	76
5.3	Funções de Avaliação Automática . . . . .	78
5.4	Banco de Dados Bossa Nova . . . . .	80
5.5	Medidas Baseadas em Altura . . . . .	81
5.5.1	Tonalidade e Dissonância . . . . .	82
5.5.2	Distribuição de Alturas . . . . .	84
5.5.3	Intervalos . . . . .	85
5.5.4	Expectativa Melódica . . . . .	88
5.5.5	Contorno . . . . .	91
5.5.6	Dissonância . . . . .	92
5.6	Medidas Baseadas em Ritmo . . . . .	93
5.7	Medidas de Padrões . . . . .	97
5.7.1	Frases . . . . .	100
5.7.2	Originalidade . . . . .	101
5.7.3	Análises de Segunda Ordem . . . . .	101
5.8	Conclusão . . . . .	102
<b>6</b>	<b>Resultados</b>	<b>105</b>
6.1	Introdução . . . . .	105
6.2	Base de Dados . . . . .	105

6.3	Geração de Novas Melodias . . . . .	106
6.4	Funções objetivo . . . . .	106
6.4.1	Altura . . . . .	107
6.4.2	Intervalo . . . . .	110
6.4.3	Contorno . . . . .	111
6.4.4	Duração . . . . .	114
6.4.5	Posição de nota . . . . .	117
6.4.6	Padrões . . . . .	119
6.5	Aplicação das Árvores de Agregação . . . . .	122
6.6	Conclusão . . . . .	128
<b>7</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>135</b>
7.1	Conclusões . . . . .	135
7.1.1	Composição Algorítmica . . . . .	135
7.1.2	Árvores de Agregação . . . . .	138
7.1.3	Medidas de Avaliação de Melodias . . . . .	139
7.2	Trabalhos Futuros . . . . .	140
7.2.1	Composição Algorítmica . . . . .	140
7.2.2	Árvores de Agregação . . . . .	141
<b>A</b>	<b>Publicações Relacionadas</b>	<b>143</b>
	<b>Referências Bibliográficas</b>	<b>147</b>
	<b>Índice Remissivo</b>	<b>165</b>

# Lista de Figuras

1.1	A complexidade de um sistema aumenta entre a ordem e a desordem (Galanter 2008) . . . . .	6
2.1	Regras de Reescrita para Geração de Frases (Steedman 1984) . . . . .	18
2.2	Exemplo de análise harmônica do sistema SPEAC, que utiliza como referência conceitos de harmonia funcional como colocação (S), preparação (P), extensão (E), antecedentes (A) e consequentes (C) (Cope 1991) . . . . .	23
3.1	Visualização em 2 dimensões . . . . .	39
3.2	Visualização em 3 dimensões . . . . .	40
3.3	Visualização do resultado de um problema de otimização multiobjetivo com 7 funções objetivo. Nesta abordagem de coordenadas paralelas, cada linha representa uma solução. . . . .	43
3.4	Número de soluções não dominadas em um conjunto aleatoriamente gerado com distribuição Gaussiana . . . . .	45
4.1	Um exemplo de conjunto de soluções com uma variedade de conflitos . . . . .	51
4.2	Árvore de Agregação para o conjunto de soluções da Figura 4.1 . . . . .	52
4.3	Coordenadas Paralelas com uma Escolha Diferente de Objetivos Adjacentes . . . . .	54

4.4	Gráfico Polar. Assim como na representação em coordenadas paralelas, cada linha representa a qualidade de uma solução em cada um dos objetivos. Valores mais próximos do centro indicam valores de função objetivo mais baixos. . . . .	55
4.5	Exemplo de vários tipos possíveis de conflito . . . . .	58
4.6	Frentes de Pareto para um Problema de Controle de Temperatura . . . .	62
4.7	Frentes de Pareto em Escala Logarítmica para um Problema de Controle de Temperatura . . . . .	63
4.8	Origem dos valores de diferença na Figura 4.5 . . . . .	71
5.1	Coeficiente de Correlação de Tons nas Melodias em já Transpostas em Dó Maior. Quanto maior a correlação, maior a probabilidade desta melodia estar neste tom, de acordo com os perfis de tonalidade. . . . .	83
5.2	Resultados Projetado em uma Rede de Mapas Auto-Organizáveis. Quanto mais altas as curvas de nível em torno de uma certa tonalidade, maior a probabilidade da melodia estar neste tom. . . . .	84
5.3	Medidas de Altura Tonal . . . . .	86
5.4	Medidas de Intervalo . . . . .	87
5.5	Expectativa Melódica . . . . .	90
5.6	Estabilidade de Contorno . . . . .	92
5.7	Dissonância . . . . .	93
5.8	Medidas de Duração de Notas . . . . .	95
5.9	Medidas de Posição de Notas . . . . .	96
5.10	Medidas de Autocorrelação . . . . .	98
5.11	Medidas de Padrões . . . . .	99
5.12	Frases . . . . .	100
6.1	Coordenadas Paralelas para o Problema de Geração de Frases Melódicas	129

6.2	Árvore de Agregação com Valores de Harmonia nos Nós. Nesta árvore houve conflito muito concentrado em regiões específicas extremas do Pareto para nenhum agrupamento de objetivos. Por este motivo, todos os nós estão representados em preto. . . . .	130
6.3	Perda associada a cada agregação. A perda associada a cada agregação está medida neste gráfico como quantidade de conflito. Cada valor pode ir então de 0 a 100. . . . .	131
6.4	Perda acumulada até cada número de agregações . . . . .	132
6.5	Grafo Polar com a soluções não dominadas para o problema proposto. . .	133
6.6	Solução com melhor valor de função objetivo na função $f_{12}$ , correspondente a expectativa melódica. . . . .	134
6.7	Solução com melhor valor de função objetivo na função $f_{22}$ , correspondente à posição no compasso da última nota. . . . .	134
6.8	Solução com melhor valor de função objetivo na função $f_w$ , combinação dos objetivos $f_8$ e $f_3$ , relacionados com a variedade de notas e de intervalos na frase. . . . .	134
6.9	Solução com melhor valor de função objetivo na função $f_v$ , grande grupo com agregação de 23 objetivos. . . . .	134



# Lista de Tabelas

4.1	Conflito entre os objetivos nas soluções da Figura 4.1. Medidas de conflito e harmonia serão apresentadas nas próximas Seções. . . . .	56
5.1	Representação de uma frase melódica do banco de dados. . . . .	81
5.2	Contorno . . . . .	91



# Lista de Algoritmos

4.1	Redução de Objetivos . . . . .	66
4.2	Construção de uma Árvore de Agregação . . . . .	73



*“A música é uma lei moral. Ela dá uma alma ao Universo, asas ao pensamento, um impulso à imaginação, um encanto à tristeza, a alegria e a vida a todas as coisas. Ela é a essência da ordem e eleva em direção a tudo o que é bom, justo e belo, de que ela é a forma invisível porém surpreendente, apaixonada, eterna.”*

— Platão, 427 BC – 347 BC



# Nomenclatura

AE	Algoritmo Evolucionário
AG	Algoritmo Genético
b	Bemol
CA	Composição Algorítmica
CE	Computação Evolutiva
dB	Decibéis
EMI	<i>Experiments in Musical Intelligence</i>
Hz	Hertz
IA	Inteligência Artificial
IC	Inteligência Computacional
IHC	Interação Homem-Computador
ME	Música Evolutiva
MIDI	<i>Musical Instrument Digital Interface</i>
MOEA	<i>Multiobjective Evolutionary Algorithm</i>
NSGA-II	<i>Non-dominated Sorting Genetic Algorithm - II</i>
RNA	Redes Neurais Artificiais
SPEAC	<i>Statement, Preparation, Extension, Antecedents, Consequents</i>
SST	<i>Sound Synthesis Technique</i>
#	Sustenido



# Capítulo 1

## Introdução

*“Acho que música em si cura. É uma expressão explosiva da humanidade. É algo pelo qual somos todos tocados. Não importa de qual cultura você vem, todos amam música.”*

— Billy Joel, 1949

Este trabalho apresenta um estudo da aplicação de Inteligência Computacional (IC) para Composição Algorítmica (CA), processo no qual determinado algoritmo pode criar uma composição de maneira assistida. Para a criação desses algoritmos, é comum estudar como padrões ocorrem em métodos tradicionais de composição e como estes influenciam a geração de novas músicas. Existem vários padrões em uma composição que podem ser algorítmicos ou não, no sentido de sua definição formal.

A criação feita desta maneira é muito intrigante, pois além de haver a necessidade de desvendar estes padrões que podem ser algorítmicos, é preciso trabalhar com dúvidas que surgem sobre os conceitos pessoais em relação à arte em questão e as expectativas sobre a habilidade de computadores para produzir obras de arte.

Os algoritmos para CA são normalmente (i) estocásticos; (ii) determinísticos e baseados em regras; ou (iii) baseados em Inteligência Artificial (IA), sendo que nos últimos se encontram os métodos evolucionários (Miranda & Biles 2007), que envolvem também estocasticidade.

Para definir quais tarefas musicais podem ser executadas ou auxiliadas por métodos computacionais, há algumas classes de tarefas musicais básicas (Biles 2007b) como composição, execução, processamento de sinais, síntese de sons e audição.

## 1.1 Motivação

Algoritmos, se vistos como regras deterministas ou estocásticas para a resolução de um problema em um número finito de passos, frequentemente foram utilizados por compositores como parte do processo criativo na música ocidental, por mais que estes processos não tenham sido formalizados. Sem embargo, quase todos os processos musicais não são fielmente baseados em algoritmos simples e deterministas, o que confere uma qualidade artística à obra. Alguns processos de composição podem levar em consideração experiências passadas como regras bem definidas ou incluir critérios baseados em aleatoriedade.

Para esclarecimento de questões sobre a codificação de aspectos que são puramente estéticos precisamos então levar em consideração a experiência pessoal dos compositores e a influência desta experiência em suas obras.

Contudo, o cálculo da aptidão, ou a qualidade, de um indivíduo na maioria dos sistemas evolucionários para arte e música requerem um julgamento estético, o que não é fácil de se modelar ou implementar através de um algoritmo. Por isso, estes sistemas normalmente utilizam algum nível de interação com o usuário, dando assim uma resposta ao sistema sobre julgamentos estéticos subjetivos, como por exemplo em (Graf & Banzhaf 1995, Lewis 2000, Moroni, Manzolli, Zuben & Gudwin 2000). Quando a arte em questão trata de imagens estáticas, a avaliação interativa não apresenta grandes problemas pois várias alternativas podem ser exibidas ao usuário em paralelo, por exemplo, em uma grade (Lewis 2008). Além disso, existem alguns estudos e estratégias para minimizar o número de opções apresentadas ao usuário, o que reduz a fadiga em Algoritmos Evolucionários (AE) interativos (Takagi 2001).

Por outro lado, em obras de arte temporais, tais como animação e música, a avaliação interativa das soluções pode exigir grande atenção do mentor, que sempre pode estar suscetível a ficar cansado, entediado, perder a atenção e outros problemas do tipo. Esse aspecto é conhecido na literatura como *fitness bottleneck*, ou gargalo de aptidão (Biles 2001).

Dada a dificuldade de se basear sistemas musicais evolucionários em avaliação humana, alguns autores estudaram o desenvolvimento de sistemas automáticos, que podiam desenvolver peças musicais ou de arte sem intervenção humana. Algumas ideias incluem co-evolução (Greenfield 2002), o desenvolvimento de medidas estéticas confiáveis (Ross, Ralph & Zong 2006) e a evolução de críticos adaptativos (Machado, Romero, Santos,

Cardoso & Manaris 2004).

No contexto musical, uma abordagem interessante para tratar o problema de gargalo de aptidão é apresentada por Biles (2001), onde o uso de aptidão como método de pressão seletiva é eliminado do sistema evolucionário, criando uma versão de seu algoritmo, o GenJam, livre de aptidão. O algoritmo inicia com uma população de compassos selecionados de um banco de dados e evolui esta população usando operadores genéticos cuidadosamente selecionados, sem qualquer avaliação de aptidão. As populações finais contêm novas melodias originais. Apesar desta abordagem levantar questões sobre a pureza do Algoritmo Evolucionário<sup>1</sup>, o autor argumenta que seu sistema ainda é um AG, ou é pelo menos baseado em Computação Evolutiva (CE) (Biles 2001).

Além disto, com abordagens baseadas em CE, é possível o desenvolvimento de novos padrões musicais, já que no processo evolutivo diferentes modelos podem levar a novos estilos ainda não explorados. Com a técnica, novos paradigmas de composição podem ser explorados então com base em análises musicais anteriores, definido-se quais podem ser considerados algorítmicos ou não. Mais ainda, abordagens para composição podem ser desenvolvidas tanto de maneira mais engessada para compositores inexperientes quanto para aqueles experientes que queiram ajuda para desenvolver ideias embrionárias de composição.

Por outro lado, quando há a intenção de se produzir soluções que são avaliadas automaticamente, várias medidas podem ser desenvolvidas para o tratamento do problema. Algumas medidas podem ter como objetivo valores específicos de altura tonal, ritmo ou padrões. Com um conjunto muito grande de objetivos paralelos, precisamos de um método que seja eficiente no tratamento destes objetivos<sup>2</sup>.

Métodos evolutivos para problemas multiobjetivos de otimização são conhecidos porém por uma pobre escalabilidade para problemas com mais de 3 objetivos. Neste contexto, além do estudo de medidas automáticas de avaliação musical, é preciso definir um método que tenha capacidade de trabalhar com vários objetivos conflituosos.

---

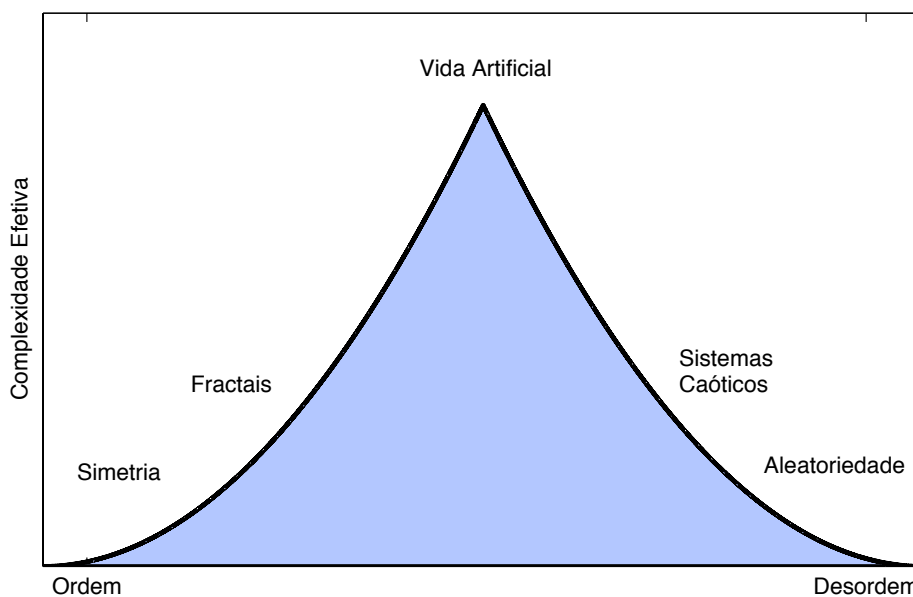
<sup>1</sup>Com “Algoritmo Evolucionário Puro”, pretendemos denotar aquele com os seguintes aspectos: representação, avaliação, seleção e operadores genéticos.

<sup>2</sup>Há duas áreas relacionadas com a otimização de problemas com mais de um objetivo. A primeira é a otimização multiobjetivo, conhecida e referenciada na literatura em inglês como *multiobjective optimization*. Quando temos um número excessivamente grande de objetivos para algoritmos baseados em relação de dominância Pareto, temos o que chamamos nesta tese de otimização multiobjetivo com muitos objetivos, conceito amplamente difundido na literatura em inglês como *many-objective optimization*.

## 1.2 Objetivos do Trabalho

O delineamento de regras de composição usualmente se torna mais claro quando comendo em um estilo particular. Assim, o espaço de busca pode ser restringido para algo mais fácil de ser explorado. É importante definir também se o sistema será criado com base em regras fixas ou entradas do usuário. Além disso, diferenças entre estilos musicais podem ser exploradas para gerar resultados surpreendentes<sup>3</sup>.

Para mostrar a complexidade efetiva na modelagem de problemas nos quais inteligência artificial está envolvida, aspectos aleatórios, como ruído branco, acabam sendo menosprezados ou representados com algoritmos muito aleatórios enquanto aspectos com estruturas representando conceitos bem definidos acabam sendo algoritmicamente comprimidos concisamente, abstraídos ou generalizados. Estruturas que não podem então ser bem comprimidas desta maneira aparecem como as mais complexas (Galanter 2008). A Figura 1.1 exemplifica esta ideia.



**Figura 1.1:** A complexidade de um sistema aumenta entre a ordem e a desordem (Galanter 2008)

<sup>3</sup>O termo estilo musical é utilizado neste trabalho mais para se referir a características concisas que definem um certo grupo de obras musicais, não necessariamente se considerando a contextualização destas obras em sua época.

Em sistemas muito ordenados, como elementos que requerem simetria, o sistema pode ser pouco complexo pois apenas trocas simples são feitas na informação original. Já em sistemas muito desordenados, como os que envolvem muita aleatoriedade, não há também uma grande necessidade de compreensão do problema já que este pode ser modelado através de variáveis muito aleatórias. Em sistemas que se encontram entre estes dois extremos, os problemas se tornam mais complexos de serem modelados.

Em música, estruturas de nível mais alto podem representar partes de uma música, como refrões e versos, ou podem definir parâmetros, como volume e timbre, enquanto estruturas de nível mais baixo podem representar elementos musicais tão simples quanto frequências. As melodias e harmonias, que estarão em estruturas de nível mais baixo, terão suas alturas guiadas pelas estruturas de nível mais alto.

Quando gera-se material musical, é necessário definir alguns fatores que devem ser penalizados ou recompensados na aptidão de um indivíduo. Contudo, estes fatores são normalmente subjetivos e se baseiam fortemente em preferências do usuário. Assim, neste trabalho, apresenta-se uma abordagem para problemas multiobjetivo onde o compromisso entre grupos de soluções e objetivos são analisados.

No caso deste trabalho, é importante basear métodos de composição em aspectos formais desenvolvidos para música tonal. Contudo, respeitar algum conjunto básico de regras não garante que o resultado terá sentido musical ou será interessante. Criatividade e novas ideias podem até mesmo surgir da violação de algumas regras, mas o problema é saber quais regras violar e quando fazê-lo. Nestes casos, é necessário a análise de algoritmos que tratem vários objetivos para encontrar um conjunto de soluções que representa um compromisso entre regras em funções objetivo diferentes, levando ao menos a um conjunto de soluções que possam ser consideradas interessantes.

Desta maneira, os principais objetivos deste trabalho consistem na análise de:

- Avaliação Musical
  - Medidas automáticas de avaliação
  - Critérios para seleção entre as diferentes soluções
  - Inclusão de preferências do compositor
- Avaliação Multiobjetivo
  - Tratamento de problemas com vários objetivos

- Visualizar o espaço de objetivos em várias dimensões
- Conflito entre objetivos
- Harmonia entre objetivos
- Reducibilidade de objetivos
- Posição de conflito

### 1.3 Organização do Texto

O texto desta tese encontra-se organizado da seguinte forma:

No Capítulo 2, apresentamos uma revisão dos métodos aplicados em CA até então. A revisão é dividida em seções dedicadas aos métodos históricos e aos métodos contemporâneos de IA.

No Capítulo 3, é apresentado o conceito e a formulação matemática de problemas de otimização com muitos objetivos. Nestes problemas, há mais de um objetivo e um conjunto de soluções deve representar o compromisso entre os objetivos para um problema dado. Em problemas com muitos objetivos, a comparação entre soluções passa a ser mais difícil e aspectos relacionados são discutidos no Capítulo.

No Capítulo 4, são apresentadas as Árvores de Agregação, uma abordagem para tratamento de problemas com muitos objetivos. A abordagem inclui o tratamento e a visualização das soluções tendo como base a quantidade de harmonia entre cada par de objetivos e entre grupos de objetivos. Para isto, considera-se a quantidade de conflito e a posição do conflito entre pares de objetivos.

No Capítulo 5, são discutidas medidas para avaliação de melodias como uma extensão natural dos desenvolvimentos anteriores. Uma extensa pesquisa na literatura tanto de regras quanto de psicologia musical é apresentada. São definidos critérios automáticos de avaliação em várias categorias, que incluem aspectos de altura tonal, ritmo e procura por padrões. São apresentados resultados de aplicação das medidas em uma base de dados criada pelo autor.

No Capítulo 6, são apresentados os experimentos e resultados. Primeiramente são apresentadas as formulações finais das medidas para avaliação de melodias implementadas como funções objetivo. Em seguida, Árvores de Agregação são aplicadas a soluções geradas e avaliadas com estes objetivos de modo a se compreender a relação entre os objetivos musicais.

No Capítulo 7, conclusões sobre o assunto são discutidas e uma série de possíveis trabalhos futuros são enumerados.

## 1.4 Contribuições

As principais contribuições desta tese são:

- Abordagem baseada em Árvores de Agregação para tratamento de problemas de otimização com muitos objetivos
- Estudo extenso de medidas automáticas para avaliação de material musical, em oposição aos métodos interativos usualmente associados com arte evolutiva

Em relação a composição algorítmica, outras contribuições relacionadas ao período de desenvolvimento desta tese foram:

- Método para geração de melodias livre de aptidão explícita incluindo:
  1. Método baseado em medida *Takeover* para controle de originalidade em composições
  2. Operadores genéticos convenientes para melodias
- Harmonizador multiobjetivo incluindo:
  1. Medidas para avaliação automática multiobjetivo de harmonias simples e dissonantes
  2. Operadores genéticos específicos para harmonias de 5 notas

Em relação aos métodos de inteligência computacional estudados durante o período de desenvolvimento desta tese, outras contribuições relacionadas foram:

- Técnica para detecção de *Takeover* em algoritmos evolucionários
- Método para solução do problema da diversidade máxima
- Medidas de diversidade na seleção de indivíduos em algoritmos genéticos
- Método para classificação de soluções baseado em Árvores de Perceptron

- Algoritmo genético híbrido, validado com aplicação em problema de logística reversa

Este trabalho é fruto de estudos tanto em composição algorítmica quanto em inteligência computacional. Assim, esta tese resultou em várias publicações listadas no Apêndice A.

# Capítulo 2

## Revisão da Literatura

*“A música não mente. Se há algo a ser mudado neste mundo, então isto só pode acontecer através da música.”*

— Jimi Hendrix, 1942–1970

### 2.1 Introdução

Neste capítulo, técnicas de inteligência computacional aplicadas a composição algorítmica são apresentadas. Como referência para terminologia musical básica, recomenda-se Kennedy & Bourne (2004).

São apresentados neste capítulo os métodos históricos mais conhecidos, assim como classes de métodos computacionais muito utilizados.

### 2.2 Métodos Históricos

Quando se pensa em CA, pode haver equívocos ao relacioná-la com métodos computacionais complexos. Na verdade, a utilização de algoritmos para criar composições é muito mais antigo. Um grande exemplo ocorre em 1757, quando Johann Kirnberger, tutelado de Bach, publica um livro para ensinar leitores a compor polonaises e minuetos (Kirnberger & Kupper 1994).

Depois da morte de Mozart, foi publicada e atribuída a ele a autoria de *Musikalisches Würfelspiel* (Jogos de Dados Musicais), método que criava composições através de números jogados em dados (Mozart 1787). O método foi publicado em 1793. O dado

é jogado para escolher aleatoriamente pequenas seções de música, que são juntadas ao final para formar uma peça inteira. Versões computacionais do mesmo método foram implementadas (Chuang 1995).

Os computadores, com certeza, criaram opções para finalidades de composição. Já no século XIX, em 1840, Ada Lovelace propõe a utilização da máquina analítica para a criação de música:

*“Supposing, for instance, that the fundamental relations of pitched sound in the signs of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.”*<sup>1</sup> (Alpern 1995)

Ainda nos métodos algorítmicos sem a utilização de computadores, John Cage, compositor inovador e polêmico por suas composições com características aleatórias, cria em 1951 um método para composição com a utilização do I-Ching como oráculo para tomar várias decisões, como qual nota usar e quando incluí-la na composição (Pritchett 1996). O método foi utilizado na composição *Imaginary Landscape n°4*, onde 24 intérpretes utilizam 12 rádios em tempo real.

## 2.3 Primeiras Composições Assistidas por Computador

A primeira composição que usualmente se considera ter sido gerada por computador é *The Illiac Suite (for String Quartet)*, criada por Lejaren Hiller e Leonard Isaacson. Vários artifícios eram usados, incluindo cadeias de Markov (Hiller 1981). A composição foi feita em 1957 e conseguiu codificar parâmetros estilísticos. O processo de composição se inicia com geração de material pelo computador e depois por modificação deste material de acordo com funções estabelecidas e por final seleção dos melhores resultados de acordo com o conjunto de regras (Alpern 1995). Esse paradigma de geração, modificação e seleção seria mais vezes utilizado no futuro. No método de Hiller, as notas são distribuídas com probabilidade uniforme e depois começam a sofrer restrições. As regras utilizadas são regras clássicas, como formação de escalas, e depois regras do serialismo (Hiller 1959).

Um outro exemplo que segue o paradigma de geração e seleção é o *formula composition* (Felder & Stockhausen 1977), criado por Karlheinz Stockhausen em 1970. Essa

<sup>1</sup>“Supondo-se, por exemplo, que as relações fundamentais entre alturas sonoras no campo da harmonia e da composição musical fossem susceptíveis de expressão e adaptação, a máquina poderia compor peças musicais elaboradas e científicas, de qualquer grau de complexidade ou extensão.”

técnica de composição é baseada em uma fórmula inicial que integra informações de vários parâmetros sonoros, agindo como uma espécie de código genético musical que se desdobra no tempo e permite o surgimento de diversas derivações. Pode ser considerada como um desenvolvimento do método serial. Não obstante, este método é sempre combinado a procedimentos mais livres e intuitivos que levaram a criação de várias óperas.

Em todos os algoritmos de CA existe dificuldade para a introdução de parâmetros externos ao processo de composição. Exemplos destes parâmetros são dados relacionados a experiências culturais do compositor, que influenciam a maneira como ele interpretará a peça, e dados presentes, como fórmulas e teorias.

Os métodos até então descritos foram totalmente estocásticos ou baseados em regras determinísticas baseadas em teoria musical, o que leva à geração de músicas muito pouco diversas na maioria das vezes. Como alternativa a estes métodos, existem os métodos baseados em Inteligência Artificial (IA), que incluem Redes Neurais Artificiais (RNA), Autômatos Celulares e Algoritmos Genéticos, o foco deste trabalho. Atualmente, grande parte dos algoritmos modernos para assistência a músicos estão nesta categoria.

## 2.4 Modelos de Markov em Composição Algorítmica

Modelos de Markov extraem probabilidades de transição entre diferentes estados de acordo com dados analisados. As probabilidades de transição em composição algorítmica são usualmente geradas de acordo com parâmetros estruturais ou com imitações de estilo a partir da análise de um acervo. A aplicação de modelos de Markov foi feita pela primeira vez na década de 1950 por Harry Olson com a análise de 11 melodias e seus modelos de Markov de primeira e segunda ordem com relação a ritmo e notas (Olson 1967).

Neste modelo de Markov, todas as músicas foram transpostas para a mesma tonalidade. Há uma grande diferença entre a distribuição das notas na cadeia de Markov de segunda ordem, já que estes refletem melodias mais naturais no sentido de contextualização dos intervalos consecutivos das notas.

Em 1956, como mencionado na seção anterior, *Illiad Suite*, composição de Lejaren Hiller e Leonard Isaacson para quarteto de cordas, foi apresentada pela primeira vez como a primeira composição gerada por computador. Cada movimento, chamado de “experimento”, é dedicado a um conceito musical especial (Hiller & Isaacson 1958). No quarto experimento, modelos de Markov de diferentes ordens são utilizados para gerar a estrutura musical. Estes modelos de Markov selecionam notas sob vários aspectos, como

diferentes intervalos e texturas em relação à harmonia corrente.

Já Iannis Xenakis utilizou modelos de Markov em 1958 (Xenakis 1992). Em sua peça, segmentos chamados de “screens” são organizados por modelos de Markov. Estes segmentos são sons com diferentes dinâmicas e grupos de instrumentos. A probabilidade de transição entre os segmentos são dadas pelos modelos de Markov.

Brooks, Hopkins, Neumann & Wright (1957) fizeram um experimento com 37 melodias de corais de estrutura rítmica similar e modelos de Markov. Todas as melodias eram 4/4, começavam no quarto tempo do compasso, não tinham valores de duração de nota mais curtos que uma colcheia, as alturas de 4 oitavas foram utilizadas, havia símbolos para criação de uma nova nota ou sustentação da nota atual e todas as melodias foram transpostas para dó maior.

Neste estudo, modelos de até oitava ordem foram gerados para todo o conjunto de dados e representados em forma de “octogramas” com a contagem da ocorrência de cada sequência de oito notas. Algumas restrições como a proibição de pausas em algumas posições do compasso ou limitação de notas para finalizar a melodia foram utilizadas para refletir outras percepções do autor.

Algumas melodias são geradas com cadeias de primeira a oitava ordem e três problemas dos modelos de Markov são apontados:

1. Aleatoriedade notável nas sequências geradas por modelos de ordens mais baixas.
2. Número limitado de opções para cadeias de ordem mais alta.
3. Quanto mais similares as melodias, mais desnecessárias são análises de ordem mais alta.

### **2.4.1 Modelos Ocultos de Markov em Composição Algorítmica**

Baseando-se em pequenos padrões transponíveis de jazz dados pelo usuário para formar o repertório do sistema, Hirzel & Soukup (2000) geraram improvisações. O modelo oculto tem uma entrada de progressão harmônica como sequência observável e o algoritmo de Viterbi gera uma sucessão de padrões melódicos apropriada como estados ocultos.

Já Farbood & Schoner (2001) utilizaram modelos ocultos para a geração de contrapontos em relação a uma melodia fixa. Em seu trabalho, eles definem regras de contra-

ponto de onde a melodia fixa é representada pela sequência observável e o algoritmo de Viterbi calcula os estados ocultos que formam os contrapontos. Neste trabalho, cada regra é unificada em uma tabela de transição onde transições proibidas têm probabilidade zero.

Em um trabalho mais recente, Bell (2011) une a capacidade de cadeias de Markov com Algoritmos Genéticos para geração de composições. Os algoritmos genéticos evoluem cadeias de Markov que são utilizadas para escolher as notas, ritmo e acordes.

## 2.4.2 Modelos de Markov para Geração de Estruturas

Já uma abordagem baseada em uma produção sem análise prévia é proposta por Jones (1981), que considera diferentes relações entre os estados do modelo e estabelece “classes de equivalência” dos estados. Uma classe forma um conjunto de elementos que são ligados em uma representação em grafo onde as transições são limitadas a poucas arestas.

As classes são divididas entre “transitórias” e “recorrentes”, sendo que classes estados em classes transitórias podem levar a outras classes, incluindo as recorrentes. Se uma classe recorrente é alcançada “por fora”, ela não pode ser mais abandonada, já que estados desta classe só podem levar a estados da mesma classe. Esta classificação leva a um modelo incompleto de Markov onde as transições de probabilidade  $p = 0$  não são consideradas. A abordagem de Jones (1981) mostra que os modelos de Markov podem ser utilizados para geração de estruturas hierárquicas melódicas em vez de simplesmente valores absolutos de notas.

Em uma outra abordagem de Ponsford, Wiggins & Mellish (1999), a estrutura harmônica é gerada por modelos de Markov baseados em 84 exemplos de sarabanda. As progressões harmônicas são representadas por graus de escala enquanto o material é dividido por compositores e modos.

Para resolver problemas devidos a cadeias de ordem mais altas, Ponsford, Wiggins & Mellish (1999) utilizaram  $n$ -gramas suavizados. A segmentação do material em frases e compassos leva a melhores resultados na peça e pedaços das peças relativos a inícios e fim são considerados independentemente para contextualização da estrutura. O autor obtém melhores resultados com cadeias de Markov de ordem 4 e sugere uma correção dos modelos de Markov pelo usuário.

Já os modelos ocultos de Markov foram também utilizados para a harmonização

de vozes (Allan & Williams 2005). A estrutura distingue frases, compassos e função harmônica. As probabilidades de emissão e transição são dependentes da base de dados e cadeias comuns de Markov de até ordem 8 são comparadas.

No melhor modelo oculto, as vozes em soprano fazem a sequência observável enquanto a harmonia corresponde aos estados ocultos. O algoritmo de Viterbi estima a sequência de estados e pode gerar uma estrutura aceitável para o coral.

Neste trabalho, a harmonização é dividida em três tarefas que são resolvidas por diferentes modelos ocultos. Primeiro, uma estrutura harmônica é construída pelos estados ocultos, onde alturas específicas não estão especificadas. A sequência observável é formada pela de soprano.

Na segunda etapa, com base em símbolos harmônicos como sequência observável, outro modelo oculto gera acordes concretos de acordo com os estados ocultos. Por final, tem-se a terceira tarefa, que é de ornamentação. Neste último modelo, símbolos indicando harmonia e notas dos tempos correntes de compasso e seguintes formam a sequência observável enquanto distintas notas de ornamentação se encontram nos estados ocultos.

## 2.5 Gramáticas Gerativas em Composição Algorítmica

Gramáticas gerativas são procedimentos baseados tanto em extração de regras gramaticais a partir de um texto quanto em reescrita de textos de acordo com as regras deduzidas ou especificadas. Dependentemente do modo de aplicação, as gramáticas podem ser aplicadas tanto para produção quanto para análise musical. Modelos podem descrever estruturas musicais genuínas, estruturas hierárquicas de análise e geração ou mesmo análises de progressões de acordes de jazz de acordo com um banco de dados. Qualquer conceito de divisão do material com a substituição de símbolos pode ser bem formulado com gramáticas gerativas.

Em alguns casos, podem-se utilizar abordagens baseadas em conhecimento quando algumas regras gerais já são assumidas. Alguns sistemas podem gerar o modelo baseando-se em um banco de dados em vez de conhecimento prévio. Neste segundo caso, tem-se uma situação de inferência gramatical.

A expressividade de modelos de Markov equivalem a gramáticas tipo 3 (Nierhaus 2010) por apenas considerarem símbolos sucessivos. Modelos de Markov também têm uma ordem fixa, definida antes da geração do modelo e usualmente incapaz de descrever

bem o contexto. Redes neurais artificiais ou algoritmos genéticos também podem ser usados para tratar de tarefas quando não há conhecimento específico no domínio. Sistemas que encontram regularidades em conjuntos de dados também existem no campo de Inteligência Artificial não supervisionada, como alguns tipos de RNAs.

Para gramáticas, a geração de um alfabeto terminal é necessária para a análise da base de dados e geração de novo material musical. Neste caso, terminais podem ser acordes, movimentos harmônicos, fragmentos de melodias, figuras rítmicas ou técnicas instrumentais.

Uma estrutura musical gerada por uma gramática poderá ser bem formada em relação à sintaxe definida mesmo quando regras musicais implicitamente esperadas pelo usuário não são respeitadas. Quando as regras de reescrita são aplicadas, não se garante que o conceito de algumas regras implícitas de estilo foram captadas pela gramática inferida.

Outro problema que pode ocorrer em gramáticas sem conhecimento específico do domínio deriva da existência de regras e condições que não podem ser cobertas por inferência gramatical. Se a base de dados representa exemplos corretos de diferentes classes musicais, este fato não será considerado pela gramática gerada. Especialmente no campo de etnologia musical, vários fatores sociológicos e ritualísticos podem influenciar a execução musical de uma maneira que será dificilmente representada por uma gramática gerativa (Blacking 1984).

Se a gramática é produzida com uma base de dados como referência, é recomendável que haja um exame do material gerado. Esta avaliação é muito importante para todos os sistemas de composição não baseados em conhecimento prévio, seja um avaliador humano, formas diferentes de resposta em aprendizado supervisionado ou uma avaliação algorítmica da saída (Nierhaus 2010).

Rader (1974) criou um modelo no qual os graus de escala I, II, III, IV, V e VI servem de base para um modelo harmônico onde há probabilidades de A% a Z% de um grau suceder o outro de acordo com regras de progressões harmônicas. O arcabouço harmônico é construído com acordes em posição da tônica e depois inversões são aplicadas para a construção da estrutura melódica.

Com múltiplos de colcheias e intervalos dentro de duas oitavas, uma melodia é gerada. Esta considera princípios de contraponto e estruturas de ritmo e alturas. Uma gramática gerativa estocástica constrói a composição.

Já em um outro experimento, Sundberg & Lindblom (1976) analisam canções para

crianças e geram uma gramática para geração das músicas. De acordo com os autores, acordes introdutórios indicam as tônicas, acordes alvo são os precedidos de dominantes, e acordes antecipatórios são dominantes seguidos de uma tônica. Em um segundo experimento, uma base de música folclórica é utilizada.

Em outros exemplos de projetos, Baroni, Brunetti, Callegari & Jacoboni (1984) e Camilleri (1984) geram estruturas musicais em diferentes estilos.

Um sistema para a criação de sequências de acordes de jazz foi criado por Steedman (1984). O sistema retorna progressões de 12 compassos que derivam de uma frase inicial. Expressões entre parênteses no lado esquerdo das regras de produção formam alternativas de acordes que devem ser substituídos da mesma maneira ao lado direito da regra. As regras estão representadas na Figura 2.1.

0:		S12(m)	→	I(m) I7	IV(m)	I(m) V7 1(m)
1:		x(m)(7)	→	x(m)	x(m)7	
2:		x(m)(7)	→	x(m)(7)	Sd <sub>x</sub>	
3a:	w	x7	→	D <sub>x</sub> (m)7	x7	
3b:	w	xm7	→	D <sub>x</sub> 7	xm7	
4:		D <sub>x</sub> 7 x(m)(7)	→	St <sub>x</sub> (m)(7)	x(m)(7)	
5:	x	x x	→	x	St <sub>x</sub> m	M <sub>x</sub> m
6:	x(m)	x(m) {D <sub>x</sub> , St <sub>x</sub> m7, L <sub>x</sub> m7}	→	x(m)	#x <sup>o</sup> 7	{D <sub>x</sub> , St <sub>x</sub> m, L <sub>x</sub> m7}

**Figura 2.1:** Regras de Reescrita para Geração de Frases (Steedman 1984)

Um símbolo  $x$  representa o acorde no qual a função harmônica é aplicada. Um  $w$  é um acorde sujeito a substituição, como na regra 3b, onde ele se torna um acorde dominante com sétima. Os acordes entre chaves na regra 6 são acordes alternativos.

Já nos aspectos de improvisações com gramáticas gerativas, um princípio para ritmo, harmonia e melodia é desenvolvido por Johnson-Laird (1991). O autor se baseia em progressões de blues para criar regras livres de contexto divididas em classes hierárquicas. Linhas de baixo são geradas por um processo duplo onde primeiramente uma gramática regular gera um contorno melódico para o qual as alturas são definidas de acordo com a progressão harmônica.

Abordagens diferentes para descrever estruturas musicais foram utilizadas por Pachet (1999), que trabalhou com o estudo de condições para estado de surpresa não só em

relação a expectativas do ouvinte mas também à habilidade de se deduzir uma sequência através de regras combinatórias. Sequências de acordes são analisadas através de um método de compressão e classificadas como esperadas ou surpreendentes.

Com as regras usuais onde a sequência da esquerda pode ser substituída pela sequência da direita, pela análise de progressões harmônicas recorrentes, como II-V-I ou I-VI-II-V, nove regras de reescrita são formuladas:

1. Repetição:  $C \rightarrow C / C$  ( $C$  e  $C \rightarrow C$  levam o mesmo tempo, “/” separam o segundo e terceiro tempo do compasso)
2. Enriquecimento de Acordes:  $C7 \rightarrow C7^9$  (para todas as notas do acorde adicionalmente possíveis)
3. Relativa Menor:  $C \rightarrow Am$
4. Substituição Tritonal:  $C7 \rightarrow F\#7$
5. Preparação:  $C \rightarrow G7 / C$
6. Preparação por Sétima Menor:  $C7 \rightarrow Gmin7/C7$
7. Transição para Quarta:  $C7 \rightarrow C7/F$
8. Retropropagação de Sétima:  $XXC7Y \rightarrow XC7YY$  (ao mesmo tempo, substituição da posição livre pelo acorde subsequente)
9. Remoção à Esquerda:  $XC7 \rightarrow XX$  (a remoção de acordes forma uma classe de regras de reescrita cuja a aplicação é decidida de acordo com o contexto)

Com estas regras de reescrita, pode, por exemplo, haver a seguinte aplicação de regras sobre o acorde  $C7$  no quarto compasso da progressão a seguir.

- $C | F | C | C7 | F \dots$  (Regra 6)
- $C | F | C | Gmin7 / C7 | F \dots$  (Regra 4)
- $C | F | C | Gmin7 / F\#7 | F \dots$  (Regra 5)
- $C | F | C | D7 / Gmin7 / F\#7 | F \dots$  (Regra 8)
- $C | F | C / D7 | Gmin7 / F\#7 | F \dots$  (Regra 9)

- C |F |D7 |Gmin7 / F#7 |F ...

Além de frases bem formadas, há interesse em se formar frases surpreendentes, utilizando-se do contexto de compressão de dados para isto, onde sempre são codificadas em um grafo as expressões de símbolos mais curtas ainda não identificadas. Para isto, a expressão é dividida em segmentos que são colocados em formato de árvore.

Antes desta codificação, sequências de acordes são representadas pela estrutura de acorde e a relação de intervalo. Na codificação, as progressões são simplificadas para indicar cada transição de acordes a partir de C. A representação em árvore mostra a frequência de cada progressão de acordes e os movimentos que ocorrem menos são considerados surpreendentes. Uma progressão, representada como nó da árvore, ocorre mais raramente quando tem menos nós filhos.

O modo como a expectativa é analisada é um exemplo interessante de como características estilísticas podem ser estudadas com extensões apropriadas de métodos de gramática gerativa.

### 2.5.1 Gramáticas Gerativas para Análise Musical

As ideias de Heinrich Schenker inspiraram Lerdahl, Jackendoff & Jackendoff (1996) a descrever um modelo extenso para representação de música tonal com gramáticas gerativas em seu livro. Uma outra possibilidade de representação com gramáticas é dada também por Roads & Wieneke (1979).

De acordo com a análise de Schenker, uma Ursatz é uma estrutura abstrata de duas vozes, com uma descida fundamental e um baixo. A descida é um movimento dentro da progressão harmônica de uma nota da tríade para a tônica do acorde. A linha de baixo indica os tons básicos da progressão harmônica. Esta Ursatz forma a estrutura de fundo da obra. A estrutura é desdobrada em uma estrutura média, que dá diferentes vozes à composição, e então em uma estrutura final, formando a partitura final.

Baseando-se também no método de Schenker, Smoliar (1980) desenvolveu um arcabouço para análise automatizada. Este sistema permite, entre outras coisas, transformações de estruturas em árvore que representam aparências diferentes de estruturas musicais profundas. O método de Schenker, porém, é polêmico e foi objeto de críticas, especialmente pelo modo como são construídos os Ursatzen, que de acordo com Meehan (1979) é uma construção teórica com pouco significado musical.

Lerdahl & Jackendoff (1983) fizeram uma abordagem para examinar música que pode

ser descrita em um contexto de funções harmônicas. Sua estrutura hierárquica é mais dividida e o material musical é tratado em relação aos conceitos definidos pelos autores de:

- Estrutura de grupo: divide a peça em motivos, frases e seções
- Estrutura métrica: controla a sequência de pulsos
- Redução de espaço de tempo: dá importância às alturas tonais
- Redução de prolongamento: atribui importância hierárquica em termos de tensão, relaxamento e duração a alturas tonais
- Regras bem formadas: descrevem as expressões corretas
- Regras de preferência: representam critérios do ouvinte

O modelo de preferências do ouvinte não tem nenhum correspondente em linguística do ponto de vista gerativo e cria um método interessante para examinar geração musical com gramáticas.

## 2.6 EMI - Experimentos em Inteligência Musical - e Redes de Transição

Um sistema muito conhecido para geração de composições em um dado estilo é o EMI (abreviação de *Experiments in Musical Intelligence*), criado em 1981 por David Cope e mantido pelo autor desde então (Cope 1987, Cope 1992, Cope 1991, Cope & Mayer 1996, Cope 2000, Cope 2004). Todos os experimentos de Cope incluem diferentes abordagens para estruturas musicais dentro do contexto do teste musical de Turing (Cope 2004).

Em seus primeiros experimentos, em analogia com os jogos musicais de dados, componentes musicais foram reordenados para formar uma nova peça. A diferença era que EMI fazia uma análise complexa de um banco de dados. De acordo com o que acreditava Cope, a recombinação de partes era o que acontecia em algum nível na cabeça de compositores, sendo a genialidade derivada não da habilidade de criar música inimaginável mas pela habilidade de reordenar as partes (Cope & Mayer 1996).

Em outros experimentos iniciais, corais de Bach foram divididos em segmentos harmônicos que eram re combinados em relação à voz principal, com as transições de segmentos que ocorriam nos corais originais. Mesmo com progressões de corais corretas, uma estrutura completa da composição não foi atingida. Para isto, outras estratégias foram aplicadas para tratar o material musical sob vários aspectos. Para obter uma estrutura que seja universalmente aceita, um coral do banco de dados serviu como modelo para as sequências de frases e progressões.

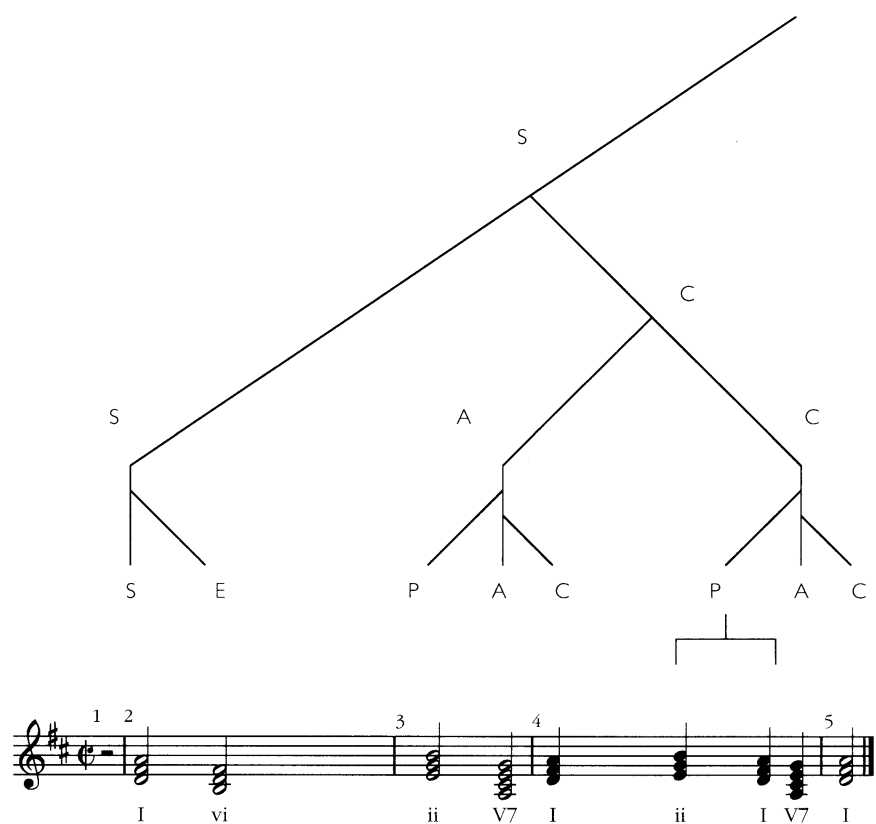
A esta estrutura foi atribuído material musical gerado pelo EMI, que continha material completo do banco de dados, dividido em segmentos de diferentes sentidos musicais. Variações musicais consideradas coerentes foram utilizadas para aumentar a capacidade gerativa do algoritmo.

