



FORMALISMOS DA COMPOSIÇÃO ALGORÍTMICA — UM EXPERIMENTO
COM CANÇÕES FOLCLÓRICAS BRASILEIRAS

Lucas Simões Maia

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Luiz Wagner Pereira Biscainho

Rio de Janeiro
Março de 2016

FORMALISMOS DA COMPOSIÇÃO ALGORÍTMICA — UM EXPERIMENTO
COM CANÇÕES FOLCLÓRICAS BRASILEIRAS

Lucas Simões Maia

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE
ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA
ELÉTRICA.

Examinada por:

Prof. Luiz Wagner Pereira Biscainho, D.Sc.

Prof. José Gabriel Rodríguez Carneiro Gomes, Ph.D.

Prof. Vítor Heloiz Nascimento, Ph.D.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2016

Maia, Lucas Simões

Formalismos da Composição Algorítmica — Um Experimento com Canções Folclóricas Brasileiras/Lucas Simões Maia. — Rio de Janeiro: UFRJ/COPPE, 2016.

XIX, 166 p.: il.; 29,7cm.

Orientador: Luiz Wagner Pereira Biscainho

Dissertação (mestrado) — UFRJ/COPPE/Programa de Engenharia Elétrica, 2016.

Referências Bibliográficas: p. 148 – 156.

1. composição algorítmica. 2. modelos de Markov. 3. gramáticas gerativas. 4. algoritmos evolucionários. I. Biscainho, Luiz Wagner Pereira. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

*“Mi terrei felice anche se ciò richiedesse il
sacrificio supremo della vita”*

(Pe. Dante Barbaro, FDP)

*“It was magnificent,” he said, as he took his
seat. “Do you remember what Darwin says
about music? He claims that the power of
producing and appreciating it existed among
the human race long before the power of speech
was arrived at. Perhaps that is why we are so
subtly influenced by it. There are vague
memories in our souls of those misty centuries
when the world was in its childhood.”*

*“That’s rather a broad idea,” I remarked.
“One’s ideas must be as broad as Nature if
they are to interpret Nature,” he answered.*

(A Study in Scarlet, Arthur Conan Doyle)

À minha família.

Agradecimentos

Agradeço, primeiro, ao Deus que tudo ordena e provê.

Agradeço a meus avós, que me forneceram um espaço tranquilo para trabalhar — meu próprio *sanctum sanctorum* — nos últimos meses desta dissertação, propiciando assim a sua conclusão.

Impossível não agradecer a meus pais e minhas irmãs, cuja preocupação velada transmutou-se em carinho e compreensão. Agradeço a toda minha família e padrinhos, que estão sempre ao redor, observando interessados. Agradeço a minha namorada, por sem querer me lembrar de viver cada dia de uma vez e por ser a dona dos abraços pelos quais anseio.

Agradeço a meu amigo-orientador, pelas conversas oficiais e não oficiais, e por não desistir de me guiar nos momentos de dificuldade.

Agradeço aos colegas da Universidade Federal do Rio de Janeiro (UFRJ), desejoso que continuemos juntos por muitos anos.

Agradeço também à professora Sara Cohen, da Escola de Música da UFRJ, por suas fundamentais contribuições no início deste trabalho.

Por fim, agradeço a outros poucos amigos e amigas, especiais ou “avacalhados”, que foram fundamentais em diversos pontos desta caminhada.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

FORMALISMOS DA COMPOSIÇÃO ALGORÍTMICA — UM EXPERIMENTO
COM CANÇÕES FOLCLÓRICAS BRASILEIRAS

Lucas Simões Maia

Março/2016

Orientador: Luiz Wagner Pereira Biscainho

Programa: Engenharia Elétrica

Este trabalho apresenta um estudo minucioso de alguns formalismos do campo da Composição Algorítmica, com uma abordagem histórica e também técnica de cada um deles. Empréstados de diferentes áreas, esses paradigmas são utilizados, em Inteligência Artificial, para produzir novas composições musicais, imitar estilos ou compositores e fornecer material bruto como fonte de inspiração.

Para ilustrar o problema de imitação e resumir todo o conhecimento que foi pesquisado, é proposto um experimento de composição de melodias conformadas ao estilo das cantigas de roda brasileiras. O sistema desenvolvido combina as forças dos algoritmos genéticos, das cadeias de Markov e das gramáticas gerativas como forma de atacar o problema em mãos.

As canções produzidas pelo algoritmo carregam traços da base de dados que foi montada como referência. Isso é digno de nota, dada a pequena quantidade de informação específica que foi codificada no sistema.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ALGORITHMIC COMPOSITION FORMALISMS — AN EXPERIMENT WITH
BRAZILIAN FOLK SONGS

Lucas Simões Maia

March/2016

Advisor: Luiz Wagner Pereira Biscainho

Department: Electrical Engineering

This work presents a thorough study of some formalisms in Algorithmic Composition. A historical but also technical approach is adopted in each of them. Borrowed from different areas, these paradigms of Artificial Intelligence are capable of producing new musical compositions, imitating genres or composers, and providing raw material to be worked upon.

In order to exemplify the imitation task and summarize the acquired knowledge, an experiment of melody composition in the style of Brazilian folk songs is proposed. The developed system mixes the strengths of genetic algorithms, Markov chains and generational grammars as a means to solve the problem at hand.

The songs produced by the algorithm hold a resemblance to the reference database — which is noteworthy, given the small amount of domain-specific information coded into the system.

Sumário

Lista de Figuras	xii
Lista de Tabelas	xiv
Lista de Símbolos	xvi
Lista de Abreviaturas	xviii
1 Introdução	1
1.1 Contextualização	2
1.1.1 Algoritmo e Composição Algorítmica	3
1.1.2 Tarefas e Paradigmas da Composição Algorítmica	4
1.1.3 Teoria da Informação	5
1.1.4 Teoria de análise-síntese	8
1.1.5 Teste de Turing	8
1.2 Problema da Composição	10
1.3 Organização deste Trabalho	10
I Dos Aspectos Teóricos e Técnicos	12
2 Modelos de Markov	13
2.1 Motivação	13
2.1.1 Leis dos grandes números	15
2.2 Cadeias de Markov	17
2.2.1 Convergência de uma cadeia de Markov	19
2.2.2 Estimadores de máxima verossimilhança	21
2.2.3 Geração de sequências	21
2.2.4 Início e término de uma cadeia	22
2.3 Aplicação à Composição	23
2.3.1 O que é um estado?	24
2.3.2 Revisão bibliográfica	24

2.3.3	Outros trabalhos	42
2.4	Limitações	44
2.4.1	Suavização (<i>Smoothing</i>)	45
3	Gramáticas Gerativas	48
3.1	Motivação	48
3.1.1	O estruturalismo de Saussure	49
3.1.2	Bloomfield e a teoria behaviorista	51
3.1.3	O gerativismo de Chomsky	53
3.2	Linguagem e Gramática	56
3.2.1	Hierarquia de Chomsky	59
3.3	Aplicações à Composição	61
3.3.1	O que é um símbolo?	63
3.3.2	Revisão bibliográfica	63
3.3.3	Outros trabalhos	75
3.4	Limitações	77
3.4.1	Gramáticas e Cadeias de Markov	78
4	Algoritmos Evolucionários	79
4.1	Motivação	79
4.1.1	Um pouco de biologia	83
4.2	Algoritmos Evolucionários	85
4.2.1	Representação	90
4.2.2	Função de Aptidão	91
4.2.3	População	92
4.2.4	Seleção de pais	93
4.2.5	Operadores de Variação	95
4.2.6	Seleção de sobreviventes	98
4.2.7	Critério de parada	99
4.2.8	Manutenção de Diversidade	100
4.2.9	A Evolução é um Processo de Otimização?	100
4.3	Aplicações à Composição	103
4.3.1	O que é um indivíduo?	105
4.3.2	Revisão bibliográfica	105
4.3.3	Outros trabalhos	111
4.4	Limitações	112

II	Dos Aspectos Experimentais	114
5	Detalhes Experimentais	115
5.1	Descrição da Base de Dados	115
5.2	Descrição do Experimento	117
5.3	Detalhes Experimentais	119
5.3.1	Uma análise preliminar do <i>corpus</i>	119
5.3.2	O problema da representação	122
6	Resultados e Discussões	125
6.1	Composição de Melodias	125
6.1.1	Caso 1: Função de aptidão unidimensional	125
6.1.2	Caso 2: Função de aptidão bidimensional	127
6.1.3	Resultados	130
6.2	Composição de Ritmos	135
6.2.1	Caso 1: Função de aptidão unidimensional	135
6.2.2	Caso 2: Função de aptidão bidimensional	136
6.2.3	Resultados	137
6.3	Composição de Melodias e Ritmos	138
6.3.1	Resultados	140
6.4	Sobre a seleção de sobreviventes	143
6.5	Arquivos	144
7	Conclusões	145
7.1	Trabalhos Futuros	146
	Referências Bibliográficas	148
A	Conjunto de Medidas para Função de Aptidão	157
A.1	Medidas de <i>Pitch</i>	157
A.2	Medidas de Tonalidade	158
A.3	Medidas de Contorno	159
A.4	Medidas Rítmicas	160
A.5	Padrões	161
A.6	Sobre Repetições de Padrões	162
B	Cálculo dos Parâmetros de Interpolação de Modelos de Markov	164
B.1	Interpolação	164
B.2	Algoritmo de Maximização de Expectativa	165