A informação musical foi processada com um modelo de análise de sentido musical proposto para segmentos de diferentes tamanhos em níveis diferentes, reconhecimento de movimentos característicos para aplicação em novas gerações, estratégias de recombinação dentro da rede de transição aumentada.

As unidades musicais foram selecionadas para recombinação de acordo com critérios sintáticos, em relação a combinações permitidas de terminais, e semânticos, em relação a algumas funções formais em sua posição. Estas funções foram basicamente derivadas de teorias de harmonia funcional.

Utilizando-se de conceitos bem definidos em harmonia funcional, há funções específicas que são associadas com notas e acordes em relação à tonalidade de uma obra. Em um sistema computacional proposto, um material pode ser semanticamente classificado nesta perspectiva por um sistema chamado SPEAC, que analisa colocação (*statement*), preparação (*preparation*), extensão (*extension*), antecedentes (*antecedents*) e consequentes (*consequents*). Antecedentes preparam a situação musical para os consequentes, enquanto preparações antecedem um sentido específico e extensões prolongam o tempo gasto neste sentido. Colocações são unidades que não existem como resultado de um destes processos. A Figura 2.2 mostra um exemplo de análise de harmonia pelo SPEAC.

Nesta partitura em tonalidade de ré maior, o primeiro acorde no compasso 2 (ré maior) é classificado como S (colocação), enquanto o acorde seguinte no mesmo compasso (si menor) é considerado extensão da tônica, por sua relação de terça com o ré maior. No compasso 3, o acorde dominante V7 (lá maior com sétima) é P (preparação) e parte de um C (consequente) em um nível mais alto em relação à tônica. As regras do SPEAC



**Figura 2.2:** Exemplo de análise harmônica do sistema SPEAC, que utiliza como referência conceitos de harmonia funcional como colocação (S), preparação (P), extensão (E), antecedentes (A) e consequentes (C) (Cope 1991)

são restritas às seguintes sucessões possíveis para um componente:

Componentes	Possíveis Sucessões
S	→ P, E, A
A	→ E, C
P	→ S, A, C
C	→ S, P, E, A
E	→ S, P, A, C

O sistema permite a interpretação de segmentos musicais em diferentes níveis. O acorde dominante da Figura 2.2, por exemplo, representa no contexto de dois acordes adjacentes a função de antecedente depois de uma preparação e sucedido por um con-

sequente. Já em um nível hierárquico mais alto, estes três acordes representam um antecedente precedido de uma confirmação e antes de uma cadência final com função de consequente.

A segmentação envolve dependências de contexto até que tenhamos uma frase final, como seria o símbolo S em uma gramática gerativa. Se durante a recombinação um símbolo terminal precisa ser inserido em alguma posição, este é selecionado de um banco de dados de acordo com o contexto.

Se símbolos terminais precisam ser inseridos com as funções “PASC”, terminais que tinham originalmente esta função são escolhidos pelo EMI. Se não há resultados para esta busca, terminais para um nível hierárquico mais baixo “PAS” são procurados. Para ter mais conformidade de estilo, EMI também processa material de significado de estrutura similar. Assinaturas (*signatures*) são frases musicais que normalmente têm componentes repetidos em uma composição. Processo de casamento de padrões com diferentes transformações são utilizados para esta tarefa. A tolerância dos limites de reconhecimento dos padrões é dada pelo número de assinaturas desejadas.

Tendências (*earmarks*) são movimentos característicos que indicam o fim ou início de novos segmentos na composição. Isto permite uma segmentação que inclua cadências características do estilo. Unificações (*unifications*) são configurações com estruturas formais importantes para a composição. Estas características são tratadas no processo de recombinação para manter integridade da estrutura.

Já um sistema de processamento de informação musical baseado em redes Petri é o ScoreSynth (Haus & Sametti 1991), onde objetos musicais com sequências de notas representados por lugares e funções de transformação manipulam informação musical. Uma sintaxe especial é desenvolvida para o ScoreSynth e subredes e redes recursivas estendem as possibilidades da aplicação.

## 2.7 Sistemas de Lindenmayer em Composição Algorítmica

Prusinkiewicz (1986) desenvolveu um sistema onde valores de nota são gerados com sistemas de Lindenmayer. Em uma representação gráfica, notas são representadas por linhas horizontais e durações por tamanhos de linhas verticais.

Em uma comparação entre abordagens estocásticas, McCormack (1996) utilizou modelos de Markov, várias gramáticas gerativas e sistemas de Lindenmayer. Um sistema permitia a aplicação de sistemas de Lindenmayer com gramáticas estruturadas hierar-

quicamente. Os sistema tem módulos de “músicos virtuais” que representam uma voz ou um instrumento, cada músico com seu próprio sistema de reescrita.

Diferentes mapeamentos de parâmetros musicais em sistemas de Lindenmayer foram descritos por DuBois (2003). Os sistemas são representados por sistemas de reescrita nos quais números de eventos musicais podem ser incluídos em cada símbolo. Eles apresentam também sistemas de estrutura polifônica em sistemas de LindenMayer sensíveis a contexto.

## 2.8 Caos e Autossimilaridade em Composição Algorítmica

Uma forma de gerar estrutura musical frequentemente aplicada a partir de conceitos de caos são os ruídos fracionais ou fractais (Leach & Fitch 1995). O termo se aplica à densidade de distribuição das frequências. Ruído branco (*White noise*) se refere a valores aleatórios não relacionados. Uma variação onde os valores são fortemente relacionados é chamada de *brownian noise*.

A forma mais interessante pode ser considerada o ruído rosa, ou  $1/f$  *noise*, com comportamento entre estes extremos. Em termos de alturas tonais, ruído rosa representaria a ideia de equilíbrio entre movimentos uniformes e pulos melódicos. Em termos de frequência, ruído rosa pode sugerir algo que seja agradável aos ouvintes (Voss & Clarke 1975).

Em um experimento com estruturas musicais (Dodge 1988), três linhas melódicas foram geradas com alturas tonais relacionadas à saída do  $1/f$  *noise*. Para cada nota na primeira linha melódica, uma sucessão de notas é produzida para a segunda linha e o processo é repetido para a terceira linha. Para determinar o ritmo, uma quarta linha artificial é envolvida para determinar a duração das notas na terceira linha. Estruturas autossimilares são retornadas como resultados.

Se seguimos a sequência de um certo valor que forma os resultados para uma variável em um sistema de equações iterativas, temos valores que usualmente se aproximarão de atratores de algum tipo. Esta sequência é chamada de trajetória ou órbita, e foram mapeadas para controlar notas convertidas para um intervalo razoável (Pressing 1988), considerando-se equações de até 4 dimensões. Outras órbitas de sistemas de 2, 3 e 4 dimensões foram mapeadas em parâmetros musicais, porém, de acordo com Bidlack (1992) os resultados podem apenas ser considerados material bruto de talvez inerente potencial de utilidade musical.

Outra abordagem de Leach & Fitch (1995) deriva uma árvore a partir das órbitas. Estas árvores consistem de arranjos hierárquicos de estruturas musicais e valores de notas são produzidos interpretando-se nós de níveis mais altos como estruturas para a progressão melódica.

## 2.9 Autômatos Celulares em Composição Algorítmica

Autômatos são modelos de máquinas de estados finitos, onde células representam estados que sofrem transições. Métodos mais simples de composição com autômatos celulares usualmente mapeiam posições de célula com notas específicas e valores de ritmo. Para aumentar a coerência musical, mais de um estado anterior das células pode ser considerado nas transformações. Sistemas também podem permitir interação com o usuário, que pode transformar os estados do autômato em tempo real. Podem ser sempre definidas regras para que o autômato também não chegue a um estado estagnado. Entre as várias combinações possíveis, por exemplo, Beyls (2003) explora em sua pesquisa uma combinação entre autômatos celulares, sistemas de Lindenmayer e algoritmos genéticos.

No trabalho de Millen (2004), é descrita uma versão de autômato celular onde é possível selecionar colunas usadas para o mapeamento e definir escalas pré-definidas para mapeamento de notas. Algumas colunas são associadas com canais MIDI para definir timbres e a duração dos eventos depende da densidade da vizinhança.

Miranda (2001), por outro lado, explora tanto composição quanto síntese de som com os autômatos. Os autômatos geram padrões complexos com as seguintes regras:

- Se a célula tem algum vizinho com estado igual ao mesmo valor da célula acrescido de 1, a célula recebe o valor do vizinho
- Células com valores máximos são alterados para valor 0 por células adjacentes de valor 0
- O valor máximo acrescido de 1 é transformado em 0

Neste sistema, o autômatos geram alturas tonais, durações e timbres. As coordenadas são interpretadas como tríades, onde o eixo x representa o intervalo entre a nota fundamental e o tom médio e o eixo y representa o intervalo entre o tom médio e a nota mais alta. A cada geração, as células ativas são analisadas e mapeadas nas respectivas tríades. Em uma extensão tridimensional deste trabalho (Burraston & Edmonds 2005), a terceira coordenada é usada para mais uma nota no acorde.

Há também aplicações de autômatos celulares para geração de estruturas rítmicas. Redes booleanas geram estruturas de nós conexos representados por bits, onde são aplicadas regras de geração para autômatos unidimensionais (Dorin 2000). Para cada célula, uma regra diferente de função booleana pode ser selecionada de acordo com sua posição no compasso. Assim, as notas são determinadas pelas notas do últimos compassos. Funções como OR, XOR, NOT, and AND são aplicadas. No sistema proposto, as redes booleanas são implementadas como um sistema de várias faixas representadas por compassos. Cada faixa pode ter diferentes regras e canais MIDI.

Já Dorin (2002) apresenta um cubo onde cada lado representa um autômato. As células próximas à borda do cubo também contam como células vizinhas de outros lados do cubo. Cada célula é atribuída a uma altura tonal enquanto os lados do cubo representam canais diferentes. Um evento de som ocorre quando um valor de célula passa de 0 para 1 e um limiar pode definir o limite de notas simultâneas.

Autômatos celulares podem também ser úteis para sonificação de processos, onde as células são mapeadas mais uma vez em alturas tonais (Bilotta & Pantano 2001). A sonificação pode fazer a atividade do autômato celular ser audível com o mapeamento musical. Em outro trabalho de Bilotta, Pantano & Talarico (2000), é identificado que em autômatos celulares unidimensionais, os resultados sonificados mostram que estruturas diferentes em relação às classes de Wolfram são bem distinguíveis.

## 2.10 Redes Neurais Artificiais em Composição Algorítmica

Em RNAs, componentes, chamados de neurônios, formam unidades de informação interligadas cuja estrutura se adapta para resolver problemas. Em RNAs para modelos de regressão, neurônios são usualmente organizados em camadas.

Hild, Feulner & Menzel (1992) criaram uma rede neural acoplada a um sistema baseado em regras para a harmonização de corais em estilo de J. S. Bach. Uma rede neural recorrente com uma camada oculta cria um arcabouço harmônico. O esquema de representação permite achar todos os acordes que contêm determinada nota e cada semínima na melodia é harmonizada de acordo com estas relações e o contexto local, que serve como entrada para a rede neural.

Nesta rede neural recorrente, uma combinação de notas  $H_t$  em um momento  $t$ , depende das combinações passadas  $H_{t-1}$ ,  $H_{t-2}$ ,  $H_{t-3}$  e do contexto melódico  $s_{t-1}$ ,  $s_{t+1}$  para formar uma harmonia. A rede também depende da posição  $H_t$  em relação ao começo ou fim da frase. Uma segunda rede neural ornamenta os resultados finais.

Restrições externas à rede neural monitoram as progressões finais e a rede é treinada através de retropropagação com dois conjuntos de corais de Bach. O primeiro conjunto tem corais em tom maior e o segundo em tom menor, cada conjunto com 20 corais.

Um fator importante de se questionar na aplicação de redes neurais para composição algorítmica é como se representará tempo (Todd 1989). Em uma possibilidade, neurônios de saída podem representar pontos no tempo que podem conter notas sucessivas e o tempo é mapeado no espaço.

Um outro modo de se tratar a questão é usar a rede para gerar notas sucessivas. A saída da rede depende então de um número de eventos passados. Neste caso, a rede deve ter uma memória de eventos passados e a saída da rede pode ser ligada a sua entrada para a geração das próximas notas.

Estas duas opções de rede se comportam de maneiras diferentes pois a primeira gera uma saída de acordo com uma entrada enquanto a segunda, gera continuamente eventos de acordo com suas conexões recorrentes.

No trabalho de Todd (1989), é utilizada uma rede Jordan recorrente com três camadas para retornar uma nota por vez com suas informações de altura tonal e duração. Alguns neurônios se ocupam de fazer o papel de memória recente da melodia e a informação de saída é sempre reenviada à rede. São também discutidas no trabalho estratégias para aumentar a capacidade de generalização da rede em relação aos conjuntos utilizados para treinamento.

Todd (1989) representa altura tonal e duração de diferentes maneiras. A altura tonal pode ser representada pela nota inicial e intervalos consecutivos ou alturas tonais. A abordagem relativa, porém, no contexto destas redes neurais recorrentes, levam a vários erros após a geração de uma altura tonal desapropriada e Todd (1989) conclui que a representação absoluta é mais apropriada e a divide em dois subgrupos: (i) distribuída com decodificação binária da saída dos neurônios mapeada em valores inteiros ou (ii) local com cada neurônio representando uma nota.

O treinamento de redes com representação distribuída, porém, se mostra muito difícil pois os valores de erro na saída da rede acabavam por ser arbitrários em relação ao intervalo entre as notas e erros igualmente sérios acabavam por ter avaliações muito diferentes.

Assim como para alturas tonais, a duração das notas pode ser representada de maneiras locais ou distribuídas. Por fim, o autor escolhe dividir o tempo em fatias iguais

de tempo, que se tornam a unidade menor de tempo. Uma sequência da mesma nota mais de uma vez pode ser considerada uma nova nota ou sustentação da nota antiga de acordo com o neurônio de saída.

As produções das rede neurais são usualmente repetitivas e Todd (1989) as compara a cadeias de Markov. Cadeias de Markov, porém, consideram probabilidades de transição sequenciais de uma dada ordem nos dados de treinamento, tratando o contexto de uma nota de maneira muito diferente das redes neurais. As redes neurais não aderem especificamente a este tratamento estatístico onde uma sequência só é gerada estritamente caso apareça nos dados de treinamento. Porém, além deste tratamento diferenciado de contexto não ser sempre uma vantagem explícita, e uma vez com os neurônios ajustados, a rede sempre retornará o mesmo evento para uma dada sequência de entrada.

Em uma outra versão de rede neural para composição (Michael 1994), melodias podem ser produzidas em paralelo com progressões harmônicas. Uma rede recorrente Elman é utilizada e um algoritmo modificado de retropropagação é utilizado. Um aspecto interessante deste trabalho é que as saídas da rede são neste caso consideradas probabilidades de seleção dos próximos valores de nota.

Um modelo multicamadas de rede permite a representação de alturas tonais, duração e função harmônica. O sistema foi utilizado para produção de material nos estilos de melodias populares, vozes de corais diversos de Bach e valsas de compositores diversos. Em um teste com participantes, o autor descreve que o sistema obtém resultados melhores que cadeias de Markov de terceira ordem. Por outro lado, o sistema ainda não é capaz de deduzir estruturas contextuais musicais de ordens hierárquicas mais altas. Assim, a relação temática é muito frequentemente perdida em peças longas.

Para a harmonização de melodias de corais barrocas, Bellgard & Tsang (1994) utilizam máquinas de Boltzmann, variantes de redes Hopfield. Nestas redes, há uma camada de neurônios completamente conectados que são treinados em contextos locais de corais barrocos, todos transpostos para o mesmo tom.

No problema contrário, ou seja, a geração de melodias para uma progressão de acordes, Eck & Schmidhuber (2002) utilizam uma rede neural recorrente com memória de longo e curto prazo para dar coerência estrutural às peças produzidas. Os valores de nota são representados localmente e os valores de duração são produzidos em múltiplos de uma unidade mínima de tempo. Apesar de contextos mais completos poderem ser considerados pela rede, esta ainda apresenta limitações em termos de peças longas.

Kazi & Bhatia (2013) comparam em seu artigo três abordagens para geração de música: uma rede neural, um algoritmo genético que utiliza redes neurais como avaliadores e um algoritmo genético que utiliza um conjunto simples de regras musicais. Eles concluem que todos os resultados gerados foram satisfatórios, porém todos métodos dependentes de redes neurais tiveram tempos de execução muito altos.

Outras aplicações de redes neurais são classificação de tonalidade (Tillmann, Bharucha & Bigand 2000) ou estilo (Kiernan 2000) em segmentos musicais.

## 2.11 Algoritmos Genéticos em Composição Algorítmica

Algoritmos Genéticos têm uma grande extensão de aplicações em ciência e engenharia, em problemas complexos para os quais uma solução específica é difícil de se encontrar. A capacidade de busca de AG chamou o interesse de várias comunidades científicas, incluindo até aplicações de arte e música (Corne & Bentley 2001, Miranda & Biles 2007, Romero & Machado 2007, Todd & Latham 1992). Houve vários estudos envolvendo computação evolucionária e arte tentando entender a influência possível de sistemas bioinspirados e arte, tecnologia e avaliação estética (Romero & Machado 2007). O uso de Algoritmos Genéticos para a evolução artificial e criação de arte e música é um ramo conhecido como Arte ou Música Evolutiva (Miranda & Biles 2007).

Várias aplicações foram desenvolvidas na área de execução musical. Madsen & Widmer (2005) estudaram como tendências estilísticas de um certo artista podem ser exploradas com um Algoritmo Evolucionário (AE). Também relativa à execução, a audiência pode influenciar uma execução, de acordo com suas respostas (Biles & Eign 1995). Esta resposta dos ouvintes também pode ocorrer *online*, como mostra o estudo feito por Putnam (1996), ou através de sensores de presença, como descrito por Woolf & Yee-King (2003). Experiências também foram feitas com *bio-feedback* para ajudar a avaliação e a escolha das músicas por um DJ eletrônico (Graham-Rowe 2001).

Existem também vários estudos para aplicações musicais mas as execuções não são exclusivamente acústicas. Estas aplicações estão baseadas nas áreas de síntese e mixagem. Em relação ao último, AEs podem ser muito úteis na aplicação de efeitos e filtros (Sharman & Esparcia-Alcázar 2003) ou para descobrir parâmetros de reverberação que correspondam às características de uma sala específica (Mrozek & Wakefield 1996). Já na área de síntese de sons, dois problemas principais se destacam: a procura de parâmetros que alcançam um som-alvo (Vuori & Välimäki 1993) e a procura por novos timbres (Horner, Beauchamp & Haken 1993). Mandelis (2001) desenvolveu uma aplicação, cha-

mada Genophone, para o desenvolvimento de novos sons sem que o usuário conheça detalhes da *Sound Synthesis Technique* (SST) empregada.

Em relação à audição de músicas, muitas aplicações úteis podem surgir, principalmente para o desenvolvimento de ouvintes avaliadores que podem co-evoluir com músicas nos AGs, já que a avaliação é um dos maiores problemas para o desenvolvimento de sistemas evolutivos de CA. Um módulo ouvido (Jacob 1995) pode ser desenvolvido como avaliadores de aptidão (Jacob 1996). Já Federman (2003) desenvolveu um sistema para prever qual a próxima nota em uma sequência melódica. Outra aplicação neste sentido envolve a coevolução de cantores machos e críticas fêmeas (Todd & Werner 1999), o que só permite a evolução de material relacionado.

Todas estas tarefas musicais estão relacionadas de algum modo com CA, que tem várias aplicações baseadas em AE. CE pode ser utilizada primeiramente para gerar novas ideias melódicas. Estas novas ideias podem se dar através de sequências de notas sem ritmo (Ralley 1995), sequências de ritmo sem notas (Horowitz 1995) ou sequências de notas e ritmos (Biles 1994). Um dos estudos importantes desta área envolve maneiras mais criativas de se tratar o problema da definição de uma função de avaliação (Waschka II 2007).

Brown (2002) descreve como várias aplicações de AG para análise musical e síntese de sons foram bem sucedidas. Uma das utilizações principais de AG neste contexto é a formação de melodias. Um algoritmo para a criação de solos de jazz em tempo real é proposto por Biles (1994), mostrando como o desenvolvimento de composição automatizada para um gênero particular de música pode ser abordado com AG. Isso é feito com um AG que considera várias tarefas musicais, como audição e improvisação. Apesar de vários resultados interessantes, a Música Evolucionária ainda encara vários desafios (McCormack 2005a).

A criação de música com programas de computador tem algumas características especiais quando vistas como problemas clássicos de otimização. É comum ver procedimentos onde um indivíduo de uma população representa um compasso que pode ser usado em uma composição final. Isso faz com que procurar uma boa população com indivíduos diversos seja mais importante que a procura de um indivíduo ótimo (Waschka II 1999). A evolução de soluções boas, em vez de uma única solução ótima, é uma característica particular a ser explorada em sistemas feitos para criar arte. Esta evolução para um conjunto bom, porém, não corresponde também a problemas clássicos de otimização multiobjetivo, já que não existe a intenção de se encontrar o melhor conjunto Pareto

possível como resultado e sim uma solução boa diferente a cada execução.

Alguns compositores podem também querer estender ideias melódicas a partir de uma ideia inicial. Neste sentido, AGs podem gerar variações melódicas ou contra melodias (Polito, Daida & Bersano-Begey 1997). Já Jacob (1995) estudou como fragmentos melódicos também podem ser combinados em linhas maiores. Além de melodias, AE também podem ser usados para determinar harmonias correspondentes a uma dada melodia. Algoritmos podem gerar partes harmônicas (McIntyre 1994) ou gerar mudanças de acordes (Horner & Ayers 1995). A tarefa de harmonização também pode ser direcionada a um estilo específico, como demonstra o trabalho de Maddox & Otten (2000).

Para a criação de arranjos, o problema não é normalmente estudado diretamente, e outras aplicações para o desenvolvimento de melodias são normalmente usados (Jacob 1995), podendo ter certos ajustes para que as melodias se adaptem a instrumentos específicos.

Assim, AGs podem ser utilizados para criação e extensão de melodias, harmonias e arranjos. Além disso, é possível criar sistemas que trabalhem com estruturas musicais de mais alto nível – como refrões e versos – de maneira *top-down* (Unemi & Senda 2002) ou *bottom-up* (Jacob 1996).

No sentido de explorar similaridades entre o processo evolutivo e o musical (mais especificamente de serem processos temporais), Moroni, Manzolli, Zuben & Gudwin (2000) desenvolveram um sistema em que a própria evolução é uma composição e a interação entre os indivíduos define o que ocorrerá na próxima geração. Isso demonstra uma faceta de vários sistemas adaptativos para CA desenvolvidos ultimamente, que é a busca por uma população de boa qualidade, em vez de uma solução ótima.

Dentre as diferentes analogias feitas com o processo biológico para geração de música, outro trabalho no qual o processo evolutivo em si representa uma música é o de Waschka II (2007), no qual cada indivíduo representa um compasso e métodos menos convencionais de avaliação das soluções são utilizados. Desta maneira, o ouvinte percebe que o material entre uma geração e a outra é relacionado, porém coerentemente modificado.

Processos de geração musical em tempo real também podem ser utilizados, como feito por Biles (2002), que desenvolveu um sistema robusto para geração de solos de jazz em performances ao vivo. Sua aplicação, *GenJam*, ou *Genetic Jammer*, faz improvisações que são calibradas com músicas cujas progressões harmônicas são complexas. A tarefa de improvisação pode ser muito estimulante pois tem características de várias tarefas

musicais, como a audição e a composição, e precisa de algoritmos robustos, já que o improvisado não pode ser editado ao final do processo.

O algoritmo de Biles (2002) pode improvisar tanto sozinho quanto em resposta a outra pessoa e tem uma base de dados com várias músicas nas quais pode fazer improvisos. Neste caso, várias ideias podem ser aproveitadas como potenciais indivíduos, como um banco de dados com compassos relacionados ao estilo e à própria música. O sistema tem a qualidade de escutar o músico com o qual está interagindo para criar novas frases. Isso ocorre com a utilização de interface que pode converter alturas tonais para códigos MIDI.

Além da CE mais comum, outros artifícios podem ser muito úteis no desenvolvimento de sistemas de CA. Por exemplo, co-evolução (Hillis 1990) pode ser utilizada para o desenvolvimento simultâneo de composições (ou cantores) e críticos, que avaliarão as músicas criadas. Nestes casos, medidas de aptidão de indivíduos podem surgir das interações entre os indivíduos.

O mesmo ocorre no trabalho de McCormack (2003), onde mundos artificiais são criados e neles seus agentes podem mover, criar ou ouvir sons; e no trabalho de Gartland-Jones (2003), no qual blocos criam frases musicais com a interação com outros blocos e cada bloco carrega sempre uma respectiva frase. Trabalhos similares a estes podem ser utilizados para criar frases que tenham uma relação temática entre si, sendo o caminho percorrido no processo evolutivo potencial material intermediário. Uma música alvo pode ser utilizada para medir a aptidão de indivíduos e forçar que estes convirjam para algo parecido com este alvo.

AEs podem ser uma ferramenta muito útil para músicos e aplicações em várias áreas podem existir com relativo sucesso, dependendo do domínio em questão. Os domínios específicos devem reger como os avanços tecnológicos se darão e esta fase se apresenta iminente, de tal modo que música já pode ser considerado um domínio no qual CE pode ser, de fato, útil.

Em (Freitas & Guimarães 2011b), o autor desta tese propõe um algoritmo genético livre de aptidão e interação com o usuário onde o processo evolutivo é guiado pelos operadores genéticos e controlados apenas pela população inicial. A população de compassos é utilizada para formar a melodia final e para controlar a convergência que levaria a uma repetição de compassos, uma matriz de *takeover* é proposta para o algoritmo.

Em outro trabalho (Freitas & Guimarães 2011a), o autor desta tese propõe um al-

goritmo multiobjetivo para harmonização de melodias. Neste algoritmo, progressões harmônicas para uma melodia são criadas com duas funções objetivo como referência: uma que leva a harmonias mais simples e outra a harmonias mais dissonantes. Ao final do processo, o usuário tem um conjunto de soluções harmônicas entre as quais pode escolher.

Em um trabalho recente, Johnson (2012) estuda a noção de aptidão em música evolutiva. Os autores analisam medidas utilizadas na literatura e propõem algumas medidas que podem ser utilizadas no futuro, como buscas de conteúdo relacionado na web.

## 2.12 Conclusão

Nesta seção vimos alguns métodos históricos importantes de composição algorítmica assim como as técnicas de inteligência artificial mais empregadas atualmente, sendo estas as cadeias de Markov, Gramáticas Gerativas, EMI, Redes de Transição, Sistemas de Lindenmayer, Caos, Autossimilaridade, Autômatos Celulares, Redes Neurais Artificiais e Algoritmos Genéticos. Cada uma destas técnicas pode ter vantagens e desvantagens em relação ao modelo proposto de composição. Nos próximos capítulos desenvolveremos nosso modelo baseado na avaliação de várias funções objetivo e estudaremos suas implicações do ponto de vista de um problema de otimização multiobjetivo.

# Capítulo 3

## Otimização Multiobjetivo

*“Há duas escolhas simples na vida: aceitar as condições como elas existem ou aceitar a responsabilidade de mudá-las.”*

— Denis Waitley, 1924—

### 3.1 Introdução

Em aplicações de engenharia e projetos de sistemas, frequentemente encaramos objetivos e critérios de qualidade múltiplos e conflitantes. Usualmente, estes são combinados em uma única função objetivo que reflete a utilidade de cada objetivo, enquanto alguns objetivos são transformados em restrições. Em Otimização Multiobjetivo, os objetivos são tratados separadamente como objetivos não comparáveis. Assim, um número de soluções, em vez de apenas uma solução, pode ser encontrado para representar o compromisso entre estes objetivos (Deb 2009, Coello, Lamont & Veldhuizen 2007, Fonseca & Fleming 1993).

As soluções que representam este compromisso formam o conjunto Pareto-ótimo, soluções que levam ao conjunto ótimo de pontos não comparáveis no espaço de objetivos. Isto significa que não há solução neste conjunto factível que é melhor que (domina) uma outra solução do conjunto Pareto-ótimo. Um problema de otimização multiobjetivo pode ser formulado como:

$$\begin{aligned} \min & (f_1(x), f_2(x), f_3(x), \dots, f_k(x)) \\ & x \in X \end{aligned} \tag{3.1}$$

onde o número inteiro  $k$  é o número de objetivos e o conjunto  $X$  é o conjunto factível de vetores de decisão. Neste tipo de problema, usualmente não existe uma solução factível que minimize todos os objetivos simultaneamente. Assim, é dada atenção às soluções Pareto ótimas, ou seja, soluções que não podem ser melhoradas em nenhum dos objetivos sem uma piora em pelo menos algum outro objetivo. Diz-se que uma solução  $x^1 \in X$  Pareto domina outra solução  $x^2 \in X$  se:

$$\begin{aligned} f_i(x^1) &\leq f_i(x^2) \text{ para todo } i \in \{1, 2, \dots, k\} \text{ e} \\ f_j(x^1) &< f_j(x^2) \text{ para algum } j \in \{1, 2, \dots, k\} \end{aligned} \quad (3.2)$$

Esta mesma solução  $x^1$  é denominada Pareto ótima se não há outra solução que a domina. O conjunto de soluções Pareto ótimas é chamado de frente de Pareto. O conjunto de soluções Pareto é composto por todas as soluções factíveis não dominadas e no contexto de dominância Pareto são soluções incomparáveis.

### 3.1.1 Métodos de Otimização Multiobjetivo

Algoritmos Evolutivos Multiobjetivo (MOEA - *Multi-objective Evolutionary Algorithms (MOEA)*) para a solução de problemas multiobjetivo têm sido desenvolvidos desde os anos 1990. A primeira geração de MOEAs (Coello, Lamont & Veldhuizen 2007) incorporavam classificação baseada em dominância Pareto e comparação para avaliação da aptidão. Desde então, MOEAs têm sido largamente reconhecidos como métodos apropriados e bem sucedidos para resolver problemas multiobjetivo neste contexto.

Uma vantagem de se utilizar algoritmos evolucionários para resolver problemas de otimização multiobjetivo é o fato de que eles funcionam com uma população de soluções candidatas. Esta evolução baseada em conjuntos oferece a possibilidade de busca por um conjunto de estimativas em apenas uma execução. Porém, há outras vantagens importantes na utilização de MOEAs:

- Generalidade e flexibilidade em relação a características do problema
  - Não há necessidade do problema ser baseado em uma função diferenciável
  - O formato do espaço de soluções ou de objetivos não é relevante
  - Qualquer tipo de variável pode ser utilizado na modelagem do problema

- Métodos evolucionários têm capacidade de continuar o processo de otimização em problemas com ruído
- Existência de mecanismos para controle e preservação da diversidade (Zitzler & Thiele 1999, Laumanns, Thiele, Deb & Zitzler 2002, Knowles & Corne 2000, Ikeda, Kita & Kobayashi 2001, Laumanns, Zitzler & Thiele 2001, Hernández-Díaz, Santana-Quintero, Coello Coello, Caballero & Molina 2006, Sato, Aguirre & Tanaka 2007, Batista, Campelo, Guimarães & Ramírez 2011)
- É possível integrar a tomada de decisão ao processo de otimização (Molina, Santana, Hernández-Díaz, Coello Coello & Caballero 2009, Luque, Miettinen, Eskelinen & Ruiz 2009, Xiong, Chen, Yang, Zhao & Xing 2012, Auger, Bader, Brockhoff & Zitzler 2012)
- É possível fazer a inclusão de métodos especializados ou abordagens híbridas (Grosan & Abraham 2007)

A segunda geração de MOEAs produziu algoritmos que são mais eficientes do ponto de vista computacional. Estes algoritmos introduziram um elitismo explícito por meio de uma população em arquivo externo, para guardar estimativas do conjunto Pareto.

O NSGA-II (*Non-Dominated Sorting Genetic Algorithm - II*) (Deb, Agrawal, Pratap & Meyarivan 2000) é um algoritmo particularmente importante para a solução de problemas de otimização multiobjetivo. O NSGA-II atribui melhores valores de aptidão a soluções não dominadas e procura soluções não dominadas recursivamente nas soluções restantes. As soluções na segunda frente de Pareto recebem piores valores de aptidão e assim por diante. Além disso, as soluções que pertencem à mesma frente têm seus valores de aptidão diferenciados pelas distâncias uns dos outros, penalizando áreas do Pareto com muitas soluções. Interessantemente diferente da atribuição de frentes, o cálculo da distância entre pontos implicitamente assume que os objetivos são comparáveis pois unidades de objetivos diferentes são somadas. Isto é um aspecto que ocorre em vários algoritmos para problemas multiobjetivo.

O SPEA2 (*Strength Pareto Evolutionary Algorithm 2*) (Zitzler, Laumanns, Thiele, Zitzler, Zitzler, Thiele & Thiele 2001) usa informação de dominância de uma maneira diferente. Ele incorpora uma medida de aptidão baseada no número de indivíduos que uma solução domina ou número de indivíduos que dominam uma solução.

Um método muito diferente para problemas de otimização multiobjetivo é o PAES

(*Pareto Archived Evolution Strategy*) (Knowles & Corne 2000), que usa uma estratégia evolutiva de busca local (1+1) para procurar por soluções novas. As soluções novas são guardadas em um arquivo de soluções não dominadas. Ao variar o número de  $\mu$  pais e  $\lambda$  filhos, versões (1+1)-PAES, (1+ $\lambda$ )-PAES e ( $\mu$ + $\lambda$ )-PAES também foram testadas.

Uma abordagem mais complexa é o PESA-II (*Pareto Envelope-based Selection Algorithm - II*) (Corne, Jerram, Knowles, Oates & J 2001), no qual o volume de hipercaixas geradas no espaço de soluções é utilizado para atribuir aptidão a indivíduos. Aptidão é então atribuída às hipercaixas e soluções são escolhidas a partir das hipercaixas selecionadas através de seleção natural. As hipercaixas levaram a conjuntos de resultados mais distribuídos pela frente de Pareto.

Para ter soluções mais distribuídas pela frente de Pareto, algoritmos que incluem mecanismos para manter diversidade no espaço de objetivos como  $\epsilon$ -MOEA e cone  $\epsilon$ -MOEA foram propostos (Batista, Campelo, Guimarães & Ramírez 2011). Estes algoritmos incluem formas relaxadas de dominância nas quais menos soluções são consideradas não dominadas em um conjunto de indivíduos vizinhos.

### 3.1.2 Visualização da Qualidade das Soluções em Problemas Multiobjetivo

Os resultados de problemas com duas funções objetivo podem ser facilmente representados em dois eixos simples como na Figura 3.1. Nesta representação, tem-se a qualidade de várias soluções, cada uma representada por um ponto. As soluções não dominadas são aquelas marcadas com número 1 na primeira frente. Estas são as soluções não comparáveis. Todas as outras soluções são piores que pelo menos uma solução na primeira frente em relação à minimização dos objetivos. Usualmente, as soluções dominadas não são mostradas no gráfico quando queremos tornar a informação mais clara.

Uma simples extensão como a da Figure 3.2 pode ser usada para problemas com 3 objetivos. Neste caso, o formato das soluções não dominadas tem uma dimensão a mais. No caso de 3 dimensões, a interpretação dos resultados através dos pontos se torna mais difícil.

## 3.2 Otimização de Problemas com Muitos Objetivos

Apesar do sucesso dos MOEAs para problemas práticos de otimização multiobjetivo, pesquisadores passaram a investigar a escalabilidade dos MOEAs quando problemas têm mais de três objetivos e foi observado que o desempenho destes algoritmos reduz drama-

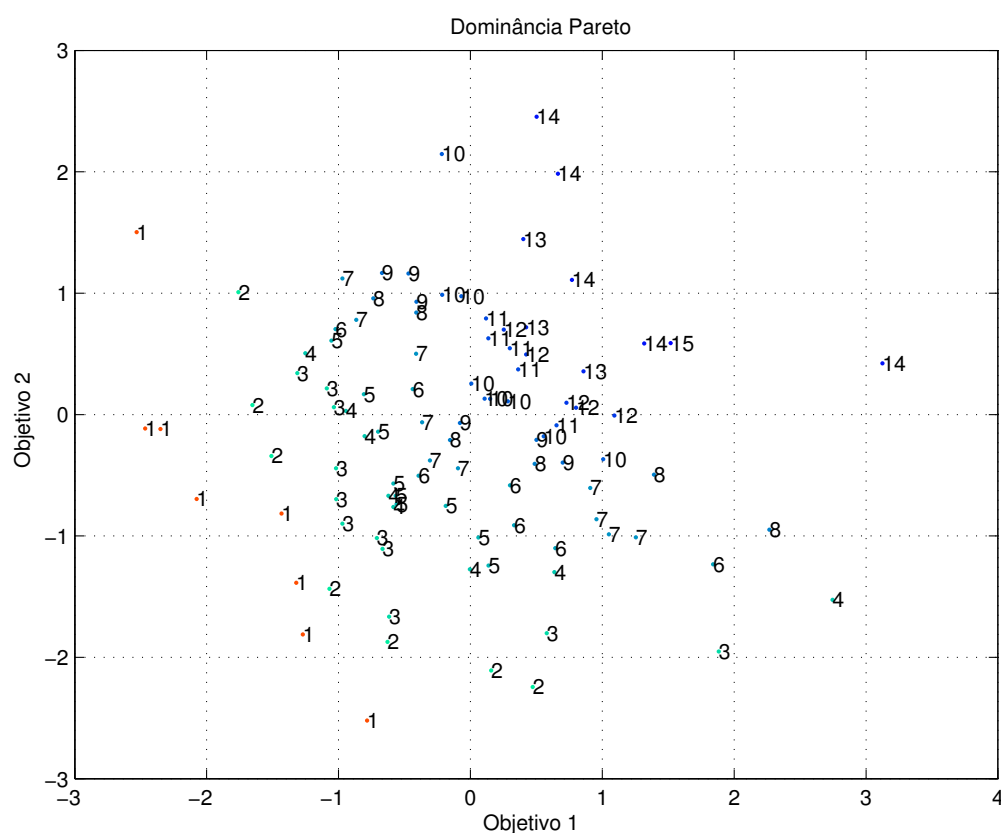
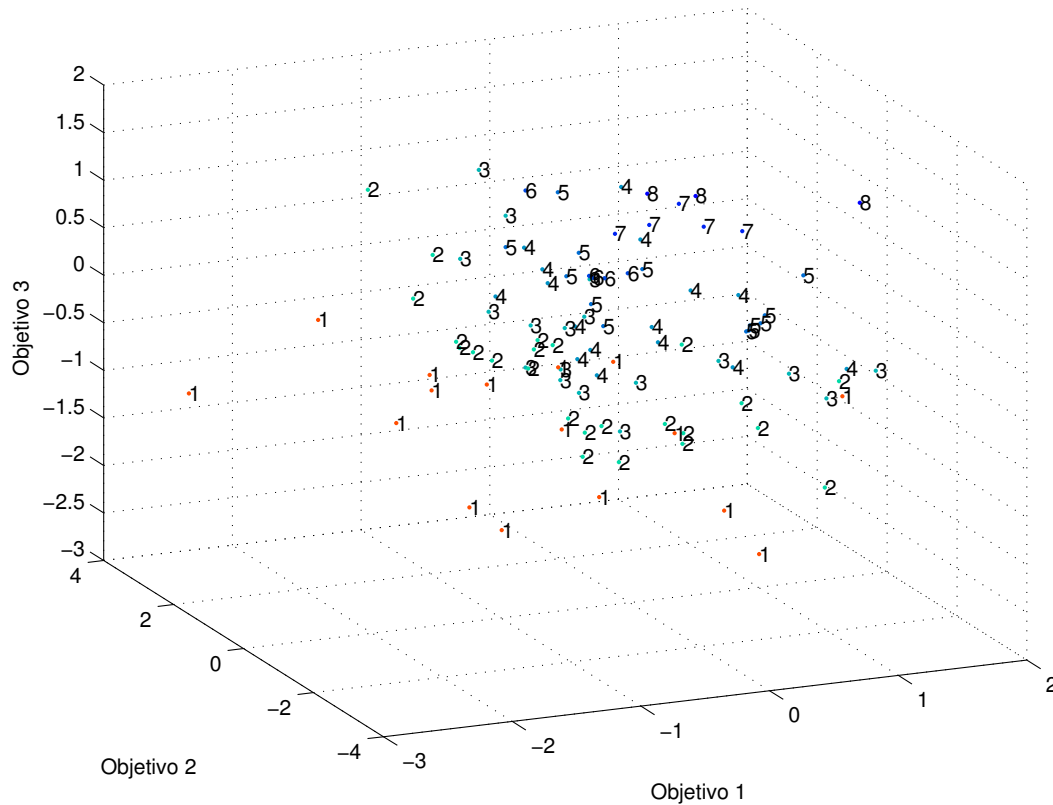


Figura 3.1: Visualização em 2 dimensões

ticamente. Este fato levou à definição de um novo termo para melhor referenciar a este contexto, definido como Problemas com Muitos Objetivos (*Many-Objective Problems*) (Ishibuchi, Tsukamoto & Nojima 2008). Em Problemas com Muitos Objetivos, menos soluções podem ser claramente reconhecidas como melhores que outras no espaço de objetivos e isto compromete a convergência dos algoritmos, já que eles dependem de comparações de dominância de Pareto.

Khare, Yao & Deb (2003) mostraram a baixa capacidade de escalabilidade dos métodos NSGA-II, PESA e SPEA2 em funções de teste escaláveis. Hughes (2005) mostrou que métodos de agregação com multipartida desempenham melhor que MOEAs. Knowles & Corne (2007) ilustraram que a habilidade dos operadores de variação de produzir soluções que dominam seus pais diminui com um número crescente de objetivos.

A aproximação da frente de Pareto precisa de um número muito grande de soluções para uma boa representação do conjunto e melhores estratégias são necessárias para uma



**Figura 3.2:** Visualização em 3 dimensões

amostragem relevante destes pontos. Além disto, é mais difícil visualizar o conjunto de soluções finais. Assim, os maiores problemas com Otimização com Muitos Objetivos são:

- Há perda de pressão seletiva pois a maior parte da população se torna não dominada, então não há tendência seletiva para direcionar a busca em qualquer direção. O aumento de soluções não dominadas à medida que o número de objetivos cresce foi identificada e reproduzida em vários artigos (Purshouse & Fleming 2007, Garza-Fabre, Toscano-Pulido, Coello Coello & Rodriguez-Tello 2011, Batista, Campelo, Guimarães & Ramírez 2011)
- A alta dimensionalidade leva a um custo computacional aumentado pois é necessário ter mais indivíduos na população para a representação da frente de Pareto
- Há dificuldade em se visualizar compromissos entre objetivos em várias dimensões

- Tomada de decisão sobre um conjunto muito grande de alternativas é um processo difícil pois requer escalabilidade das técnicas de tomada de decisão

Ideias para resolver estes problemas incluem relações de dominância diferentes para que mais soluções sejam dominadas a cada geração (Batista, Campelo, Guimarães & Ramírez 2011), outras medidas de aptidão envolvendo indicadores (Beume, Naujoks & Wagner 2006) ou otimizar várias funções objetivo escalarizadas simultaneamente (Ishibuchi, Doi & Nojima 2006). Um algoritmo recente com bom desempenho em problemas com até 7 objetivos é o PICEA-g (Wang, Purshouse & Fleming 2012). Basicamente as abordagens propostas para estes problemas são classificadas como:

- Relações de dominância diferentes tais que mais soluções sejam dominadas a cada geração (Batista, Campelo, Guimarães & Ramírez 2011, Sato, Aguirre & Tanaka 2007, Saxena, Ray, Deb & Tiwari 2009, Batista, Campelo, Guimarães & Ramírez 2011)
- Modificar métodos de classificação de soluções (Drechsler, Drechsler & Becker 2001, Zou, Zhang & Wang 2008, Sato, Aguirre & Tanaka 2009, Wang, Wu & Mao 2007, Knowles & Corne 2007)
- Outras medidas de aptidão envolvendo indicadores (Beume, Naujoks & Wagner 2006, Zitzler & Künzli 2004, Beume, Naujoks & Emmerich 2007, Brockhoff & Zitzler 2006, Brockhoff & Zitzler 2007, Auger, Bader, Brockhoff & Zitzler 2012, Saxena & Deb 2007)
- Otimizar várias funções objetivo escalarizadas por vez (Hughes 2007, Ishibuchi, Doi & Nojima 2006, Ishibuchi & Nojima 2007, Wickramasinghe & Li 2009)
- Redução de dimensionalidade em objetivos redundantes (Brockhoff & Zitzler 2006, Brockhoff & Zitzler 2007, Saxena & Deb 2007, Singh & Singh 2011)

Brockhoff & Zitzler (2007) propuseram um método para redução de objetivos equipado para integração na busca evolutiva. O algoritmo depende de uma mudança na estrutura de dominância. Eles também apresentam o problema de se achar o subconjunto mínimo de objetivos, mantendo a estrutura de dominância dada com um dado erro. Um algoritmo guloso remove objetivos se as relações de dominância não mudam com a remoção do objetivo porque ele é considerado um objetivo sem conflito.

Saxena & Deb (2007) empregam Análise de Componentes Principais (PCA - *Principal Component Analysis*) aos valores objetivo para identificar objetivos relacionados. O PCA é adaptado para manejar relações não lineares entre os objetivos e a técnica é então embutida em um NSGA-II. Outros artigos apresentaram novas técnicas baseadas em desdobramento de variância máxima e PCA para redução de objetivos não lineares (Saxena, Duro, Tiwari, Deb & Zhang 2013). Estas técnicas removem dependências nas soluções não dominadas.

López Jaimes, Coello Coello & Chakraborty (2008) empregaram seleção de características não supervisionada para reduzir o número de objetivos de acordo com sua correlação. Objetivos distantes são tratados como conflitantes. Similarmente a Brockhoff & Zitzler (2007), o algoritmo também procura pelo subconjunto mínimo de objetivos com erro mínimo.