Lista de Figuras

1.1	Modelo de um sistema de comunicação genérico	5
1.2	Entropia de uma variável aleatória binária com probabilidades p e $(1 - p)$	7
1.3	Modelo de análise-síntese	9
2.1	Experimento de Bernoulli	14
2.2	Representações possíveis para uma cadeia de Markov	19
2.3	Diagrama de blocos do <i>Electronic Music Synthesizer</i>	25
2.4	Representação interna do algoritmo de Brooks <i>et al.</i> para um octograma	30
2.5	Exemplo de anotação do <i>corpus</i>	37
2.6	Exemplos de árvores de prefixos no treinamento do <i>Continuator</i>	40
3.1	Modelo básico de um signo de Saussure, com significado e significante	50
3.2	Exemplo de análise de constituintes imediatos da sentença “o menino adora batata-doce”	52
3.3	Hierarquia de Chomsky	60
3.4	Exemplo de <i>Ursatz</i> — <i>Urfinie</i> e <i>Bassbrechung</i> — de uma análise schenkeriana	64
3.5	Exemplo de árvore reducional	66
3.6	Exemplo de melodia e estrutura de frases em concordância com o sistema de codificação empregado por Bod	69
3.7	Representação da estrutura de uma progressão de acordes	74
3.8	Parte de uma máquina de estados para geração de contornos melódicos	76
4.1	Explicação de Lamarck para as características físicas das girafas	81
4.2	Cromossomos homólogos nas fases da meiose	84
4.3	Superfície de aptidão	86
4.4	Funcionamento de um algoritmo evolucionário	88
4.5	Curva de desempenho típica de um algoritmo evolucionário com base na aptidão média da população	89
4.6	Exemplo de conjunto de Pareto	93
4.7	Exemplo de seleção de pais pelo método da roleta	95

4.8	Operador de mutação básico	96
4.9	Exemplo de mutação (<i>bit-flipping</i>) sobre uma cadeia binária	97
4.10	Operador de recombinação básico	98
4.11	Exemplo de recombinação com cadeias binárias: <i>crossover</i> de um ponto	99
5.1	Perfil dos <i>pitches</i> da base	120
5.2	Histograma das diferentes figuras presentes no <i>corpus</i>	120
5.3	Histograma das diferentes durações presentes no <i>corpus</i>	121
5.4	Histogramas dos comprimentos (em termo do número de compassos) das canções de diferentes <i>subcorpora</i>	122
6.1	Diferentes tipos de representação no sistema	129
6.2	Curva de desempenho típica na composição de melodias com função de aptidão unidimensional	130
6.3	Curvas de desempenho típicas na composição de melodias com função de aptidão bidimensional	131
6.4	Média das distâncias de cada medida à média esperada no teste melódico	132
6.5	Indivíduos melódicos gerados somente por meio das heurísticas	133
6.6	Fronteira de soluções não dominadas encontrada após a última geração	134
6.7	Indivíduos melódicos retirados de diferentes pontos da fronteira de soluções não dominadas	134
6.8	Curva de desempenho típica na composição de ritmos com função de aptidão unidimensional	137
6.9	Média das distâncias de cada medida à média esperada no teste rítmico	138
6.10	Curvas de desempenho típicas na composição de ritmos com função de aptidão bidimensional	138
6.11	Sequência rítmica gerada somente com as heurísticas	139
6.12	Indivíduos rítmicos retirados de diferentes pontos da fronteira de soluções não dominadas	139
6.13	Curva de desempenho típica na composição de melodias e ritmos com função de aptidão unidimensional	141
6.14	Composição completa gerada somente com as heurísticas	141
6.15	Composições completas geradas pelo algoritmo multiobjetivo em duas rodadas diferentes	142
6.16	Composições completas retiradas de diferentes gerações em uma mesma rodada do algoritmo	143

Lista de Tabelas

2.1	Octogramas ordenados e um exemplo de formação das tabelas de probabilidade	31
2.2	Exemplos de padrões de extensão dos N -gramas através da análise de uma sequência de notas	32
2.3	Referências para a Seção 2.3.2	43
3.1	Regras da gramática de Steedman para geração de frases de 12 compassos	72
3.2	Regras da gramática de Johnson-Laird para geração sequências de acordes em oito compassos	74
4.1	Exemplo de ponte temática	106
4.2	Operadores de mutação para compassos do <i>GenJam</i>	109
4.3	Operadores de mutação para frases do <i>GenJam</i>	109
5.1	<i>Corpus</i> transcrito para o experimento	116
5.2	Diferentes <i>subcorpora</i> e análise do número de compassos	117
6.1	Valor médio e de desvio das cantigas do <i>subcorpus</i> SC_5 após análise pelo conjunto de heurísticas do teste melódico	126
6.2	Configuração do teste de composição de melodias com função de aptidão unidimensional	128
6.3	Configuração do teste de composição de melodias com função de aptidão bidimensional	130
6.4	Valor médio e de desvio das cantigas do <i>subcorpus</i> SC_6 após análise pelo conjunto de heurísticas do teste rítmico	135
6.5	Configuração do teste de composição de ritmos com função de aptidão bidimensional	136
6.6	Valor médio e de desvio das cantigas do <i>subcorpus</i> SC_2 após análise pelo conjunto de heurísticas do teste completo	140

A.1 Graus de dissonância associados a cada tipo de intervalo (medido em semitons)	158
---	-----

Lista de Símbolos

$H(X)$	Entropia de uma variável aleatória X , p. 6
L	Comprimento de uma sequência s , p. 37
Q	Espaço de estados, p. 17
R	Conjunto de regras de reescrita, p. 58
T	Conjunto de funções de transição, p. 18
V	Vocabulário ou alfabeto, p. 56
V_N	Conjunto de símbolos não terminais, p. 57
V_T	Conjunto de símbolos terminais, p. 57
X	Variável aleatória, p. 6
b	Bemol, p. 69
κ	Tamanho do torneio, p. 95
λ	Filhos de uma população, p. 95
\mathbf{P}	Matriz de transição, p. 18
\mathcal{G}	Gramática, p. 58
$\mathcal{L}(\mathcal{G})$	Linguagem, p. 57
$E\{\cdot\}$	Valor esperado de uma variável aleatória, p. 15
SC_i	<i>Subcorpus i</i> , p. 119
μ	Tamanho da população, p. 93
\sharp	Sustenido, p. 34
p_c	Probabilidade de recombinação entre os pais, p. 98

p_m	Probabilidade de mutação em cada gene, p. 96
p_{ij}	Probabilidade da transição $q_i \rightarrow q_j$, p. 17
q_m	Estado m , p. 17
s	Sequência, p. 22
s_m^n	Subsequência compreendida entre o m -ésimo e o n -ésimo elementos de s ($n \geq m$), p. 42
w_m	m -ésimo elemento de uma sequência s , p. 42

Lista de Abreviaturas

ANN	<i>Artificial Neural Network</i> , p. 5
ATN	<i>Augmented Transition Network</i> , p. 77
CAAC	<i>Computer-Aided Algorithmic Composition</i> , p. 5
CD	<i>Crowding distance</i> , p. 128
CSP	<i>Constraint Satisfaction Problem</i> , p. 4
DEC	<i>Dinamically Expanding Context</i> , p. 76
DNA	<i>Deoxyribonucleic Acid</i> , p. 83
EMI	<i>Experiments in Musical Intelligence</i> , p. 77
EM	<i>Expectation Maximization</i> , p. 164
EP	<i>Evolutionary Programming</i> , p. 87
ES	<i>Evolution Strategies</i> , p. 87
FPS	<i>Fitness Proportional Selection</i> , p. 94
FSM	<i>Finite State Machine</i> , p. 18
GA	<i>Genetic Algorithm</i> , p. 87
GPR	<i>Grouping Preference Rule</i> , p. 67
GP	<i>Genetic Programming</i> , p. 87
GWFR	<i>Grouping Well-Formedness Rule</i> , p. 67
HMM	<i>Hidden Markov Model</i> , p. 43
LBA	<i>Linear Bounded Automaton</i> , p. 61
MIDI	<i>Musical Instrument Digital Interface</i> , p. 36

MLE	<i>Maximum Likelihood Estimator</i> , p. 21
MOEA	<i>Multiobjective Evolutionary Algorithm</i> , p. 91
NFA	<i>Nondeterministic Finite Automaton</i> , p. 61
NFL	<i>No Free Lunch</i> , p. 89
NFSM	<i>Nondeterministic Finite State Machine</i> , p. 61
NLP	<i>Natural Language Processing</i> , p. 45
NPA	<i>Nondeterministic Pushdown Automaton</i> , p. 61
NTM	<i>Nondeterministic Turing Machine</i> , p. 60
PCA	<i>Principal Component Analysis</i> , p. 110
PDF	<i>Probability Density Function</i> , p. 6
RNA	<i>Ribonucleic Acid</i> , p. 83
SLLN	<i>Strong Law of Large Numbers</i> , p. 16
SMT	Laboratório de Sinais, Multimídia e Telecomunicações, p. 1
U.T.	Unidade de tempo, p. 115
UFRJ	Universidade Federal do Rio de Janeiro, p. vi
WLLN	<i>Weak Law of Large Numbers</i> , p. 15
i.i.d.	Independentes e identicamente distribuídas, p. 14

Capítulo 1

Introdução

O objetivo desta dissertação é apresentar, em detalhe, diferentes formalismos da Composição Algorítmica e desenvolver um experimento para produção de canções ao modo de cantigas de roda do folclore brasileiro. Dessa forma, almeja-se contribuir com os trabalhos nesse campo e ao mesmo tempo fomentar esse tipo de pesquisa no país, particularmente, no laboratório de Sinais, Multimídia e Telecomunicações (SMT-UFRJ). O experimento computacional elaborado combina um algoritmo genético — uma classe especial dos algoritmos evolucionários — com modelos probabilísticos e gramaticais na intenção de imitar o estilo musical já mencionado.

Lejaren Hiller e Leonard Isaacson iniciam o livro “*Experimental Music*”, de 1959, com o seguinte questionamento: “Por que programar um computador digital para gerar música?” [1]. Mais de 50 anos depois, a pergunta parece não fazer tanto sentido, uma vez que os computadores modernos dividem tarefas e não raro substituem os seres humanos em diferentes atividades. Hiller e Isaacson seguem argumentando, porém, que a resposta não é simples, porque traz à tona outras questões relativas à natureza da música e das estruturas musicais, à expressão dos princípios estéticos (de maneira que um computador “entenda”) e, justamente, à capacidade criativa dos computadores [1]. Como não há um modelo completo da criatividade humana e, muitas vezes, os próprios artistas não sabem explicar as origens de suas decisões criativas [2], uma discussão sobre a computabilidade da criatividade, i.e., se esta pode ser tratada sistematicamente, encontra-se ainda em aberto [2]. De fato, no âmbito da “criatividade” computacional ainda há muito a ser explorado, como ilustra o crescimento do campo da Inteligência Artificial nas últimas décadas.