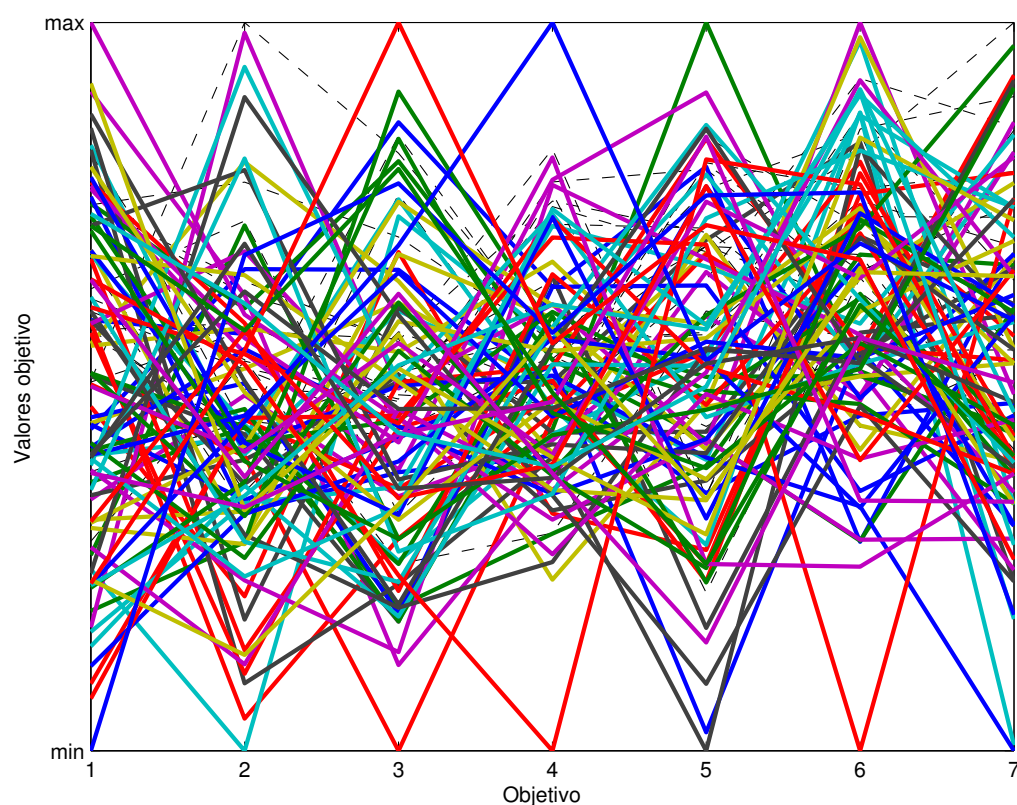
Singh, Isaacs & Ray (2011) apresentam o *Pareto Corner Search*. Este algoritmo foca apenas em cantos entre as soluções não dominadas. Depois disto, o número de soluções não dominadas resultante de uma redução nas soluções alcançadas indica se os objetivos são importantes ou se eles podem ser reduzidos.

Como veremos nos próximos capítulos, a formulação de problemas com muitos objetivos é de fundamental importância para esta tese devido ao grande número de critérios que podem ser considerados em problemas de composição algorítmica.

### 3.2.1 Visualização dos Resultados para Problemas com Muitos Objetivos

Para visualizar a qualidade de soluções de problemas de otimização com muitos objetivos, a abordagem mais comum é a de coordenadas paralelas, como apresentado na Figura 3.3 para 7 objetivos.

Neste gráfico, cada linha representa a qualidade de uma solução e os valores das funções objetivo são normalizados entre o máximo e mínimo. Esta normalização é frequentemente dos valores mínimos até o valor máximo conhecidos ou em relação a uma meta mínima esperada pelo usuário para cada objetivo. Quando há várias linhas se cruzando entre dois objetivos adjacentes, isto significa que há conflito entre estes objetivos. Neste gráfico, também, cores diferentes são atribuídas às linhas para tornar a visualização mais fácil e soluções dominadas são representadas com uma linha preta fina pontilhada.



**Figura 3.3:** Visualização do resultado de um problema de otimização multiobjetivo com 7 funções objetivo. Nesta abordagem de coordenadas paralelas, cada linha representa uma solução.

Neste exemplo da Figura 3.3, 82 das 100 soluções são não dominadas, ou não comparáveis de acordo com a dominância Pareto. Isto significa que um algoritmo baseado em dominância Pareto não teria qualquer referência para classificar soluções diferentes a não ser que usemos alguma medida que implique em comparabilidade entre os objetivos, seja de acordo com a variação entre os valores de máximo e mínimo, a área dominada por uma solução, medidas de distância ou até mesmo valores absolutos em cada objetivo. Medidas de área, distância ou volume dependem das unidades que compõem a métrica e conseqüentemente precisam implicar em comparabilidade entre os objetivos também ou a medida em si será multiobjetivo novamente. Assim, ao utilizar estas métricas, o algoritmo já tende a dar preferência a um objetivo sobre o outro. Nesta tese, é tomada a decisão de se desenvolver um trabalho que não use medidas que impliquem em comparabilidade entre os objetivos, ou seja, baseado estritamente em dominância Pareto para

comparação de objetivos.

Outro problema com o método de representação usado na Figura 3.3 é que nós podemos perceber conflito entre objetivos adjacentes. Para representar soluções em dimensões altas mais eficientemente, outras abordagens frequentemente envolvem mapear os objetivos em dimensões mais baixas (Obayashi & Sasaki 2003, Yoshikawa & Furuhashi 2007, Koppen & Yoshida 2007).

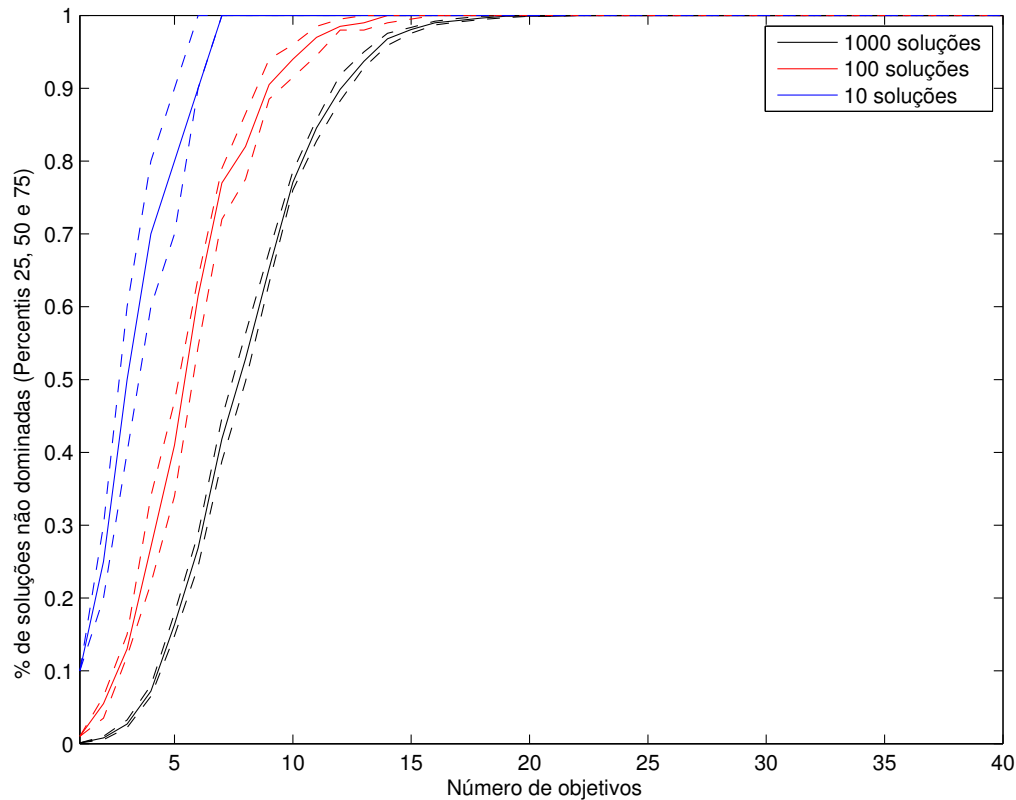
### 3.2.2 Problema de escalabilidade

Os valores na Figura 3.3 são valores Gaussianos gerados em volta de zero para simular valores no espaço de objetivos. Quando executamos um algoritmo nestas soluções para gerarmos novas soluções iterativamente com o intuito de termos uma convergência em direção à frente de Pareto, cada vez mais soluções serão não dominadas e incomparáveis em relação à dominância Pareto.

Isto torna impossível uma otimização posterior das soluções correntes. Quanto mais objetivos existirem, mais incomparáveis se tornam as soluções por estarem não dominadas, a não ser que usemos uma medida que implica em comparabilidade entre os objetivos em sua definição, como distância, área ou hipervolume. Estas medidas não são então escaláveis e qualquer tipo de normalização nos valores de função objetivo alteraria os resultados do problema. Outra medida útil para implicar em comparabilidade entre objetivos é usar o número de soluções dominadas para atribuir aptidão a indivíduos pois isto beneficia soluções menos extremas em relação aos objetivos obtidos do espaço de soluções.

Se geramos 1000 pontos Gaussianos em um número variado de dimensões, podemos perceber o forte aumento no número mediano de soluções não dominadas como mostra a Figura 3.4. Na geração desta Figura, foram feitas 100 repetições do experimento para cada número de objetivos.

Cada conjunto colorido de 3 linhas, incluindo 2 linhas pontilhadas cada, representam este rápido aumento com 10, 100 e 1000 soluções, respectivamente. As linhas pontilhadas em cada conjunto representam o 25º e o 75º percentis, que são sempre bem próximos da mediana, representada pela linha não pontilhada. Por exemplo, podemos ver da Figura 3.4 que, se um conjunto de 1000 soluções for retornado ao tomador de decisão como solução de um problema multiobjetivo, podemos dizer muito conservadoramente que é alta a probabilidade de só termos soluções incomparáveis para mais de 25 objetivos. Ainda assim, esta afirmação é muito conservadora pois:



**Figura 3.4:** Número de soluções não dominadas em um conjunto aleatoriamente gerado com distribuição Gaussiana

1. Se menos soluções (10 ou 100, por exemplo) são retornadas, ainda menos objetivos (12 e 17, respectivamente) são necessários para tornar todas as soluções intrinsecamente incomparáveis para todas as 100 réplicas do experimento.
2. Para uma população inicial de soluções aleatórias, a maior parte dos problemas não têm a distribuição de seus valores de função objetivo tão bem espaçados no espaço de objetivos como a distribuição Gaussiana de valores usada neste experimento. Para soluções geradas por um algoritmo de otimização, a distribuição da população ainda menos uniforme e tendendo a ter mais soluções não dominadas.
3. Mesmo se algumas soluções da população são comparáveis, isto ainda não dá uma boa referência de para os algoritmos de otimização baseados em dominância Pareto pois as piores soluções, dominadas, são descartadas a cada geração.

Assim, aplicando-se os valores da Figura 3.4 a um modelo de regressão simples,

problemas com mais de  $5.2n^{\frac{1}{5}} + 3.6$  objetivos conflitantes, onde  $n$  é o número de soluções, são muito conservadoramente prováveis que tenham apenas soluções incomparáveis. Este modelo de regressão simples não tem a intenção de dar uma definição forte de Problemas com Muitos Objetivos com soluções incomparáveis mas sim de servir como referência para este trabalho.

Pode-se perceber quão conservadora é também esta regressão com os resultados para  $n = 1, 2$ , que são 8, 87 e 9, 64. Uma classificação forte de problemas com muitos objetivos como tendo apenas soluções incomparáveis deve depender também da percepção de incomparabilidade das soluções. Uma regressão menos conservadora com os valores medianos 7, 14 e 22 para 10, 100 e 1000 objetivos seria  $42.88n^{0.05799} - 42$ . Frizamos aqui a falta de precisão para valores  $n < 10$  ou  $n > 1000$  como para qualquer interpolação de modelo de regressão. Assim, pode-se perceber porque tanto a otimização quanto visualização de soluções para problemas com muitos objetivos ainda são problemas em aberto.

Uma última maneira de olhar para o problema é que se temos apenas um objetivo, precisaríamos de apenas uma solução para representar o melhor valor de função objetivo possível. Para 2 objetivos conflitantes, se queremos uma granularidade que representa apenas as soluções extremas, precisaríamos de 2 soluções da curva formada pela frente de Pareto: um com o melhor valor para cada um dos objetivos. Para 3 objetivos conflitantes, a frente de Pareto é então um plano e precisaríamos de 4 soluções para representar as soluções extremas. Do hiperplano formado por soluções em 4 objetivos, precisaríamos então de 8 soluções apenas para representar soluções extremas e assim por diante. Assim, mesmo no caso em que guardamos apenas as soluções extremas, que usualmente não são as mais apropriadas, precisamos de pelo menos  $2^{M-1}$  soluções para representar este compromisso, onde  $M$  é o número de objetivos do problema.

Esta relação exponencial mostra a importância de se tratar otimização com muitos objetivos, com apenas soluções incomparáveis, como uma classe diferente de problemas. Nestes cenários, é mais frutífero entender a relação entre os objetivos e facilitar a função dos tomadores de decisão do que investir na tarefa não polinomial de se encontrar uma frente apropriada com as melhores soluções.

Independentemente do número de objetivos, há temos um problema que formalmente tem muitos objetivos quando a maior parte da população se torna não dominada e então há perda de pressão seletiva, seja com 5, 10 ou 35 objetivos. Quando chegamos ao ponto onde as técnicas correntes não podem resolver o problema adequadamente, estamos no

domínio dos problemas com muitos objetivos. Adicionalmente, para tarefas diferentes, este limiar pode se alterar. Por exemplo, para visualização, problemas começam a ocorrer para mais de 3 objetivos.

### 3.3 Conclusão

Dado um conjunto de soluções incomparáveis em relação a todos os objetivos, usamos métodos de visualização como os da Figura 3.3 para representar o compromisso entre os objetivos. Tendo estes compromissos, o tomador de decisão pode então decidir qual é uma boa abordagem para resolver o problema reduzindo o espaço de preferências, eliminando objetivos ou mesmo mudando fatores do mundo real que resultam em um novo modelo para o problema.

O problema com coordenadas paralelas, tais como a da Figura 3.3 é que podemos apenas ver a diferença entre objetivos adjacentes e nossa percepção de conflito e harmonia entre objetivos pode ser tendenciosa pela normalização feita nos resultados. Também, não é possível executar qualquer mapeamento em uma dimensão mais baixa sem perda de informação.

Considerando qualquer problema com muitos objetivos e as características que o tomador de decisão precisará ver nos dados, precisamos de um método de visualização no qual é possível perceber facilmente os mesmos componentes que perceberíamos em um gráfico de 2 objetivos, tais como:

1. A quantidade de conflito entre os objetivos
2. A quantidade de harmonia entre os objetivos
3. A globalidade ou localidade destas características no conjunto de pontos
4. A posição destas características, se elas forem locais

A última característica é importante para o tomador de decisão pois se há apenas conflito para soluções que são ruins em relação a dois objetivos dados, o tomador de decisões provavelmente tomaria decisões muito diferentes do caso em que há conflitos entre as soluções boas.

Além disto, em otimização com muitos objetivos temos feições que não estão presentes em otimização biobjetivo, como:

1. Existe a necessidade de se comparar qual o conflito entre diferentes combinações de objetivos e não apenas 2 objetivos.
2. Alguns objetivos podem ser agregados para uma conveniente redução de dimensionalidade. Precisamos então neste cenário identificar qual combinação e redução de objetivos em uma função agregada causaria menos distorção na representação final da frente de Pareto.
3. Da mesma maneira que há necessidade de se identificar a diferença de conflito entre os objetivos, há também a necessidade de se identificar grupos de objetivos que sejam conflitantes.

A resposta apropriada para estas questões daria informação o suficiente para o tomador de decisão entender o problema e então tomar medidas apropriadas para filtrar soluções e decidir em uma solução única. Com uma indicação de quais objetivos têm menos conflito entre si, torna-se mais fácil decidir uma área de preferência conveniente até que tenhamos menos soluções e finalmente uma solução única. Podemos ver que as medidas acuradas das características mencionadas acima envolveriam:

1. Um cálculo robusto de conflito e harmonia entre objetivos, que são frequentemente em unidades diferentes.
2. Definição da causa dos valores finais de conflito e harmonia, para definir localidade e globalidade.
3. Um método de visualização que possa conter toda esta informação de maneira fácil de se interpretar.

# Capítulo 4

## Árvores de Agregação

*“Felicidade é quando o que você pensa, diz e faz estão em harmonia.”*

— Mahatma Gandhi, 1869–1948

### 4.1 Introdução

Apresenta-se neste Capítulo uma abordagem chamada Árvores de Agregação para tratamento de problemas de otimização com muitos objetivos. Esta abordagem se baseia na organização de objetivos em ramos de uma árvore de modo que representa as melhores possibilidades de agregação de objetivos. Os melhores candidatos a agregação são definidos de acordo com a harmonia entre cada par de objetivos.

Otimização Multiobjetivo trata de objetivos como não comparáveis e um número de soluções pode ser encontrado para representar o compromisso entre estes objetivos. Algoritmos evolucionários têm sido aplicados com sucesso a problemas práticos mas sua escalabilidade para problemas com muitos objetivos não funciona bem e seus desempenhos reduzem significativamente. Neste contexto, apresenta-se aqui uma estratégia para redução e visualização de relação entre estes objetivos em tais problemas de acordo com uma medida de sua harmonia <sup>1</sup>.

Apresentam-se então formas de se mensurar conflito e harmonia, sendo a última usada para agrupar iterativamente os objetivos mais harmônicos em um novo objetivo

---

<sup>1</sup>O conceito de harmonia entre objetivos no contexto de otimização não é relacionado ao conceito de harmonia musical

composto. Em contraste com outras medidas simples de conflito, a medida de harmonia apresentada é não paramétrica e conseqüentemente pode ser robusta independentemente da relação de valor entre os objetivos.

Assim, o método de Árvores de Agregação:

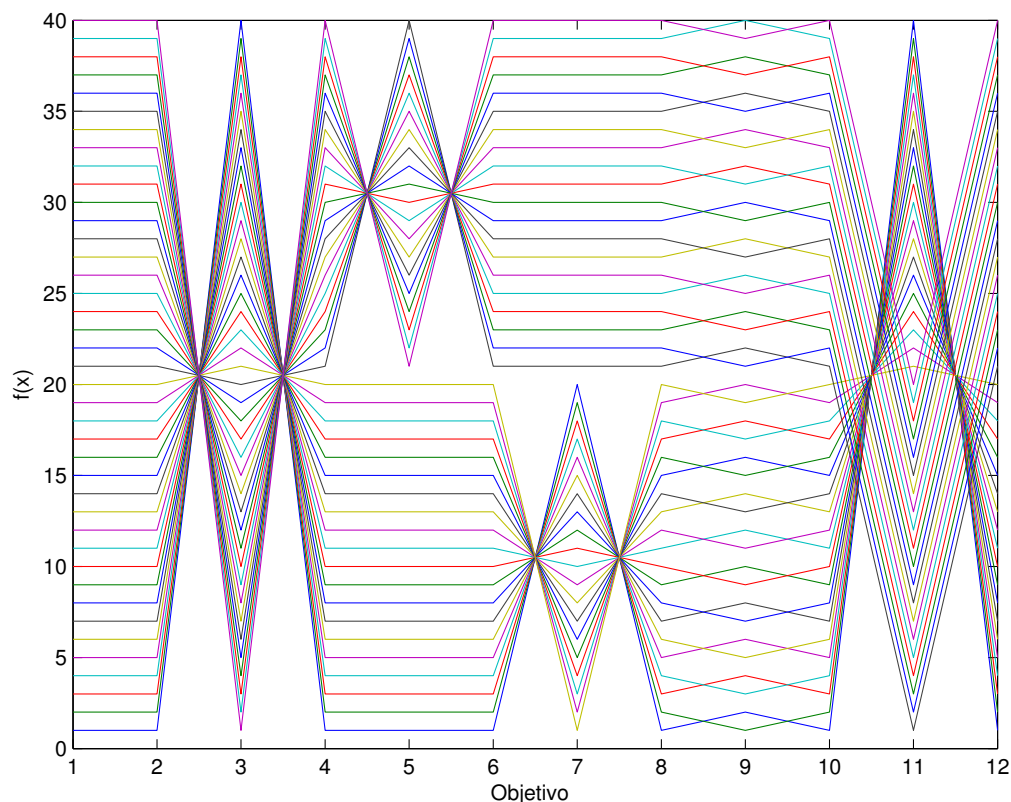
- É um método que permite uma visualização fácil das relações entre objetivos com um esquema de redução iterativa e cálculo de harmonia e conflito.
- Independência de qualquer relação de linearidade entre os objetivos para desempenhar a agregação de objetivos.
- Tem a capacidade de funcionar bem com qualquer número de objetivos a serem reduzidos a critério do tomador de decisão.
- É um algoritmo fácil de ser implementado.

Para exemplificar a utilidade da Árvore de Agregação que será proposta, considere o conjunto de soluções apresentado em coordenadas paralelas na Figura 4.1. Todos os pares de objetivos neste conjunto de soluções contêm algum dos 6 tipos de conflito apresentados na Figura 4.5 com uma granularidade de 40 pontos. Os objetivos 1 e 2 estão em harmonia completa. Os objetivos 2, 4, 6, 8, 10 e 12 estão também em harmonia completa pois para cada par de objetivos representando um tipo de conflito, os valores retornam aos valores do objetivo 1 antes de se passar para o próximo tipo de conflito adjacente.

A Figura 4.2 representa a relação entre estes objetivos e sua reducibilidade em uma estrutura de árvore. Estes grupos são iterativamente formados com os objetivos que estão com mais harmonia enquanto porcentagens nos nós pais mostram o conflito entre os objetivos filho. Medidas de conflito são discutidas na Seção 4.2.

Mais especificamente, esta Árvore de Agregação tem as seguintes propriedades:

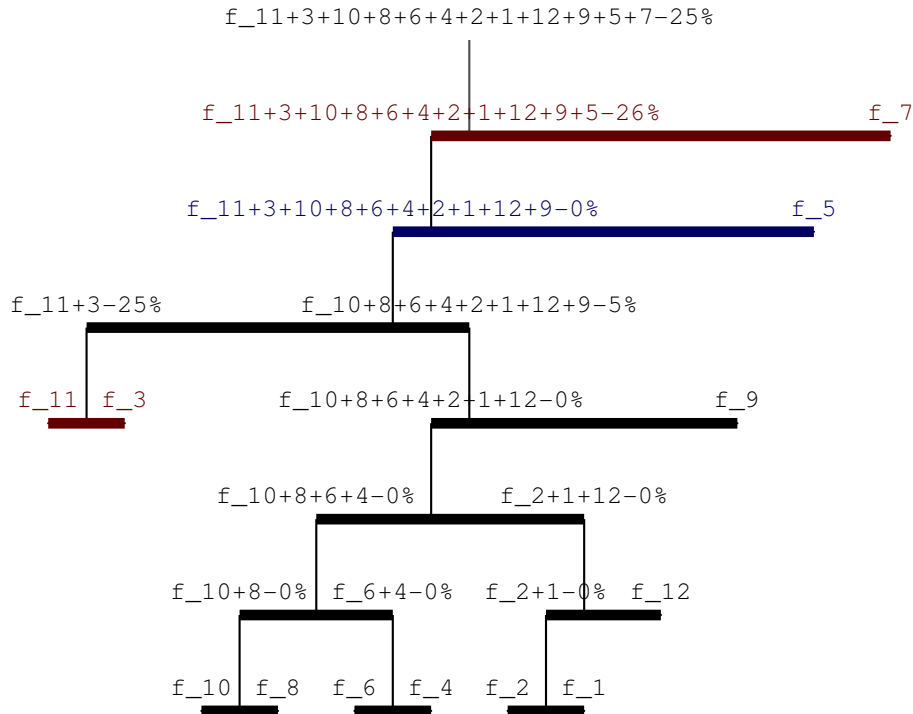
- Nós folha representam objetivos  $f_n$  na forma  $f\_n$ , onde  $n$  é o número do objetivo.
- Pais de nós folha representam um objetivo composto formado pela soma dos objetivos e renormalização de seus valores. Este é representado na forma  $f\_n1+n2-c$  (agregação de  $f_{n1}$  e  $f_{n2}$  com conflito  $c$ ), onde  $n1$  e  $n2$  são os objetivos utilizados na agregação e  $c$  é o conflito entre estes objetivos. Por exemplo, o nó  $f\_10+8-0\%$



**Figura 4.1:** Um exemplo de conjunto de soluções com uma variedade de conflitos

representa um objetivo composto com os objetivos  $f_{10}$  e  $f_8$ . Este nó também mostra que há 0% de conflito entre estes dois objetivos e eles são agrupados pois estão em harmonia completa.

- Outros nós representam objetivos compostos de ordens mais altas formados de maneira similar. Os valores de conflito nestes nós incluem apenas o conflito entre os objetivos já compostos de seus filhos. Por exemplo, além de representar o objetivo composto formado com os objetivos 10, 8, 6 e 4, o nó  $f_{10+8+6+4-0\%}$  também mostra que há 0% de conflito entre os objetivos compostos  $f_{10+8}$  e  $f_{6+4}$ . O conflito 0% entre objetivos pares acontece porque os valores de função objetivo para objetivos pares na Figura 4.1 sempre retornam para os mesmos valores do objetivo 2, enquanto objetivos ímpares estão sendo usados para didaticamente demonstrar os tipos de conflito da Figura 4.5.



**Figura 4.2:** Árvore de Agregação para o conjunto de soluções da Figura 4.1

- Objetivos são respectivamente agrupados de acordo com sua harmonia e, conseqüentemente, primos distantes na árvore representam objetivos menos harmoniosos.
- Nós em preto representam conflito global. Valores em vermelho representam conflito local mais concentrado em bons valores para estes objetivos. Valores em azul representam conflito para soluções ruins nestes objetivos.

Quando dois objetivos compostos são agrupados em um novo objetivo composto, o conflito nos objetivos compostos anteriores não é considerado no conflito do objetivo novo. Assim, um recurso adicional não incluído neste exemplo para facilitar a visualização é ter o tamanho de cada ramo da árvore proporcional ao conflito de modo que o somatório de conflito na esquerda mostre a quantidade de conflito acumulado que temos do nó raiz até qualquer posição da árvore.

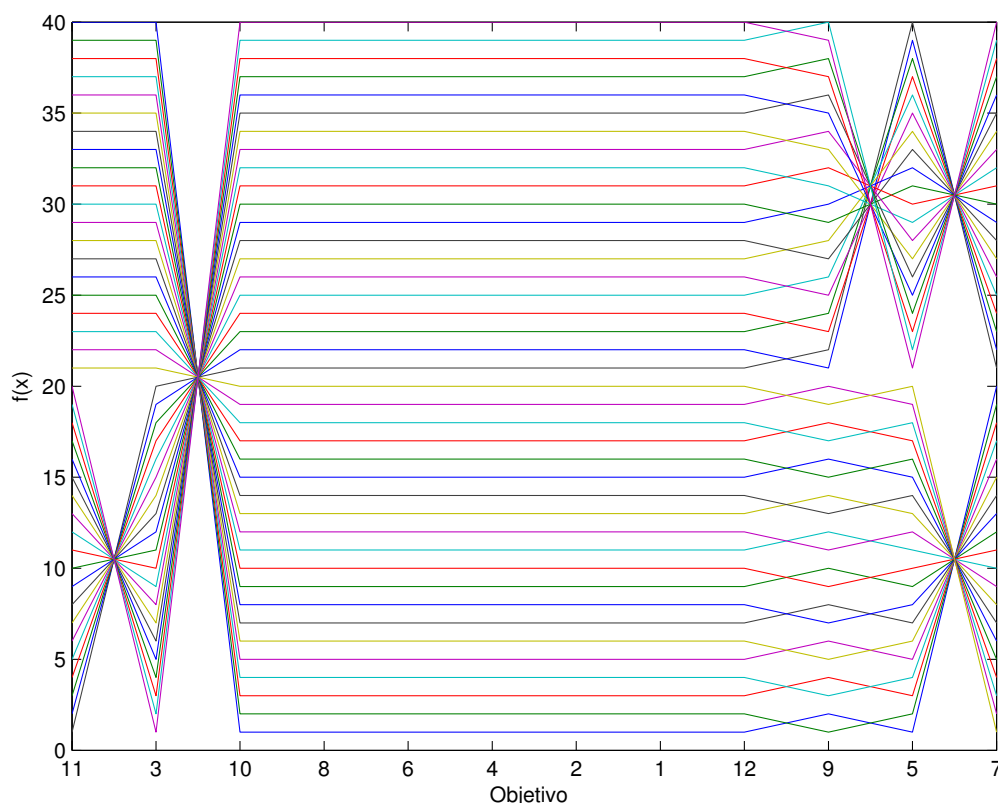
Os valores de conflito na árvore podem ser normalizados se valores altos de conflitos dificultarem a cognição. Opções para esta normalização são:

- Normalizar os valores em relação aos valores máximo e mínimo de conflito
  - Tem a vantagem de não depender da medida de conflito usada
  - Tem a desvantagem de não representar bem a quantidade geral de conflito no problema para o qual há uma quantidade similar de conflito entre todos os objetivos
  - Não representa bem a quantidade de conflito em relação às possibilidades de conflito em outros problemas
- Definir o valor máximo de conflito para aquela medida de conflito e representar conflito em termos percentuais ou unidades de conflito máximo
  - Tem a vantagem de dar boa referência cognitiva em relação aos valores possíveis de conflito
  - Tem a desvantagem de depender da medida de conflito sendo utilizada

Na visualização da relação entre objetivos em Árvores de Agregação, há várias vantagens:

- É fácil de se ver diretamente o conflito entre os objetivos
- Objetivos são agrupados de acordo com sua harmonia
  - Se o tomador de decisão quiser criar objetivos compostos de acordo com sua harmonia (e conseqüentemente sua reducibilidade) os nós dão informação sobre quais seriam os melhores para se agrupar
- Em vez de apresentar apenas o conflito em cada par de objetivos, a árvore mostra também o conflito entre grupos de objetivos que podem ser agrupados antes de se mostrar seu conflito com soluções menos compatíveis
- Os nós em preto ou coloridos representam tanto sua globalidade do conflito entre aquelas duas soluções ou a região do espaço de objetivos onde estas características estão concentradas.
- Os nós em vermelho ou azul representam a posição do conflito

Mais uma aplicação útil da árvore é derivada da ideia de que quanto mais distantes dois nós folha primos são, menos harmonia há entre estes objetivos. Assim, a posição final dos nós folha dão uma maneira conveniente de se escolher bons objetivos adjacentes em coordenadas paralelas, assim como mostrado na Figura 4.3.

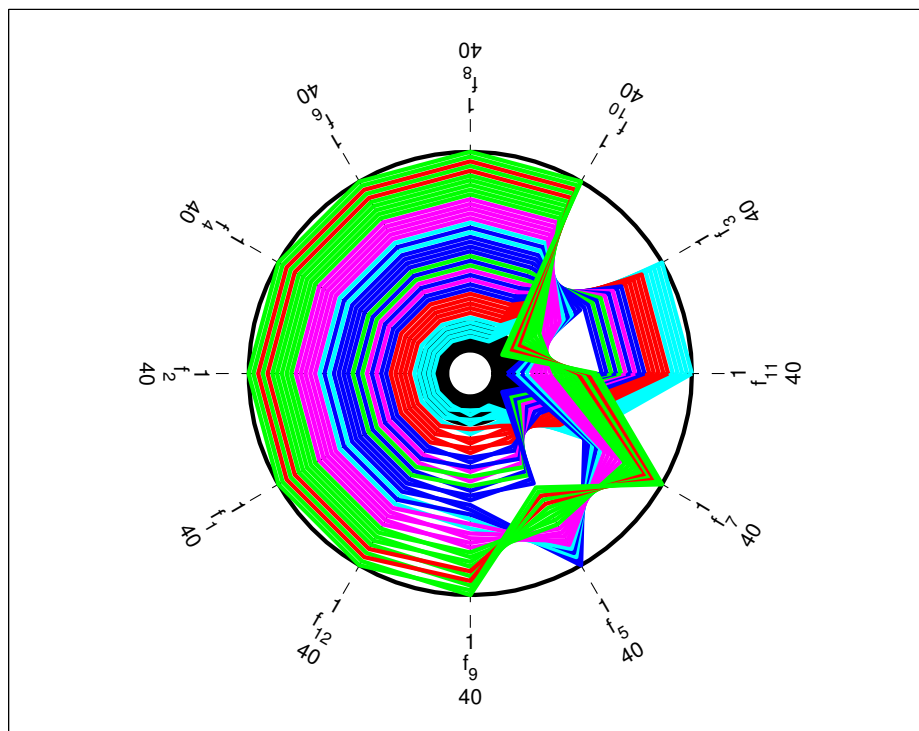


**Figura 4.3:** Coordenadas Paralelas com uma Escolha Diferente de Objetivos Adjacentes

Nesta Figura, é mais fácil identificar a relação entre os objetivos em termos de valores objetivos porque objetivos harmoniosos estão próximos, não rompendo a representação de objetivos conflitantes como acontece em cada par de objetivos na Figura 4.1.

Dada a utilidade desta ordenação de objetivos, propomos uma representação ainda melhor para os valores absolutos em um Gráfico Polar, como o da Figura 4.4.

Similarmente às coordenadas paralelas, cada linha do Gráfico Polar representa uma solução e a principal diferença entre eles é que o Gráfico Polar tem uma estrutura toroidal que permite a representação de objetivos extremos nas coordenadas paralelas, que de



**Figura 4.4:** Gráfico Polar. Assim como na representação em coordenadas paralelas, cada linha representa a qualidade de uma solução em cada um dos objetivos. Valores mais próximos do centro indicam valores de função objetivo mais baixos.

acordo com a árvore devem ser de interesse pois representam aqueles objetivos que são os primos mais distantes possíveis.

Além disto, o interior do círculo representa valores pequenos para cada um dos objetivos e a parte mais externa representa valores altos. Os valores são normalizados linearmente e a faixa de valores absolutos em cada objetivo é representada fora do círculo. Uma técnica de agrupamento, o PSA (*Part-and-Select Algorithm*) (Salomon, Avigad, Goldvard & Schütze 2013), é também aplicada aqui para facilitar a visualização dos resultados.

Assim, em nosso exemplo, a estrutura toroidal do Gráfico Polar torna possível ver a relação de conflito entre os objetivos 11 e 7. Também, à medida que primos próximos na árvore, representando objetivos em harmonia, são colocados juntos e primos distantes

**Tabela 4.1:** Conflito entre os objetivos nas soluções da Figura 4.1. Medidas de conflito e harmonia serão apresentadas nas próximas Seções.

$f$	1	2	3	4	5	6	7	8	9	10	11	12
1		0	1	0	.25	0	.25	0	.05	0	1	0
2	0		1	0	.25	0	.25	0	.05	0	1	0
3	1	1		1	1	1	1	1	1	1	.25	1
4	0	0	1		.25	0	.25	0	.05	0	1	0
5	.25	.25	1	.25		.25	.5	.25	.275	.25	1	.25
6	0	0	1	0	.25		.25	0	.05	0	1	0
7	.25	.25	1	.25	.5	.25		.25	.275	.25	1	.25
8	0	0	1	0	.25	0	.25		.05	0	1	0
9	.05	.05	1	.05	.275	.05	.275	.05		.05	1	.05
10	0	0	1	0	.25	0	.25	0	.05		1	0
11	1	1	.25	1	1	1	1	1	1	1		1
12	0	0	1	0	.25	0	.25	0	.05	0	1	

são colocados longe uns dos outros no gráfico, é possível visualizar o efeito de cada troca de um ramo da árvore para o outro. Por exemplo, torna-se fácil visualizar que soluções em verde que em geral têm valores altos nos objetivos harmônicos  $f_{10}$ ,  $f_8$ ,  $f_6$ ,  $f_4$ ,  $f_2$ ,  $f_1$ ,  $f_{12}$ ,  $f_1$ , and  $f_9$ . Pode-se perceber que estes valores altos nos objetivos harmoniosos são compensados por valores baixos em  $f_3$  e valores médios para  $f_{11}$  e  $f_5$ . Ainda, valores baixos nos objetivos harmoniosos em vermelho são compensados nos objetivos  $f_{11}$  and  $f_3$ .

Mais um uso interessante dos dados provenientes desta análise de conflito é o conflito direto de primeira ordem entre cada par de objetivos iniciais, como apresentado na Tabela 4.1. Nesta Tabela estão apresentados os valores de conflito de 0 a 1 entre cada par de objetivos. Todos estes valores são calculados com as medidas de conflito que serão apresentadas em seções subsequentes.

Em suma, a Árvore de Agregação é uma ferramenta para auxiliar tomadores de decisão a computar e visualizar redundância e conflito entre objetivos. Assim, sua aplicação principal é o agrupamento de objetivos, para redução de dimensionalidade do problema, e visualização das análises usando árvores e gráficos polares. O código fonte de todos estes algoritmos está disponível em <https://sourceforge.net/projects/harmonytrees/>. Na seção 4.5, descrevemos em detalhe o algoritmo para construção da

árvore.

## 4.2 Medidas de Conflito e Harmonia

Em termos simples, há conflito (Purshouse & Fleming 2003, Purshouse & Fleming 2007) entre dois objetivos quando valores bons em um deles implicam em valores ruins para o outro. Apesar de parecer simples, é um conceito que depende do conjunto de soluções possíveis, do formato da frente de Pareto e até de expectativas do tomador de decisão. A Figura 4.5 mostra vários tipos de conflito em coordenadas paralelas de dois objetivos. Cada linha representa uma solução enquanto o eixo  $x$  representa cada um dos objetivos e o eixo  $y$  representa os valores de função objetivo. Pode-se perceber dos gráficos que não apenas a quantidade de conflito varia entre os objetivos mas também que o conflito existente pode ser concentrado em certas regiões do espaço de objetivos.

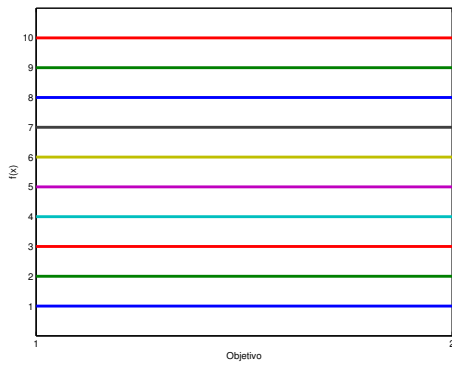
Harmonia (Purshouse & Fleming 2003, Purshouse & Fleming 2007, Brockhoff & Zitzler 2007, Giagkiozis, Purshouse & Fleming 2012), por outro lado, é um conceito que se aplica quando melhora no primeiro objetivo levaria a melhora no segundo objetivo. De acordo com estes autores, o conceito de harmonia está relacionado a reducibilidade e possibilidade de agregação de objetivos.

Objetivos harmoniosos são indicados em coordenadas paralelas por linhas que não se cruzam. Assim, é relacionado com a possibilidade de agregar objetivos através de soma sem perda de qualidade na frente de Pareto. Se vamos agrupar dois objetivos em um novo objetivo composto, é melhor agrupar os objetivos que tenham o máximo de harmonia mesmo se há conflito entre eles. A Figura 4.5(a) mostra um exemplo de harmonia completa.

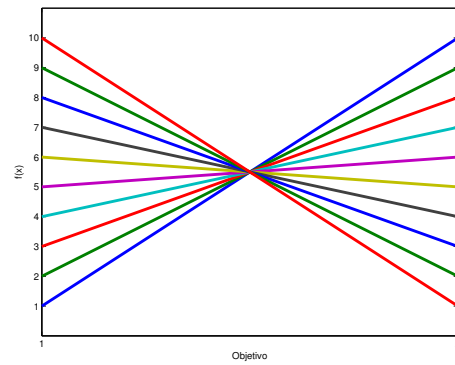
É comum que se há harmonia entre objetivos não há conflito. Contudo, harmonia e conflito não são antônimos perfeitos, como veremos nas medidas de harmonia e conflito nas próximas Seções.

Nesta tese, é proposta uma técnica de redução de objetivos não-paramétrica que considera o fato de que quanto mais harmoniosos dois objetivos são, melhores opções eles são para serem reduzidos ou agrupados em um novo valor único de função objetivo escalarizado.

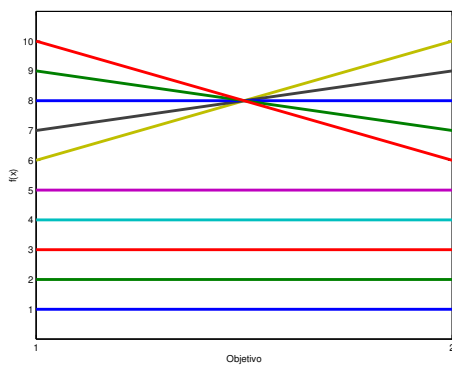
Aumentar o número de objetivos harmoniosos não tem efeito na convergência da superfície de Pareto. Para o desenvolvimento deste trabalho, desenvolveu-se medidas de conflito e harmonia que podem ser utilizadas pelas Árvores de Agregação (Freitas,



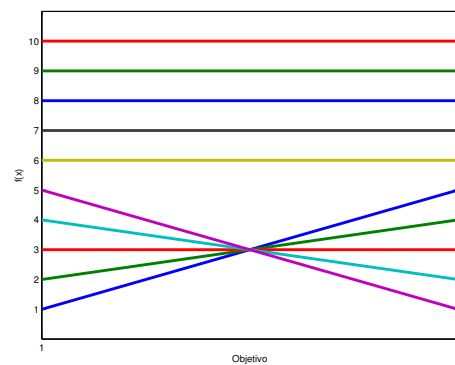
(a) Harmonia Total



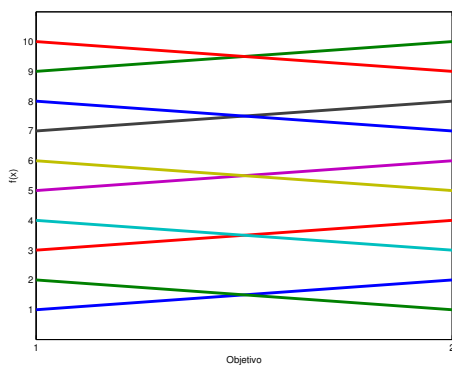
(b) Conflito Máximo Global



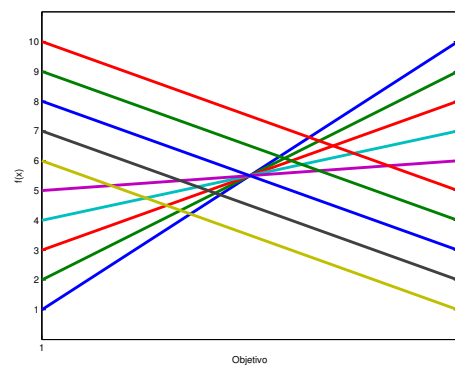
(c) Conflito Concentrado em Valores Altos



(d) Conflito Concentrado em Valores Baixos



(e) Conflito Médio Global



(f) Conflito Local Concentrado em Locais Diferentes em cada Objetivo

**Figura 4.5:** Exemplo de vários tipos possíveis de conflito

Guimarães & Fleming 2013). Cada medida tem características específicas que serão discutidas nas seções subsequentes.

### 4.2.1 Conflito Direto

A primeira medida de conflito é a diferença de ganho ou perda em termos absolutos para os objetivos. Esta medida pode ser particularmente interessante se ambos os objetivos estão nas mesmas unidades. Esta medida de conflito  $c$  é a diferença absoluta entre os valores para os objetivos. Se a faixa de valores é muito diferente para cada objetivo, os valores objetivo de  $\mathbf{X}$  podem ser normalizados para zero em um novo  $\mathbf{X}'$  formado pela subtração dos valores mínimos em  $\mathbf{X}$  para cada objetivo.

Assim, sendo  $\mathbf{X}_{ij}$  o valor para o objetivo  $j$  na solução  $i$ , o conflito direto  $C_{ab}$  entre os objetivos  $a$  e  $b$  é:

$$C_{ab} = \sum_i |\mathbf{X}'_{ia} - \mathbf{X}'_{ib}| \quad (4.1)$$

$$\mathbf{X}'_{ij} = \mathbf{X}_{ij} - \min(\mathbf{X}_{.j})$$

A aplicabilidade desta medida de conflito é fortemente dependente do tipo de problema analisado. Com esta normalização para cálculo de conflito, a medida de conflito direto é:

- Insensitiva a somas ou subtrações nos valores objetivo originais
- Útil quando os objetivos estão nas mesmas unidades
- A soma dos objetivos poderia igualmente resolver o problema porque os objetivos são igualmente importantes
- Útil quando o tomador de decisão quer entender os compromissos envolvidos

A faixa de valores possíveis de conflito vai de  $c_{\min} = 0$  até um valor  $c_{\max}$  dependente do problema. Isto ocorre porque os valores absolutos são diretamente comparados. O valor  $c_{\max}$  é então relacionado à faixa original de valores nos objetivos.

A medida de conflito responde à seguinte questão:

- Dado que os ganhos em todos os objetivos em termos absolutos (e não relativos em termos de faixas de valores conhecidas) são igualmente importantes, qual o conflito existente entre estes objetivos?

Como otimização multiobjetivo é usada para tratar objetivos que tenham unidades de medida incomparáveis no caso geral, pode parecer não fazer sentido comparar diretamente os valores objetivo. Contudo, isto pode ser útil quando os objetivos têm as mesmas unidades.

Por exemplo, pode haver decisões em uma companhia que podem ser monetarizadas. Os objetivos provavelmente poderia ser agrupados com um somatório já que a companhia quer auferir o maior lucro possível independentemente de qual objetivo fornece o lucro. Porém, o tomador de decisão pode querer analisar os resultados de um algoritmo de otimização multiobjetivo para que possa entender os compromissos envolvidos na decisão.

#### 4.2.2 Conflito Maxmin

Esta segunda medida de conflito também assume comparabilidade entre os objetivos. Ela é útil quando temos qualquer referência para implicar que todos os objetivos são igualmente importantes, onde a comparabilidade e importância de cada objetivo é proporcional à faixa conhecida de valores para um deles. Isto implica que os objetivos não são completamente incomparáveis mas são igualmente importantes em relação à faixa conhecida para seus valores. Isto quer dizer que quanto maior a faixa de valores conhecida para um objetivo, menos importante se torna uma diferença constante entre dois valores de função objetivo. A formulação matemática deste conflito maxmin é:

$$C_{ab} = \sum_i |\mathbf{X}'_{ia} - \mathbf{X}'_{ib}|$$

$$\mathbf{X}'_{ij} = \frac{\mathbf{X}_{ij} - \min(\mathbf{X}_{.j})}{\max(\mathbf{X}_{.j}) - \min(\mathbf{X}_{.j})} \quad (4.2)$$

Esta medida é normalizada e nos deixa com valores entre 0 e 1 para todos os valores objetivo. A medida maxmin é:

- Insensitiva a qualquer normalização linear anterior
- Útil quando todos os objetivos são igualmente importantes

- Implica-se que a importância de cada objetivo é inversamente proporcional ao tamanho da faixa de valores conhecidos para ele na área de preferência

Sendo  $n$  o número de soluções analisadas, a faixa de valores possíveis de conflito entre dois objetivos vai de  $c_{\min} = 0$  até  $c_{\max}$ , onde:

$$c_{\max} = \sum_{i=1}^n 1 \quad (4.3)$$

$$c_{\max} = n$$

O conflito maxmin responde a seguinte questão:

- Dado que todos os objetivos são igualmente importantes em proporção à sua faixa de valores conhecidos, qual é o conflito existente entre estes objetivos?