É possível satisfazer parcialmente, contudo, o questionamento principal de Hiller e Isaacson ao considerar como o ambiente do computador permitiu e, por certo, ainda permite investigar rapidamente as estruturas musicais sob diferentes óticas, de forma a caminhar no sentido da compreensão dos aspectos criativos e estéticos que perpassam a mente de um compositor [3]. A Composição Algorítmica busca elucidar, portanto, essas características inerentes ao processo humano de criação,

auxiliando a atividade da composição e distinguindo a máquina do ser humano, em vez de substituir o compositor no seu ofício [1].

1.1 Contextualização

Música é, ao mesmo tempo, arte e ciência [4]. Pode ser definida como a arte de produzir “combinações agradáveis, expressivas ou inteligíveis de tons” [4]. Ou ainda como a ciência que organiza um conjunto “de operações [lógicas] e de relações [de equivalência, de implicação e quantitativas] elementares entre as entidades sônicas ou funções destas entidades” [5]. A própria origem latina da palavra composição (*componere*, “pôr junto”/“unir”) sintetiza essa noção combinacional e organizacional da música, que também é percebida nos processos de improvisação e interpretação.

De maneira genérica, a produção musical consiste na sequência dos seguintes processos [4]: a notação simbólica no papel (pelo compositor); sua tradução em sinais sonoros por meio da voz ou de instrumentos musicais (intérprete); e a atuação destes sinais sobre o mecanismo da audição humana (ouvinte). A ação do compositor, centro de interesse do presente trabalho, ora conforma-se às estruturas estéticas e estilísticas bem estabelecidas, ora quebra estas regras, movida pela inspiração. A criatividade, portanto, seja na música, na pintura ou em qualquer outra arte, se apresenta em dois distintos “sabores” [6]: talento e esforço. Enquanto o primeiro tipo ainda carece de uma definição formal (de caráter psicológico ou cognitivo) [6], o segundo pode ser facilmente desempenhado pelos computadores, seja por “força bruta” ou através de algoritmos mais “inteligentes”. Computadores podem ser considerados a ferramenta ideal para esse tipo de tarefa “regrada”, i.e., atrelada às regras estéticas preestabelecidas, uma vez que somente respondem ao que foram programados para fazer¹.

O advento da Composição Algorítmica pode ser traçado até o início do próprio desenvolvimento da notação musical moderna no Ocidente [7]. Em “*Micrologus*”, o monge beneditino Guido d’Arezzo (ca. 990–1033), cujas principais contribuições são os esquemas de solfejo e a notação musical medieval² (em particular, o emprego do tetragrama para uma representação mais precisa da altura das notas), descreve um sistema para geração de estruturas melódicas a partir de texto. Mapeando automaticamente letras e sílabas em notas e frases melódicas (neumas), o sistema de Guido já delineava o conceito de algoritmo aplicado à música. Outro exemplo notório são os jogos musicais de dados, muito populares no século XVIII, e que, nas suas mais variadas formas, permitiam ao leigo em música “escrever minuetos,

¹A introdução de uma espécie “livre arbítrio” com o uso de geradores de números pseudoaleatórios confere ao computador certa imprevisibilidade; esta não é, contudo, equivalente ao processo de criatividade de um ser humano [1].

²Uma abordagem histórica detalhada da evolução da notação musical pode ser vista em [8].

marchas, polonesas, contradanças, valsas e assim por diante pela seleção de pedaços pré-fabricados de música através de operações de sorte” [9]. Alguns desses jogos são atribuídos a compositores famosos, e.g., Haydn e Mozart, mas estas autorias são passíveis de contestação [9].

Contemporaneamente, os computadores desempenham diversas tarefas musicais, que vão desde a síntese sonora à modelagem física de espaços acústicos, da notação automática à edição de partituras [2]. A Composição Algorítmica, campo ao qual este trabalho se restringe, também muito se beneficiou com a velocidade de processamento das máquinas e sua capacidade de lidar com tarefas cada vez mais complexas. Assim, o computador ou faz as vezes de compositor ou, pelo menos, auxilia no processo de composição, fornecendo motivos, frases ou movimentos inteiros para a escolha do compositor humano.

1.1.1 Algoritmo e Composição Algorítmica

De forma geral, o objetivo de um algoritmo é determinar, em um número finito de passos, a resposta a uma questão ou a solução de um problema [7]. Similarmente, um algoritmo pode ser entendido como um procedimento sistemático através do qual se calcula a mesma saída sempre que um determinado dado ou conjunto de dados lhe é apresentado como entrada.

As definições acima destacam duas fundamentais condições sobre os algoritmos “clássicos”, a saber: (1) terminação (o algoritmo deve produzir a solução em um tempo finito); e (2) determinismo (o algoritmo deve produzir sempre o mesmo e válido resultado para uma dada entrada). Esse tipo de definição, contudo, não pode ser utilizado para caracterizar os métodos que compõem o arcabouço da Composição Algorítmica.