Com esta medida de conflito, temos mais perda de informação devido à falta de comparabilidade direta na mesmas unidades. Por outro lado, é uma medida mais robusta porque há menos suposições, tal como a própria possibilidade de ter diferentes unidades de medida.

Com uma adaptação pequena, a normalização pode também incluir preferências do tomador de decisão. Dado que  $\mathbf{P}_j$  é a meta do tomador de decisão para cada objetivo  $j$  que define a área de preferência, uma simples adaptação desta medida de conflito para problemas de minimização pode ser:

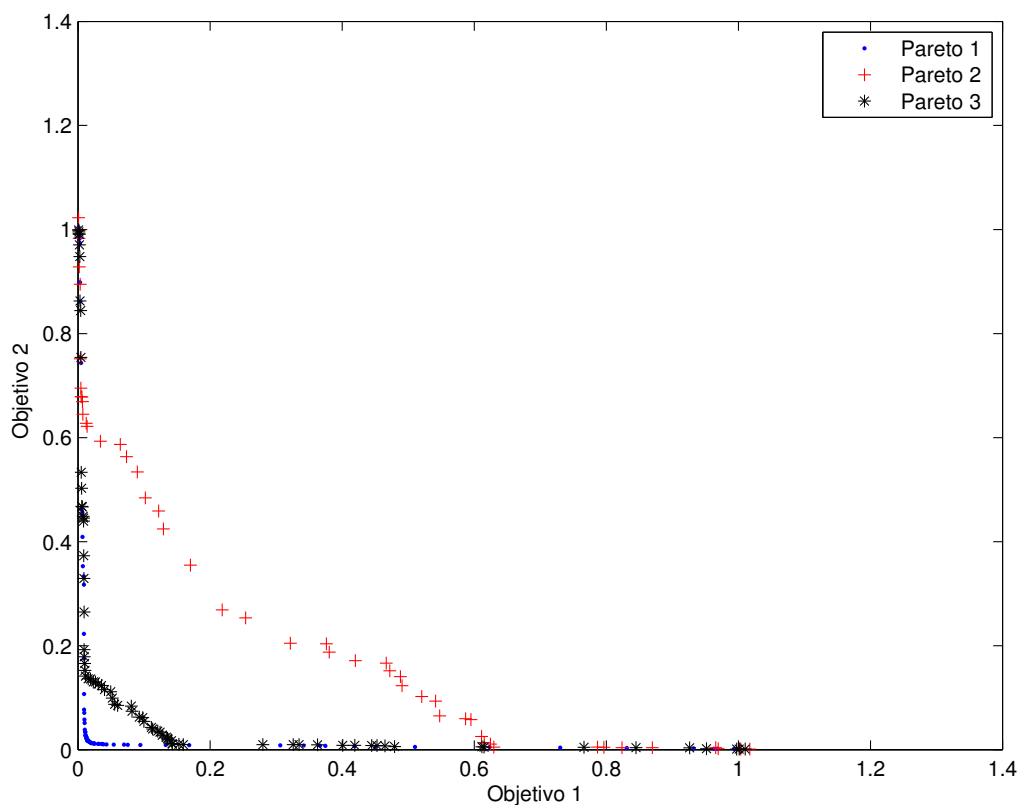
$$C_{ab} = \sum_i |\mathbf{X}'_{ia} - \mathbf{X}'_{ib}| \quad (4.4)$$

$$\mathbf{X}'_{ij} = \frac{\mathbf{X}_{ij} - \min(\mathbf{X}_j)}{\mathbf{P}_j - \min(\mathbf{X}_j)}$$

Neste caso, a importância dos objetivos definida pela faixa de valores conhecidos também considera as expectativas do tomador de decisão.

### 4.2.3 Conflito Exponencial ou Logarítmico

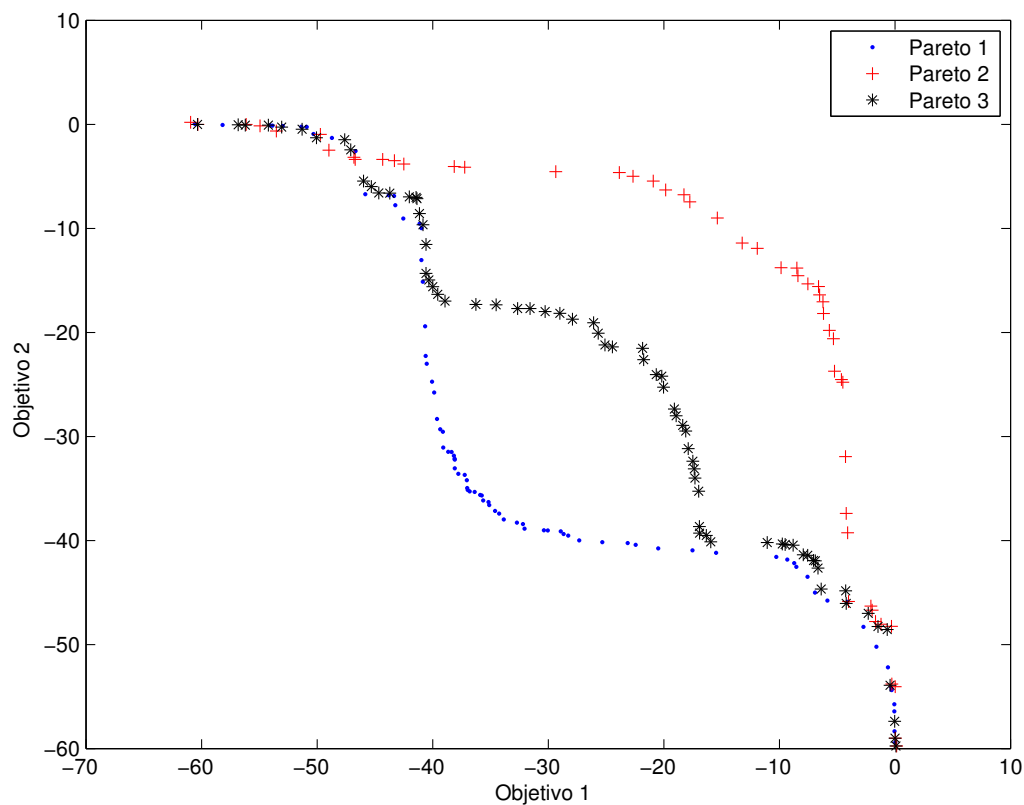
Pode haver casos nos quais não faz sentido analisar resultados não normalizados e a interpretação direta dos resultados podem iludir. Por exemplo, no caso de limites de sensibilidade Pareto-ótimos para sistemas de controle de temperatura de Eitelberg (2006), temos 3 conjuntos de soluções para 3 sistemas diferentes na Figura 4.6. Destes resultados podemos ver que quase não há conflito para o primeiro conjunto e há um pouco mais de conflito para o terceiro conjunto.



**Figura 4.6:** Frentes de Pareto para um Problema de Controle de Temperatura

Contudo, uma vez que os resultados são obtidos, não faz sentido interpretar os resultados diretamente neste contexto específico. Para melhor visualização, uma escala logarítmica é utilizada. Ao se normalizar os resultados de  $\mathbf{X}_{ij}$  com  $\mathbf{X}_{ij} \leftarrow 20 \log_{10} \mathbf{X}_{ij}$ , obtemos a representação na Figura 4.7.

Nesta segunda representação das soluções, pode-se interpretar que há conflito em todas as frentes de Pareto. Contudo, a percepção de conflito deriva apenas da repre-



**Figura 4.7:** Frentes de Pareto em Escala Logarítmica para um Problema de Controle de Temperatura

sentação cognitiva das mesmas soluções. Isto mostra como a percepção de conflito pode depender das preferências do tomador de decisão.

Para estes tipos de problemas nos quais técnicas de normalização mais avançadas são requeridas, apenas a interferência do tomador de decisão pode levar a uma medida apropriada de conflito porque quando as medidas de conflito minmax ou direto são aplicadas logo depois, isto envolveria comparabilidade entre os objetivos.

Além disto, não seria apropriado estender mais ainda em direção a medidas de normalização de conflito que sejam logarítmicas ou exponenciais no contexto de otimização, como fizemos com somatórios e multiplicação nas últimas subseções. A razão para isto é que, em geral, estes tipos de transformações podem fazer com que valores inicialmente crescentes se tornem decrescentes após a normalização. Em outras palavras, no caso geral, soluções dominadas podem até mesmo se tornar não-dominadas após uma

normalização deste tipo.

Assim, normalizações posteriores poderiam alterar a relação original de harmonia entre os objetivos. Chamaremos estas de *normalizações disruptivas*. Contudo, estas normalizações disruptivas mostram a necessidade de uma medida mais robusta que reconheceria conflito em todas as frentes da Figura 4.6, independentemente de interpretações cognitivas complexas dos resultados.

#### 4.2.4 Conflito Não-Paramétrico de Ranqueamento

No exemplo das Figuras 4.6 e 4.7, técnicas diferentes de normalização teriam resultados diferentes mas uma medida não-paramétrica de conflito poderia ser insensitiva a estas percepções do tomador de decisão. Esta robustez, contudo, é obtida ao custo de se perder informação em um processo de ranqueamento, o que pode precisar ser compensado com um aumento da granularidade do conjunto de soluções.

Uma medida não-paramétrica de conflito funciona sem a suposição de comparabilidade entre os objetivos. Ela é útil quando não se tem qualquer referência para implicar uma relação de importância entre os objetivos e temos a meta de entender a relação entre objetivos incomparáveis sem erroneamente implicar ou declarar que eles são igualmente importantes. As diferenças de posição de ranking nos objetivos são usadas para comparar os resultados sem se considerar a distância entre estes valores. Assim, a formulação matemática do conflito não-paramétrico é:

$$C_{ab} = \sum_i |\mathbf{X}'_{ia} - \mathbf{X}'_{ib}|$$

$$\mathbf{X}'_{ij} = \mathbf{R}_{ij} \tag{4.5}$$

$$\mathbf{R}_{ij} = \text{posicionamento de } \mathbf{X}_{ij} \text{ em } \mathbf{X}_{.j}$$

Sendo  $n$  o número de soluções sendo analisadas, esta normalização nos deixa com valores de 0 a  $n$  para cada valor objetivo e a medida de conflito não paramétrica é:

- Insensitiva a qualquer normalização não disruptiva
- Útil quando os objetivos usam unidades diferentes e não são comparáveis de qualquer forma
- Um valor da importância de cada objetivo não pode ser inferido mas queremos

entender a relação entre objetivos

A faixa de valores possíveis de conflito entre dois objetivos vai de  $c_{\min} = 0$  até  $c_{\max}$ :

$$c_{\max} = \sum_{i=1}^n |2i - n - 1| \quad (4.6)$$

$$c_{\max} = 2 \left( \lceil n/2 \rceil (n + 1) - \frac{2(1 + \lceil n/2 \rceil) \lceil n/2 \rceil}{2} \right)$$

O conflito não-paramétrico de ranqueamento responde a seguinte questão:

- Independentemente de qualquer normalização não-disruptiva anterior, um aumento de valor em um objetivo implicaria em perda no segundo objetivo dentro da área de preferência?

De acordo com a definição dada na Seção 4.2 e a questão relacionada a esta medida de conflito, pode-se perceber que o conflito de ranqueamento pode também ser utilizado como uma medida de harmonia, que é a relação correspondente de aumento ou diminuição de modo geral para os objetivos, independentemente do formato da frente de Pareto. Dado o valor  $\frac{C_{ab}}{c_{\max}}$  de conflito não-paramétrico normalizado entre 0 e 1 entre os objetivos  $a$  e  $b$ , temos que a harmonia normalizada  $H_{ab}$  entre estes dois objetivos é:

$$H_{ab} = 1 - C_{ab} \quad (4.7)$$

A normalização não-paramétrica de ranking é mais robusta pois é mais insensitiva a qualquer normalização anterior em geral. Como em qualquer medida não-paramétrica, isto envolve perda de informação como o custo de se ter menos suposições. A perda de informação advém da falta de comparabilidade direta entre os objetivos, que parece apropriada para o caso geral em otimização multiobjetivo. Por outro lado, é uma medida mais robusta pois há menos suposições, tais como a possibilidade de haver unidades diferentes de medida sem qualquer conversão entre elas.

Junto às medidas de conflito direto e maxmin, este é um dos três tipos gerais e relevantes de normalizações considerados neste trabalho. Contudo, apenas a normalização

de rankeamento é robusta o suficiente para também medir harmonia e para trabalhar com qualquer tipo de dado em geral. Desta maneira, com esta medida não-paramétrica, conflito e harmonia são complementares.

### 4.3 Algoritmo de Redução de Objetivos

Como mencionado, quanto mais harmoniosos são dois objetivos, mais eles podem ser reduzidos ou agrupados em um novo valor objetivo escalonado sem deformação do formato da frente de Pareto. Por esta razão, para cada grupo de objetivos, achamos os dois objetivos mais harmoniosos e os agrupamos como um novo objetivo escalonado de soma ponderada. Assim, o tomador de decisão ou o usuário pode fazer a decisão sobre o número de objetivos que desejam considerar com o Algoritmo 4.1.

---

#### Algoritmo 4.1: Redução de Objetivos

---

**Entrada:** Matriz  $\mathbf{X}_{n \times m}$  com conjunto de valores de função objetivo para  $m$  objetivos e  $n$  soluções, Número  $k$  de objetivos a serem reduzidos

**Resultado:** Valores de função objetivo  $\mathbf{X}$  com  $k$  objetivos a menos

- 1 **para** cada um dos  $m$  objetivos **faça**
  - 2     |   Rankear os valores neste objetivo;
  - 3 **fim**
  - 4 **enquanto** o usuário ainda quer reduzir mais um dos  $k$  objetivos **faça**
  - 5     |   Calcular  $C_{i,j}$  para cada par de objetivos;
  - 6     |   Achar o menor valor de  $C_{i,j}$ ;
  - 7     |   Reduzir os objetivos com conflito não-paramétrico mínimo  $C_{i,j}$  (ou harmonia máxima  $H_{i,j}$ );
  - 8 **fim**
  - 9 **retorna** Novo conjunto reduzido de objetivos  $\mathbf{X}$ ;
- 

Rankear os valores em um objetivo tem custo  $O(n \log n)$  ou  $O(n)$ , dependendo do algoritmo de ordenação utilizado. Calcular  $C_{i,j}$  para cada par de objetivos tem custo  $O(nm^2)$ . Achar o menor valor de  $C_{i,j}$  tem custo  $O(m^2)$ . Achar o menor valor de  $C_{i,j}$  tem custo  $O(m^2)$ . Reduzir os objetivos com harmonia máxima tem custo  $O(n)$ .

Rankear os valores de um objetivo tem o mesmo custo de se ordenar um conjunto de elementos, ou seja:

- $O(n \log n)$  para algoritmos baseados em comparação

- $O(n)$  para algoritmos baseados em contagem, como o Radix sort

Assim, o primeiro laço do algoritmo custa  $O(mn \log n)$  ou  $O(mn)$ .

Cada iteração do segundo laço custa  $O(\max(nm^2, m^2, n)) = O(nm^2)$ . Como temos  $k$  reduções, o custo total do segundo laço é  $O(knm^2)$ . Assim, a complexidade total do algoritmo é  $O(\max(mn \log n, knm^2))$  para algoritmos de ordenação baseados em comparação ou  $O(\max(mn, knm^2)) = O(knm^2)$  para algoritmos de ordenação baseados em contagem.

Para propósitos práticos, temos os seguintes casos:

- Se considerarmos que  $k$  e  $n$  são constantes, o custo do algoritmo se torna  $O(m^2)$  para qualquer caso
- Se queremos reduzir todos os objetivos em apenas um objetivo, então  $k = m$  e o algoritmo é  $O(nm^3)$  ou  $O(\max(mn \log n, nm^3))$
- Se queremos reduzir todos os objetivos em apenas um objetivo e considerarmos que  $n$  é constante, então  $k = m$  e o algoritmo é sempre  $O(m^3)$
- Para  $k = 1$ , com apenas uma redução por vez, a ferramenta pode ser eficientemente embutida em um MOEA com custo  $O(nm^2)$
- Para  $k = 1$ , com apenas uma redução por vez e considerando  $n$  constante, a ferramenta pode ser eficientemente embutida em um MOEA com custo  $O(m^2)$

Como exemplo, consideramos o conjunto de soluções representado em coordenadas paralelas na Figura 4.1. Cada par de objetivos neste conjunto de soluções contém todos os seis tipos de conflito apresentados na Figura 4.5 com uma granularidade de 40 pontos.

Neste exemplo, o algoritmo sugere então a redução de todos os objetivos  $f_1, f_2, f_4, f_6, f_8, f_{10}, f_{12}$  no primeiro passo. Todos estes objetivos estão em harmonia completa porque eles tem os mesmos valores de função objetivo. Normalizando todos estes valores em posições de ranking novamente e somando seus valores em um novo objetivo composto  $f_a$ , temos agora 6 objetivos restantes.

Aplicando mais uma redução,  $f_a$  é agrupado com o objetivo  $f_9$  com 5% de conflito não-paramétrico (ou 95% de harmonia).

Definindo que  $f_b = f_9 + f_a$ , dos 5 objetivos restantes, o algoritmo de redução agrupa então nos próximos passos:

1.  $f_c = f_{11} + f_3$  com 25% de conflito
2.  $f_d = f_b + f_c$  com 0% de conflito, resultando em um problema com 3 objetivos
3.  $f_e = f_d + f_5$  com 26% de conflito
4.  $f_f = f_e + f_7$ , finalmente, com 25% de conflito

O objetivo  $f_f$  representa a agregação de todo o problema como um problema mono-objetivo na forma de uma soma ponderada.

## 4.4 Localidade de Conflito

Já que o conflito não-paramétrico é medido como uma soma de diferenças de posições ranking, a posição destas diferenças é também geralmente uma informação relevante. É importante para o tomador de decisão saber se o conflito ocorre para os piores ou melhores valores dos objetivos sendo comparados. Esta informação é importante porque se o conflito ocorre para soluções com valores objetivo ruins, isto significa que não há conflito inerente entre os objetivos sendo analisados em si pois as soluções com valores objetivo ruins para os dois objetivos estão apenas na frente de Pareto por causa de outros objetivos.

Por exemplo, a Figura 4.5 apresenta vários exemplos de compromissos entre dois objetivos em coordenadas paralelas. Todos os resultados são mostrados em termos de ranks. Na Figura 4.5(a), uma melhora no objetivo 1 sempre leva a uma melhor no objetivo 2. Neste caso não há conflito entre os objetivos e este comportamento é global. Este é o melhor caso para fazer um novo objetivo composto com a simples simetria dos objetivos.

Se há pouco conflito e muita harmonia, a maior parte do comportamento ainda será global pois a harmonia é global. Na Figura 4.5(b), o comportamento é também global pois há conflito em todos os locais. Este é um caso extremo de conflito onde melhora em um objetivo necessariamente causa piora em outro.

As Figuras 4.5(c) e 4.5(d), por outro lado, são exemplos nos quais o conflito é apenas local. No primeiro caso, da Figura 4.5(c), o conflito apenas acontece para soluções que

são ruins em relação a estes objetivos, considerando-se um problema de minimização. Nestes casos, a remoção de outros objetivos eventualmente causaria um desaparecimento do compromisso, pois apenas para os objetivos sendo analisados, apenas uma solução estaria no frente de Pareto.

Na Figura 4.5(d), contudo, a eliminação de outros objetivos não poderia levar a menos conflito entre os objetivos sendo considerados pois 5 soluções sempre estarão na primeira frente de Pareto pois o conflito entre estas soluções nestes objetivos é inerente. Isto mostra a importância de se representar não apenas a globalidade do conflito mas também sua posição.

Ter menos que conflito máximo não quer dizer necessariamente que conflito seja local. O exemplo na Figura 4.5(e) mostra um caso de conflito global médio.

Outro caso especial é o representado na Figura 4.5(f), na qual há conflito local mas a posição do conflito é diferente para os objetivos. Neste caso, haveria uma frente de Pareto mesmo se apenas estes dois objetivos fossem considerados pois o conflito é localizado em torno de boas soluções para o primeiro objetivo. Neste caso, o conflito é localizado em bons valores para o objetivo 1 e valores ruins para o objetivo 2.

Já que harmonia é medida em termos de diferenças acumuladas de ranks nas soluções, é importante então ver de onde estas diferenças vêm e se estas diferenças estão localizadas em torno de soluções boas ou ruins. A Figura 4.8 mostra a diferença de valores para os mesmos exemplos da Figura 4.5. O eixo  $x$  tem o rank de cada solução no objetivo 1 enquanto o eixo  $y$  tem a diferença absoluta para a solução correspondente no objetivo 2. O gráfico mostra cada valor de diferença e a área destes valores é o conflito entre os objetivos, como descrito anteriormente na Equação 4.7. Poderíamos fazer também este gráfico para o objetivo 2 e teríamos resultados diferentes por causa de situações como a da Figura 4.5(f), onde podemos ter o mesmo conflito localizado em regiões diferentes para objetivos diferentes.

No primeiro caso da Figura 4.8(a), não há diferença alguma entre todas as soluções pois não há conflito. No caso oposto, de conflito máximo, a Figura 4.8(b) mostra como a globalidade de conflito leva a uma diferença de valores que é simétrica no gráfico. Também pode-se perceber que no caso de conflito máximo, não ocorre valores altos de diferença em todas as posições pois soluções medianas têm menos conflito.

As Figuras 4.8(c) e 4.8(d) mostram casos de conflito que são especificamente locais. Nestes casos, as diferenças são respectivamente concentradas nos piores e melhores

valores objetivo. O formato dos gráficos de valores de diferenças neste caso reflete simetricamente em uma escala menor as diferenças da Figura 4.8(b) pois mesmo sendo o conflito local, ele tem localmente o mesmo comportamento do conflito máximo na área onde ocorre. Estes dois gráficos dão uma ideia de como a distribuição de diferenças é concentrada em algumas áreas se o conflito for local.

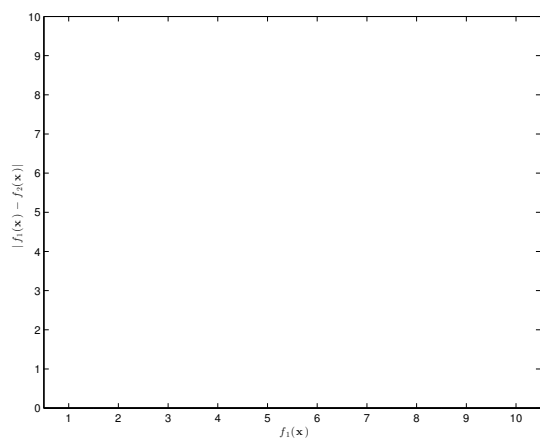
A Figura 4.8(e) é um caso de conflito global mais uma vez. Similarmente aos dois primeiros casos, a distribuição das diferenças está por todos os lados. Contudo, a área é menor do que na Figura 4.8(b) pois há menos conflito.

O último caso, da Figura 4.8(f), é o conflito localizado em diferentes posições para dois objetivos. Em relação ao objetivo 1, o conflito é para soluções de baixo rank e os valores de diferença formam áreas de valores similares pois mesmo que as linhas para os valores altos de rank estejam em paralelo, há uma grande diferença nos ranks para o grupo e, assim, conflito para todo o grupo. Contudo, o que indica a localidade de conflito no gráfico é que a área está mais concentrada nos valores de rank baixos para as primeiras soluções e que a área para as soluções de rank alto sejam bem distribuídas, indicando que quanto mais baixo o rank de uma solução, mais conflito há.

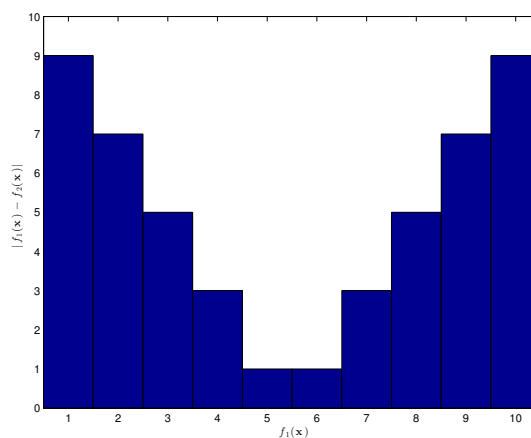
A partir de gráficos como os apresentados, temos então que definir se os valores de conflito estão vindo de valores baixos ou altos. Para fazer isto, os valores de rank  $f_1(\mathbf{x}_i)$  para as soluções  $\mathbf{x}_i$  no objetivo  $f_1$  são linearmente normalizados de -1 a 1 em uma nova variável  $f_1(\mathbf{x}'_i)$ . Os novos valores normalizados agora representam o peso desta solução para soluções de rank baixo (-1) ou alto (+1) e a seguinte medida de localidade de conflito  $L(a, b)$  do objetivo  $a$  em relação ao objetivo  $b$  é utilizada para as Árvores de Agregação.

$$L(a, b) = \frac{\sum_{i=1}^n |f_a(\mathbf{x}_i) - f_b(\mathbf{x}_i)| f_a(\mathbf{x}'_i)}{L(a, b)_{\max}} \quad (4.8)$$

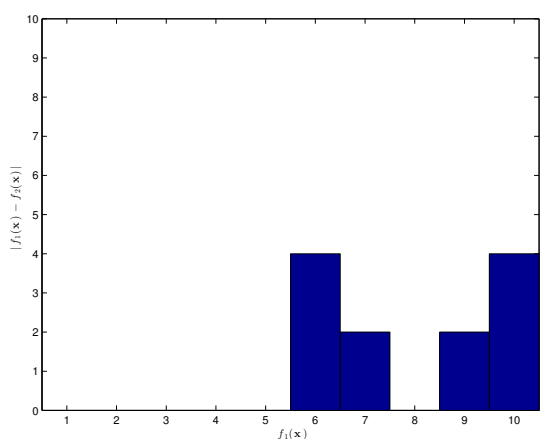
As diferenças de rank são ponderadas por sua localidade e são somadas. Finalmente, os valores são normalizados para a faixa entre -1 e 1 novamente com a divisão por  $L(a, b)_{\max}$ . A normalização é feita com o valor máximo de  $\sum_{i=1}^n |f_a(\mathbf{x}_i) - f_b(\mathbf{x}_i)| f_a(\mathbf{x}'_i)$ , que é o que ocorre com conflito máximo global apenas para metade das soluções.



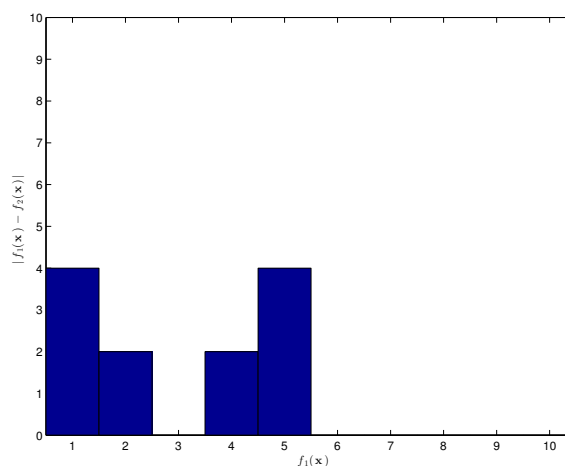
(a) Conflito Global Mínimo



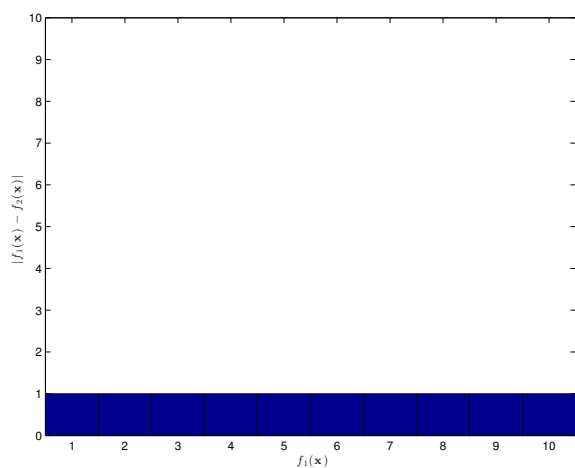
(b) Conflito Global Máximo



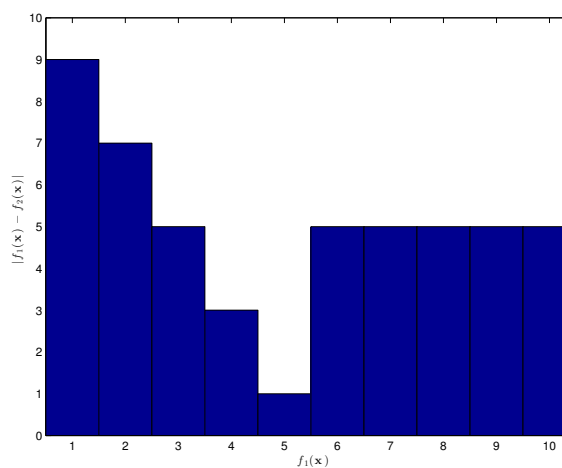
(c) Conflito Local para Valores Altos



(d) Conflito Local para Valores Baixos



(e) Conflito Global Médio



(f) Conflito Local em Posições Diferentes

Figura 4.8: Origem dos valores de diferença na Figura 4.5

$$L(a, b)_{\max} = \sum_{i=\lfloor n/2 \rfloor + 1}^n |2i - \lfloor 3n/2 \rfloor - 1| f_a(\mathbf{x}'_i) \quad (4.9)$$

Valores negativos indicam desarmonia mais concentrada em soluções de rank baixo enquanto valores positivos indicam desarmonia concentrada em torno de valores de rank alto. Esta informação é importante para o tomador de decisão pois se o conflito é concentrado nas melhores soluções, isto significa que há conflito inerente entre os objetivos mesmo se outros objetivos não estão presentes. Este caso é representado em vermelho na Árvore de Agregação. Quando o conflito é mais concentrado nas soluções ruins, o nó da árvore é representado em azul. A intensidade destas cores é definida de acordo com o valor  $0 < |L(a, b)| < 1$ . Se não há concentração de localidade, o nó é representado então em preto.

## 4.5 Construindo uma Árvore de Agregação

O processo para construir uma Árvore de Agregação envolve medidas de conflito e harmonia entre objetivos assim como localidade de conflito. Estas medidas foram discutidas em seções anteriores. A estrutura do algoritmo para gerar a árvore é descrita no Algoritmo 4.2.

Na linha 1, a estrutura da árvore é inicializada com um nó raiz como pai de todos os objetivos. Na linha 2, todos os valores dos objetivos são normalizados como demandado por uma das medidas de conflito descritas na Seção 4.2. A partir da linha 3, um laço iterativo começa. A cada iteração deste laço, os dois objetivos (entre originais ou compostos) mais harmoniosos que são filhos do nó raiz são agrupados em um novo nó pai da árvore, que se torna filho da raiz. Na linha 4, uma nova versão dos valores dos objetivos é criada para a iteração do laço. Esta nova versão  $X'$  já inclui o somatório dos objetivos agrupados até o momento. Na linha 5, esta nova versão é normalizada mais uma vez para uma comparação justa dos objetivos. A normalização é feita de acordo com a medida de conflito desejada que, no caso de conflito não-paramétrico, corresponde a trocar o valor por sua posição de *rank*.

Nas linhas 6 e 7, o par de objetivos com mais harmonia é calculado de acordo com a medida de harmonia apresentada na Seção 4.2.4. Nas linhas 8 e 9, o conflito e a localidade do conflito são calculados para o par de objetivos mais harmonioso. Na linha

10, um novo nó é incluído na árvore como um filho do nó raiz.

---

**Algoritmo 4.2:** Construção de uma Árvore de Agregação
 

---

**Entrada:** Matriz  $\mathbf{X}_{n \times m}$  com conjunto de valores de função objetivo para  $m$  objetivos e  $n$  soluções

**Resultado:** Árvore  $t$

- 1 Inicializar a árvore  $t$  com um nó raiz e todos os objetivos como filhos;
  - 2  $\mathbf{X} \leftarrow \text{normalizar}(\mathbf{X})$ ;
  - 3 **enquanto** *ainda há objetivos a serem agrupados* **faça**
  - 4      $\mathbf{X}' \leftarrow \text{reduzir}(\mathbf{X})$ ;
  - 5      $\mathbf{X}' \leftarrow \text{normalizar}(\mathbf{X}')$ ;
  - 6      $\mathbf{H} \leftarrow \text{matriz\_de\_harmonia}(\mathbf{X}')$ ;
  - 7      $a, b \leftarrow$  nós folha ou nós com objetivos compostos de  $\mathbf{X}'$  com maior harmonia;
  - 8      $c \leftarrow \text{conflito}(\mathbf{X}', a, b)$ ;
  - 9      $L \leftarrow \text{localidade}(\mathbf{X}', a, b)$ ;
  - 10     $t$  recebe um novo nó  $nn$ ;
  - 11     $nn$  recebe  $a$  e  $b$  como filhos;
  - 12     $nn$  guarda os valores  $(c, L)$ ;
  - 13     $a$  and  $b$  são agrupados. (A próxima geração tem um objetivo a menos);
  - 14 **fim**
  - 15 Imprimir a Árvore de Agregação  $t$ ;
  - 16  $order \leftarrow$  nós folha de  $t$  na ordem como aparecem em  $t$ ;
  - 17 Imprimir o Gráfico Polar considerando  $order$ ;
- 

Na linha 11, este novo nó recebe os nós que estavam representando os objetivos mais harmoniosos até o momento. Na linha 12, os valores de conflito e localidade de conflito para os objetivos mais harmônicos são guardados também neste novo nó. Neste ponto, o nó raiz tem um objetivo a menos e uma nova iteração do algoritmo se inicia.

A ideia por trás do algoritmo é que a cada iteração os dois objetivos mais harmônicos são agrupados em um novo objetivo composto até que haja apenas um objetivo composto, que representaria um somatório simples de todos os objetivos para otimização mono-objetivo. Durante o processo de se construir a árvore, precisamos de matrizes simples com informação sobre o conflito entre cada par de objetivos. Após o processo iterativo, nas linhas 15–17, apresentamos a Árvore de Agregação resultado e seu Gráfico Polar correspondente.

Em relação a custo computacional, este método tem o mesmo comportamento assintótico do método de redução de objetivos quando  $k = m$ .

## 4.6 Conclusão

Neste Capítulo discutimos a abordagem de Árvores de Agregação para problemas de otimização com muitos objetivos. As árvores são baseadas em uma medida não-paramétrica de harmonia para agrupar objetivos. As agregações mais convenientes são representadas graficamente em uma árvore, o que provê informação sobre a relação entre os objetivos.

Vantagens do método são (i) a facilidade de se visualizar a relação entre objetivos, (ii) a não dependência de nenhuma relação de linearidade entre objetivos e (iii) custo computacional baixo.

Na próxima Seção é apresentado um problema de otimização com muitos objetivos que é utilizado para experimentos com a Árvore de Agregação.

# Capítulo 5

## Avaliação de Melodias

*“Na vida de um indivíduo, uma sensibilidade estética é tanto mais autêntica quanto mais louvável do que uma política ou religiosa.”*

— Tom Robbins, 1936—

### 5.1 Introdução

Neste capítulo, analisamos ideias para a avaliação automática de características usuais em composições. Não só os métodos evolutivos, mas métodos de composição algorítmica em geral, usualmente dependem de métodos secundários de avaliação para definir probabilidades de aplicação de operadores de mudança, sejam eles critérios para geração e extensão de material melódico ou operadores de mutação.

Contudo, a avaliação de material envolve codificação de critérios que são estéticos, o que é um processo muito complexo se quisermos delinear procedimentos automáticos que são capazes de computar a adequação de melodias em um determinado contexto.

Assim, apresentamos neste capítulo uma pesquisa de numerosas ideias para examinar e avaliar melodias, sendo algumas destas ideias baseadas em teoria musical. Estas ideias têm sido utilizadas em análise musical porém usualmente negligenciadas em procedimentos de composição algorítmica.

A avaliação de melodias em música evolucionária é um problema aberto que foi enfrentado por vários autores com avaliação interativa, algoritmos livres de avaliação e até mesmo redes neurais. Porém, todas as abordagens baseadas em uma análise formal

de um corpo de dados ou teoria musical foram apenas parciais, o que é algo de se esperar de um problema complexo como este.

Neste contexto, descreve-se aqui uma lista de medidas divididas em várias categorias. Em paralelo a estas medidas de pesquisa de musicologia, mostram-se também resultados destas medidas em um banco de dados. Discutem-se também cada uma das medidas como métodos de análise musical.

Mesmo havendo ainda muita pesquisa a fazer nesta área, pretende-se auxiliar aqui compositores a definirem métodos de avaliação musical mais sofisticados e úteis. Assim, apresentam-se aqui várias medidas que podem ser utilizadas na avaliação de melodias e seus resultados práticos aplicados a um banco de dados criado com melodias de Bossa Nova.

Primeiramente, é analisado o problema da avaliação de melodias e em seguida, soluções nas variadas categorias são apresentadas. Apesar de que tem-se a intenção de estender as ideias possíveis para compositores evolucionários, haverá ainda muita possibilidade de pesquisa a ser desenvolvida no campo e cada gênero de música sempre precisará de medidas de qualidade específicas.

## 5.2 Avaliação Melódica

A codificação de critérios estéticos é uma tarefa complexa e o maior problema em composição evolucionária (Galanter 2010). Abordagens comuns para atacar o problema da codificação estética são usualmente:

- Avaliação interativa (McCormack 2005b, Todd & Werner 1999, Biles 1994)
- Algoritmos genéticos livres de aptidão (Freitas & Guimarães 2011b)
- Redes neurais (Biles, Anderson & Loggi 1996, Phon-Amnuaisuk, Law & Kuan 2007)

Cada uma destas abordagens tem várias desvantagens.

Como a avaliação de música inclui o problema da codificação de critérios estéticos, a definição de métodos automáticos para o cálculo de características melódicas é por consequência uma tarefa complexa. Por esta razão, a avaliação de melodias é um dos maiores problemas de métodos de composição algorítmica baseados em geração e teste

porque não é claro como julgamentos estéticos podem ser expressados por uma função de avaliação (Galanter 2010).

Problemas sérios com seleção estética automática e interativa em computação evolucionária (McCormack 2005b) são que:

1. O tamanho da população é limitado à habilidade do usuário realizar comparações que requerem tempo, atenção e boa referência
2. Comparação interativa é um processo lento e subjetivo onde pode levar várias horas para a avaliação de poucas gerações
3. Pequenas mudanças no genótipo podem levar a mudanças radicais no fenótipo e qualidade dos indivíduos
4. A complexidade do genótipo é limitada e peças complexas podem levar um longo tempo para serem processadas, mesmo quando não é possível distinguir quais as soluções raras que levarão a resultados interessantes

Como a música é uma arte temporal, a avaliação interativa leva a um compromisso entre a legitimidade da avaliação e o tamanho das amostras mostradas ao mentor, problema chamado de gargalo de aptidão (Todd & Werner 1999, Biles 1994). Uma abordagem ousada para lidar com o gargalo de aptidão são os algoritmos genéticos livres de aptidão (Freitas & Guimarães 2011b), como o descrito no Capítulo 2. Nestes algoritmos, permite-se que a originalidade emergja a partir de um ambiente livre de avaliação de aptidão à medida que é controlado fortemente através de operadores genéticos específicos e uma população inicial adequada.

Além de evitar o estudo das nuances do problema, esta abordagem é limitada no sentido em que o algoritmo não “sabe” que direção está tomando e tudo o que se pode fazer é esperar que a população inicial seja boa o suficiente e que os operadores genéticos não vão romper as melodias em algo menos aceitável que as melodias originais de referência. Para garantir isto, os operadores genéticos precisam ser inevitavelmente mais conservadores, em contraste com a audácia da ideia inicial, e eles não serão autorizados a alterar as soluções fortemente.

Outro substituto para o mentor humano é a avaliação através de redes neurais, que ainda não alcançou muito sucesso pois seu resultado típico é uma rede incapaz de generalizar além dos conjuntos de treinamento ou mesmo avaliar os conjuntos devidamente (Biles, Anderson & Loggi 1996).

O problema é frequentemente baseado na falta de informação ou falta de características facilmente computáveis no material musical (Biles 2007b). Outras tentativas da utilização de tais modelos conexionistas foram focadas na avaliação de música simples (Phon-Amnuaisuk, Law & Kuan 2007), imagens digitais (Gedeon 2008), e pinturas (Machado, Romero & Manaris 2008).

Portanto, um problema aberto na área é a formalização de funções de avaliação automáticas (Galanter 2010). Estas funções devem ser representáveis em linguagem de máquina, capazes de medir propriedades estéticas humanas e computáveis em termos práticos. Funções de avaliação não devem apenas definir o que é mais provável de ocorrer em melodias mas devem também permitir criatividade quando se consideram todos os objetivos estéticos diferentes para gerar ideias não imaginadas antes (Freitas & Guimarães 2011a). Avaliação estética computacional é um problema em aberto que é distintamente não trivial (Galanter 2010).

### 5.3 Funções de Avaliação Automática

Vários exemplos do uso de, ao menos, medidas formais simples para avaliação automática podem ser encontradas na literatura. Características heurísticas vêm de percepções do compositor enquanto outras características são baseadas em teoria musical (Freitas, Guimarães & Barbosa 2012). A diferença entre abordagens heurísticas e por regras deve-se ao processo: no caso das regras, parte-se de uma definição prévia da teoria musical; nas abordagens heurísticas é preciso experimentar a partir de certas intuições e verificar se o resultado é adequado. No final, chega-se também a regras, porém com um processo diferente.

Houve tentativas parciais de se desenvolver medidas estéticas automáticas de qualidade (Galanter 2010). Em uma pesquisa extensa, Towsey, Brown, Wright & Diederich (2001) consideram alguns aspectos simples para modelar uma função de avaliação e analisar quais destes impõem restrições fortes nas melodias. Com o menor desvio padrão entre estas características em um conjunto de dados, a variedade de alturas aparece como uma restrição forte nas melodias analisadas.

Medidas estéticas foram desenvolvidas para evoluir harmonizações barrocas de quatro partes para melodias de entrada (McIntyre 1994) e os resultados foram limitados à escala de dó maior para reduzir o espaço de busca. Algumas medidas formais de avaliação automática foram também utilizadas na geração de melodias de jazz (Papadopoulos & Wiggins 1998). Neste trabalho, há algumas medidas como intervalos, padrões, sus-

pensões, posição e duração das notas e contorno.

Embora estejamos descrevendo características a serem exploradas pela função de aptidão, é importante notar que outros resultados (Phon-Amnuaisuk & Wiggins 1999, Freitas & Guimarães 2011b) sugerem que alguns aspectos musicais precisam ser considerados não apenas pela função de avaliação mas também em métodos de manipulação conscientes que possam manter as soluções factíveis. Caso contrário, sistemas baseados em regras que não avaliam soluções iterativamente podem ser mais adequadas para estas tarefas (Phon-Amnuaisuk & Wiggins 1999).

Além de medidas que indicam como pessoas percebem arte cognitivamente, comparações com valores alvo também foram utilizadas para avaliar música, como tem sido utilizado para gerar vozes de sintetizadores similares a sons alvo (McDermott, Griffith & O'Neill 2005).

Apesar de aspectos melódicos alvo como padrões de sons poderem ser utilizados para avaliar melodias (Khalifa & Foster 2006, Horner & Goldberg 1991), no caso de composição algorítmica o problema é muito complexo pois ele envolve a necessidade de originalidade.

Medidas empíricas podem também ser utilizadas para formular conceitos de avaliação. Por exemplo, a lei de Zipf (Manaris, Machado, McCauley, Romero & Krehbiel 2005) e análises de fractais (Mori, Endou & Nakayama 1996) foram utilizadas para a avaliação de música e arte. Há também trabalhos que enfrentam estética em geral e uma medida estética foi definida em relação à tarefa de processamento de imagens do cérebro (Machado & Cardoso 1998), na qual o valor estético de uma obra de arte seria diretamente ligada à complexidade da imagem e inversamente conectada à complexidade de processamento.

Como pode-se ver da literatura, várias medidas foram utilizadas individualmente para avaliar melodias em computação evolucionária (Biles 2007a). Muito frequentemente, estes algoritmos genéticos não produziram resultados satisfatórios porque o contexto mais amplo das melodias não foi considerado.

Isto motiva o esforço neste trabalho para considerar pesquisas de musicologia para definir funções de avaliação com critérios melódicos estéticos. Definem-se aqui várias medidas para avaliação de aptidão divididas em várias seções seguintes. A partir de uma análise da literatura, a maior parte dos algoritmos também não examinam uma relação entre as categorias de medidas possíveis (Freitas, Guimarães & Barbosa 2012).

É importante mencionar que a aplicabilidade das medidas de aptidão apresentadas neste capítulo são dependentes do gênero de música com o qual o compositor pretende trabalhar. Alguns critérios comuns, como a penalização de intervalos verticais muito grandes, por exemplo, não se aplicam a várias melodias barrocas, que podem ter intervalos verticais muito mais largos que várias melodias renascentistas.

Para outras melodias, a tonalidade não pode ser tão facilmente implicada e isto torna a transposição de todas as melodias para um mesmo tom menos factível. Assim, conquanto alguma adaptação ao material com o qual o leitor estará trabalhando é esperada, a maior parte das observações no texto são adequadas para um repertório clássico ou barroco de melodias tonais e elas também se aplicam a um espectro largo de melodias populares do século XX.

## 5.4 Banco de Dados Bossa Nova

Com várias análises de melodias de um banco de dados em diferentes perspectivas, pode-se comparar as melodias do banco de dados às potenciais soluções de um algoritmo de otimização. Alguns resultados indicam parâmetros com uma distribuição normal, como a distribuição de alturas. A distribuição normal pode ser testada em uma solução candidata com um teste de Jarque-Bera (Jarque & Bera 1987).

Alguns resultados podem mostrar que parâmetros vêm de outra distribuição, tal como na distribuição de proporção rítmica, que pode ser comparada a uma solução candidata com um teste de Kolmogorov-Smirnov com duas amostras (Massey Jr 1951).

Alguns parâmetros podem apenas representar valores categóricos, que podem ser comparados com um teste estatístico nominal (Bashkansky, Gadrich & Kuselman 2012).

Finalmente, outros resultados indicam potenciais valores alvo individuais para as melodias, como o tempo de cada melodia, que pode ser diretamente incluída nos valores de função objetivo como distância de um alvo.

Assim, há várias maneiras possíveis de se mapear os resultados das medidas em funções de avaliação específicas. Para dar uma boa representação das medidas mencionadas, criamos um corpo de dados com 458 frases melódicas de 26 músicas de Bossa Nova retiradas de um songbook de Tom Jobim (Jobim & Chediak 1990).

Todas as melodias foram manualmente codificadas em uma esquema de representação baseado em matrizes de  $n$  linhas e 3 colunas, onde cada linha representa uma nota e cada coluna representa, respectivamente, a posição, duração e altura da nota.