De maneira simplificada, o objetivo de aplicações algorítmicas musicais é criar estruturas melódicas simbólicas a partir de elementos musicais, de acordo com um conjunto de regras particular. Um algoritmo compositor não obedece, necessariamente, a uma condição de terminação e nem sempre é determinístico, e.g., arranjos produzidos para instalações sonoras podem ser executados continuamente e os métodos estocásticos, cujas decisões envolvem probabilidades, permitem gerar composições diferentes com os mesmos valores iniciais [7]. Nierhaus atualiza a definição de algoritmo em aplicações de composição de maneira recursiva. Segundo ele, um algoritmo compositor é “um procedimento abstrato e formalizável que, quando aplicado na geração de estruturas musicais, determina o campo de aplicação da chamada composição algorítmica” [7]. Ademais, o algoritmo compositor está delimitado pela representação simbólica da estrutura musical gerada, ou seja, o trabalho de síntese e a execução da composição não fazem parte de seu escopo.

1.1.2 Tarefas e Paradigmas da Composição Algorítmica

A composição, a interpretação e a experiência auditiva de uma peça musical se concentram, de forma não exaustiva, em aspectos como melodia e ritmo, fraseamentos, contrapontos e harmonia [10]. A função do compositor é combinar, geralmente de forma característica (seu estilo próprio), todos esses aspectos. Tradicionalmente, entende-se que o compositor se guia por um conjunto de regras técnicas e também por sua “sensibilidade auditiva” [1].

Também a máquina, no papel de compositor, deve organizar esses elementos de acordo com uma ordem específica, gerando assim composições agradáveis ou que transmitam uma intenção. Embora as regras estilísticas possam ser descritas diretamente no computador, sua “sensibilidade auditiva” depende inteiramente do que foi formalizado e codificado pelo programador. Grosso modo, dois tipos de tarefas direcionam a ordenação dos elementos sonoros nas composições algorítmicas [7, 11]:

- A imitação de um estilo musical específico, e.g., o estilo de um *corpus*;
- A composição automática em diferentes níveis de automação — desde a criação de ferramentas (e inspirações motivicas) para o compositor até a geração de composições inteiramente novas (ditas “genuínas”).

Essas tarefas se diferenciam, geralmente, pela qualidade das “informações específicas” que são fornecidas ao algoritmo, i.e., se são extraídas ou formuladas³.

As metodologias de Composição Algorítmica no nível mais alto de automação, i.e., com mínima ou nenhuma intervenção humana, sejam elas estratégias para imitação ou para composição automática (genuína), são “emprestadas” de diferentes áreas do campo da Inteligência Artificial. Os principais paradigmas podem ser divididos da seguinte maneira⁴:

- Sistemas de regras e restrições (e.g., problemas de satisfação de restrição, em inglês “*constraint satisfaction problems* — CSPs);
- Métodos estocásticos (e.g., cadeias de Markov);
- Gramáticas gerativas;
- Abordagens evolucionárias;

³Nierhaus demonstra como a fronteira entre essas duas tarefas é, na verdade, nebulosa [7].

⁴A divisão proposta não intenciona ser taxonômica, uma vez que a classificação dessas abordagens não é tratada de forma única na literatura. Para exemplificar, Nierhaus agrupa os algoritmos evolucionários e as redes neurais sob a alcunha de “aprendizagem de máquina”, enquanto relega os modelos de Markov e as gramáticas às técnicas de “processamento de linguagem” [7]. Já Fernández e Vico colocam redes neurais e cadeias de Markov sob o título de “aprendizagem de máquina” [11]; ainda, métodos evolucionários são tratados como sistemas de “otimização” (o que não é uma classificação adequada, como se discute na Seção 4.2.9 deste trabalho) e as gramáticas são colocadas junto aos sistemas de regras.

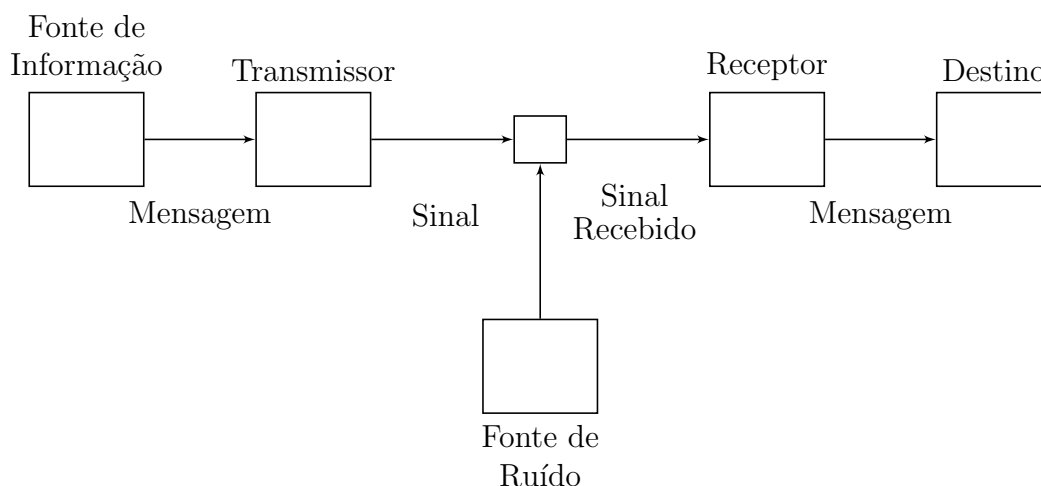


Figura 1.1: Modelo de um sistema de comunicação genérico. Adaptado de [12]