Posição	Duração	Nota
1	0,5	74
1,5	1	77
2,5	1	81
3,5	2,5	80
6	2	81
8	2,5	77
10,5	1	74
11,5	1	77
12,5	3,5	74

**Tabela 5.1:** Representação de uma frase melódica do banco de dados.

Um exemplo deste esquema de representação está apresentado na Tabela 5.1. A posição considera um compasso de 4 pulsos enquanto as notas seguem a notação MIDI, onde um 60 representa um dó e cada variação de uma unidade representa uma variação de um semi-tom na nota.

Todos os dados desta base de dados estão disponíveis no endereço <http://www.alandefreitas.com/>.

## 5.5 Medidas Baseadas em Altura

Altura é como humanos diferenciam sons de diferentes frequências. Valores de altura são provavelmente os aspectos principais a serem analisados em uma melodia. Um corpo de dados musical grande foi produzido sobre altura (Vos & Troost 1989) para determinar a relevância perceptiva de diferentes possibilidades de conjuntos de altura.

Uma primeira abordagem intuitiva para analisar alturas seria um histograma de distribuição de alturas, que podem revelar as notas mais utilizadas em todas as melodias. Quando melodias vêm de peças em tons diferentes, é provavelmente mais útil transpor as melodias ao mesmo tom antes de gerar o histograma. O histograma de distribuição de alturas transpostas pode então dar uma sugestão das escalas sendo mais utilizadas nas melodias. Estes histogramas podem também considerar uma ou mais oitavas, onde a mesma altura seria contada separadamente se estiver sendo executada em uma frequência diferente.

É importante analisar também a variedade de alturas em cada melodia pois todas

as notas reveladas pelo histograma podem não estar ocorrendo na mesma proporção em todas as melodias. Um método para se medir a variedade de alturas (Towsey, Brown, Wright & Diederich 2001) seria dividir o número de alturas distintas pelo número de alturas possíveis para a melodia.

Outro aspecto da variedade de altura é o âmbito de altura (Towsey, Brown, Wright & Diederich 2001), que é a diferença entre a nota mais grave e mais aguda na melodia.

Melodias são divididas em frases musicais, que podem também ter alguma tendência a se iniciar ou terminar com alturas específicas. Assim, seria importante estudar histogramas de tons iniciais e tons finais. Quando melodias são geradas para sequências de acorde específicas, é particularmente importante estudar a relação entre estas notas e os acordes sendo executados.

Assim como utilizado em algumas abordagens, como para definir a tonalidade de melodias (Krumhansl 2001), pode ser útil fazer histogramas de distribuição de alturas ponderadas, onde a relevância de uma altura é proporcional a sua duração. Isto faz sentido intuitivamente já que se ouve mais as alturas com duração mais longa.

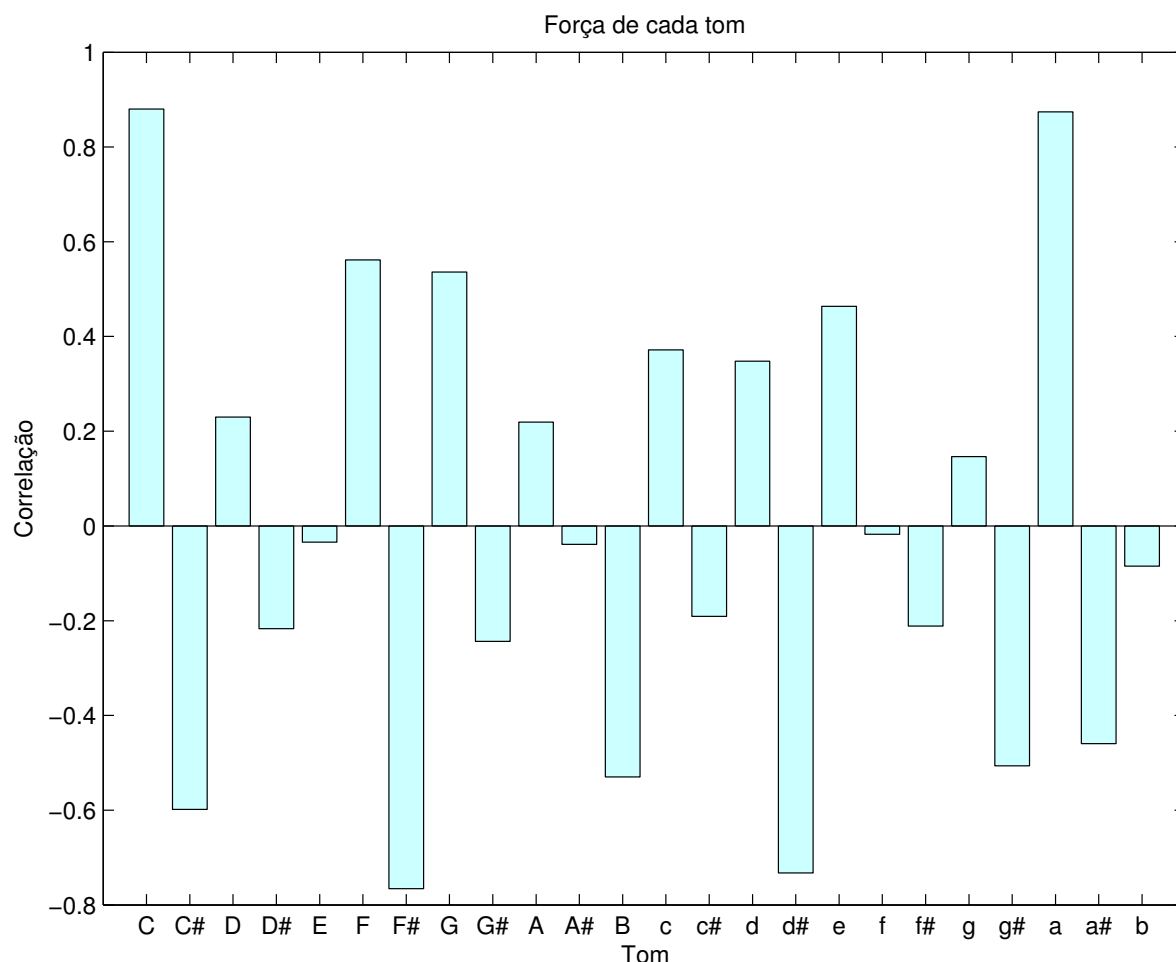
### 5.5.1 Tonalidade e Dissonância

Em relação a melodias, é importante definir a tonalidade da melodia se esta informação não está disponível porque isto muda as notas que serão desejáveis quando os intervalos são decididos. O algoritmo K-S de descoberta de tonalidade (Krumhansl 2001) se utiliza de perfis de tonalidade para encontrar o tom de uma música baseando-se em suas alturas.

Detectamos a tonalidade de cada melodia em nosso banco de dados com o algoritmo K-S. Como as melodias podem ter até mesmo mudanças de tom, esta pode ser uma abordagem simples. Porém, todos os tons detectados corresponderam às armaduras nas partituras e os resultados podem de qualquer maneira dar uma ideia das tonalidades em relação aos perfis do algoritmo, como podemos ver na Tabela ???. Nesta tabela estão apresentadas a correlação entre as melodias e cada uma das tonalidades, onde quanto maior a correlação, maior a probabilidade da melodia estar neste tom.

Assim, transpomos todas as músicas para dó maior, ou lá menor, o seu relativo menor. Isto torna análises dependentes de tonalidade possíveis, tais como a detecção de dissonâncias. A correlação dos perfis de tonalidade do algoritmo à distribuição de alturas nas peças leva a uma representação da força de cada tonalidade, como na Figura

5.1.

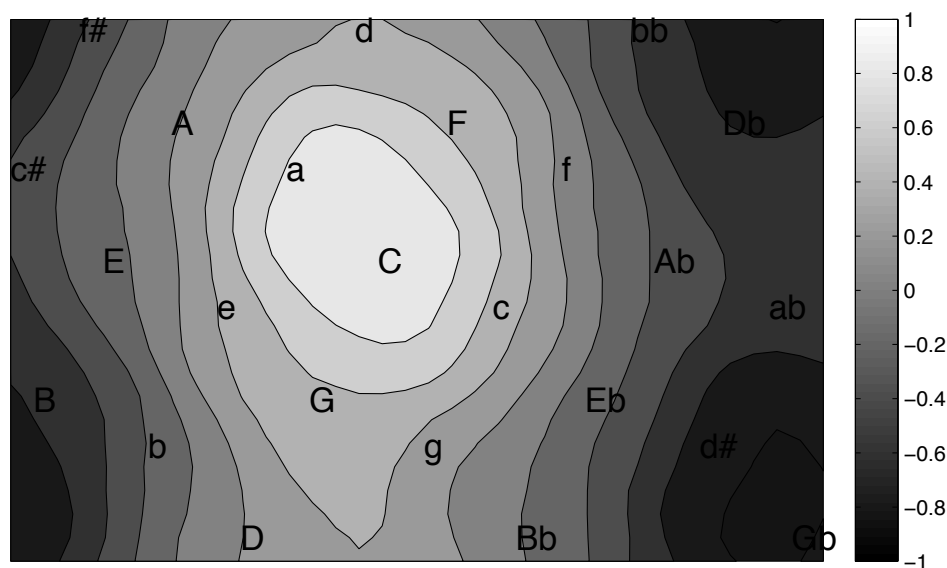


**Figura 5.1:** Coeficiente de Correlação de Tons nas Melodias em já Transpostas em Dó Maior. Quanto maior a correlação, maior a probabilidade desta melodia estar neste tom, de acordo com os perfis de tonalidade.

Os valores de correlação são significativamente maiores para os perfis de tonalidade de C maior e A menor, indicando alguma relevância do método aplicado.

A tonalidade de uma melodia pode também ser induzida e projetada em uma rede neural baseada em mapas auto-organizáveis de tonalidade treinada com 24 perfis de tonalidade, como descrito por Toiviainen, Krumhansl et al. (2003). Os resultados projetados em um mapa auto-organizável treinado com os perfis de tonalidade estão na Figura 5.2. No mapa apresentado, quanto maior a altura apresentada pelas curvas de nível, maior a probabilidade de uma melodia estar naquele tom de acordo com os perfis de tonalidade. A tonalidade da melodia pode também ser analisada através da melodia

com menores janelas de tempo.



**Figura 5.2:** Resultados Projetado em uma Rede de Mapas Auto-Organizáveis. Quanto mais altas as curvas de nível em torno de uma certa tonalidade, maior a probabilidade da melodia estar neste tom.

Conforme identifica-se a tonalidade, pode-se usualmente reconhecer a ocorrência de cada dissonância, ou notas infrequentes, usadas na melodia e a variedade de dissonância. É importante medir que dissonâncias são atraídas à nota mais próxima pertencente à escala apropriada ou tendo outra função.

Outros aspectos já empregados na literatura (Towsey, Brown, Wright & Diederich 2001) são (i) centros de tonalidade (proporção de notas tônicas ou dominantes), (ii) notas fora da escala e (iii) intervalos dissonantes.

### 5.5.2 Distribuição de Alturas

As alturas utilizadas em todas as melodias estão na Figura 5.3(a). Estas alturas estão em codificação MIDI. Os resultados mostram que a distribuição destas notas é

muito normal.

Contudo, ao transpor estas melodias à mesma tonalidade, tem-se uma grande diferença de ocorrência entre notas consecutivas, como na Figura 5.3(b). Isto é devido às notas dissonantes, que são estranhas à escala principal.

Dadas as 12 classes de altura, o resto de um valor de altura dividido por 12 é a classe desta nota. A ocorrência destas classes dão uma ideia melhor das escalas utilizadas na melodia, como na Figura 5.3(c). Pode-se notar uma ocorrência maior de notas na escala diatônica de dó.

A variedade de notas é também diferente para cada melodia. Dividindo-se o número de notas distintas por 12, temos na Figura 5.3(d) que a variedade de notas é muito diferente entre as melodias, mas todas as melodias usam mais que 70% das notas possíveis.

Outro aspecto de variedade de altura é o alcance de altura, ou a diferença entre as notas mais grave e mais aguda. Isto está representado na Figura 5.3(e), onde se vê que o alcance das notas tem uma distribuição mais normal, centrada na faixa de 16 notas.

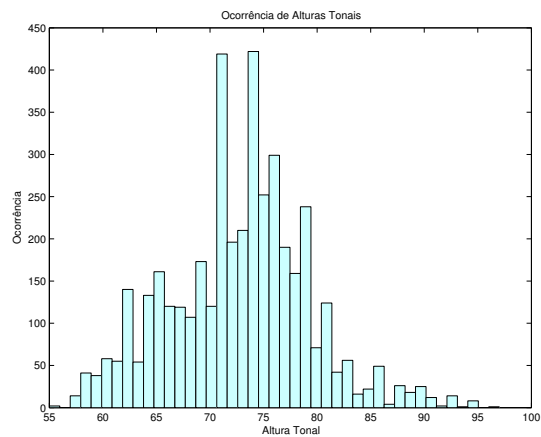
Outras medidas úteis são as alturas iniciais e finais assim como notas com distribuição ponderada em relação à duração. Para nosso banco de dados, esta medida não representou muita diferença, como se vê na Figura 5.3(f).

### 5.5.3 Intervalos

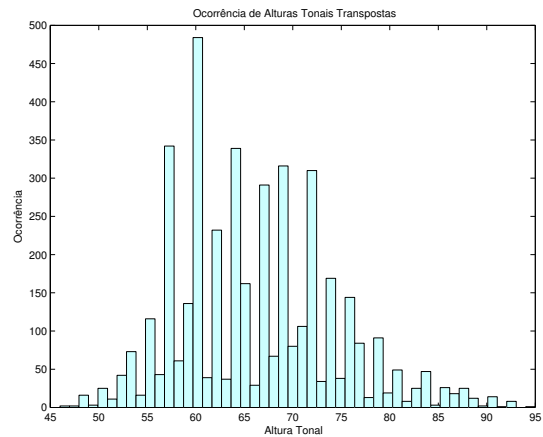
Outra classe de informação melódica, talvez mais importante que os valores absolutos de altura, se relaciona com os intervalos entre alturas. Estes intervalos são usualmente mensurados em termos de graus da escala ou semitons e são a base de vários algoritmos de composição algorítmica.

Assim, a primeira informação que pode-se extrair nesta categoria é um histograma de distribuição de intervalos e a distribuição de tamanhos de intervalo. A Figura 5.4(a) mostra os intervalos presentes nas melodias.

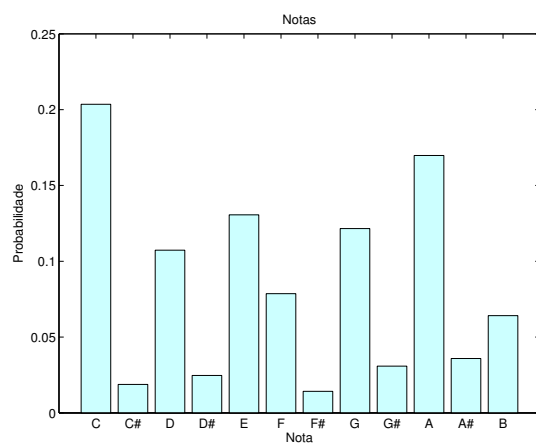
Um modelo teórico da ocorrência de cada intervalo (Dowling & Harwood 1986) indica que intervalos curtos devem ocorrer mais frequentemente que intervalos largos. Assim, outra maneira de analisar os intervalos seria a comparação da diferença entre os intervalos e o modelo teórico. De fato, é prática comum penalizar intervalos muito grandes na avaliação de soluções (Papadopoulos & Wiggins 1998). Porém, isto pode ser apenas aplicável a alguns gêneros de música e uma abordagem baseada em uma análise mais



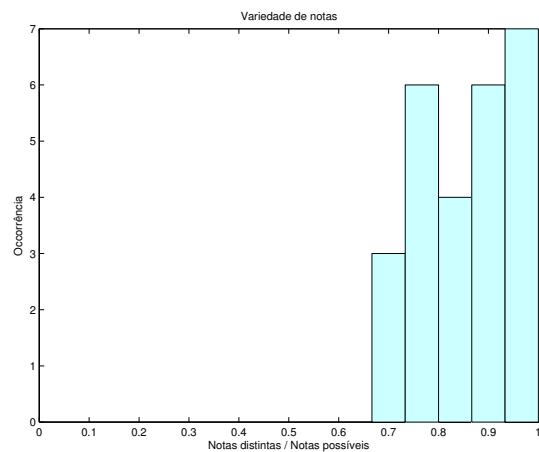
(a) Alturas



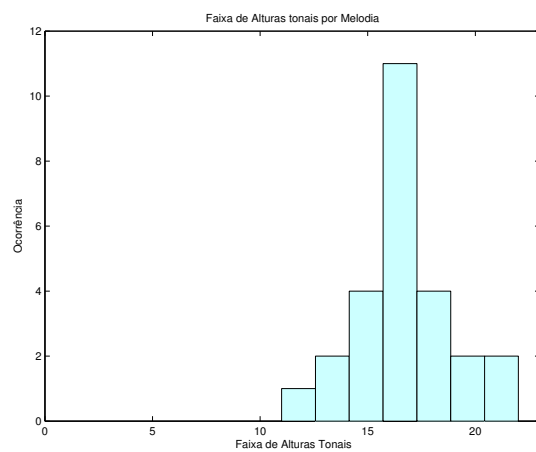
(b) Alturas Transpostas



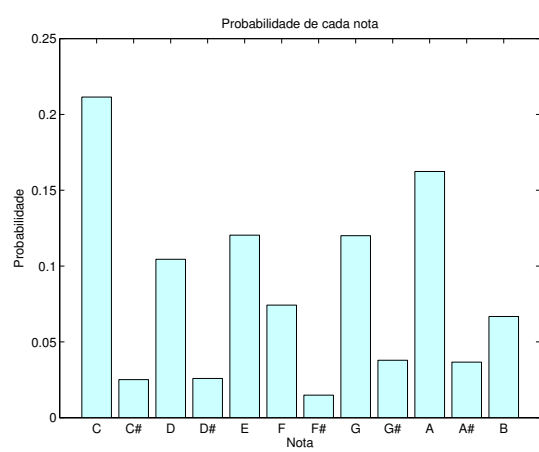
(c) Classes de Notas



(d) Variedade de Notas

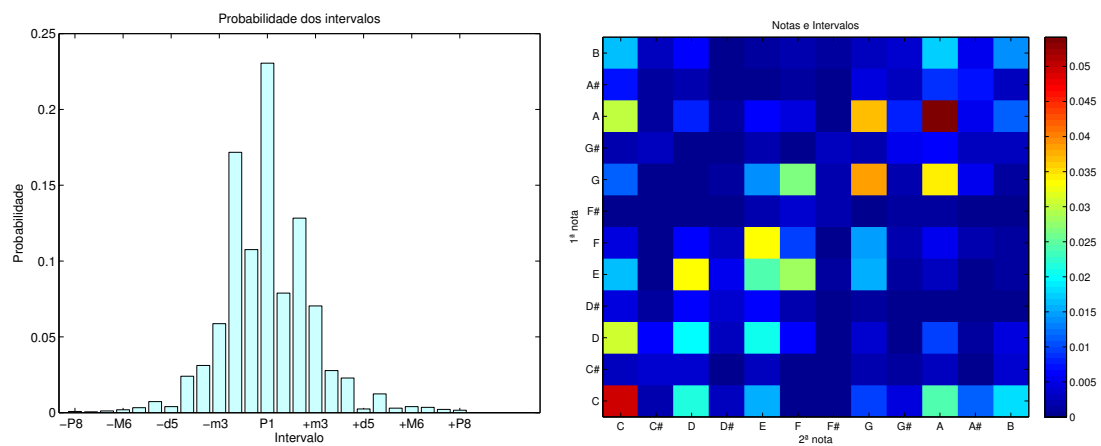


(e) Alcance de Notas



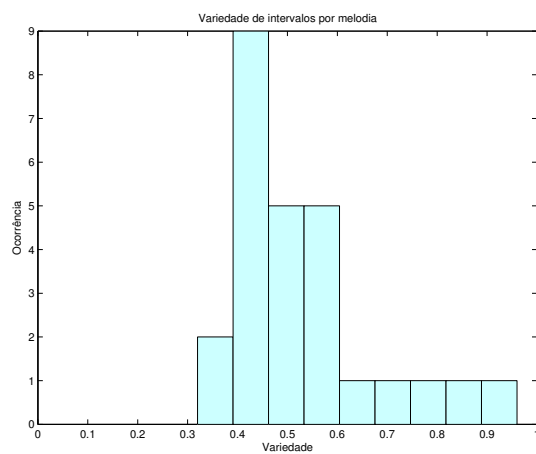
(f) Notas Ponderadas pela Duração

**Figura 5.3:** Medidas de Altura Tonal



(a) Intervalos

(b) Notas e Intervalos



(c) Variedade de Intervalos

**Figura 5.4:** Medidas de Intervalo

profunda é recomendada.

Quando se combina a informação das notas aos intervalos, encontra-se que a probabilidade da próxima nota depende da nota corrente, como mostrado na Figura 5.4(b).

Como descrito para as alturas, é também importante estudar a variedade de intervalos quando se extrai características das melodias. Isto está representado na Figura 5.4(c). A variedade de intervalos pode ser calculada como o número de intervalos divididos pelo número de intervalos possíveis, definidos em um limite prático.

Há também evidência de que intervalos melódicos curtos tendem a ser predominantemente descendentes enquanto os largos tendem a ser ascendentes (Vos & Troost 1989). Então um histograma de ascendentes e descendentes, assim como a proporção de intervalos ascendentes são aspectos bons para analisar a direção dos intervalos.

Para saber a forma da melodia, outra medida é a estabilidade em relação ao último intervalo (Towsey, Brown, Wright & Diederich 2001). Isto representaria quanto os intervalos tendem a seguir a mesma direção, à medida que frases musicais focam em alcançar tons mais altos ou mais baixos.

Como intervalos muito largos costumam ser raros em grande parte dos estilos de música tonal, é uma prática comum penalizar intervalos muito largos na avaliação das soluções (Papadopoulos & Wiggins 1998). Apesar de isto ser uma prática comum, lembra-se que esta regra pode ser muito específica de alguns gêneros e usualmente fazer menos sentido ainda quando o intervalo largo ocorre entre duas frases.

#### 5.5.4 Expectativa Melódica

Quando alguém escuta uma melodia, é comum haver uma expectativa do que está por vir. Um estudo importante no assunto é o modelo de implicação-realização de Narmour (Narmour 1992), que pode prever expectativa melódica razoavelmente bem.

O modelo se baseia em princípios que consideram a expectativa do ouvinte após um dado intervalo. Há quantificações destes princípios na literatura (Krumhansl 1995, Eerola & Toiviainen 2004). Assim, pode-se penalizar melodias que desrespeitam o modelo ou mensurar o quanto novas melodias devem seguir o modelo. Embora o modelo completo seja muito complexo, seus aspectos principais são mencionados aqui.

O primeiro princípio do modelo é direção de registro e ele implica que intervalos curtos tendem a ser seguidos por uma continuação na direção da altura enquanto intervalos

grandes dão expectativa de uma mudança em direção.

Diferença de intervalos é o princípio de que intervalos pequenos implicam em outros intervalos de tamanho similar enquanto intervalos grandes implicam em intervalos menores.

Quando um segundo intervalo leva a uma altura próxima da altura inicial, tem-se o princípio de retorno de registro. Assim, ouvintes esperam saltos que retornem a uma altura similar.

O princípio de proximidade simplesmente define que intervalos pequenos são mais esperados que intervalos grandes. O último princípio é encerramento e ele ocorre quando há mudanças na direção da melodia ou quando um intervalo grande é seguido de um pequeno.

O modelo foi quantificado por Krumhansl (1995). Em seu trabalho, ela propõe, além dos princípios já citados, um novo princípio chamado consonância. Este princípio declara que uníssonos, quartas perfeitas, quintas e oitavas são intervalos favorecidos. Contudo, isto é uma afirmação muito específica para alguns contextos pois as melodias podem ter muito mais intervalos cromáticos ou dissonantes, dependendo de seu gênero.

Claro que a tonalidade das melodias também influencia as expectativas dos ouvintes, então os valores de expectativa podem ser mudados para se conformar mais aos valores de estabilidade tonal (Krumhansl & Kessler 1982).

Além de que, a altura correntemente usada em uma melodia pode influenciar o tamanho do próximo intervalo já que ela pode tender a ser seguida por uma nota que pertence à tonalidade. Como tons instáveis tendem a ser atraídos por tons estáveis, atração melódica (Lerdahl 1996) pode também ser analisada nas melodias.

Duas reformulações no modelo (Von Hippel 2000) são tessitura, que declara que as notas seguintes tendem à altura mediana, e mobilidade, que usa correlação para prever se a próxima altura é previsível em relação aos intervalos prévios.

Uma implementação quantificada de todas estas medidas está disponível em Eerola & Toiviainen (2004).

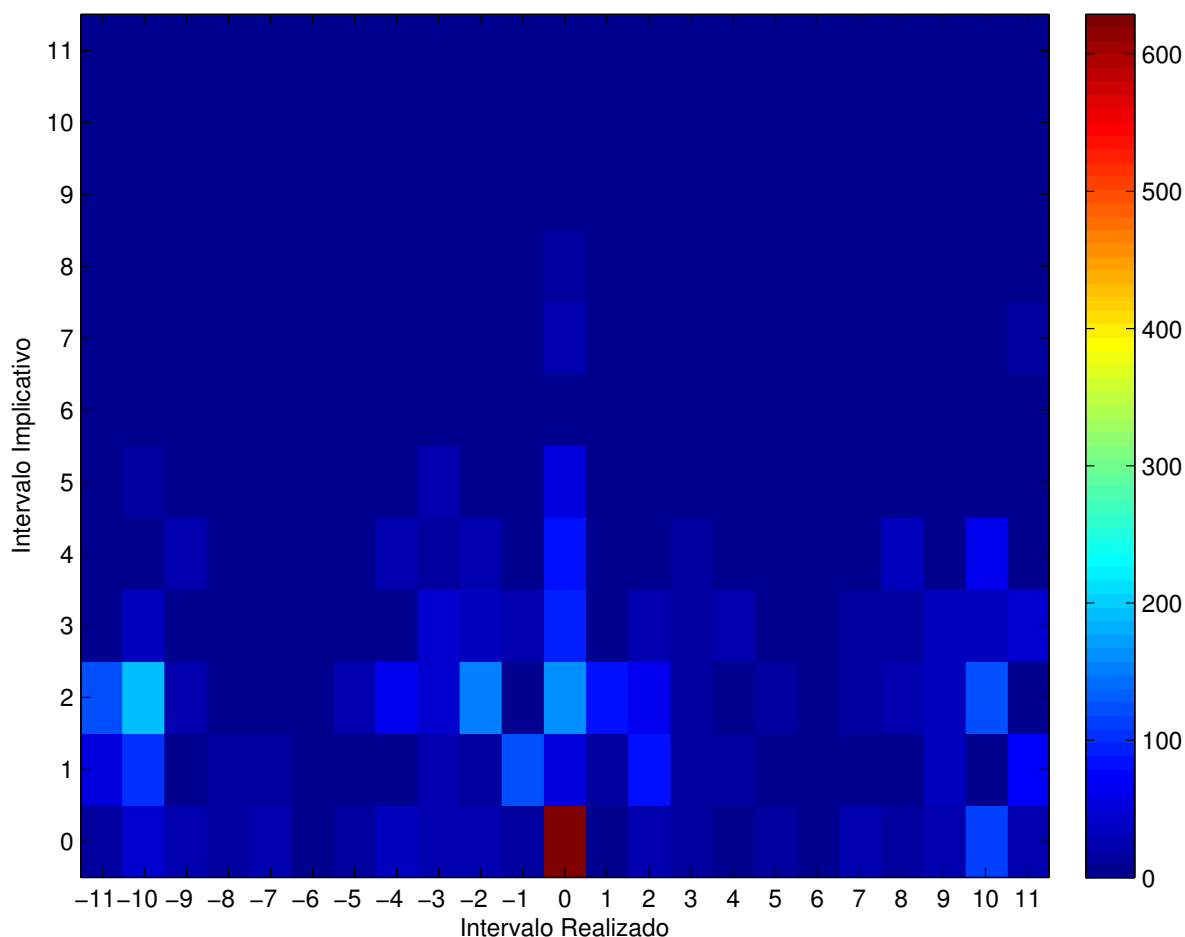
Há modos de se utilizar estas expectativas em uma função de avaliação:

1. Usando o modelo diretamente e penalizando melodias que não seguem o modelo
2. Estudando a relação entre um corpo grande de dados e o modelo antes de penalizar

melodias que não tem uma relação similar

Como o modelo original pode ser específico para certos contextos ou ineficientes para considerar intervalos estáveis em relação ao tom (Krumhansl & Kessler 1982), podemos usar os intervalos nas melodias para inferir nosso próprio modelo de expectativa que seria específico para nossos objetivos.

A Figura 5.5 mostra tal modelo, onde as linhas representam os intervalos implicativos e as colunas representam os intervalos realizados. O modelo confirma a expectativa de intervalos pequenos.



**Figura 5.5:** Expectativa Melódica

Atração melódica também deve ser considerada por este modelo de expectativa já que temos respostas diferentes para notas diferentes (Lerdahl 1996). Um modo de fazer isto seria inferir 12 modelos diferentes de acordo com a nota corrente.

	Ascendentes	Uníssonos	Descendentes
Após um ascendente	<b>30.17%</b>	18.78%	51.03%
Após um uníssonos	19.92%	<b>55.24%</b>	24.82%
Após um descendente	45.36%	14.57%	<b>40.06%</b>
Em geral	33.52%	26.95%	39.52%

Tabela 5.2: Contorno

### 5.5.5 Contorno

Contorno representa os tipos de movimentos sendo realizados pelas melodias. O contorno de uma melodia é usualmente mais fácil de ser lembrado do que todos os intervalos (Dowling 1978).

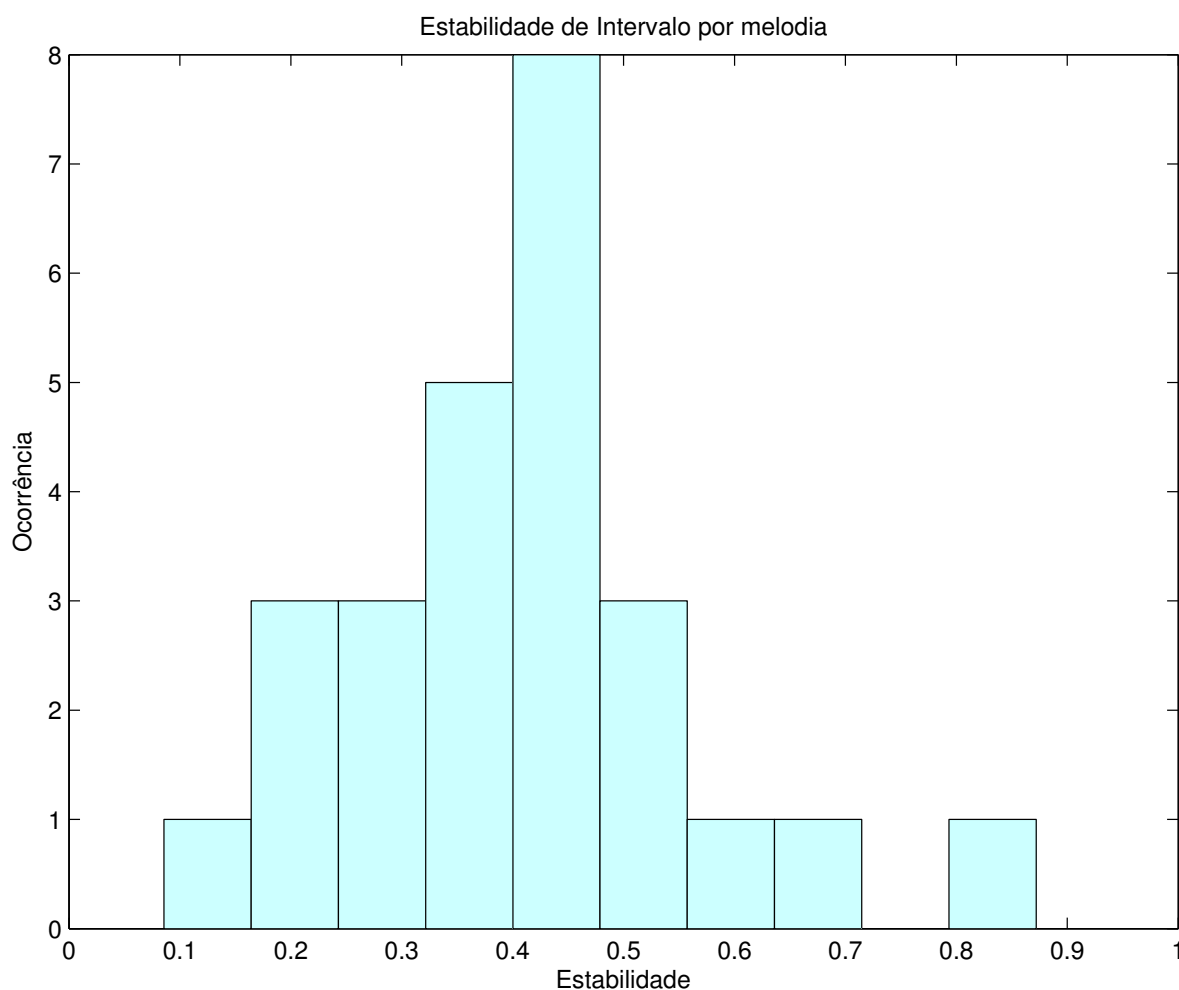
Há vários tipos de contorno melódico e a ocorrência de cada um pode ser utilizada para avaliar melodias. Pode-se identificar algumas formas de contorno (Nettl 1956) em melodias: (i) ascendente, (ii) descendente, (iii) ondulante, (iv) pêndulo, (v) em cascata ou (vi) arco. Naturalmente, outro aspecto a se analisar é a variedade de tipos de contorno. Medidas de contorno já empregadas (Towsey, Brown, Wright & Diederich 2001) são:

- Direção média de contorno: a direção dos intervalos
- Estabilidade do contorno: quantos intervalos mudam de direção
- Movimentos por passos: a quantidade de intervalos diatônicos
- Retornos em salto: intervalos grandes não seguidos de um intervalo de retorno
- Força de clímax: o inverso do número de vezes que uma nota climática é repetida em uma melodia

Uma forma fácil de analisar contorno é medir quantos intervalos são ascendentes ou descendentes e a estabilidade em relação à direção. A Tabela 5.2 mostra os valores de ascendentes e descendente em geral ou em relação ao último intervalo.

Os valores em negrito representam a estabilidade de contorno, que é um critério que também já foi utilizado em algoritmos evolucionários (Towsey, Brown, Wright &

Diederich 2001). Cada vez que dois intervalos ocorrem na mesma direção, isto é contabilizado como uma ocorrência de estabilidade. Estes valores de estabilidade para cada música estão representados na Figura 5.6.



**Figura 5.6:** Estabilidade de Contorno

Outra forma de controlar contorno é através da direção média de contorno (Papadopoulos & Wiggins 1998, Towsey, Brown, Wright & Diederich 2001).

### 5.5.6 Dissonância

Analisando-se tonalidade e alturas em conjunto, pode-se ter uma ideia das dissonâncias usadas nas músicas. Na Figura 5.3(c) pode-se perceber que seria muito razoável analisar a dissonância em termos da proporção de notas que não pertencem à escala diatônica.

Assim, a probabilidade de uma nota dissonantes é 30.53%, mas a Figura 5.7, que mostra a ocorrência de dissonância dividida pelo número de notas dissonantes possíveis, mostra como este valor pode variar consideravelmente. Nesta figura cada vez que uma nota fora da escala diatônica ocorre, isto é contabilizado como uma ocorrência de dissonância.

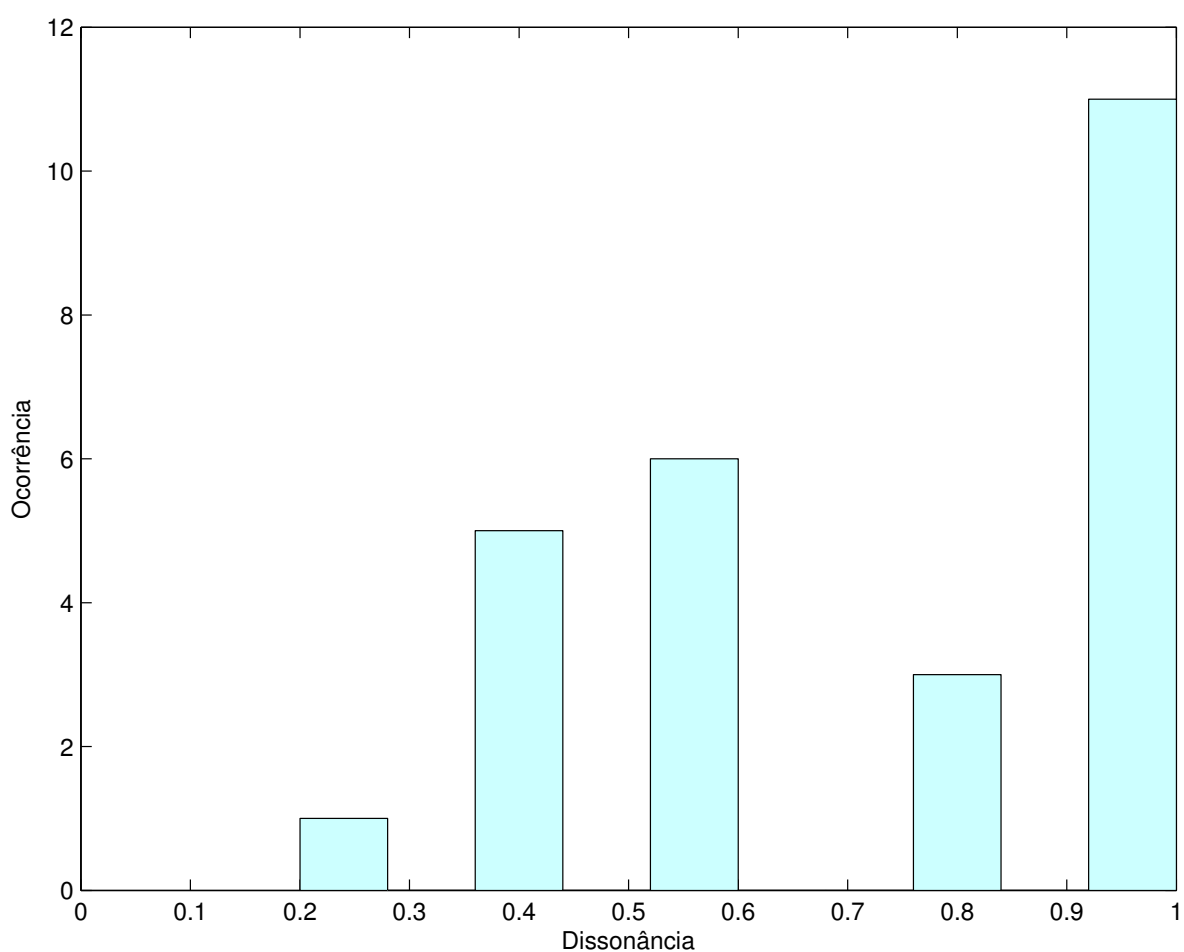


Figura 5.7: Dissonância

A atração de notas dissonantes a tons estáveis acontece a 55.25% das dissonâncias. Contudo, esta medida se sobrepõe com a medida de notas de segunda ordem, como a mostrada na Figura 5.4(b).

## 5.6 Medidas Baseadas em Ritmo

O primeiro aspecto a ser analisado em relação ao ritmo é a duração das notas e variedade da duração. Para fazer isto, todas as durações possíveis devem ser determinadas

a um limite prático.

A Figura 5.8(a) mostra uma análise de segunda ordem da proporção da duração das notas. De 36 valores possíveis de duração presentes nas melodias, o histograma é baseado nas durações mais curtas que 4 pulsos e mais longas que  $1/4$  de pulso que ocorrem nas melodias, independentes de suas durações (Eerola & Toiviainen 2004). Os padrões mostram uma tendência à repetição na duração das notas.

Outras características a serem estudadas aqui são as durações das primeiras e últimas notas em cada frase musical. A duração das primeiras notas é mostrada na Figura 5.8(b). As primeiras notas tendem a ter duração maior que as últimas notas, mostradas na Figura 5.8(c).

A proporção rítmica ou a diferença rítmica das melodias, ou a proporção ou diferença entre as notas mais longa e mais curta, como mostrado na Figura 5.8(d), é também algo a ser estudado.

Como fizemos para as alturas, pode-se também calcular a variedade de duração nas melodias, como na Figura 5.8(e).

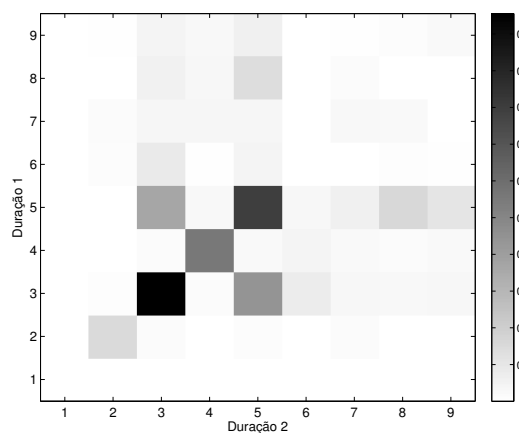
Também tão importante quanto analisar a ocorrência de alturas em uma melodia é perceber a ocorrência de silêncio nas melodias já que ele pode mudar dramaticamente seus sentidos. Na Figura 5.8(f), temos a quantidade de silêncio de até 2 pulsos por melodia. Em alguns casos, até mais de 10% da melodia pode ser silenciosa.

Outra característica que forma o ritmo é o momento em que as notas são executadas. Duas feições importantes são a localização das notas no compassos e a localização das notas no pulso. Foi revelado que uma grade hierárquica de localizações de notas podem existir na expectativa de melodias ocidentais (Palmer & Krumhansl 1990).

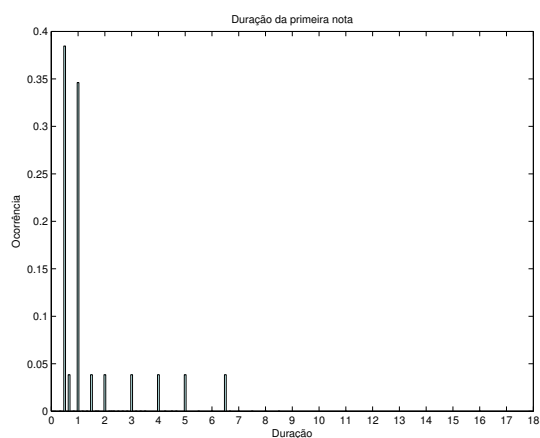
Esta medida também pode ser útil para encontrar ritmos sincopados na base de dados. É também relevante saber a localização das primeiras notas e a localização das últimas notas. Por exemplo, as posições das notas no compasso musical estão representadas na Figura 5.9(a).

As primeiras e últimas notas também podem ter posições diferentes, como no exemplo da Figura 5.9(b). De fato, apenas 6 valores de posição são utilizados como primeira nota enquanto 16 valores são usados para todas as notas.

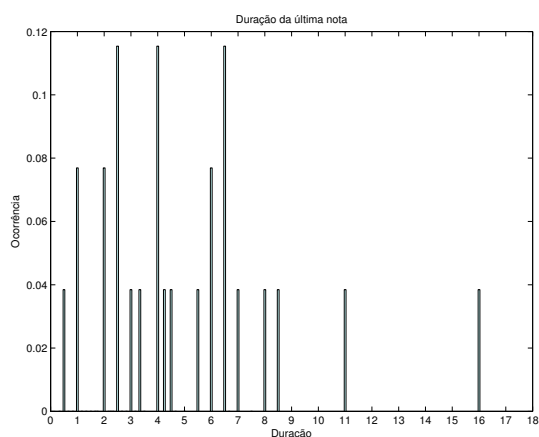
Os dados relacionados à localização das notas pode também ser comparado aos dados relacionados à posição das notas, ponderados por sua duração, já que notas com maior



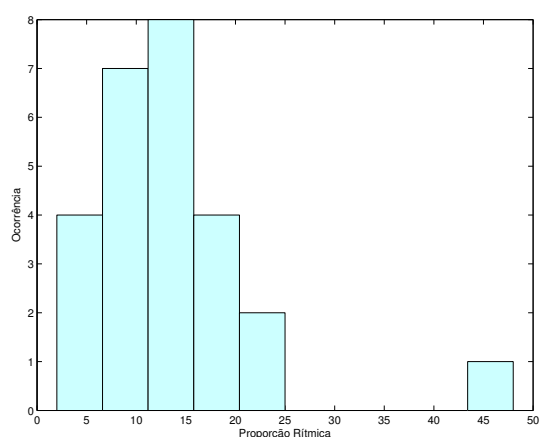
(a) Duração das Notas



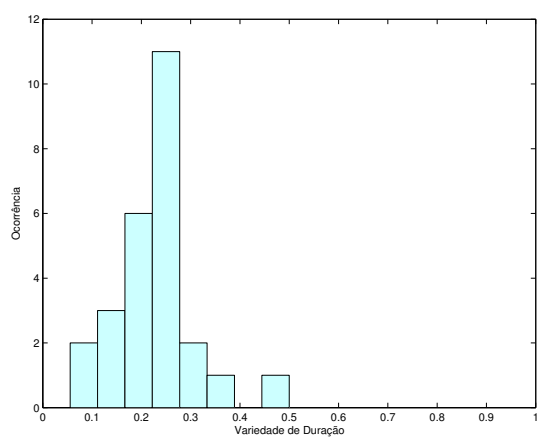
(b) Duração das Primeiras Notas



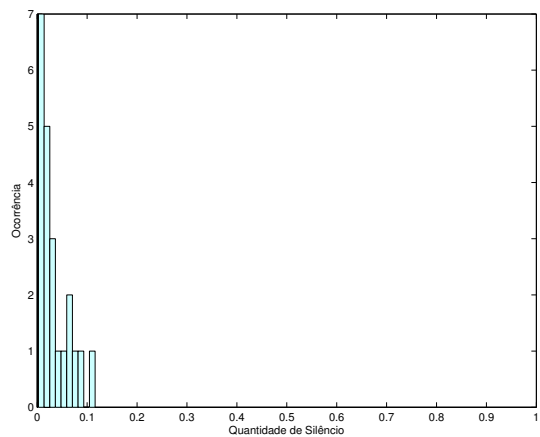
(c) Duração das Últimas Notas



(d) Proporção Rítmica

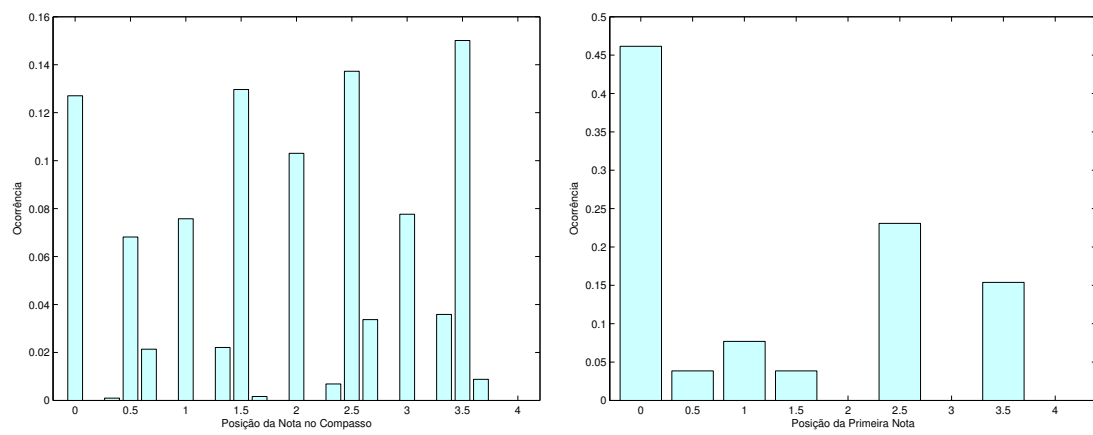


(e) Variedade de Duração



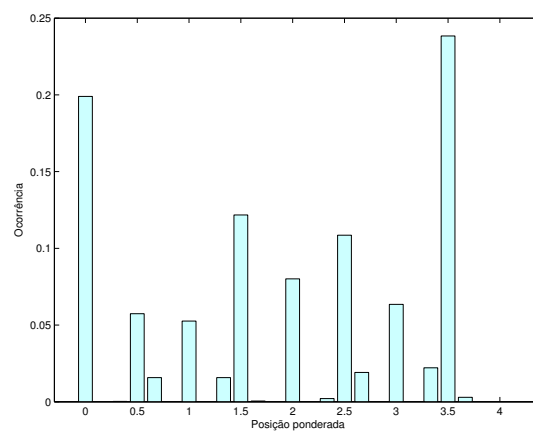
(f) Variedade de duração de silêncio

**Figura 5.8:** Medidas de Duração de Notas



(a) Posição das Notas

(b) Primeira Posição das Notas



(c) Posição Balanceada das Notas

**Figura 5.9:** Medidas de Posição de Notas

duração são mais notadas pelos ouvintes (Thompson 1994). A Figura 5.9(c) mostra a relação entre estes dois componentes.