- Aprendizagem de máquina (e.g., redes neurais artificiais, em inglês “*artificial neural networks*” — ANNs).

No nível mais baixo da automação encontra-se a subárea usualmente intitulada “composição algorítmica assistida por computador” (CAAC — *Computer-Aided Algorithmic Composition*) [11], cuja base são as linguagens, *frameworks* e ferramentas gráficas para execução de tarefas de composição [11]. Na CAAC também são colocados os programas que fornecem inspiração para o compositor [11]. As páginas deste projeto fazem menção a diversas pesquisas em Composição Algorítmica automática (em imitação de estilo e composição genuína) e assistida.

Nas seções a seguir são destacados e discutidos brevemente algumas teorias e trabalhos que auxiliam na fundamentação dos conceitos principais em Composição Algorítmica. Em primeiro lugar, discute-se como os conceitos de teoria da informação são aplicados na música. Em seguida, são apresentados alguns obstáculos que devem ser superados nos processos de análise-síntese e também uma forma de avaliação das produções musicais na subárea de imitação de estilo.

1.1.3 Teoria da Informação

Os trabalhos iniciais em Composição Algorítmica beberam da fonte da Teoria da Informação, delineada pelo matemático Claude E. Shannon em “*A Mathematical Theory of Communication*” [12]. Posto de modo simples, nesse artigo seminal da teoria da comunicação, Shannon se dedica a estudar os limites teóricos de compressão e de transmissão de dados, i.e., mensagens selecionadas a partir de um conjunto de diferentes possibilidades [12].

O modelo de um sistema de comunicação é composto por uma fonte de informação e o transmissor, cuja mensagem deve ser enviada através de um canal com

ruído até o receptor (cf. Figura 1.1). Shannon mostra que é possível recuperar o sinal com probabilidade de erro próxima de zero para qualquer taxa de transmissão inferior à capacidade do canal, que pode ser calculada com base nas características do ruído [13]. Ele também argumenta que processos aleatórios como a fala e a música possuem uma “complexidade irreduzível” além da qual o sinal não pode ser comprimido [13] — a essa medida Shannon deu o nome de entropia [12].

Em teoria da comunicação, entropia⁵ expressa o conceito de quantidade de informação, de incerteza ou, similarmente, de liberdade de escolha [12]. Se a fonte pode escolher entre muitas e diferentes possibilidades de evento, há bastante informação. Se, ao contrário, há somente uma possibilidade a ser transmitida, o receptor não tem incertezas. Shannon ilustra esse descrevendo um conjunto de possíveis eventos com probabilidades de ocorrência p_1, p_2, \dots, p_k , extraídos de uma variável aleatória X [12]. O receptor não sabe qual evento será transmitido, embora conheça suas probabilidades. A medida da quantidade de escolha envolvida na seleção do evento, i.e., a incerteza associada à transmissão deste evento, pode ser descrita pela entropia:

$$H(X) = - \sum_{i=1}^k p_i \log_2 p_i, \quad (1.1)$$

em bits⁶. A Figura 1.2 apresenta a variação do valor da entropia de uma variável aleatória binária, em função das probabilidades p e $q = 1 - p$ de seus eventos. A Equação 1.1 e a Figura 1.2 clarificam os conceitos acima destacados. Quando um evento tem probabilidade unitária e todos os demais eventos têm probabilidade nula, então $H(X) = 0$ e não há informação⁷. Caso os eventos sejam todos equiprováveis, i.e., a variável aleatória tem função de densidade de probabilidade (PDF — “*probability density function*”) uniforme, então a entropia é máxima (há mais escolha). Em última instância, a entropia representa o comprimento esperado de um código ótimo para uma variável aleatória, no qual o comprimento de código associado a cada possível evento é dado por $\ell_i^* = -\log_2 p_i$ [13]. Esse processo de codificação ótima também é conhecido como compressão e objetiva remover do sinal da fonte todo tipo de redundância, preservando apenas o que é indispensável à sua posterior reconstrução no receptor [13].

O conceito de “informação” na teoria da comunicação não está relacionado ao uso cotidiano desta palavra, e.g., do significado de uma mensagem, mas às diferentes possibilidades de escolha que são apresentadas à fonte (e ao receptor) [1] ou, como

⁵Não confundir com o conceito de entropia da Física, que é uma medida de “desordem” de um sistema termodinâmico.

⁶A unidade de medida da entropia está relacionada à base do logaritmo. Se, em vez da base binária, fosse utilizada a base natural, a entropia seria medida em nats [13].