Outras medidas interessantes (Papadopoulos & Wiggins 1998, Towsey, Brown, Wright & Diederich 2001) são:

- Suspensões: notas que atravessam acordes consecutivos
- Notas em tempos fortes: já que estes são posições significativas no compasso
- Notas longas: notas que duram demais
- Densidade da nota: a dispersão ou ocorrência frequente na melodia
- Densidade de intervalo: duração total de silêncio
- Variedade rítmica: as durações usadas comparadas às durações possíveis
- Alcance rítmico: a diferença de posição da nota mais longa até a nota mais curta, dividida pelo número de durações possíveis

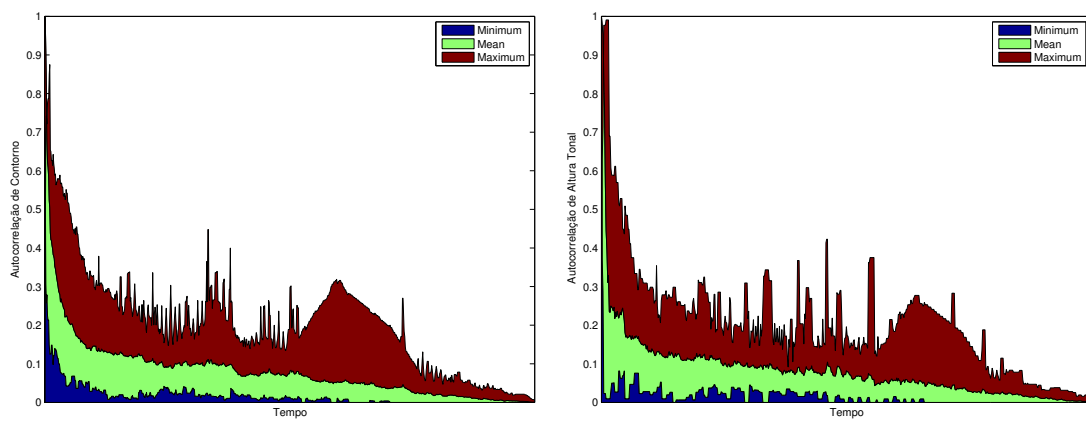
## 5.7 Medidas de Padrões

A relação entre altura e ritmo pode ser usada para fazer uma identificação de padrões ou frases melódicas que sejam psicologicamente motivadas (Leman 1997). Uma técnica de autocorrelação (Eerola, Himberg, Toiviainen & Louhivuori 2006) pode identificar estes padrões correlacionando séries com uma cópia deslocada da melodia.

Os valores de correlação vão de 0 a 1 e estes valores são sempre 1 no ponto 0, quando comparamos a forma de uma melodia com uma cópia de si mesmo, como mostrado na Figura 5.10(a). As três áreas representam a correlação máxima, mediana e mínima. Semelhantemente à forma do contorno, podemos aplicar esta técnica apenas às notas ou durações, cujos resultados estão nas Figuras 5.10(b) e 5.10(c) .

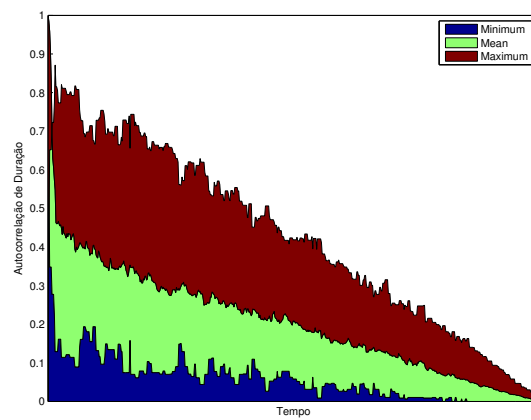
Esta técnica de autocorrelação pode ser usada para identificar padrões de contorno, altura ou ritmo. Outra ideia para encontrar padrões em alturas é usar casamento de padrões (Papadopoulos & Wiggins 1998) para fazer um histograma do número de padrões de altura encontrados para cada tamanho de melodia.

Outra maneira de se perceber padrões é identificar o número de padrões de um tamanho específico na melodia. Pode-se analisar isto na Figura 5.11(a), onde cada



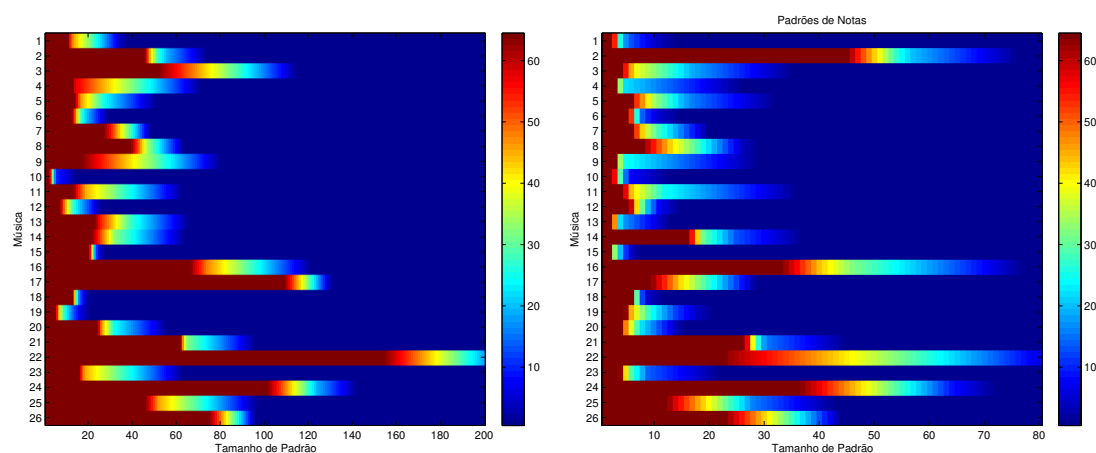
(a) Autocorrelação de Altura e Duração

(b) Autocorrelação de Altura



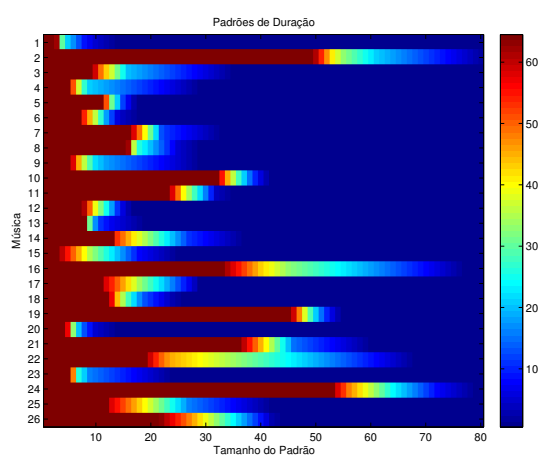
(c) Autocorrelação de Duração

**Figura 5.10:** Medidas de Autocorrelação



(a) Padrões de Contorno

(b) Padrões de Altura



(c) Padrões de Duração

**Figura 5.11:** Medidas de Padrões

linha representa uma melodia, cada coluna representa um tamanho do padrão e as cores representam a quantidade deste padrão. Padrões curtos são naturalmente mais comuns já que padrões longos e raros podem representar a repetição de frases longas na melodia. A mesma medida pode ser aplicada a notas e valores de duração, como nas Figuras 5.11(b) e 5.11(c).

Outras medidas (Towsey, Brown, Wright & Diederich 2001) são o número de alturas repetidas em relação ao número de intervalos usados e valores rítmicos repetidos. Extensões desta ideia são o número de alturas repetidas de padrão rítmico de  $n$  notas.

### 5.7.1 Frases

Melodias usualmente consistem de várias frases musicais. Dividindo a melodia em frases, pode-se analisar (i) o tamanho das frases nas melodias tanto quanto (ii) a relação entre frases e (iii) todas as características mencionadas até aqui individualmente para as frases. A última é atraente para perceber se a extensão de possibilidades usadas em todas as músicas também se aplicam a diferentes frases da mesma música.

Na maior parte dos casos, a informação sobre a segmentação das melodias não está disponível. Neste caso, há abordagens baseadas em regras para segmentação melódica baseadas em psicologia de Gestalt (Tenney & Polansky 1980) ou extensões destas ideias (Cambouropoulos 1997). Há também abordagens estatísticas baseadas na análise de melodias (Bod 2002). A Figura 5.12 mostra o número de frases por melodia de acordo com a abordagem baseada em regras (Tenney & Polansky 1980).

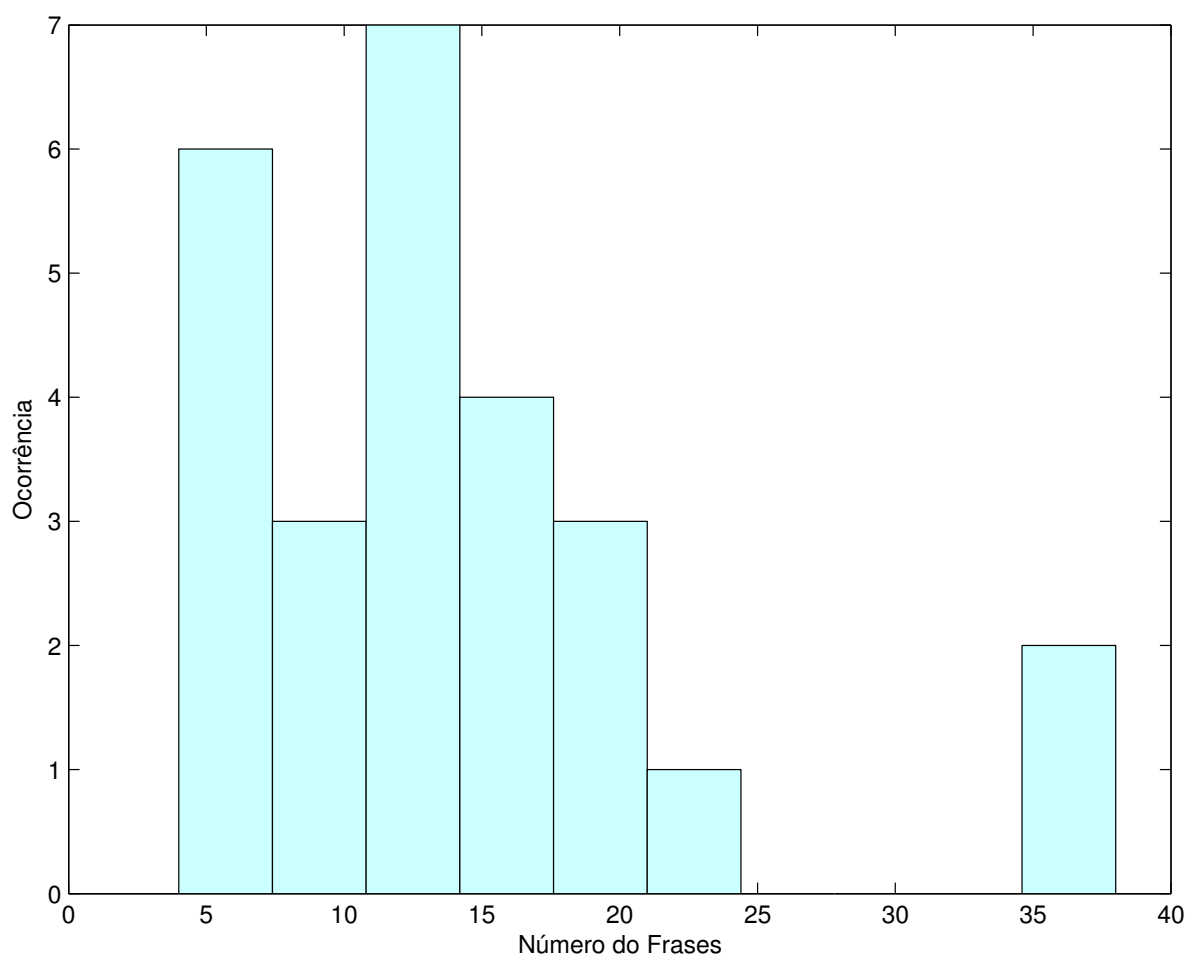


Figura 5.12: Frases

Semelhantemente ao modo como descrevemos a possibilidade de se usar autocorrelação para encontrar padrões na melodia, uma vez com os limites das frases, pode-se também usar correlação para detectar similaridade entre frases. Dividir a melodia em unidades menores permite uma análise da variação de alcance de altura entre frases, que é um dos recursos importantes para se criar variedade melódica. Outras características são a distância entre frases e o número de frases por melodia.

### 5.7.2 Originalidade

Vários métodos tentaram gerar melodias que não apenas seguem as regras mas que também são originais e diversas (Freitas & Guimarães 2011b). Algoritmos de composição algorítmica não podem ser considerados eficientes se eles sempre retornam uma mesma melodia considerada como ótima por seus critérios. Há também uma relação forte entre a originalidade de temas e sua popularidade (Simonton 1994). Os temas mais simples podem ser considerados muito banais enquanto os mais originais podem ser considerados complexos demais.

Apesar do conceito de originalidade em estética ser muito complexo, alguma originalidade de uma melodia em relação a várias outras composições pode ser inferida por suas probabilidades de transição em relação a uma melodia mediana (Simonton 1984).

O modelo baseado em expectativa (Eerola & North 2000) usa teorias de expectativa melódica para definir a complexidade de uma melodia. Outra maneira de medir a originalidade de uma melodia é monitorar os parâmetros melódicos, como quaisquer dos parâmetros citados neste capítulo. Ao se monitorar estes parâmetros, é possível detectar uma mudança de comportamento na melodia quando comparada aos parâmetros usados até certo ponto.

### 5.7.3 Análises de Segunda Ordem

Como música é uma arte contextual, faz sentido analisar a influência dos parâmetros uns nos outros. Por exemplo, é possível que uma sequência particular de alturas leve a determinadas expectativas de duração das notas. Neste caso, é importante ter uma análise de segunda, terceira ou  $n$ -ésima ordem, de modo que possamos descobrir estas interações entre parâmetros.

Estas análises usualmente não são simples e levam a uma tarefa que é também importante nas análises de primeira ordem: a importância da medida. Há tantas medidas que é importante distinguir quais são as mais importantes, independente de sua ordem.

Uma estratégia proposta para medir a importância de uma característica é analisar o seu desvio padrão (Towsey, Brown, Wright & Diederich 2001), já que baixos desvios padrões em um parâmetro indicam que este implica em uma restrição forte nas melodias.

## 5.8 Conclusão

Talvez o fato mais importante a ser notado no desenvolvimento de sistemas evolucionários para CA é que esta é uma tarefa na qual discussão no domínio relevante é estritamente necessária. Qualquer tentativa de se criar um sistema composicional sem discussão sobre teoria musical é desnecessária. Os algoritmos propostos neste trabalho mostram que é possível criar novas composições e misturar elementos conhecidos relacionados a novas músicas.

Como mencionado neste capítulo, as abordagens para formalizar uma função de avaliação para gerar melodias têm sido apenas parciais e a intenção deste trabalho é estender as opções de ideias nas quais compositores auxiliados por computadores podem estar trabalhando no momento. Apesar de que pode haver muito mais aspectos a serem pesquisados, este trabalho pode expandir as possibilidades de categorias características nas quais os compositores têm pensado.

Um procedimento de composição algorítmica não será capaz de manejar todas as medidas mencionadas aqui. Uma simples soma ponderada das funções objetivo ou a simples otimização multiobjetivo de todas as funções podem não ser as melhores respostas para este problema e estas podem ser na verdade soluções ingênuas. Estudos precisam incluir como colocar estas medidas juntas para avaliar melodias.

Para colocar as medidas juntas para sua utilização em um algoritmo genético, é natural que uma análise de quais são os aspectos mais importantes assim como uma análise dos limites práticos destas medidas devem ser desenvolvidos.

Assim, é necessário analisar várias melodias e definir quais são as restrições fortes para um gênero específico de música ou até para se desenvolver modelos teóricos. Naturalmente, as medidas mais importantes serão dependentes do estilo das melodias analisadas. Além de definir quais são as medidas mais importantes, um estudo de quais medidas se sobrepõem pode ajudar a manusear todos os aspectos mencionados aqui.

Quando tivermos uma boa solução para manejar todos os critérios mencionados neste capítulo, outro problema a se resolver é a diversidade e a originalidade das soluções geradas pelo algoritmo, já que não se quer que o algoritmo retorne sempre a mesma

melodia “ótima” (Freitas & Guimarães 2011b) ou ignore a originalidade necessária em obras primas (Simonton 1994). Uma vez que se consideram estas questões, podemos focar na aplicação de métodos estatísticos para extrair mais informação a ser usada na forma de valores de função objetivo.

É importante esclarecer que embora tenham-se mencionado medidas gerais de avaliação automática, há sempre dificuldades específicas em cada repertório que devem ser consideradas para uma boa análise das músicas. Todas as medidas apresentadas podem levar a diferentes modelos de acordo com os gêneros específicos de música.

Estes modelos, em alguns casos, podem até levar a problemas que são simples de se resolver em tempo polinomial. Neste caso, computação evolucionária poderia ser até mesmo inadequada para o problema de composição. Por outro lado, com toda a informação a ser considerada na geração de composições, é pouco provável que exista um bom modelo de composição que seja trivial.



# Capítulo 6

## Resultados

*“A música cria ordem a partir do caos: pois o ritmo impõe unanimidade sobre os divergentes, a melodia impõe continuidade sobre os disjuntos e a harmonia impõe compatibilidade sobre os incongruentes.”*

— Yehudi Menuhin, 1916–1999

### 6.1 Introdução

Nesta seção aplicamos nossa metodologia para tratamento de problemas com muitos objetivos a um problema de geração de frases melódicas que utilizam as medidas apresentadas no capítulo anterior. Para isto, as medidas são formuladas como funções objetivo e estas são consideradas individualmente para um conjunto de soluções geradas.

### 6.2 Base de Dados

Para os testes deste capítulo, a mesma base de dados do Capítulo 5 foi utilizada. Porém, esta base de dados foi manualmente dividida em frases melódicas, em um processo que resultou em 458 frases.

Estas frases foram utilizadas com as medidas apresentadas no Capítulo 5 de modo que servissem de referência para funções objetivo para geração de novas frases melódicas.

### 6.3 Geração de Novas Melodias

Para estudo do espaço de objetivos, é preciso definir um método que gere melodias a serem avaliadas. Para este estudo de comparação entre os objetivos, novas melodias semi-aleatórias baseadas em cadeias de Markov de ordem 0 são geradas. Abordagens mais aleatórias foram evitadas para se evitar que muitas melodias fossem mal avaliadas do ponto de vista de notas possíveis e suas ocorrências. Cadeias de Markov de ordem zero são equivalentes a seleção aleatória ponderada.

Em termos específicos, infere-se a probabilidade de cada tamanho  $|p_i|$  de frase  $p_i$  a partir do banco de dados de frases. Um número aleatório com esta distribuição é utilizado para definir o tamanho da nova frase a ser gerada. O tamanho da frase define o número de eventos existentes na frase, sendo cada evento composto de posição de início, duração e altura.

Para definição dos valores de eventos, é escolhida uma música  $s_1$  entre as 26 músicas do banco de dados. É estimada então a probabilidade de cada duração  $d_{ij}$  de nota  $j$  nesta música  $s_1$  considerando-se todas as suas frases  $i$ . Para cada evento é gerada uma duração aleatoriamente selecionada com esta distribuição.

O primeiro valor de posição de nota também é gerado com o mesmo procedimento e utilizando-se a música  $s_1$ . Uma posição de nota inicial é selecionada entre as posições de notas iniciais para a música  $s_1$ . As posições seguintes são então derivadas da posição de evento anterior acrescida da duração do evento respectivo.

Já para geração dos valores de alturas, uma nova música  $s_2$  é aleatoriamente selecionada. As probabilidades de cada altura nesta música  $s_2$  são estimadas e os valores de altura para cada evento são aleatoriamente selecionados com base nesta distribuição.

### 6.4 Funções objetivo

Definem-se medidas de análise de melodias como funções objetivo que têm a intenção de avaliar a capacidade das frases melódicas geradas de copiar o estilo das frases melódicas originais. Todas as frases foram consideradas na modelagem do problema e resultados da análise destas melodias formam a instância do problema. Todas estas informações ficam pré-processadas para serem utilizadas como função objetivo. Neste trabalho foram implementadas 27 funções objetivo que são analisadas simultaneamente.

### 6.4.1 Altura

Para os objetivos de altura, as músicas são todas transpostas para a tonalidade de dó maior ou seu relativo lá menor, para melodias maiores ou menores, respectivamente. Um teste estatístico ORDANOVA então é executado para se analisar se as distribuições de ocorrências de notas são diferentes nos dois tipos de melodias. No caso das melodias em nossa base de dados, o valor- $p < 0,05$  deste teste indica que há diferença significativa na distribuição das notas.

Como há diferença na distribuição de notas em melodias em tonalidades menor e maior, elas são consideradas separadamente. É estimada a probabilidade de uma frase ser maior ou menor e uma destas possibilidades é aleatoriamente selecionada para nossa função objetivo.

#### Função $f_1$

O primeiro objetivo  $f_1$  implementado é relacionado com a distribuição das notas. Para esta função, a ocorrência de cada nota  $i$  (ou classe de altura) possível entre as frases sendo consideradas (maiores ou menores) é registrada. Isso ocorre tanto para as frases  $source_i$  do banco de dados consideradas quanto para a frase  $sol_i$  sendo avaliada.

Assim, a probabilidade  $p\_source_i$  de uma nota ocorrer na base de dados pode ser obtida através da equação  $p\_source_i \leftarrow source_i / \sum_{i=1}^{12} source_i$ . Já a probabilidade  $p\_sol_i$  de uma nota ocorrer na solução pode ser obtida através da equação  $p\_sol_i \leftarrow sol_i / \sum_{i=1}^{12} sol_i$ .

Finalmente, a função  $f_1$  é definida pela Equação 6.1.

$$f_1 = \frac{\sum_{i=1}^{12} |p\_source_i - p\_sol_i|}{12} \quad (6.1)$$

A minimização desta função leva a probabilidades similares de notas entre as melodias sendo geradas e as melodias do banco de dados.

#### Função $f_2$

O segundo objetivo  $f_2$  implementado é bastante similar ao primeiro, porém relacionado com a distribuição das alturas. Para esta função, a ocorrência de cada altura  $i$  possível entre 120 possíveis nas frases sendo consideradas (maiores ou menores) é registrada. Isso ocorre tanto para as frases  $source_i$  do banco de dados consideradas quanto

para a frase  $sol_i$  sendo avaliada. Para facilidade de notação entre todas as funções, os nomes de algumas variáveis, especialmente as relacionadas a probabilidade, são reutilizados de uma função objetivo para outra.

Assim, a probabilidade  $p\_source_i$  de uma altura ocorrer na base de dados pode ser obtida através da equação  $p\_source_i \leftarrow source_i / \sum_{i=1}^{120} source_i$ . Já a probabilidade  $p\_sol_i$  de uma nota ocorrer na solução pode ser obtida através da equação  $p\_sol_i \leftarrow sol_i / \sum_{i=1}^{120} sol_i$ .

Finalmente, a função  $f_2$  é definida pela Equação 6.2.

$$f_2 = \frac{\sum_{i=1}^{120} |p\_source_i - p\_sol_i|}{120} \quad (6.2)$$

A minimização desta função leva a probabilidades similares de alturas entre as melodias sendo geradas e as melodias do banco de dados.

### Função $f_3$

Diferentemente dos objetivos já apresentados,  $f_3$  considera a variedade de notas em uma melodia como um valor alvo. Para tal, o número de notas utilizadas em cada frase  $i$  é dividido pelo número possível de notas (12) e este valor é armazenado em  $v\_source_i$ . Com a distribuição de valores em  $v\_source_i$ , um valor aleatório de variedade é aleatoriamente gerado e guardado como valor alvo  $t$ .

A variedade de notas na melodia sendo testada também é calculada e guardada como  $v\_sol$ . Finalmente, a função  $f_3$  é definida pela Equação 6.3.

$$f_3 = |t - v\_sol| \quad (6.3)$$

A minimização desta função leva uma variedade de notas na frase gerada que é similar a uma das frases do banco de dados.

### Função $f_4$

A função  $f_4$ , assim como a anterior, utiliza um valor alvo. Nesta função, a faixa utilizada em cada frase melódica para valores de altura é guardada em  $r\_source_i$  e um destes valores é utilizado como alvo  $t$ .

A faixa de alturas na melodia sendo avaliada também é calculada e guardada como  $r\_sol$ . Finalmente, a função  $f_4$  é definida pela Equação 6.4.

$$f_4 = |t - r\_sol| \quad (6.4)$$

A minimização desta função leva uma faixa utilizada de alturas na frase gerada que é similar a uma das frases do banco de dados.

### Função $f_5$

O objetivo  $f_5$  considera novamente a distribuição de notas nas frases, assim como no objetivo  $f_1$ . Porém, no objetivo  $f_5$ , as notas são ponderadas em relação a sua duração. A ideia por trás desta função é que notas que duram mais são mais percebidas.

Para esta função, a ocorrência de cada nota  $i$  possível entre 12 notas nas frases sendo consideradas (maiores ou menores) é multiplicada por sua duração e somada a uma variável  $source_i$ . Isso ocorre tanto para as frases  $source_i$  do banco de dados consideradas quanto para a frase  $sol_i$  sendo avaliada.

Assim, a probabilidade ponderada  $p\_source_i$  de uma nota estar soando entre as notas que soam durante as músicas pode ser obtida através da equação  $p\_source_i \leftarrow source_i / \sum_{i=1}^{12} source_i$ . Já a probabilidade  $p\_sol_i$  de um evento similar na solução sendo avaliada pode ser obtida através da equação  $p\_sol_i \leftarrow sol_i / \sum_{i=1}^{12} sol_i$ .

Finalmente, a função  $f_5$  é definida pela Equação 6.5.

$$f_5 = \frac{\sum_{i=1}^{12} |p\_source_i - p\_sol_i|}{12} \quad (6.5)$$

A minimização desta função leva a probabilidades similares de notas normalizadas por suas durações entre as melodias sendo geradas e as melodias do banco de dados.

## 6.4.2 Intervalo

### Função $f_6$

O objetivo  $f_6$  considera a distribuição de tamanho de intervalos nas frases. São considerados intervalos de -12 até 12, ou seja, 25 intervalos possíveis, considerando-se o intervalo zero. Intervalos mais distantes que 12 notas são contados como intervalos para a mesma classe altura em um tom mais próximo da nota original. Por exemplo, um intervalo -14 é contado como um intervalo  $-14 + 12 = -2$ .

Para esta função, cada intervalo  $i$  que ocorre em uma frase é guardado por uma variável  $source_i$ . Isso ocorre tanto para as frases  $source_i$  do banco de dados consideradas quanto para a frase  $sol_i$  sendo avaliada.

Assim, a probabilidade  $p\_source_i$  de um intervalo nas frases pode ser obtida através da equação  $p\_source_i \leftarrow source_i / \sum_{i=-12}^{12} source_i$ . Já a probabilidade  $p\_sol_i$  de um evento similar na solução sendo avaliada pode ser obtida através da equação  $p\_sol_i \leftarrow sol_i / \sum_{i=-12}^{12} sol_i$ .

Finalmente, a função  $f_6$  é definida pela Equação 6.6.

$$f_6 = \frac{\sum_{i=-12}^{12} |p\_source_i - p\_sol_i|}{25} \quad (6.6)$$

A minimização desta função leva a probabilidades similares intervalos entre as melodias sendo geradas e as melodias do banco de dados.

### Função $f_7$

O objetivo  $f_7$  considera também a distribuição de tamanho de intervalos nas frases. Porém, nesta função, a probabilidade dos intervalos é considerada em uma análise de segunda ordem na distribuição das notas. Para isto é utilizada uma matriz  $12 \times 12$  com a probabilidade de cada nota dada a nota anterior.

Para esta função, cada nota  $j$  que ocorre em uma frase após uma nota  $i$  é guardada por uma variável  $source_{ij}$ . Isso ocorre tanto para as frases  $source_{ij}$  do banco de dados consideradas quanto para a frase  $sol_{ij}$  sendo avaliada.

Assim, a probabilidade  $p\_source_{ij}$  de uma nota  $j$  após uma nota  $i$  nas frases pode ser obtida através da equação  $p\_source_{ij} \leftarrow source_{ij} / \sum_{i=1}^{12} \sum_{j=1}^{12} source_{ij}$ . Já a proba-

bilidade  $p\_sol_{ij}$  de um evento similar na solução sendo avaliada pode ser obtida através da equação  $p\_sol_{ij} \leftarrow sol_{ij} / \sum_{i=1}^{12} \sum_{j=1}^{12} sol_{ij}$ .

Finalmente, a função  $f_7$  é definida pela Equação 6.7.

$$f_7 = \frac{\sum_{i=1}^{12} \sum_{j=1}^{12} |p\_source_{ij} - p\_sol_{ij}|}{12.12} \quad (6.7)$$

A minimização desta função leva a probabilidades similares de notas em segunda ordem entre as melodias sendo geradas e as melodias do banco de dados.

### Função $f_8$

Similarmente à função  $f_3$ , a função  $f_8$  considera agora a variedade de intervalos em uma melodia como um valor alvo. Para tal, o número de diferentes intervalos utilizados em cada frase  $i$  é dividido pelo número possível de intervalos (com intervalos limitados entre -12 e 12, como na função  $f_6$ ) e este valor é armazenado em  $v\_source_i$ . Com a distribuição de valores em  $v\_source_i$ , um valor aleatório de variedade é aleatoriamente gerado e guardado como valor alvo  $t$ .

A variedade de intervalos na melodia sendo testada também é calculada e guardada como  $v\_sol$ . Finalmente, a função  $f_8$  é definida pela Equação 6.8.

$$f_8 = |t - v\_sol| \quad (6.8)$$

A minimização desta função leva uma variedade de intervalos na frase gerada que é similar à uma das frases do banco de dados.

## 6.4.3 Contorno

### Função $f_9$

O objetivo  $f_9$  considera a proporção de ascendentes e de descendentes em cada frase. Para isto uma frase  $t$  é aleatoriamente selecionada no banco de dados e sua proporção de ascendentes e descendentes é analisada.  $source\_c_1$  guarda o número de intervalos descendentes,  $source\_c_2$  guarda o número de intervalos uníssonos e  $source\_c_3$  guarda o número de intervalos ascendentes. Isso ocorre tanto para as frases  $source\_c_i$  do banco

de dados consideradas quanto para a frase  $sol\_c_i$  sendo avaliada.

Assim, a probabilidade  $p\_source_1$  de um intervalo descendente ocorrer pode ser obtida através da equação  $p\_source_1 \leftarrow source\_c_1 / \sum_{i=1}^3 source\_c_i$ . Analogamente, podemos encontrar as probabilidades de intervalos uníssonos e ascendentes através dos valores  $p\_source_2$  e  $p\_source_3$ , respectivamente.

Já a probabilidade  $p\_sol_i$  de um evento similar na solução sendo avaliada pode ser obtida através da equação  $p\_sol_i \leftarrow sol\_c_i / \sum_{i=1}^3 sol\_c_i$ .

Finalmente, a função  $f_9$  é definida pela Equação 6.9.

$$f_9 = \frac{\sum_{i=1}^3 |p\_source_i - p\_sol_i|}{3} \quad (6.9)$$

A minimização desta função leva uma variedade de tipos de intervalos na frase gerada que é similar à uma das frases do banco de dados.

### Função $f_{10}$

O objetivo  $f_{10}$  considera também a proporção de ascendentes e de descendentes em cada frase. Para isto uma frase  $t$  é aleatoriamente selecionada no banco de dados e sua proporção de ascendentes e descendentes é analisada. Porém, nesta função, os valores de segunda ordem são utilizados.

Assim, os valores 1, 2 e 3 ainda representam intervalos ascendentes, uníssonos e descendentes, enquanto os valores  $source_{ij}$  representam um intervalo do tipo  $i$  seguido por um intervalo do tipo  $j$ . Isso ocorre tanto para as frases  $source_{ij}$  do banco de dados consideradas quanto para a frase  $sol_{ij}$  sendo avaliada.

A probabilidade  $p\_source_{ij}$  de um intervalo do tipo  $i$  ocorrer e ser seguida por um intervalo do tipo  $j$  pode ser obtida através da equação  $p\_source_{ij} \leftarrow source_{ij} / \sum_{i=1}^3 \sum_{j=1}^3 source_{ij}$ . Já a probabilidade  $p\_sol_{ij}$  de um evento similar na solução sendo avaliada pode ser obtida através da equação  $p\_sol_{ij} \leftarrow sol_{ij} / \sum_{i=1}^3 \sum_{j=1}^3 sol_{ij}$ .

Finalmente, a função  $f_{10}$  é definida pela Equação 6.10.

$$f_{10} = \frac{\sum_{i=1}^3 \sum_{j=1}^3 |p\_source_{ij} - p\_sol_{ij}|}{3.3} \quad (6.10)$$

A minimização desta função leva uma variedade de tipos de intervalos em segunda ordem na frase gerada que é similar à uma das frases do banco de dados.

### Função $f_{11}$

Já o objetivo  $f_{11}$  considera apenas a proporção de intervalos uníssonos em relação a todos os intervalos. Para isto uma frase  $t$  é aleatoriamente selecionada no banco de dados e sua proporção de uníssonos é analisada.  $source\_u$  guarda o número de intervalos uníssonos enquanto  $source\_t$  guarda o número total de intervalos. Isso ocorre tanto para as frases  $source\_u$  do banco de dados consideradas quanto para a frase  $sol\_u$  sendo avaliada.

Assim, a probabilidade  $p\_source$  de um intervalo uníssonos ocorrer pode ser obtida através da equação  $p\_source \leftarrow source\_u/source\_t$ . Já a probabilidade  $p\_sol$  de um evento similar na solução sendo avaliada pode ser obtida através da equação  $p\_sol \leftarrow sol\_u/sol\_t$ .

Finalmente, a função  $f_{11}$  é definida pela Equação 6.11.

$$f_{11} = |p\_source - p\_sol| \quad (6.11)$$

A minimização desta função leva uma quantidade de intervalos uníssonos na frase gerada que é similar à uma das frases do banco de dados.

### Função $f_{12}$

O objetivo  $f_{12}$  considera a expectativa melódica em relação aos intervalos nas frases. Nesta função, valores de segunda ordem são utilizados. Os valores  $source_{i,j}$  representam um intervalo do tipo  $i$  seguido por um intervalo do tipo  $j$ , onde  $0 \leq i \leq 11$  e  $-11 \leq j \leq 11$ .

Para os valores de  $i$ , um intervalo é do tipo  $i$  se tem tamanho  $i$ . Já um intervalo de tamanho  $j$  é do tipo  $j$ , se ele segue a mesma direção de  $i$ , ou do tipo  $-j$ , se segue direção contrária ao intervalo  $i$ . Isso ocorre tanto para as frases  $source_{ij}$  do banco de

dados consideradas quanto para a frase  $sol_{ij}$  sendo avaliada.

A probabilidade  $p\_source_{ij}$  de um intervalo do tipo  $i$  ocorrer e ser seguido por um intervalo do tipo  $j$  pode ser obtida através da equação  $p\_source_{ij} \leftarrow source_{ij} / \sum_{i=0}^{11} \sum_{j=-11}^{11} source_{ij}$ . Já a probabilidade  $p\_sol_{ij}$  de um evento similar na solução sendo avaliada pode ser obtida através da equação  $p\_sol_{ij} \leftarrow sol_{ij} / \sum_{i=0}^{11} \sum_{j=-11}^{11} sol_{ij}$ .

Finalmente, a função  $f_{12}$  é definida pela Equação 6.12.

$$f_{12} = \frac{\sum_{i=0}^{11} \sum_{j=-11}^{11} |p\_source_{ij} - p\_sol_{ij}|}{12.23} \quad (6.12)$$

A minimização desta função leva uma proporção de expectativa intervállica em segunda ordem na frase gerada que é similar às frases do banco de dados.

#### 6.4.4 Duração

Para as medidas relacionadas a duração, todas as possibilidades de duração utilizadas na base de dados são listadas e guardadas em uma lista de tipos possíveis de duração para utilização nas melodias geradas. No caso deste experimento,  $d = 36$  valores de duração possíveis foram encontrados na base de dados.

##### Função $f_{13}$

O objetivo  $f_{13}$  considera as durações existentes nas frases de uma música  $t$  aleatoriamente escolhida no banco de dados. Assim,  $source_i$  guarda o número de notas com duração do tipo  $i$  entre os  $d$  tipos possíveis de duração. Isso ocorre tanto para as frases  $source_i$  do banco de dados consideradas quanto para a frase  $sol_i$  sendo avaliada.

Assim, a probabilidade  $p\_source_i$  de uma nota ter duração do tipo  $i$  pode ser obtida através da equação  $p\_source_i \leftarrow source_i / \sum_{i=1}^d source_i$ . Já a probabilidade  $p\_sol_i$  de um evento similar na solução sendo avaliada pode ser obtida através da equação  $p\_sol_i \leftarrow sol_i / \sum_{i=1}^d sol_i$ .

Finalmente, a função  $f_{13}$  é definida pela Equação 6.13.

$$f_{13} = \frac{\sum_{i=1}^d |p\_source_i - p\_sol_i|}{d} \quad (6.13)$$

A minimização desta função leva a uma distribuição de durações na frase gerada que é similar a uma das músicas do banco de dados.

### Função $f_{14}$

O objetivo  $f_{14}$ , similarmente ao objetivo  $f_{13}$ , considera as durações existentes nas frases da mesma música  $t$  escolhida no banco de dados. Porém, dados de segunda ordem são considerados, fazendo com que  $source_{ij}$  guarde o número de notas com duração do tipo  $i$  seguidas por notas do tipo  $j$  entre os  $d$  tipos possíveis de duração. Isso ocorre tanto para as frases  $source_{ij}$  do banco de dados consideradas quanto para a frase  $sol_{ij}$  sendo avaliada.

Assim, a probabilidade  $p_{source_{ij}}$  de uma nota ter duração do tipo  $i$  seguida por duração do tipo  $j$  pode ser obtida através da equação  $p_{source_{ij}} \leftarrow source_{ij} / \sum_{i=1}^d \sum_{j=1}^d source_{ij}$ . Já a probabilidade  $p_{sol_{ij}}$  de um evento similar na solução sendo avaliada pode ser obtida através da equação  $p_{sol_{ij}} \leftarrow sol_{ij} / \sum_{i=1}^d \sum_{j=1}^d sol_{ij}$ .

Finalmente, a função  $f_{14}$  é definida pela Equação 6.14.

$$f_{14} = \frac{\sum_{i=1}^d \sum_{j=1}^d |p_{source_{ij}} - p_{sol_{ij}}|}{d^2} \quad (6.14)$$

A minimização desta função leva a uma distribuição em segunda ordem de durações na frase gerada que é similar à da música  $t$  do banco de dados.

### Função $f_{15}$

O objetivo  $f_{15}$  considera a duração da primeira nota em uma frase  $t2$  das frases da mesma música  $t$  escolhida no banco de dados. Esta frase é aleatoriamente escolhida com a distribuição das primeiras notas na música escolhida.

O tipo de duração desta primeira nota é definido como um alvo  $t3$ . Dado que as possíveis durações estão em ordem crescente de duração e que a primeira nota da melodia sendo testada é  $n$ , a função  $f_{15}$  é definida pela Equação 6.15.

$$f_{15} = \frac{|t3 - n|}{d} \quad (6.15)$$

A minimização desta função leva a notas iniciais na frase gerada que são de duração similar às primeiras de uma frase  $t2$  na música  $t$  do banco de dados.

### Função $f_{16}$

Já o objetivo  $f_{16}$  considera a duração da última nota em uma frase  $t2$  das frases da mesma música  $t$  escolhida no banco de dados. Esta frase é aleatoriamente escolhida com a distribuição das últimas notas na música escolhida.

O tipo de duração desta última nota é definido como um alvo  $t3$ . Dado que as possíveis durações estão em ordem crescente de duração e que a última nota da melodia sendo testada é  $n$ , a função  $f_{16}$  é definida pela Equação 6.16.

$$f_{16} = \frac{|t3 - n|}{d} \quad (6.16)$$

A minimização desta função leva a notas finais na frase gerada que são de duração similar às últimas de uma frase  $t2$  na música  $t$  do banco de dados.

### Função $f_{17}$

Já o objetivo  $f_{17}$  considera a proporção rítmica em uma frase  $t2$  das frases da mesma música  $t$  escolhida no banco de dados. A proporção rítmica corresponde à duração da nota de maior duração  $d_{\max}$  dividida pela duração da nota de menor duração  $d_{\min}$ . O resultado desta divisão passa a ser o valor alvo  $t_{rp} \leftarrow d_{\max}/d_{\min}$ .

Assim, dado que temos também  $sol_{rp}$  como a proporção rítmica da melodia sendo avaliada, a função  $f_{17}$  é definida pela Equação 6.17.

$$f_{17} = |t_{rp} - sol_{rp}| \quad (6.17)$$

A minimização desta função leva a uma proporção entre a maior e menor notas na frase gerada que é similar à da frase  $t2$  na música  $t$  do banco de dados.

### Função $f_{18}$

Similarmente às funções  $f_3$  e  $f_8$ , a função  $f_{18}$  considera a variedade de durações na frase  $t2$  da música  $t$  como um valor alvo. Para tal, o número de diferentes durações

utilizados na frase é dividido pelo número possível de durações (com valores limitados à durações presentes no banco de dados) e este valor é armazenado em  $t_{dv}$ .

A variedade de intervalos na melodia sendo testada também é calculada e guardada como  $sol_{dv}$ . Finalmente, a função  $f_{18}$  é definida pela Equação 6.18.

$$f_{18} = |t_{dv} - sol_{dv}| \quad (6.18)$$

A minimização desta função leva a uma variedade de durações na frase gerada que é similar a uma das frases do banco de dados.

### Função $f_{19}$

A função  $f_{19}$  considera a quantidade de silêncio na frase  $t2$  da música  $t$  como um valor alvo. Para isto, a partir da primeira nota procuramos a diferença entre o tempo em que termina esta nota e se inicia a próxima nota e assim por diante com todas as outras nota. O somatório de todos este valores é guardado por uma variável  $t_s$ .

A quantidade de silêncio na melodia sendo testada também é calculada e guardada como  $sol_s$ . Finalmente, a função  $f_{19}$  é definida pela Equação 6.19.

$$f_{19} = |t_s - sol_s| \quad (6.19)$$

A minimização desta função leva a uma quantidade de silêncio na frase gerada que é similar à de uma das frases do banco de dados.

## 6.4.5 Posição de nota

Para estas funções, assim como para as funções de duração, as possíveis posições de nota dentro de um compasso são pré-processadas.

### Função $f_{20}$

A função  $f_{20}$  considera a posição das notas dentro de seu respectivo compasso em todas as frases da mesma música  $t$  selecionada para os objetivos de duração. A ocorrência de cada posição possível é atribuída a  $source_i$ , sendo que  $i$  é a  $i$ -ésima posição possível. Assim, a probabilidade de uma nota na posição possível  $i$  é  $p\_source_i \leftarrow source_i / \sum_{i=1}^p source_i$ ,

onde  $p$  é o número possível de posições de uma nota no compasso. O mesmo é feito para o cálculo de ocorrências na solução em  $sol_i$  e suas probabilidades  $p_{sol_i}$ .

Finalmente, a função  $f_{20}$  é definida pela Equação 6.20.

$$f_{20} = \frac{\sum_{i=1}^p |p_{source_i} - p_{sol_i}|}{p} \quad (6.20)$$

A minimização desta função leva a um posicionamento das notas no compasso que é similar aos de uma das frases do banco de dados.

### Função $f_{21}$

O objetivo  $f_{21}$  considera a posição no compasso da primeira nota em uma frase  $t2$  das frases da mesma música  $t$  escolhida no banco de dados. Esta frase é aleatoriamente escolhida com a distribuição das primeiras notas na música escolhida.

O tipo de posição desta primeira nota é definido como um alvo  $t3$ . Dado que as possíveis posições estão em ordem crescente de valor e que a primeira nota da melodia sendo testada está na posição  $n$ , a função  $f_{21}$  é definida pela Equação 6.21.

$$f_{15} = \frac{|t3 - n|}{p} \quad (6.21)$$

A minimização desta função leva a notas iniciais na frase gerada que estão em posição similar às primeiras de uma frase  $t2$  na música  $t$  do banco de dados.

### Função $f_{22}$

O objetivo  $f_{22}$  considera a posição no compasso da última nota em uma frase  $t2$  das frases da mesma música  $t$  escolhida no banco de dados. Esta frase é aleatoriamente escolhida com a distribuição das primeiras notas na música escolhida.

O tipo de posição desta última nota é definido como um alvo  $t3$ . Dado que as possíveis posições estão em ordem crescente de valor e que a última nota da melodia sendo testada está na posição  $n$ , a função  $f_{22}$  é definida pela Equação 6.22.

$$f_{22} = \frac{|t3 - n|}{p} \quad (6.22)$$

A minimização desta função leva a notas finais na frase gerada que estão em posição similar às últimas notas de uma frase  $t2$  na música  $t$  do banco de dados.

### 6.4.6 Padrões

#### Função $f_{23}$

O objetivo  $f_{23}$  considera autocorrelação entre as alturas da frase. Dada uma frase  $sol$  de tamanho  $n$ , são guardados todos os valores de correlação  $sol\_c_i$ , para  $i = 1, 2, 3...n$ .

O valor de correlação  $sol\_c_1$  é igual ao número de alturas iguais em posições iguais quando comparamos uma melodia com ela mesma dividido pelo número  $n$  de notas. O valor de correlação  $sol\_c_2$  é igual ao número de elementos iguais quando comparamos uma melodia com uma cópia dela mesma deslocada em um elemento dividido pelo número de notas. O valor de correlação  $sol\_c_i$  é igual ao número de elementos iguais quando comparamos uma melodia com uma cópia dela mesma deslocada em  $i - 1$  elementos dividido pelo número de notas.

O mesmo é feito para todas as frases do banco de dados e o resultado é guardado em uma variável  $db\_c_i$ . Sendo  $m$  o tamanho da maior frase no banco de dados, a função  $f_{23}$  é definida pela Equação 6.23.

$$f_{23} = \sum_{i=1}^{\min(n,m)} |sol\_c_i - db\_c_i| \quad (6.23)$$

A minimização desta função leva a uma relação de autocorrelação de alturas na frase gerada que é similar à correlação geral nas frases do banco de dados.

#### Função $f_{24}$

O objetivo  $f_{24}$  considera autocorrelação entre as durações da frase. Dada uma frase  $sol$  de tamanho  $n$ , são guardados todos os valores de correlação  $sol\_c_i$ , para  $i = 1, 2, 3...n$ .

O valor de correlação  $sol\_c_1$  é igual ao número de durações iguais em posições iguais

quando comparamos uma melodia com ela mesma dividido pelo número  $n$  de notas. O valor de correlação  $sol\_c_2$  é igual ao número de elementos iguais quando comparamos uma melodia com uma cópia dela mesma deslocada em um elemento dividido pelo número de notas. O valor de correlação  $sol\_c_i$  é igual ao número de durações iguais quando comparamos uma melodia com uma cópia dela mesma deslocada em  $i - 1$  elementos dividido pelo número de notas.

O mesmo é feito para todas as frases do banco de dados e o resultado é guardado em uma variável  $db\_c_i$ . Sendo  $m$  o tamanho da maior frase no banco de dados, a função  $f_{24}$  é definida pela Equação 6.24.

$$f_{24} = \sum_{i=1}^{\min(n,m)} |sol\_c_i - db\_c_i| \quad (6.24)$$

A minimização desta função leva a uma relação de autocorrelação de duração na frase gerada que é similar à correlação geral nas frases do banco de dados.

### Função $f_{25}$

O objetivo  $f_{25}$  considera o número de padrões de altura de um dado tamanho na frase. Dada uma frase  $sol$  de tamanho  $n$ , é guardada a quantidade de padrões de tamanho  $i$  encontradas na melodia em  $sol\_c_i$ , para  $i = 1, 2, 3...n$ .

Neste algoritmo, para cada valor  $i$ , percorremos a frase com um loop  $j$  de 1 até  $n - i$  que contém outro laço interno com  $k$  de  $j + 1$  até  $n - i + 1$ . A cada combinação de  $j$  e  $k$ , comparamos as frases de tamanho  $i$  se iniciando em  $j$  e  $k$  e se as frases forem iguais  $sol\_c_i$  é acrescido de um elemento.

O mesmo é feito para todas as frases do banco de dados e o resultado é guardado em uma variável  $db\_c_i$ . Sendo  $m$  o tamanho da maior frase no banco de dados, a função  $f_{25}$  é definida pela Equação 6.25.

$$f_{25} = \sum_{i=1}^{\min(n,m)} |sol\_c_i - db\_c_i| \quad (6.25)$$

A minimização desta função leva a uma relação de padrões de altura na frase gerada

que é similar à correlação geral nas frases do banco de dados.

### Função $f_{26}$

O objetivo  $f_{26}$  considera o número de padrões de duração de um dado tamanho na frase. Dada uma frase  $sol$  de tamanho  $n$ , é guardada a quantidade de padrões de tamanho  $i$  encontradas na melodia em  $sol\_c_i$ , para  $i = 1, 2, 3...n$ .

Neste algoritmo, para cada valor  $i$ , percorremos a frase com um loop  $j$  de 1 até  $n - i$  que contém outro laço interno com  $k$  de  $j + 1$  até  $n - i + 1$ . A cada combinação de  $j$  e  $k$ , comparamos as frases de tamanho  $i$  se iniciando em  $j$  e  $k$  e se as frases forem iguais  $sol\_c_i$  é acrescido de um elemento.

O mesmo é feito para todas as frases do banco de dados e o resultado é guardado em uma variável  $db\_c_i$ . Sendo  $m$  o tamanho da maior frase no banco de dados, a função  $f_{26}$  é definida pela Equação 6.26.

$$f_{26} = \sum_{i=1}^{\min(n,m)} |sol\_c_i - db\_c_i| \quad (6.26)$$

A minimização desta função leva a uma relação de padrões de duração na frase gerada que é similar à correlação geral nas frases do banco de dados.

### Função $f_{27}$

O objetivo  $f_{27}$  considera o tamanho da frase sendo gerada através de um valor alvo  $t$ , que é gerado aleatoriamente com a distribuição dos tamanhos de frase no banco de dados. Dado que a frase sendo testada tem tamanho  $n$ , a função  $f_{27}$  é definida pela Equação 6.27.

$$f_{27} = |t - n| \quad (6.27)$$

A minimização desta função leva a um tamanho de frase gerada que é similar ao de uma das frases do banco de dados.

## 6.5 Aplicação das Árvores de Agregação

Para uma análise da relação intrínseca entre os objetivos, foram geradas 100 instâncias de problema de geração de melodias. Cada instância foi gerada com diferentes combinações de valores alvo para os objetivos porém com a mesma base de dados. Para cada uma destas 100 instâncias, foram geradas 100 soluções através de uma heurística construtiva baseada em cadeias de Markov, como as descritas na Seção 6.3. Nesta heurística, uma cadeia de Markov de ordem 0 é utilizada. A probabilidade da quantidade de notas primeiramente é definida e frases melódicas são geradas de acordo com a distribuição de alturas, durações e posições no banco de dados.

Como diferentes instâncias podem ter relações diferentes entre objetivos, os resultados das instâncias foram agrupados, e uma frente de Pareto apenas foi analisada. Desta maneira, apenas objetivos que não podem ser conciliados em situação alguma são declarados conflituosos. A ocorrência harmonia na frente de Pareto indica então apenas que é possível obter soluções que sejam boas naquele agrupamento de objetivos e não que soluções boas em parte do agrupamento sempre serão boas em outra parte deste mesmo agrupamento.

Inicialmente, temos na Figura 6.1 o espaço de objetivos das soluções representado em coordenadas paralelas. A complexidade de interpretação dos resultados para esta aplicação prática mostra claramente a desvantagem deste modo de representação.

Assim, a Figura 6.2 apresenta a Árvore de Agregação resultante destas soluções. Para facilitar a leitura, os valores percentuais de harmonia estão representados em cada nó. Para cada valor de harmonia  $h_\beta$  em uma agregação  $\beta$  de um nó, os valores de conflito não paramétrico  $c_\beta$  são sempre  $100 - h_\beta$ .

Analisando a árvore podemos perceber que os dois objetivos mais harmônicos na primeira iteração são inicialmente  $f_{13}$  e  $f_{14}$ . Isso indica que não há muito conflito na relação de duração das notas quando consideramos este parâmetro em primeira ou segunda ordem. O mesmo pode se dizer dos objetivos  $f_1$  e  $f_5$ , que representam as distribuições de primeira ordem das classes de alturas, sendo que o objetivo  $f_5$  pondera estes valores por suas durações. Isto indica haver pouca relação entre durações e notas específicas nas melodias analisadas.

Já considerando a agregação destes objetivos em novas funções  $f_a = f_{14} + f_{13}$  e  $f_b = f_5 + f_1$ , o próximo objetivo mais harmônico é então a combinação destas duas  $f_c = f_a + f_b$ . Isso indica também ausência de conflito entre a distribuição de notas e a

duração destas notas em geral, considerando-se análises de primeira e segunda ordem.

A função de agregação pode ponderar cada um dos dois objetivos de qualquer maneira. Assim, um objetivo é multiplicado por um valor  $0 < \alpha < 1$  enquanto o outro é multiplicado por um valor  $1 - \alpha$ . Um valor  $\alpha = 0,5$  dá importância igual a cada um dos objetivos. Em alguns casos, pode ser conveniente simplesmente remover completamente um dos objetivos, se tivermos uma relação de precedência entre os objetivos. Neste caso, isto equivale a uma função de agregação no caso extremo onde  $\alpha = 0$ .

Neste trabalho, as funções agregadas são normalizadas com a mesma técnica não paramétrica apresentada para a redução de conflito e estes valores normalizados são somados. Estes valores são normalizados mais uma vez então para que fiquem em uma faixa de valores igual à dos objetivos não agrupados. Assim, agregamos objetivos não parametricamente, mantendo a estratégia básica do método sem dar preferência a nenhum deles.

Após este agrupamento de 4 objetivos, há mais um agrupamento de 4 objetivos quase tão harmoniosos quanto. Primeiro,  $f_{10}$  é agrupado com  $f_9$ , formando  $f_d = f_{10} + f_9$  e depois temos uma agregação  $f_e = f_6 + f_7$ . A primeira destas agregações mostra ausência de conflito entre a proporção de ascendentes e descendentes com a proporção de uníssonos nas frases, como é se esperar que não haja conflito entre estes objetivos. Já a agregação de  $f_e = f_6 + f_7$  mostra que não há conflito na distribuição de intervalos se compararmos os resultados de primeira e segunda ordem. Estas duas funções  $f_d$  e  $f_e$  são agregadas em um novo objetivo  $f_f = f_e + f_d$  com ainda mais harmonia do que as agregações iniciais por não haver conflito entre os tipos de intervalos e a quantidade de ascendentes, uníssonos e descendentes.

Até o momento temos dois agrupamentos isolados de 4 objetivos cada um:  $f_c$  com funções que consideram a distribuição de notas com suas durações e  $f_f$  com funções que consideram tamanhos e tipos de intervalos. A próxima agregação feita é então  $f_g = f_c + f_f$ . Isso indica que é possível se obter soluções que respeitem simultaneamente as distribuições desejadas de notas, duração e intervalos. De fato, estes são objetivos que podem ser facilmente percebidos como subproblemas que podem ser resolvidos separadamente. Porém, como a análise é feita sobre elementos do Pareto, isto não indica que uma solução com bons valores em um dos objetivos terá inerentemente bons valores em outro. Por este motivo, é importante a utilização de funções de agregação moderadas que aproveitem a harmonia para otimizar mais objetivos simultaneamente em vez de descartar objetivos e desnecessariamente abrir espaço para soluções arbitrariamente

ruins em alguns objetivos sem ganho computacional.

A próxima dupla mais harmônica de objetivos não agregados é  $f_h = f_{20} + f_{21}$  que são, respectivamente, a distribuição da posição das notas no compasso e a posição da primeira nota no compasso. A sobreposição dos dois objetivos ainda é clara mas há neste objetivo uma harmonia um pouco menor, que é justificável pois o objetivo  $f_{21}$  escolhe uma frase melódica aleatória da música como valor alvo, o que pode tornar a combinação dos elementos mais difícil em alguns casos.

Agrupamos até o momento dois grandes blocos  $f_g$  e  $f_h$  com 8 e 2 objetivos, respectivamente. Estes dois blocos são então agrupados em um bloco  $f_i = f_h + f_g$ , indicando a possibilidade de se encontrar soluções que sejam boas em todos estes objetivos, que são relativamente independentes para os objetivos definidos, que consideram suas distribuições. As próximas agregações são feitas em  $f_j = f_{16} + f_{17}$  e  $f_k = f_{23} + f_{24}$ , respectivamente, com aproximadamente a mesma harmonia em cada uma delas.  $f_j$  agrupa a duração da última nota em uma frase com a proporção rítmica e  $f_k$  agrupa autocorrelação de alturas e de durações nas frases. Esta é uma sobreposição de objetivos que já é mais difícil de se perceber. Em seguida, estes dois objetivos são agrupados em  $f_l = f_j + f_k$ .

A função  $f_i$  com várias relações de distribuição é então agrupada com  $f_2$ , que corresponde à distribuição das alturas. O novo objetivo  $f_m = f_i + f_2$  é então agrupado com o outro grande agrupamento  $f_l$  em  $f_n = f_m + f_l$ . Este agrupamento nos deixa com apenas um objetivo composto e outros 12 objetivos do problema original. Neste ponto, temos apenas opções de agregação com menos de 50% de harmonia.

O próximo agrupamento feito é então  $f_o = f_n + f_{11}$ , que agrupa os objetivos que já temos com o objetivo relativo à proporção de intervalo uníssonos. Deste ponto em diante, temos as seguintes agregações:

- $f_p = f_{26} + f_{25}$
- $f_q = f_o + f_p$
- $f_r = f_{27} + f_{18}$
- $f_s = f_r + f_q$
- $f_t = f_s + f_4$
- $f_u = f_t + f_{15}$

- $f_v = f_u + f_{19}$
- $f_w = f_8 + f_3$
- $f_x = f_w + f_v$
- $f_y = f_x + f_{22}$
- $f_z = f_y + f_{12}$

A cada vez que dois objetivos são agregados, ocorre uma perda em representação que é minimizada ao se escolher o objetivo com maior harmonia. Se considerarmos que os valores de conflito não paramétrico são sempre  $1 - h_\beta$ , sendo  $h_\beta$  o valor de harmonia na agregação corrente, podemos utilizar este valor de conflito como uma estimativa da perda de representação que ocorre a cada agregação. A Figura 6.3 mostra a diferença de perda entre as primeiras e últimas reduções.

Podemos então somar estas medidas de perda a cada iteração de agregação para medirmos o quanto foi perdido até o momento para um dado número de agregações, o que é mostrado na Figura 6.4.

Assim, o método de Árvores de Agregação permite que se escolha o número de objetivos a serem agregados em relação à sua perda associada. Pode-se escolher também um limite de perda para a redução de objetivos.

Durante todo o processo de redução destes objetivos, um grande grupo de objetivos sempre esteve presente agrupando grupos menores de no máximo 4 objetivos. Isto indica que os objetivos analisados neste problema são conflitantes entre si mas não têm tendência a uma divisão grande em dois ou mais grupos claros de objetivos que são muito conflitantes entre si.

Se fizermos o processo contrário ao de construção da árvore, ou seja, partir de um objetivo  $f_z$  que corresponde à agregação de todos os objetivos e separar da agregação principal os objetivos mais conflitantes do resto, os objetivos conflitantes desagrupados seriam respectivamente  $f_{12}$ ,  $f_{22}$ ,  $f_8$ ,  $f_3$ ,  $f_{19}$ ,  $f_{15}$ ,  $f_4$ ,  $f_{27}$ ,  $f_{18}$ ,  $f_{26}$ ,  $f_{25}$ ,  $f_{26}$ ,  $f_{25}$ ,  $f_{11}$ ,  $f_{24}$ ,  $f_{23}$ ,  $f_{17}$ ,  $f_{16}$ ,  $f_2$ ,  $f_{21}$ ,  $f_{20}$ ,  $f_7$ ,  $f_6$ ,  $f_{10}$ ,  $f_9$ ,  $f_{14}$ ,  $f_{13}$ ,  $f_5$  e  $f_1$ . Em relação ao primeiro objetivo a ser desagregado por ser conflitante com o resto do grupo,  $f_{12}$  contém uma medida bastante diferente dos outros objetivos por se tratar de expectativa melódica.

Em relação aos valores absolutos de função objetivo, a Figura 6.5 apresenta o grafo polar da árvore de agregação apresentada. Neste grafo podemos ver a relação entre os

objetivos que acabaram sendo agrupados em  $f_n$ , que vão de  $f_{21}$  a  $f_{23}$ . Neste grupo de objetivos, os valores de função objetivo da frente de Pareto estão espalhados entre os valores máximo e mínimo, enquanto as outras funções conseguem atingir valores baixos mais facilmente. Porém, o agrupamento de funções em cores no grafo polar não mostra uma relação muito clara entre os objetivos devido ao grande número de objetivos em relação ao número de grupos de cores. Assim, grupos de uma cor acabam na maior parte das vezes tendo valores baixos e altos para o mesmo objetivo, o que faz com que apenas a árvore consiga representar melhor a relação entre os objetivos.

Como a árvore apresentada foi gerada com as soluções não dominadas, harmonia entre dois objetivos representa que é possível obter valores que são bons nos dois objetivos porém não indica que soluções com bons valores em um objetivo terão bons valores no outro. Deste modo, a harmonia não implica em causalidade entre os valores de função objetivo.

Se fizermos uma nova árvore com todos os valores de função objetivo, incluindo os valores dominados, teremos uma árvore na qual harmonia entre dois objetivos indica que bons valores em um objetivo estão correlacionados a bons valores no outro. Para a árvore apresentada, porém, 9988 de 10000 soluções, ou 99,88% delas, eram não dominadas. Isto ocorre devido ao grande número de funções objetivo sendo consideradas. Assim, as duas árvores (não incluindo e incluindo os valores dominados) têm resultados muito similares, onde em nenhum dos nós da árvore há uma diferença de harmonia maior do que 0,1% em relação a outra árvore.

Isto mostra características importantes de problemas com muitos objetivos. Quando há muitos objetivos e a maior parte das soluções se tornam não dominadas, os resultados da comparação entre apenas dois objetivos se torna cada vez mais próxima da comparação entre estes dois objetivos para soluções aleatórias. No caso da árvore apresentada,  $9988/10000 = 99,88\%$  das soluções são não dominadas e são consideradas em qualquer agrupamento.

Por outro lado, se apenas os objetivos  $f_{14}$  e  $f_{13}$  fossem considerados no problema, o seu agrupamento consideraria apenas  $9/10000 = 0,09\%$  das soluções na análise de conflito e sua junção teria uma harmonia de 100%, valor justificável pois os dois objetivos não são conflitantes de modo algum em sua definição matemática. O maior número de pontos considerados ocorre pois outros objetivos fazem com que soluções conflitantes entre dois objetivos específicos entrem no Pareto por serem pontos bons em outras combinações de objetivos. Assim, a harmonia entre os dois objetivos passa a ser menor devido ao número

de objetivos, processo que pode ser apenas reduzido por uma relação intrínseca entre os objetivos de modo que bons valores em um objetivo causem necessariamente valores bons em outro objetivo, o que acontece em funções objetivo que consideram critérios semelhantes como  $f_{14}$  e  $f_{15}$ . Isto aproxima então a Árvore de Agregação que considera apenas soluções não dominadas à árvore que considera todas as soluções à medida que o número de objetivos aumenta, o que torna mais rara a obtenção de valores altos de harmonia.

Como exemplo de resultados do conjunto não dominado, fazemos então a agregação de todos os objetivos em apenas 4 objetivos agregados mais conflitantes através da informação dada pela Árvore de Agregação. Assim, estes 4 objetivos são  $f_{12}$ ,  $f_{22}$ ,  $f_w = f_8 + f_3$  e  $f_v$ , com a agregação de todos os outros objetivos.

Na Figura 6.6 temos a solução com melhor valor de função objetivo na função  $f_{12}$ , relacionada com a expectativa melódica inferida do conjunto de dados. Como apenas a medida de expectativa melódica é considerada na avaliação desta soluções, pode-se perceber que a relação entre os valores rítmicos são arbitrários.

Na Figura 6.7 temos a solução com melhor valor de função objetivo na função  $f_{22}$ , relacionada com a posição no compasso da última nota. A solução obteve  $f_{22} = 0$  pois a nota ocorreu na posição esperada. Mais uma vez, apenas um objetivo é considerado e neste caso atender o objetivo individualmente é trivial pois é um objetivo baseado em valor alvo. Este objetivo terá seu melhor valor enquanto a posição da última nota for a definida como valor alvo, não importando mais uma vez se os outros critérios sejam atendidos.

Na Figura 6.8 temos a solução com melhor valor de função objetivo na função  $f_w = f_8 + f_3$ , relacionada com a variedade de notas na frases em relação a um valor alvo e a variedade de intervalos também com um valor alvo. A solução obteve tanto  $f_3 = 0$  como  $f_8 = 0$  pois os alvos foram atingidos. Ambos os valores alvo foram atingidos mais uma vez.

Na Figura 6.9 temos a solução com melhor valor de função objetivo na função  $f_v$ , relacionada com os outros objetivos agregados com mesmo peso. Neste caso, todas as funções são consideradas porém não é dada preferência a nenhum dos objetivos.

## 6.6 Conclusão

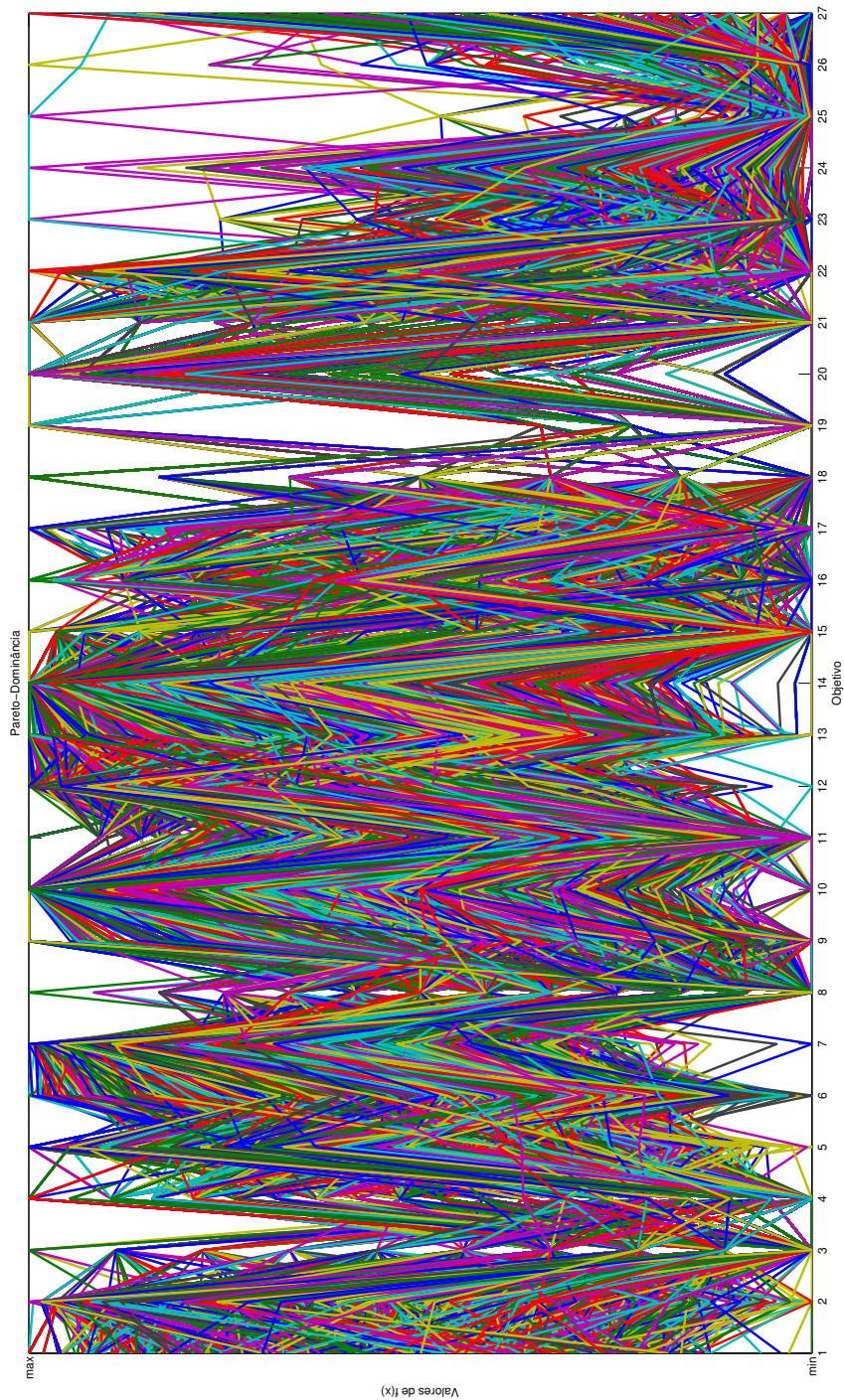
Neste Capítulo, são apresentadas 27 formas de se analisar melodias em forma de função objetivo. Estas funções são então utilizadas para a análise de soluções geradas através de cadeias de Markov para um problema de CA.

Em um exemplo da aplicação das Árvores de Agregação para a redução do problema a 4 objetivos pode-se perceber um grande grupo de 23 objetivos que foi agrupado ( $f_v$ ) a parte de um outro grupo com outros 4 objetivos menos harmoniosos com o grande grupo.

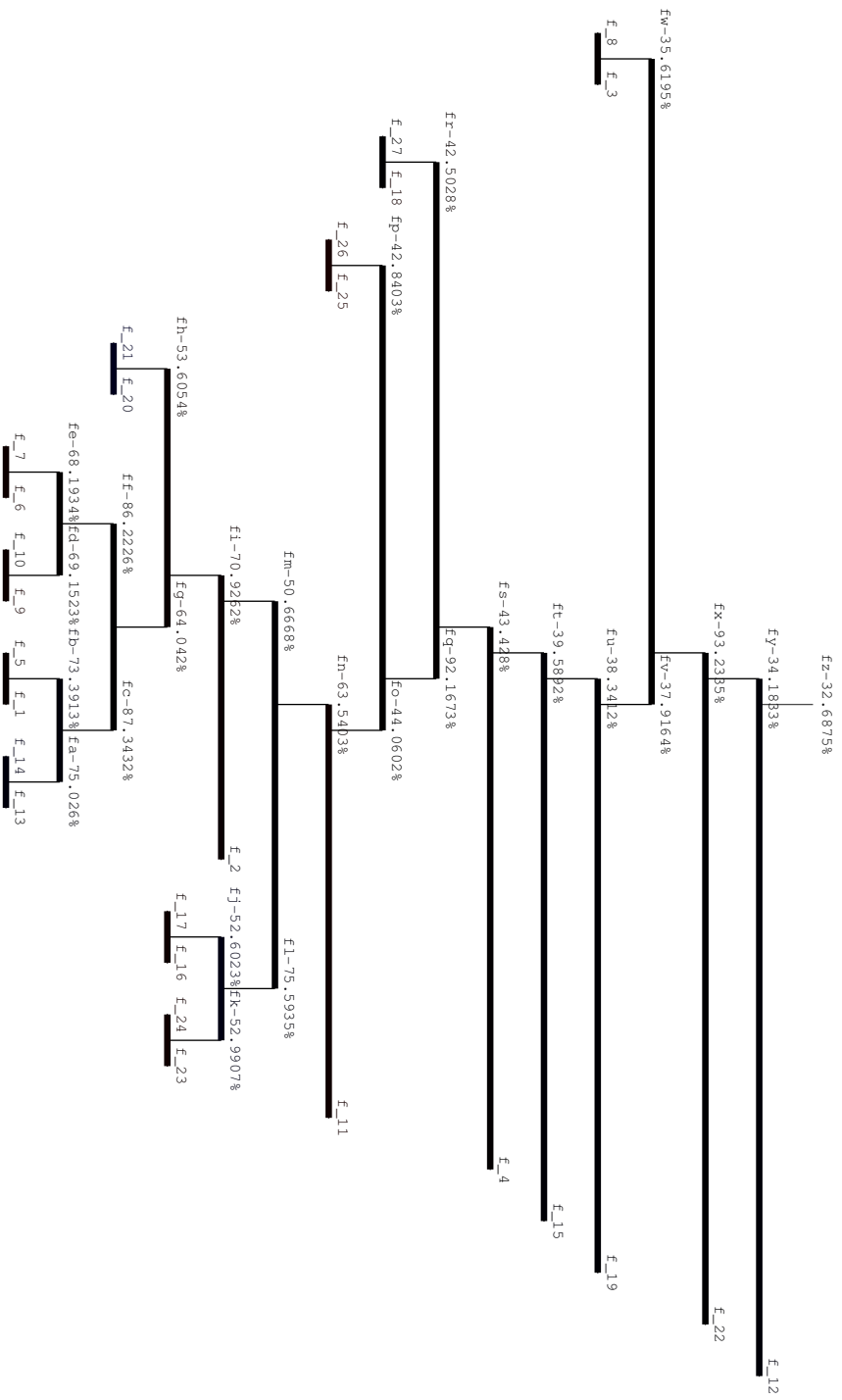
Soluções que não consideraram o grande grupo em  $f_v$  em sua avaliação frequentemente tiveram aspectos arbitrários em relação duração e posição das notas por não conterem objetivos relacionados em suas agregações. Contudo, a Árvore de Agregação resultante demonstra qual seria a melhor combinação de agregações para qualquer número final de objetivos definido pelo usuário.

Em uma segunda análise, a perda envolvida em cada agregação é apresentada em um outro gráfico. Isto é importante pois em um ambiente prático o usuário pode decidir que o número de objetivos reduzidos será definido em função de um limiar máximo de perda por agregação.

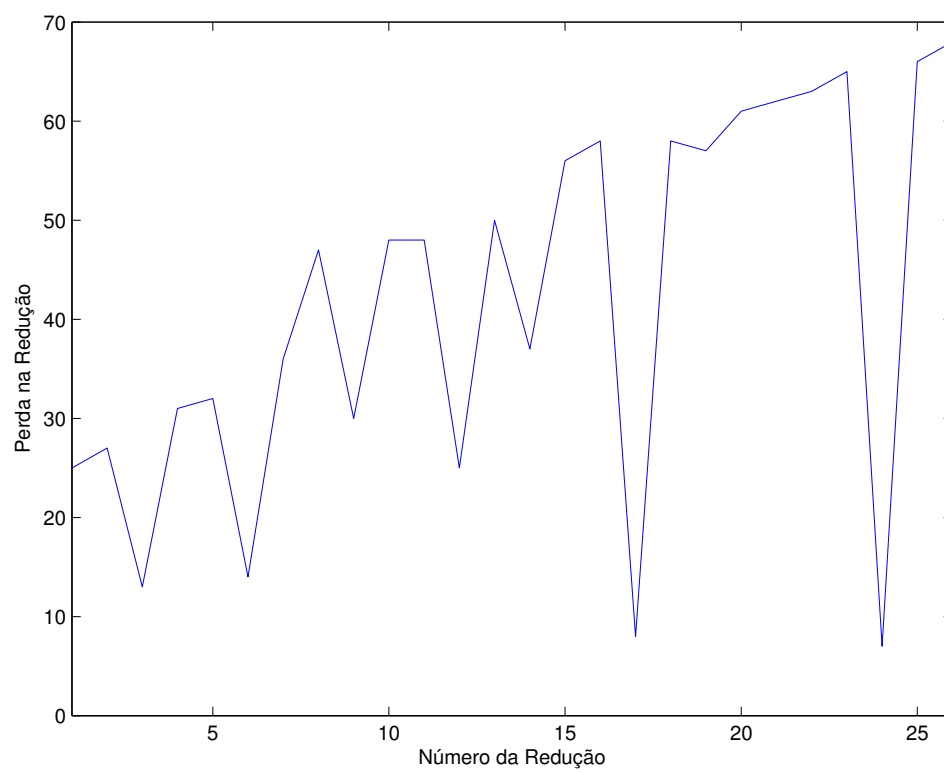
A Árvore de Agregação resultante também mostra como o processo de composição melódica é complexo, como problema com muitos objetivos que é. Por isto, é interessante como ela consegue representar a relação entre os objetivos conflitantes.



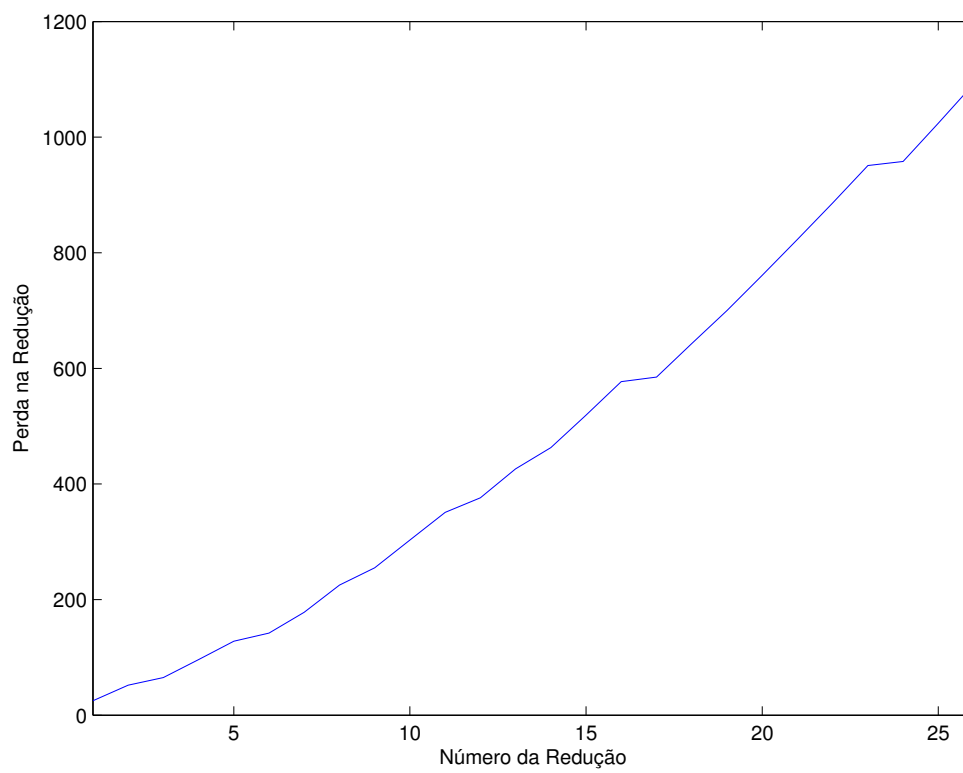
**Figura 6.1:** Coordenadas Paralelas para o Problema de Geração de Frases Melódicas



**Figura 6.2:** Árvore de Agregação com Valores de Harmonia nos Nós. Nesta árvore houve conflito muito concentrado em regiões específicas extremas do Pareto para nenhum agrupamento de objetivos. Por este motivo, todos os nós estão representados em preto.



**Figura 6.3:** Perda associada a cada agregação. A perda associada a cada agregação está medida neste gráfico como quantidade de conflito. Cada valor pode ir então de 0 a 100.



**Figura 6.4:** Perda acumulada até cada número de agregações

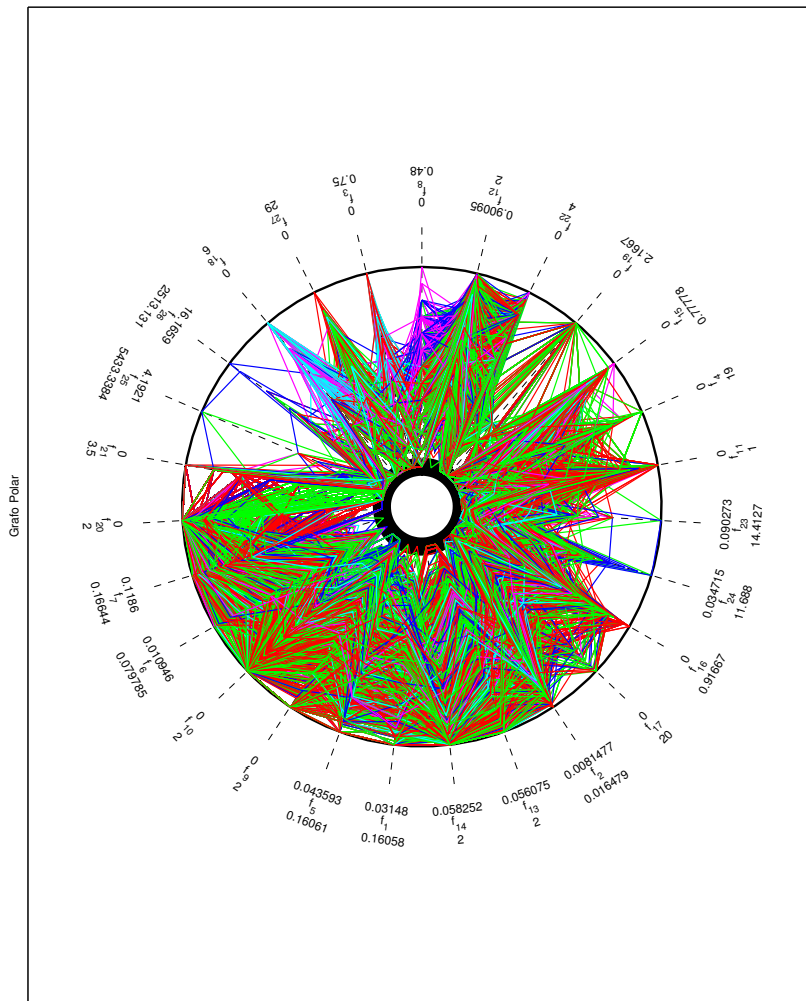


Figura 6.5: Gráfico Polar com a soluções não dominadas para o problema proposto.



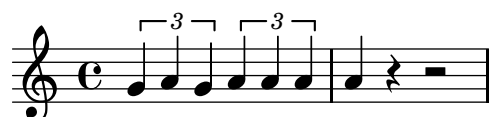
**Figura 6.6:** Solução com melhor valor de função objetivo na função  $f_{12}$ , correspondente a expectativa melódica.



**Figura 6.7:** Solução com melhor valor de função objetivo na função  $f_{22}$ , correspondente à posição no compasso da última nota.



**Figura 6.8:** Solução com melhor valor de função objetivo na função  $f_w$ , combinação dos objetivos  $f_8$  e  $f_3$ , relacionados com a variedade de notas e de intervalos na frase.



**Figura 6.9:** Solução com melhor valor de função objetivo na função  $f_v$ , grande grupo com agregação de 23 objetivos.

# Capítulo 7

## Conclusões e Trabalhos Futuros

*“Quando examino a mim mesmo e meus métodos de pensamento, chego à conclusão de que o dom da fantasia me significou mais do que qualquer talento para pensamento abstrato e positivo.”*

— Albert Einstein , 1879–1955

### 7.1 Conclusões

#### 7.1.1 Composição Algorítmica

Os algoritmos podem ser utilizados para composições originais ou para cópia de estilo. A maior parte dos trabalhos, porém, foca em composições que sigam determinados estilos (Nierhaus 2010). Estilos genuínos de música em composição algorítmica não podem ser definidos precisamente já que podem haver imitações de estilo sobre o primeiro estilo original. Isto pode ocorrer quando os métodos são utilizados para realizar o mesmo conceito estético em vários trabalhos.

Além disto, a integração de algoritmos comuns para composição em um novo algoritmo, pode ser vista também como imitação de estilo em um nível estrutural. Fatores culturais também podem influenciar estas definições pois os conceitos podem depender do período e local de criação do conceito e abordagem histórica para classificar as regras de composição como novas ou imitação. Assim, imitação de estilos e obras genuínas não podem ser separadas tão facilmente.

Por outro lado, pode-se traçar uma divisória em relação à motivação inicial dos métodos aplicados. Neste sentido, os algoritmos apresentados neste trabalho têm a clara intenção de seguir um estilo determinado de música. Não é necessário que todas as regras estejam formuladas no algoritmo, até mesmo porque os métodos podem não ser baseados em conhecimento explícito. Porém, o objetivo embutido no algoritmo de se atingir obras no estilo determinado é o fator determinante.

Na maior parte dos algoritmos, apenas altura tonal, duração e dinâmica são usualmente tratadas. Isso pode ocorrer pela dominância do protocolo MIDI. Os termos “composição algorítmica” e “composição assistida por computador” têm usualmente sido utilizados como sinônimos na literatura. Se os princípios do algoritmo são aplicados com ou sem a ajuda do computador é um fato de pouca importância.

Por outro lado, a extensão em que os algoritmos auxiliam o compositor é um fator importante. Os modelos algorítmicos podem ser utilizados para estender as possibilidades do compositor mais do que oferecer composições prontas. Os algoritmos de geração e teste são um bom exemplo desta expansão de possibilidades, mesmo que as ideias não sejam utilizadas.

Outra possibilidade é utilizar o algoritmo na posição de substituir o compositor. Isso questiona os papéis sociais do compositor, do intérprete e do ouvinte. Além disso, estratégias que tentam definir a validade estética de estruturas de composição se utilizando de áreas além da música, como a psicologia, também podem entrar nesta categoria. Porém, como a arte é uma tarefa amplamente humana em sentido estrito, o papel do compositor é passado sempre para o criador dos algoritmos de auxílio ao compositor. Assim, estes trabalhos podem não ter a capacidade de substituir os compositores mas podem sim alterar profundamente seus papéis no processo. Deste modo, a aplicação do algoritmo representa apenas um método menos usual de composição, enquanto nem validade axiomática nem a substituição do compositor são inerentes da abordagem, assim como no caso de qualquer outra abordagem para trazer à vida uma obra de arte.

Além disto, uma boa motivação para sistemas de composição algorítmica é a capacidade generativa de computadores, que podem produzir material novo continuamente em um certo contexto. Indústrias que se beneficiariam disto são aquelas em que a geração de material contextualizado e interativo seja propícia, como no caso dos jogos eletrônicos. Um exemplo histórico de métodos de geração contínua é o jogo musical de dados (Chuang 1995).

Alguns conceitos algorítmicos ocorrem em vários gêneros musicais. Técnicas de con-

traponto são exemplos clássicos disto. Apesar destes métodos não serem entendidos como composição algorítmica, talvez pela falta de necessidade de se ver o problema por este viés, as regras são formuladas nas técnicas como condições de restrição. Por este mesmo motivo, estas técnicas estão usualmente inseridas em vários algoritmos para cópia de estilo.

Além de toda esta discussão, há também métodos de composição algorítmica que servem fundamentalmente a interesses comerciais. O jogo de dados musicais foi comercializado mais tarde com o nome de *Kaleidacousticon System*, em uma versão adaptada para cartas. Sistemas como BandInABox também foram desenvolvidos com a intenção de atingir um grande público. O sistema produz várias peças musicais de diferentes gêneros de acordo com progressões harmônicas dadas pelo usuário.

Música gerada algoritmicamente pode ter vários interesses comerciais como música funcional para lojas de departamento, tons de espera em sistemas de telefonia, toques para celulares e outras atividades do tipo podem ser ramos atrativos de negócio. Soluções baseadas em software podem retornar composições agradáveis como se estivessem ordenadas em linha e pudessem ser acessadas sem educação musical e acima de tudo, sem pagamento de direitos autorais.

Em nosso trabalho passado sobre geração de melodias de maneira livre de aptidão, apesar da influência da população inicial no resultado final das melodias, o modo como o algoritmo é controlado pode levar a resultados completamente novos já que regras estritas (como uma função de aptidão usual) não são aplicadas no processo evolucionário.

Ainda no gerador de melodias, mudanças através das gerações têm mais importância do que a procura por um resultado particular que satisfaz melhor um conjunto específico de regras. Assim, é mais significativo ter operadores cientes musicalmente robustos que funções de aptidão baseadas em regras que podem ao contrário remover alguma criatividade dos resultados. Neste, uma matriz de *takeover* também é abordagem desenvolvida para encontrar um bom conjunto de soluções relacionadas entre si.

As melodias, como um resultado do processo criativo, não podem ser criadas baseadas apenas em regras musicais simples, o sistema deve ter alguma experiência implícita sobre o que compositores humanos fazem. Mais do que isto, um processo que cria melodias com menos regras pode ser muito mais fácil de ser implementado e usado quando comparado com algoritmos que usam o artifício de um mentor humano para avaliar a população inteira. Este algoritmo pode não apenas ser útil para desenvolver novas ideias mas também para estender e mesclar ideias previamente criadas, que podem ser competitivas

com ideias geradas por humanos.

Melodias, como resultado do processo criativo que são, podem ser criadas sem base em velhas regras musicais que limitam a criatividade da composição. Além disto, o processo de criação de melodias com menos regras e avaliação automática pode funcionar de maneira mais fácil em que avaliação interativa devido ao problema do gargalo de avaliação da aptidão que é automaticamente resolvido e acaba com o *trade-off* entre tamanho da amostra e qualidade da avaliação.

Já para o algoritmo biobjetivo também desenvolvido como trabalho do autor, os experimentos com duas funções de aptidão para avaliar as harmonias tornaram claros aspectos importantes deste tipo de algoritmo. A abordagem possibilita que o algoritmo ignore preferências estritamente específicas do usuário e gere um conjunto de soluções factíveis.

O algoritmo convergiu para várias soluções interessantes que são resultado da flexibilidade embutida no sistema biobjetivo.

Modelos baseados em regras podem ser mais eficientes em alguns casos específicos de harmonização, especialmente quando comondo em um estilo específico. Não obstante, sistemas evolucionários têm claras vantagens de flexibilidade e possibilidades de novas soluções criativas. A abordagem multiobjetivo demonstra como estas possibilidades de novas soluções criativas podem ser utilizadas.

### 7.1.2 Árvores de Agregação

Uma vantagem deste método é que é mais fácil visualizar a relação entre os objetivos pois a cada passo podemos ver o par de objetivos que são mais harmoniosos e a quantidade de conflito entre eles. A estratégia de agrupar objetivos um a um também torna fácil a interpretação de resultados e quais seriam as melhores escolhas de redução de objetivos dado um número desejado de objetivos finais. A medida de conflito também diz ao tomador de decisão quanta penalização está envolvida na redução destes objetivos.

Talvez a feição mais importante deste método é que ele não é restrito a relações lineares entre os objetivos e não é nem mesmo dependente de qualquer relação não-linear específica entre os objetivos pois apenas sua harmonia é usada como critério para decidir quais objetivos reduzir. Esta é uma feição conveniente pois formas não-paramétricas de conflito podem levar o tomador de decisão a pensar que há harmonia onde pode não haver. A característica de um objetivo poder ser reduzido por vez torna conveniente

para o tomador de decisão escolher quando parar de agrupar os objetivos. As medidas de conflito podem tornar esta decisão mais fácil.

O método proposto é também de fácil implementação e tem custo computacional baixo no contexto de problemas multiobjetivo. O custo computacional é baixo o suficiente para permitir que seja embutido em um MOEA com uma interface para que o tomador de decisões escolha em tempo real quantos objetivos devem ser reduzidos.

Quando o algoritmo não está embutido em um MOEA, uma discussão importante é se usaremos apenas soluções não dominadas para analisar a relação entre objetivos, qualquer conjunto de soluções (dominadas e não dominadas), ou soluções limitadas à área de preferência. Não há melhor resposta para esta questão mas sabe-se que utilizar soluções não-dominadas mostrariam apenas que os objetivos podem ser otimizados de uma maneira na qual melhoramos ambas as soluções. Usar qualquer conjunto de soluções poderia mostrar que há harmonia inerente entre os objetivos. De qualquer maneira, à medida que o número de objetivos cresce, esta diferença tende a desaparecer pois a maior parte das soluções serão não-dominadas.

### 7.1.3 Medidas de Avaliação de Melodias

Em relação às medidas desenvolvidas nesta tese, a avaliação do material gerado é um aspecto fundamental das aplicações de composição algorítmica (Nierhaus 2010). Algumas abordagens avaliam os algoritmos para a geração de certas estruturas e outras abordagens avaliam as saídas dos algoritmos. Neste trabalho, apresentamos métodos que avaliavam as melodias tanto implicitamente através dos operadores genéticos quanto explicitamente através de análise da saída.

A avaliação pode ser feita de forma algorítmica ou por um usuário, o que faz a avaliação difícil em termos práticos. Por isto, vários artigos tem até mesmo de descrever a qualidade dos resultados obtidos quando a intenção do algoritmo é de se gerar composições genuínas. Quando composições são avaliadas por um grupo de indivíduos, pode ser difícil haver coerência de julgamento para levar a um estilo de música intermediário que agrade uma grande audiência.

Além disto, o gosto de uma audiência, comumente utilizada em alguns trabalhos, pode ser uma medida de aceitação de uma composição, mas não é exatamente uma boa medida para a qualidade de uma obra de arte. Com base no teste de Turing, houve também a utilização de avaliação feita por humanos em uma série de experimentos para distinguir entre as composições geradas por máquina ou por humanos.

Neste trabalho, medidas de avaliação são basicamente focadas na avaliação das saídas do algoritmo, um aspecto importante e negligenciado em várias abordagens da área. Um erro grave é a aceitação das estruturas geradas por redes neurais ou algoritmos genéticos sem uma avaliação criteriosa dos resultados. Não se pode julgar que há sucesso simplesmente pela geração de saídas. Com este trabalho, podemos estender consideravelmente as possibilidades de ideias para avaliação de melodias.

## 7.2 Trabalhos Futuros

Seguem sugestões de trabalhos futuros que podem ser realizados em todas as etapas de desenvolvimento deste trabalho.

### 7.2.1 Composição Algorítmica

Métodos que permitissem que o AGs tivessem uma ideia de como a música inteira deve ser e que tivessem algum tipo de influência externa poderiam provavelmente levar a melhores abordagens. Atualmente, a maior parte das condições de avaliação de uma solução consideram apenas o acorde seguinte, no máximo. Seria interessante imaginar sistemas que pudessem analisar os compassos como um todo antes de aplicar os operadores genéticos.

Existe trabalho potencial na representação das soluções. A mesma notação musical que faz a comunicação mais fácil entre música pode ser eficiente quando se procura uma solução ótima com um AG.

Na geração de melodias, abordagens podem ser estudadas para a manutenção da diversidade da população. Entre elas, algumas seriam (i) abordagens multiobjetivo, (ii) organização da população em nichos e (iii) a utilização de sistemas imunológicos.

Um planejamento de mais alto nível antes da inicialização das soluções pode ser útil para extrair mais informação sobre o problema. Isso permitiria o desenvolvimento de operadores específicos para o domínio e um processo musicalmente ciente mais eficiente. Os algoritmos atuais aqui descritos operam apenas em níveis temporais locais e não globais, como a organização da música com repetições, frases ou motivos.

Em trabalhos futuros, uma definição mais aprofundada de quais aspectos são regras podem levar a uma abordagem ainda criativa que poderia ser mais baseada em regras.

Outro pontencial trabalho futuro desta tese seria uma biblioteca de análise de música tonal. Esta biblioteca pode ser formada com as medidas apresentadas de forma que

usuários possam analisar suas próprias músicas.

### 7.2.2 Árvores de Agregação

À medida que mais soluções se tornam não dominadas em otimização com muitos objetivos, explicar as relações entre estes objetivos se torna mais importante do que otimizá-los neste estágio. Pode ser também interessante utilizar soluções extremas em cada um dos objetivos para se ter uma representação significativa do potencial de cada objetivo.

Quando os objetivos são agrupados um a um, há o mínimo de conflito possível a cada passo de redução mas estes podem ser escolhas gulosas que não levam a uma redução ótima quando nós consideramos os objetivos que já foram agrupados. Neste sentido, pode ser que a ordem de redução para este problema de redução de objetivos seja apenas um ótimo local em relação a um certo número de objetivos a serem reduzidos se vários objetivos estão sendo reduzidos.

Outra questão a ser considerada são os modos de visualização destas reduções e como os resultados finais serão apresentados ao tomador de decisões já que há ainda uma necessidade de se mostrar valores absolutos e os objetivos reduzidos.



# Apêndice A

## Publicações Relacionadas

*“Quem não gosta de samba bom sujeito não é. É ruim da cabeça ou doente do pé.”*

— Samba da Minha Terra, Dorival Caymmi, 1914–2008

Segue neste Apêndice uma lista das publicações relacionadas e geradas ao longo do desenvolvimento desta tese.

- Composição Algorítmica
  - Freitas, A. R. R. ; Guimarães, F. G. ; Barbosa, R. V. . Automatic Evaluation Methods in Evolutionary Music: An Example with Bossa Melodies. Lecture Notes in Computer Science. 1ed.: Springer Berlin Heidelberg, 2012, v. , p. 458-467. *International Conference on Parallel Problem Solving From Nature*, (PPSN 2012), Taormina.
  - Freitas, A. R. R. ; Guimarães, F. G. ; Barbosa, R. V. . Ideas in Automatic Evaluation Methods for Melodies in Algorithmic Composition. In: *Sound and Music Computing Conference, 2012*, Copenhagen. *Proceedings of the Sound and Music Computing Conference, 2012*.
  - Freitas, A. R. R., Guimarães, F. G. (2011). Originality and Diversity in the Artificial Evolution of Melodies. *Genetic and Evolutionary Computation Conference (GECCO 2011)*, Dublin.
  - Freitas, A. R. R., Guimarães, F. G. (2011). Melody Harmonization in Evolutionary Music Using Multiobjective Genetic Algorithms. *8th Sound and Music Computing Conference (SMC 2011)*, Pádua.

- Freitas, A. R. R. ; Guimarães, F. G. ; Ruela, A. (2011). Computação Evolutiva Multiobjetivo para Harmonização de Melodias. *Simpósio Brasileiro de Pesquisa Operacional (SBPO 2011)*, Ubatuba.
- Freitas, A. R. R. ; Guimarães, F. G. (2011). Geração de Melodias Originais Usando Algoritmos Genéticos. *Congresso Brasileiro de Inteligência Computacional*, (CBIC 2011), Fortaleza.
- Inteligência Computacional
  - Freitas, A. R. R.; Fleming, P. J.; Guimarães, F. G. (2013). A Non-Parametric Harmony-Based Objective Reduction Method for Many-Objective Optimization. *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2013)*, Manchester.
  - Freitas, A. R. R. ; Silva, V. M. R. ; Campelo, F.; Guimarães, F. G. Optimizing two-level reverse distribution networks with hybrid memetic algorithms. *Optimization Letters (Print)*, p. 1-10, 2013.
  - Freitas, A. R. R. ; Guimarães, F. G. ; Pedrosa Silva, R. C. ; Souza, M. J. F.; Memetic self-adaptive evolution strategies applied to the maximum diversity problem. *Optimization Letters (Print)*, v. 7, p. 1-10, 2013.
  - Freitas, A. R. R. ; Silva, V. M. R. ; Guimarães, F. G. ; Campelo, F. . Genetic Algorithms Applied to Reverse Distribution Networks. *Advances in Intelligent Systems and Computing*. 1ed.: Springer Berlin Heidelberg, 2013, v. , p. 317-326.
  - Freitas, A. R. R. ; Pedrosa Silva, R. C. ; Guimarães, F. G. . Differential Evolution and Perceptron Decision Trees for Fault Detection in Power Transformers. *Advances in Intelligent Systems and Computing*. 1ed.: Springer Berlin Heidelberg, 2013, v. , p. 143-152.
  - Lopes, R. A. ; Freitas, A. R. R. ; Pedrosa-Silva, R. C. ; Guimarães, Frederico Gadelha . Differential Evolution and Perceptron Decision Trees for Classification Tasks. *Lecture Notes in Computer Science*. 1ed.: Springer Berlin Heidelberg, 2012, v. , p. 550-557.
  - Pedrosa-Silva, R. C. ; Lopes, R. A. ; Freitas, A. R. R. ; Guimarães, F. G. . Performance Comparison of Parameter Variation Operators in Self-Adaptive Differential Evolution Algorithms. In: *Brazilian Symposium on Neural Networks, 2012, Curitiba*. Proceeding of the Brazilian Symposium on Neural Networks, 2012.

- Silva, V. M. R. ; Freitas, A. R. R. ; Campelo, F. ; Guimarães, F. G. . Logística Reversa Capacitada em Dois Níveis - Uma Abordagem Utilizando Algoritmos Genéticos. In: Simpósio Brasileiro de Pesquisa Operacional (SBPO), 2012, Rio de Janeiro. Anais do Simpósio Brasileiro de Pesquisa Operacional (SBPO), 2012.
- Oliveira, M. V. ; Freitas, A. R. R. ; Guimarães, F. G. . Uma estratégia de Ranking baseada em diversidade em Algoritmos Genéticos. In: Simpósio Brasileiro de Pesquisa Operacional (SBPO), 2012, Rio de Janeiro. Anais do Simpósio Brasileiro de Pesquisa Operacional (SBPO), 2012.
- Eitelberg, E. ; Freitas, A. R. R. . Pareto Optimal Sensitivity Bounds for a Strongly Interacting 2-Loop Brazing Temperature Control System. In: The ORT Braude College 8th Interdisciplinary Research Conference, 2012, Karmi'el. Proceedings of the The ORT Braude College 8th Interdisciplinary Research Conference. Karmi'el, 2012.
- Freitas, A. R. R. ; Lopes, R. A. ; Pedrosa-Silva, R. C. ; Guimarães, F. G. . Classificação de Dados Através de Árvores de Perceptrons Geradas por Evolução Diferencial. In: Encontro Nacional de Inteligência Artificial, 2012, Curitiba. Anais do Encontro Nacional de Inteligência Artificial, 2012.
- Ruela, A. ; Aquino, A. ; Guimarães, F. G. ; Freitas, A. R. R. . Um Algoritmo Coevolutivo Cooperativo para Configuração de Redes de Sensores sem Fio. In: XLIII Simpósio Brasileiro de Pesquisa Operacional (SBPO), 2011, Ubatuba. Anais do XLIII Simpósio Brasileiro de Pesquisa Operacional, 2011.



# Referências Bibliográficas

- Allan, M. & Williams, C. (2005). Harmonising chorales by probabilistic inference, *Advances in Neural Information Processing Systems* **17**: 25–32.
- Alpern, A. (1995). Techniques for algorithmic composition of music, <http://hamp.hampshire.edu/~adaF92/algocomp/algocomp> **95**.
- Auger, A., Bader, J., Brockhoff, D. & Zitzler, E. (2012). Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications, *Theoretical Computer Science* **425**: 75–103.
- Baroni, M., Brunetti, R., Callegari, L. & Jacoboni, C. (1984). A grammar for melody: Relationships between melody and harmony, *Musical Grammars and Computer Analysis* pp. 201–218.
- Bashkansky, E., Gadrich, T. & Kuselman, I. (2012). Interlaboratory comparison of test results of an ordinal or nominal binary property: analysis of variation, *Accreditation and Quality Assurance: Journal for Quality, Comparability and Reliability in Chemical Measurement* pp. 1–5.
- Batista, L., Campelo, F., Guimarães, F. & Ramírez, J. (2011). Pareto cone  $\epsilon$ -dominance: improving convergence and diversity in multiobjective evolutionary algorithms, *Evolutionary Multi-Criterion Optimization*, Springer, pp. 76–90.
- Bell, C. (2011). Algorithmic music composition using dynamic markov chains and genetic algorithms, *J. Comput. Sci. Coll.* **27**(2): 99–107.  
**URL:** <http://dl.acm.org/citation.cfm?id=2038836.2038850>
- Bellgard, M. & Tsang, C. (1994). Harmonizing music the boltzmann way, *Connection Science* **6**(2-3): 281–297.

- Beume, N., Naujoks, B. & Emmerich, M. (2007). Sms-emoa: Multiobjective selection based on dominated hypervolume, *European Journal of Operational Research* **181**(3): 1653–1669.
- Beume, N., Naujoks, B. & Wagner, T. (2006). Pareto-, aggregation-, and indicator-based methods in many-objective optimization.
- Beyls, P. (2003). Selectionist musical automata: Integrating explicit instruction and evolutionary algorithms, *IX Brazilian Symposium on Computer Music*, Brazilian Computing Society.
- Bidlack, R. (1992). Chaotic systems as simple (but complex) compositional algorithms, *Computer Music Journal* pp. 33–47.
- Biles, J. (1994). Genjam: A genetic algorithm for generating jazz solos, *Proceedings of the International Computer Music Conference*, Citeseer, pp. 131–131.
- Biles, J. (2001). Autonomous genjam: eliminating the fitness bottleneck by eliminating fitness, *Proceedings of the GECCO-2001 Workshop on Non-routine Design with Evolutionary Systems*, San Francisco.
- Biles, J. (2002). Genjam: evolutionary computation gets a gig, *Proceedings of the 2002 Conference for Information Technology Curriculum, Rochester, New York, Society for Information Technology Education*, Citeseer.
- Biles, J. (2007a). Improvizng with genetic algorithms: Genjam, *Evolutionary Computer Music* pp. 137–169.
- Biles, J. A. (2007b). Evolutionary computation for musical tasks, in E. R. Miranda & J. A. Biles (eds), *Evolutionary Computer Music*, Springer London, pp. 28–51.  
**URL:** [http://dx.doi.org/10.1007/978-1-84628-600-1\\_2](http://dx.doi.org/10.1007/978-1-84628-600-1_2)
- Biles, J., Anderson, P. & Loggi, L. (1996). Neural network fitness functions for a musical iga, *International ICSC Symposium on Intelligent Industrial Automation (IIA'96) and Soft Computing (SOCO'96)*, International Computing Sciences Conferences (ICSC), pp. B39–44.
- Biles, J. & Eign, W. (1995). Genjam populi: Training an iga via audience-mediated performance, *Proceedings of the 1995 International Computer Music Conference*, Vol. 12.

- Bilotta, E. & Pantano, P. (2001). Artificial life music tells of complexity, *ALMMA 2001: Proceedings of the workshop on artificial life models for musical applications*, pp. 17–28.
- Bilotta, E., Pantano, P. & Talarico, V. (2000). Music generation through cellular automata: How to give life to strange creatures, *Proceedings of Generative Art GA 2000*.
- Blacking, J. (1984). What languages do musical grammars describe, *Musical grammars and computer analysis* pp. 363–370.
- Bod, R. (2002). Memory-based models of melodic analysis: Challenging the gestalt principles, *Journal of New Music Research* **31**(1): 27–36.
- Brockhoff, D. & Zitzler, E. (2006). Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization, *Parallel Problem Solving from Nature-PPSN IX* pp. 533–542.
- Brockhoff, D. & Zitzler, E. (2007). Dimensionality reduction in multiobjective optimization: The minimum objective subset problem, *Operations Research Proceedings 2006* pp. 423–429.
- Brooks, F., Hopkins, A., Neumann, P. G. & Wright, W. V. (1957). An experiment in musical composition, *Electronic Computers, IRE Transactions on* **EC-6**(3): 175–182.
- Brown, A. R. (2002). Opportunities for evolutionary music composition, *Proceedings of the Australasian Computer Music Conference*, Melbourne, pp. 27–34.
- Burraston, D. & Edmonds, E. (2005). Cellular automata in generative electronic music and sonic art: a historical and technical review, *Digital Creativity* **16**(3): 165–185.
- Cambouropoulos, E. (1997). Musical rhythm: A formal model for determining local boundaries, accents and metre in a melodic surface, *Music, gestalt, and computing* pp. 277–293.
- Camilleri, L. (1984). A grammar of the melodies of schubert’s lieder’, *Musical Grammars and Computer Analysis* pp. 229–36.
- Chuang, J. (1995). Mozarts musikalisches wurfelspiel, *URL: <http://sunsite.univie.ac.at/Mozart/dice>* .

- Coello, C., Lamont, G. & Veldhuizen, D. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*, Genetic and evolutionary computation series, Springer.  
**URL:** <http://books.google.com.br/books?id=rXIuAMw3lGAC>
- Cope, D. (1987). An expert system for computer-assisted composition, *Computer Music Journal* pp. 30–46.
- Cope, D. (1991). *Computers and musical style*, AR Editions, Inc.
- Cope, D. (1992). On algorithmic representation of musical style, *Understanding music with AI*, MIT Press, pp. 355–363.
- Cope, D. (2000). *The algorithmic composer*, Vol. 16, AR Editions, Inc.
- Cope, D. (2004). *Virtual music: computer synthesis of musical style*, MIT press.
- Cope, D. & Mayer, M. (1996). *Experiments in musical intelligence*, Vol. 12, AR Editions Madison, WI.
- Corne, D. W. & Bentley, P. J. (eds) (2001). *Creative Evolutionary Systems*, The Morgan Kaufmann Series in Artificial Intelligence, Morgan Kaufmann.
- Corne, D. W., Jerram, N. R., Knowles, J. D., Oates, M. J. & J, M. (2001). Pesa-ii: Region-based selection in evolutionary multiobjective optimization, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, Morgan Kaufmann Publishers, pp. 283–290.
- Deb, K. (2009). *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley-interscience series in systems and optimization, Wiley.  
**URL:** <http://books.google.com.br/books?id=U0dnPwAACAAJ>
- Deb, K., Agrawal, S., Pratap, A. & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii, *Parallel Problem Solving from Nature PPSN VI*, Springer, pp. 849–858.
- Dodge, C. (1988). "profile": A musical fractal, *Computer Music Journal* **12**(3): 10–14.
- Dorin, A. (2000). Boolean networks for the generation of rhythmic structure, *proceedings of Interfaces: The Australasian Computer Music Conference*, AR Brown and R. Wilding, eds. Brisbane: Australasian Computer Music Association, pp. 38–45.

- Dorin, A. (2002). Liquiprism: Generating polyrhythms with cellular automata, *Proceedings of the 2002 International Conference on Auditory Display, Kyoto, Japan, July 2*, Vol. 5.
- Dowling, W. (1978). Scale and contour: Two components of a theory of memory for melodies., *Psychological Review* **85**(4): 341.
- Dowling, W. & Harwood, D. (1986). *Music cognition*, Vol. 19986, Academic Press New York.
- Drechsler, N., Drechsler, R. & Becker, B. (2001). Multi-objective optimisation based on relation favour, *Evolutionary Multi-Criterion Optimization*, Springer, pp. 154–166.
- DuBois, R. (2003). *Applications of generative string-substitution systems in computer music*, PhD thesis, Columbia University.
- Eck, D. & Schmidhuber, J. (2002). A first look at music composition using lstm recurrent neural networks, *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale* .
- Eerola, T., Himberg, T., Toivainen, P. & Louhivuori, J. (2006). Perceived complexity of western and african folk melodies by western and african listeners, *Psychology of Music* **34**(3): 337–371.
- Eerola, T. & North, A. (2000). Expectancy-based model of melodic complexity, *Proceedings of the Sixth International Conference on Music Perception and Cognition. Keele, Staffordshire, UK: Department of Psychology. CD-ROM*.
- Eerola, T. & Toivainen, P. (2004). *MIDI Toolbox: MATLAB Tools for Music Research*, University of Jyväskylä, Jyväskylä, Finland.  
**URL:** [www.jyu.fi/musica/miditoolbox/](http://www.jyu.fi/musica/miditoolbox/)
- Eitelberg, E. (2006). On multiloop interaction and relative and bristol gains, *Journal of Dynamic Systems, Measurement, and Control* **128**(4): 929–937.  
**URL:** <http://link.aip.org/link/?JDS/128/929/1>
- Farbood, M. & Schoner, B. (2001). Analysis and synthesis of palestrina-style counterpoint using markov chains, *Proceedings of the International Computer Music Conference*, pp. 471–474.
- Federman, F. (2003). The nextpitch learning classifier system: Representation, information theory and performance, *Leonardo* **36**(1): 47–50.

- Felder, D. & Stockhausen, K. (1977). An interview with karlheinz stockhausen, *Perspectives of New Music* **16**(1): 85–101.
- Fonseca, C. & Fleming, P. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, *Proceedings of the fifth international conference on genetic algorithms*, Vol. 1, San Mateo, California, p. 416.
- Freitas, A. & Guimarães, F. (2011a). Melody harmonization in evolutionary music using multiobjective genetic algorithms, *Proceedings of the Sound and Music Computing Conference (SMC)*.
- Freitas, A. & Guimarães, F. (2011b). Originality and diversity in the artificial evolution of melodies, *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, ACM, pp. 419–426.
- Freitas, A., Guimarães, F. & Barbosa, R. (2012). Ideas in automatic evaluation methods for melodies in algorithmic composition, *Proceedings of the Sound and Music Computing Conference (SMC)*.
- Freitas, A. R. R., Guimarães, F. G. & Fleming, P. J. (2013). A non-parametric harmony-based objective reduction method for many-objective optimization., *IEEE International Conference on Systems, Man, and Cybernetics*.
- Galanter, P. (2008). Complexism and the role of evolutionary art, *The Art of Artificial Evolution* pp. 311–332.
- Galanter, P. (2010). The problem with evolutionary art is..., *Applications of Evolutionary Computation* pp. 321–330.
- Gartland-Jones, A. (2003). Musicblox: a real-time algorithmic composition system incorporating a distributed interactive genetic algorithm, *Proceedings of the 2003 international conference on Applications of evolutionary computing*, Springer-Verlag, pp. 490–501.
- Garza-Fabre, M., Toscano-Pulido, G., Coello Coello, C. & Rodriguez-Tello, E. (2011). Effective ranking+ speciation= many-objective optimization, *Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE*, pp. 2115–2122.
- Gedeon, T. (2008). Neural network for modeling esthetic selection, *Neural Information Processing*, Springer, pp. 666–674.

- Giagkiozis, I., Purshouse, R. & Fleming, P. (2012). Generalized decomposition and cross entropy methods for many-objective optimization.
- Graf, J. & Banzhaf, W. (1995). Interactive evolution of images, *in* D. B. Fogel (ed.), *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pp. 53–65.
- Grahan-Rowe, D. (2001). Computer dj uses biofeedback to pick tracks, *New Scientist* .
- Greenfield, G. R. (2002). Simulated aesthetics and evolving artworks: a coevolutionary approach, *Leonardo* **35**(3).
- Grosan, C. & Abraham, A. (2007). Hybrid line search for multiobjective optimization, *High Performance Computing and Communications* pp. 62–73.
- Haus, G. & Sametti, A. (1991). Scoresynth: a system for the synthesis of music scores based on petri nets and a music algebra, *Computer* **24**(7): 56–60.  
**URL:** <http://dx.doi.org/10.1109/2.84837>
- Hernández-Díaz, A., Santana-Quintero, L., Coello Coello, C., Caballero, R. & Molina, J. (2006). A new proposal for multi-objective optimization using differential evolution and rough sets theory, *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, ACM, pp. 675–682.
- Hild, H., Feulner, J. & Menzel, D. (1992). Harmonet: a neural net for harmonising chorales in the style of js bach.
- Hiller, L. (1959). Computer music, *Scientific American* **201**(6): 109–120.
- Hiller, L. (1981). Composing with computers: A progress report, *Computer Music Journal* **5**(4): 7–21.
- Hiller, L. & Isaacson, L. (1958). Musical composition with a high-speed digital computer, *Machine models of music* pp. 9–21.
- Hillis, W. (1990). Co-evolving parasites improve simulated evolution as an optimization procedure, *Physica D: Nonlinear Phenomena* **42**(1-3): 228–234.
- Hirzel, M. & Soukup, D. (2000). Project writeup for csci 5832 natural language processing.
- Horner, A. & Ayers, L. (1995). Harmonisation of musical progression with genetic algorithms, *ICMC Proceedings 1995*, pp. 483–484.

- Horner, A., Beauchamp, J. & Haken, L. (1993). Methods for multiple wavetable synthesis of musical instrument tones, *Journal-Audio Engineering Society* **41**: 336–336.
- Horner, A. & Goldberg, D. (1991). Genetic algorithms and computer-assisted music composition, *Urbana* **51**(61801): 14.
- Horowitz, D. (1995). Generating rhythms with genetic algorithms, *Proceedings of the national conference on artificial intelligence*, John Wiley & Sons LTD, pp. 1459–1459.
- Hughes, E. (2005). Evolutionary many-objective optimisation: many once or one many?, *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, Vol. 1, IEEE, pp. 222–227.
- Hughes, E. (2007). Msops-ii: A general-purpose many-objective optimiser, *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, IEEE, pp. 3944–3951.
- Ikeda, K., Kita, H. & Kobayashi, S. (2001). Failure of pareto-based moeas: Does non-dominated really mean near to optimal?, *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, Vol. 2, IEEE, pp. 957–962.
- Ishibuchi, H., Doi, T. & Nojima, Y. (2006). Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms, *Parallel Problem Solving from Nature-PPSN IX* pp. 493–502.
- Ishibuchi, H. & Nojima, Y. (2007). Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning, *International Journal of Approximate Reasoning* **44**(1): 4–31.
- Ishibuchi, H., Tsukamoto, N. & Nojima, Y. (2008). Evolutionary many-objective optimization: A short review, *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, IEEE, pp. 2419–2426.
- Jacob, B. (1995). Composing with genetic algorithms, *Proceedings of the 1995 International Computer Music Conference* .
- Jacob, B. (1996). Algorithmic composition as a model of creativity, *Organised Sound* **1**(03): 157–165.
- Jarque, C. & Bera, A. (1987). A test for normality of observations and regression residuals, *International Statistical Review/Revue Internationale de Statistique* pp. 163–172.

- Jobim, A. & Chediak, A. (1990). *Songbook Tom Jobim*, Vol. 1, Irmãos Vitale.
- Johnson, C. (2012). Fitness in evolutionary art and music: What has been used and what could be used?, in P. Machado, J. Romero & A. Carballal (eds), *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, Vol. 7247 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 129–140.  
**URL:** [http://dx.doi.org/10.1007/978-3-642-29142-5\\_12](http://dx.doi.org/10.1007/978-3-642-29142-5_12)
- Johnson-Laird, P. (1991). Jazz improvisation: A theory at the computational level, *Representing musical structure*, London pp. 291–325.
- Jones, K. (1981). Compositional applications of stochastic processes, *Computer Music Journal* **5**(2): 45–61.
- Kazi, N. & Bhatia, S. (2013). Various artificial intelligence techniques for automated melody generation, *International Journal of Engineering* **2**(7).
- Kennedy, M. & Bourne, J. (2004). *The concise Oxford dictionary of music*, Oxford University Press, USA.
- Khalifa, Y. & Foster, R. (2006). A two-stage autonomous evolutionary music composer, *Applications of Evolutionary Computing* pp. 717–721.
- Khare, V., Yao, X. & Deb, K. (2003). Performance scaling of multi-objective evolutionary algorithms, *Evolutionary Multi-Criterion Optimization*, Springer, pp. 72–72.
- Kiernan, F. (2000). Score-based style recognition using artificial neural networks, *Pythagoras, Graduate School, Department of Music, University of Jyväskylä, Finland*.
- Kirnberger, J. & Kupper, H. (1994). *Der allezeit fertige Polonoisen-und Menuettencomponist*, Musikwiss. Inst.
- Knowles, J. & Corne, D. (2000). Approximating the nondominated front using the pareto archived evolution strategy, *Evolutionary computation* **8**(2): 149–172.
- Knowles, J. & Corne, D. (2007). Quantifying the effects of objective space dimension in evolutionary multiobjective optimization, *Evolutionary Multi-Criterion Optimization*, Springer, pp. 757–771.
- Koppen, M. & Yoshida, K. (2007). Visualization of pareto-sets in evolutionary multi-objective optimization, *Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on*, IEEE, pp. 156–161.

- Krumhansl, C. (1995). Music psychology and music theory: Problems and prospects, *Music Theory Spectrum* pp. 53–80.
- Krumhansl, C. (2001). *Cognitive foundations of musical pitch*, Oxford University Press, USA.
- Krumhansl, C. & Kessler, E. (1982). Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys., *Psychological Review* **89**(4): 334.
- Laumanns, M., Thiele, L., Deb, K. & Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization, *Evolutionary computation* **10**(3): 263–282.
- Laumanns, M., Zitzler, E. & Thiele, L. (2001). On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization, *Evolutionary multi-criterion optimization*, Springer, pp. 181–196.
- Leach, J. & Fitch, J. (1995). Nature, music, and algorithmic composition, *Computer Music Journal* **19**(2): 23–33.
- Leman, M. (1997). *Music, gestalt, and computing: Studies in cognitive and systematic musicology*, Vol. 1, Springer Verlag.
- Lerdahl, F. (1996). Calculating tonal tension, *Music Perception* pp. 319–363.
- Lerdahl, F. & Jackendoff, R. (1983). An overview of hierarchical structure in music, *Music Perception* pp. 229–252.
- Lerdahl, F., Jackendoff, R. & Jackendoff, R. (1996). *A generative theory of tonal music*, The MIT Press.
- Lewis, M. (2000). Aesthetic evolutionary design with data flow networks, *Proceedings of Generative Art*, Milan, Italy.
- Lewis, M. (2008). Evolutionary visual art and design, in P. Machado & J. Romero (eds), *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, Springer, pp. 3–37.
- López Jaimes, A., Coello Coello, C. A. & Chakraborty, D. (2008). Objective reduction using a feature selection technique, *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, GECCO '08, ACM, New York, NY, USA,

pp. 673–680.

URL: <http://doi.acm.org/10.1145/1389095.1389228>

- Luque, M., Miettinen, K., Eskelinen, P. & Ruiz, F. (2009). Incorporating preference information in interactive reference point methods for multiobjective optimization, *Omega* **37**(2): 450–462.
- Machado, P. & Cardoso, A. (1998). Computing aesthetics, *Advances in Artificial Intelligence* pp. 105–119.
- Machado, P., Romero, J. & Manaris, B. (2008). Experiments in computational aesthetics, *The Art of Artificial Evolution* pp. 381–415.
- Machado, P., Romero, J., Santos, M., Cardoso, A. & Manaris, B. (2004). Adaptive critics for evolutionary artists, *Applications of evolutionary computing* pp. 437–446.
- Maddox, T. & Otten, J. (2000). Using an evolutionary algorithm to generate four-part 18th century harmony, *Mathematics and Computers in Modern Science: Acoustics and Music, etc* pp. 83–89.
- Madsen, S. & Widmer, G. (2005). Exploring similarities in music performances with an evolutionary algorithm, *Proceedings of The 18th International FLAIRS Conference, AAAI Press*.
- Manaris, B., Machado, P., McCauley, C., Romero, J. & Krehbiel, D. (2005). Developing fitness functions for pleasant music: Zipf’s law and interactive evolution systems, *Applications of Evolutionary Computing* pp. 498–507.
- Mandelis, J. (2001). Genophone: An evolutionary approach to sound synthesis and performance, *Proceedings ALMMA* pp. 37–50.
- Massey Jr, F. (1951). The kolmogorov-smirnov test for goodness of fit, *Journal of the American statistical Association* pp. 68–78.
- McCormack, J. (1996). Grammar based music composition, *Complex systems* **96**: 321–336.
- McCormack, J. (2003). Evolving sonic ecosystems, *Kybernetes* **32**(1): 184–202.
- McCormack, J. (2005a). Open problems in evolutionary music and art, *Applications on Evolutionary Computing* pp. 428–436.

- McCormack, J. (2005b). Open problems in evolutionary music and art, *Applications of Evolutionary Computing* pp. 428–436.
- McDermott, J., Griffith, N. & O’Neill, M. (2005). Toward user-directed evolution of sound synthesis parameters, *Applications of Evolutionary Computing* pp. 517–526.
- McIntyre, R. (1994). Bach in a box: The evolution of four part baroque harmony using the genetic algorithm, *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, IEEE, pp. 852–857.
- Meehan, J. (1979). An artificial intelligence approach to tonal music theory, *Proceedings of the 1979 annual conference*, ACM, pp. 116–120.
- Michael, C. (1994). Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing, *Connection Science* **6**(2-3): 247–280.
- Millen, D. (2004). An interactive cellular automata music application in cocoa, *International Computer Music Conference*.
- Miranda, E. (2001). Evolving cellular automata music: From sound synthesis to composition, *Proceedings of 2001 Workshop on Artificial Life Models for Musical Applications*.
- Miranda, E. R. & Biles, J. A. (2007). *Evolutionary computer music*, Springer Verlag.
- Molina, J., Santana, L., Hernández-Díaz, A., Coello Coello, C. & Caballero, R. (2009). g-dominance: Reference point based dominance for multiobjective metaheuristics, *European Journal of Operational Research* **197**(2): 685–692.
- Mori, T., Endou, Y. & Nakayama, A. (1996). Fractal analysis and aesthetic evaluation of geometrically overlapping patterns, *Textile research journal* **66**(9): 581.
- Moroni, A., Manzolli, J., Zuben, F. & Gudwin, R. (2000). Vox populi: An interactive evolutionary system for algorithmic music composition, *Leonardo Music Journal* pp. 49–54.
- Mozart, W. (1787). Musikalisches würfelspiel: Anleitung so viel walzer oder schleifer mit zwei würfeln zu componieren ohne musikalisch zu seyn noch von der composition etwas zu verstehen, *Köchel Catalog of Mozarts Work KV1 Appendix 294d or KV6 516f*.

- Mrozek, E. & Wakefield, G. (1996). Perceptual matching of low-order models to room transfer functions, *Proceedings of the International Computer Music Conference*, International Computer Music Association, pp. 111–113.
- Narmour, E. (1992). *The analysis and cognition of melodic complexity: The implication-realization model*, University of Chicago Press.
- Nettl, B. (1956). *Music in primitive culture.*, Harvard University Press.
- Nierhaus, G. (2010). *Algorithmic Composition: Paradigms of Automated Music Generation*, Springer.
- Obayashi, S. & Sasaki, D. (2003). Visualization and data mining of pareto solutions using self-organizing map, *Evolutionary multi-criterion optimization*, Springer, pp. 71–71.
- Olson, H. (1967). *Music, physics and engineering*, Dover Pubns.
- Pachet, F. (1999). Surprising harmonies, *International Journal of Computing Anticipatory Systems* **4**: 139–161.
- Palmer, C. & Krumhansl, C. (1990). Mental representations for musical meter., *Journal of Experimental Psychology: Human Perception and Performance; Journal of Experimental Psychology: Human Perception and Performance* **16**(4): 728.
- Papadopoulos, G. & Wiggins, G. (1998). A genetic algorithm for the generation of jazz melodies, *Proceedings of STeP* **98**.
- Phon-Amnuaisuk, S., Law, E. & Kuan, H. (2007). Evolving music generation with som-fitness genetic programming, *Applications of Evolutionary Computing* pp. 557–566.
- Phon-Amnuaisuk, S. & Wiggins, G. (1999). The four-part harmonisation problem: a comparison between genetic algorithms and a rule-based system, *Proceedings of the AISB99 Symposium on Musical Creativity*, pp. 28–34.
- Polito, J., Daida, J. & Bersano-Begey, T. (1997). Musica ex machina: composing 16th-century counterpoint with genetic programming and symbiosis, *Evolutionary Programming VI*, Springer, pp. 113–123.
- Ponsford, D., Wiggins, G. & Mellish, C. (1999). Statistical learning of harmonic movement, *Journal of New Music Research* **28**(2): 150–177.
- Pressing, J. (1988). Nonlinear maps as generators of musical design, *Computer Music Journal* **12**(2): 35–46.

- Pritchett, J. (1996). *The Music of John Cage*, Cambridge Univ Pr.
- Prusinkiewicz, P. (1986). Score generation with l-systems, *Proceedings of the 1986 International Computer Music Conference*, pp. 455–457.
- Purshouse, R. & Fleming, P. (2003). Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation, in C. Fonseca, P. Fleming, E. Zitzler, L. Thiele & K. Deb (eds), *Evolutionary Multi-Criterion Optimization*, Vol. 2632 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 16–30.  
**URL:** [http://dx.doi.org/10.1007/3-540-36970-8\\_2](http://dx.doi.org/10.1007/3-540-36970-8_2)
- Purshouse, R. & Fleming, P. (2007). On the evolutionary optimization of many conflicting objectives, *Evolutionary Computation, IEEE Transactions on* **11**(6): 770–784.
- Putnam, J. (1996). A grammar-based genetic programming technique applied to music generation, *Evolutionary Programming V, MIT Press, Cambridge, MA* pp. 363–368.
- Rader, G. (1974). A method for composing simple traditional music by computer, *Communications of the ACM* **17**(11): 631–638.
- Ralley, D. (1995). Genetic algorithms as a tool for melodic development, *Urbana* **101**: 61801.
- Roads, C. & Wieneke, P. (1979). Grammars as representations for music, *Computer Music Journal* **3**(1): 48–55.
- Romero, J. J. & Machado, P. (2007). *The art of artificial evolution: A handbook on evolutionary art and music*, Natural Computing Series, Springer.
- Ross, B. J., Ralph, W. & Zong, H. (2006). Evolutionary image synthesis using a model of aesthetics, *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2006*, Vancouver, BC, pp. 1087–1094.
- Salomon, S., Avigad, G., Goldvard, A. & Schütze, O. (2013). Psa – a new scalable space partition based selection algorithm for moeas, in O. Schütze, C. A. Coello Coello, A.-A. Tantar, E. Tantar, P. Bouvry, P. Del Moral & P. Legrand (eds), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, Vol. 175 of *Advances in Intelligent Systems and Computing*, Springer Berlin Heidelberg, pp. 137–151.  
**URL:** [http://dx.doi.org/10.1007/978-3-642-31519-0\\_9](http://dx.doi.org/10.1007/978-3-642-31519-0_9)

- Sato, H., Aguirre, H. & Tanaka, K. (2007). Controlling dominance area of solutions and its impact on the performance of moeas, *Evolutionary Multi-Criterion Optimization*, Springer, pp. 5–20.
- Sato, H., Aguirre, H. & Tanaka, K. (2009). Local dominance moea including control of dominance area of solutions on 0/1 multiobjective knapsack problems, *Information and Media Technologies* **4**(1): 33–43.
- Saxena, D. & Deb, K. (2007). Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding, *Evolutionary Multi-Criterion Optimization*, Springer, pp. 772–787.
- Saxena, D., Duro, J., Tiwari, A., Deb, K. & Zhang, Q. (2013). Objective reduction in many-objective optimization: Linear and nonlinear algorithms, *Evolutionary Computation, IEEE Transactions on* **17**(1): 77–99.
- Saxena, D., Ray, T., Deb, K. & Tiwari, A. (2009). Constrained many-objective optimization: a way forward, *Evolutionary Computation, 2009. CEC'09. IEEE Congress on, IEEE*, pp. 545–552.
- Sharman, K. & Esparcia-Alcázar, A. (2003). Evolutionary methods for designing digital filters, *Contemporary Music Review* **22**(3): 5–19.
- Simonton, D. (1984). Melodic structure and note transition probabilities: A content analysis of 15,618 classical themes, *Psychology of Music* **12**(1): 3–16.
- Simonton, D. (1994). Computer content analysis of melodic structure: Classical composers and their compositions, *Psychology of Music* **22**(1): 31–43.
- Singh, H., Isaacs, A. & Ray, T. (2011). A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems, *Evolutionary Computation, IEEE Transactions on* **15**(4): 539–556.
- Singh, S. & Singh, V. (2011). Three-level ahp-based heuristic approach for a multi-objective facility layout problem, *International Journal of Production Research* **49**(4): 1105–1125.
- Smoliar, S. (1980). A computer aid for schenkerian analysis, *Computer Music Journal* **4**(2): 41–59.

- Steedman, M. (1984). A generative grammar for jazz chord sequences, *Music Perception* pp. 52–77.
- Sundberg, J. & Lindblom, B. (1976). Generative theories in language and music descriptions, *Cognition* **4**(1): 99–122.
- Takagi, H. (2001). Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation, *Proceedings of the IEEE* **89**(9): 1275–1296.
- Tenney, J. & Polansky, L. (1980). Temporal gestalt perception in music, *Journal of Music Theory* **24**(2): 205–241.
- Thompson, W. (1994). Sensitivity to combinations of musical parameters: Pitch with duration, and pitch pattern with durational pattern, *Attention, Perception, & Psychophysics* **56**(3): 363–374.
- Tillmann, B., Bharucha, J. & Bigand, E. (2000). Implicit learning of tonality: a self-organizing approach., *Psychological review* **107**(4): 885.
- Todd, P. (1989). A connectionist approach to algorithmic composition, *Computer Music Journal* **13**(4): 27–43.
- Todd, P. & Werner, G. (1999). Frankensteinian methods for evolutionary music composition, *Musical networks: Parallel distributed perception and performance* pp. 313–339.
- Todd, S. & Latham, W. (1992). *Evolutionary art and computers*, Academic Press, Orlando, FL, USA.
- Toiviainen, P., Krumhansl, C. et al. (2003). Measuring and modeling real-time responses to music: The dynamics of tonality induction, *PERCEPTION-LONDON* **32**(6): 741–766.
- Towsey, M., Brown, A., Wright, S. & Diederich, J. (2001). Towards melodic extension using genetic algorithms, *Educational Technology & Society* **4**(2): 54–65.
- Unemi, T. & Senda, M. (2002). A study on building a support system for music composition by simulated breeding method. extension to multi-part., *Joho Shori Gakkai Kenkyu Hokoku* **2002**(40): 133–138.
- Von Hippel, P. (2000). Redefining pitch proximity: Tessitura and mobility as constraints on melodic intervals, *Music Perception* pp. 315–327.

- Vos, P. & Troost, J. (1989). Ascending and descending melodic intervals: Statistical findings and their perceptual relevance, *Music Perception* pp. 383–396.
- Voss, R. & Clarke, J. (1975). 1/f noise in music and speech, *Nature* **258**: 317–318.
- Vuori, J. & Välimäki, V. (1993). Parameter estimation of non-linear physical models by simulated evolution-application to the flute model, *Proceedings of the International Computer Music Conference*, International Computer Music Association, pp. 402–402.
- Wang, R., Purshouse, R. & Fleming, P. (2012). Local preference-inspired co-evolutionary algorithms, *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, ACM, pp. 513–520.
- Wang, T., Wu, Z. & Mao, J. (2007). A new method for multi-objective tdma scheduling in wireless sensor networks using pareto-based pso and fuzzy comprehensive judgement, *High Performance Computing and Communications* pp. 144–155.
- Waschka II, R. (1999). Avoiding the fitness bottleneck: Using genetic algorithms to compose orchestral music, *International Computer Music Conference ICMC99*, pp. 201–203.
- Waschka II, R. (2007). Composing with genetic algorithms: Gendash, *Evolutionary Computer Music* pp. 117–136.
- Wickramasinghe, U. & Li, X. (2009). Using a distance metric to guide pso algorithms for many-objective optimization, *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, ACM, pp. 667–674.
- Wolf, S. & Yee-King, M. (2003). Virtual and physical interfaces for collaborative evolution of sound, *Contemporary Music Review* **22**(3): 31–41.
- Xenakis, I. (1992). *Formalized Music: Thought and mathematics in composition*, Pendragon Pr.
- Xiong, J., Chen, Y., Yang, K., Zhao, Q. & Xing, L. (2012). A hybrid multiobjective genetic algorithm for robust resource-constrained project scheduling with stochastic durations, *Mathematical Problems in Engineering* **2012**.
- Yoshikawa, T. & Furuhashi, T. (2007). Visualization techniques for mining of solutions, *ISIS 2007 Proceedings of the 8TH Symposium on Advanced Intelligent Systems*.

- Zitzler, E. & Künzli, S. (2004). Indicator-based selection in multiobjective search, *Parallel Problem Solving from Nature-PPSN VIII*, Springer, pp. 832–842.
- Zitzler, E., Laumanns, M., Thiele, L., Zitzler, E., Zitzler, E., Thiele, L. & Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm.
- Zitzler, E. & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, *Trans. Evol. Comp* **3**(4): 257–271.  
**URL:** <http://dx.doi.org/10.1109/4235.797969>
- Zou, S., Zhang, H. & Wang, K. (2008). Fuzzy-ga and multi-objective transportation optimization, *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*, IEEE, pp. 270–273.

# Índice Remissivo

- Árvores de Agregação, 49
- Algoritmos Genéticos em Composição Algorítmica, 30
- Análises de Segunda Ordem, 101
- Aplicação das Árvores de Agregação, 122
- Autômatos Celulares em Composição Algorítmica, 26
- Avaliação de Melodias, 75
- Avaliação Melódica, 76
- Banco de Dados Bossa Nova, 80
- Base de Dados, 105
- Caos e Autossimilaridade em Composição Algorítmica, 25
- Conclusão da Revisão da Literatura, 34
- Conclusão sobre Árvores de Agregação, 74
- Conclusão sobre Medidas de Avaliação Automática, 102
- Conclusão sobre os Resultados, 128
- Conclusão sobre Otimização com Muitos Objetivos, 47
- Conclusões, 135
- Conclusões sobre Composição Algorítmica, 135
- Conclusões sobre Medidas de Avaliação de Melodias, 139
- Conflito Não-Paramétrico de Ranqueamento, 64
- Contorno, 91
- Contribuições, 9
- Dissonância, 92
- EMI, 21
- Expectativa Melódica, 88
- Experimentos em Inteligência Musical, 21
- Experiments in Musical Intelligence, 21
- Frases, 100
- Funções de Avaliação Automática, 78
- Funções objetivo, 106
- Funções objetivo de Altura, 107
- Funções objetivo de Contorno, 111
- Funções objetivo de Duração, 114
- Funções objetivo de Intervalo, 110
- Funções objetivo de padrões, 119
- Funções objetivo de posição de nota, 117
- Geração de Novas Melodias, 106
- Gramáticas Gerativas em Composição Algorítmica, 16
- Gramáticas Gerativas para Análise Musical, 20
- Intervalos, 85
- Introdução, 3
- Introdução à Revisão da Literatura, 11
- Introdução a Árvores de Agregação, 49

- Introdução a Medidas de Avaliação de Medidas Aplicadas à Bossa Nova, 75
- Introdução a Problemas de Otimização Multiobjetivo, 35
- Métodos de Otimização Multiobjetivo, 36
- Métodos Históricos, 11
- Medidas Baseadas em Altura, 81
- Medidas Baseadas em Ritmo, 93
- Medidas de Padrões, 97
- Modelos de Markov em Composição Algorítmica, 13
- Modelos de Markov para Geração de Estruturas, 15
- Modelos Ocultos de Markov em Composição Algorítmica, 14
- Motivação, 4
- Objetivos do Trabalho, 6
- Organização do Texto, 8
- Originalidade, 101
- Otimização Multiobjetivo, 35
- Prefácio, xiii
- Primeiras Composições Assistidas por Computador, 12
- Publicações Relacionadas, 143
- Redes de Transição, 21
- Redes Neurais Artificiais em Composição Algorítmica, 27
- Resultados, 105
- Sistemas de Lindenmayer em Composição Algorítmica, 24
- Tonalidade e Dissonância, 82
- Trabalhos Futuros, 140
- Visualização da Qualidade das Soluções em Problemas Multiobjetivo, 38
- Visualização dos Resultados para Problemas com Muitos Objetivos, 42