⁷Por continuidade, convencionou-se que $0 \log 0 = 0$, uma vez que $\lim_{x \rightarrow 0} x \log x = 0$ [13].

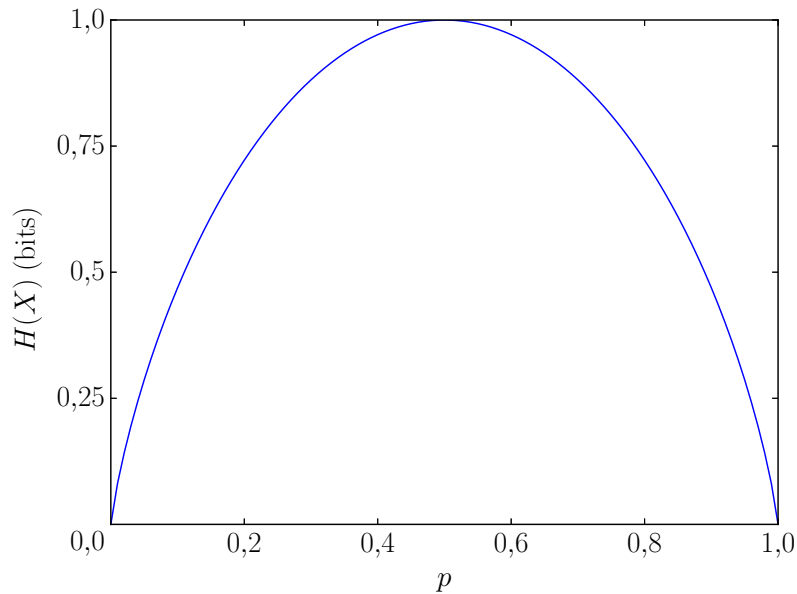


Figura 1.2: Entropia de uma variável aleatória binária com probabilidades p e $(1-p)$.

coloca Shannon, “os aspectos semânticos da comunicação são irrelevantes para o problema de engenharia” [12]. Apesar disso, diversas pesquisas foram conduzidas na tentativa de associar o conceito de entropia aos conceitos de beleza e melodia [1, 4]. Sob a ótica da teoria da informação, a música poderia ser, então, entendida como um sistema discreto⁸ de comunicação [4], e cada composição corresponderia a uma mensagem transmitida pelo compositor (a fonte) para um ouvinte (o receptor) [5].

A interseção entre música e teoria da informação reside, possivelmente, não na interpretação semântica das “mensagens musicais”, mas sobretudo no estudo de sua dimensão sintática [14]. A melodia, descrita como “uma sucessão agradável ou organização de tons” [4], contém elementos de similaridade, regularidade ou simetria. Toda música possui algum grau de redundância [3], que está diluída na expressão da entropia. A redundância é um dos elementos da ordem, da previsibilidade, e o trabalho do compositor, portanto, se assemelha justamente a organizar os tons de forma a extrair “ordem a partir do caos” [1, 3, 4]. Embora todas as nuances do processo de composição (e, a bem da verdade, de qualquer processo criativo) não sejam compreendidas completamente, pode-se intuir que a entropia de uma melodia deve ser baixa o suficiente para que certos padrões possam ser estabelecidos [4]. Ao mesmo tempo, deve ser alta a ponto de incorporar elementos de surpresa, complexidade e sofisticação [4].

Uma pesquisa recente [15] sugere que a “agradabilidade” de uma composição pode ser predita pela facilidade com a qual ela é comprimida. Se a composição é muito redundante (e.g., a repetição de uma única nota ou uma escala ascendente)

⁸Um sistema discreto é, segundo Shannon, um sistema no qual tanto a mensagem quanto o sinal que a representa são sequências de símbolos discretos [12]. Na notação musical, diferentes elementos discretos operam simultaneamente: *pitch*, duração, timbre etc.

ou por demais aleatória (e.g., ruído branco) — os dois extremos do espectro da compressão — seu efeito não será agradável ao ouvinte. A boa música encontra-se entre a ordem e a desordem absolutas, e a medida da entropia pode, portanto, auxiliar a compreensão da estética composicional (se não os “significados” musicais, ao menos as regras sintáticas de suas estruturas).

1.1.4 Teoria de análise–síntese

Conforme mencionado, a informação necessária para um processo de síntese musical em estilo específico (ou, genericamente, o processo de síntese dedutiva de novos elementos em uma classe bem definida) pode ser extraída e representada de diferentes formas, e.g., através de regras de formação e restrições explícitas ou a partir de descrições estatísticas de seus elementos. De uma maneira ou de outra, conforme destacam Brooks *et al.* em [16], o sucesso de tal empreendimento depende tanto do método e da sofisticação do processo de análise como de características do conjunto de amostras selecionadas — o *corpus*.

Evidentemente, um primeiro indicador qualitativo do *corpus* é sua própria a “coerência interna”, i.e., o quanto suas amostras realmente podem ser classificadas como pertencentes a uma mesma classe. Brooks *et al.* apontam, contudo, outras dificuldades de caráter mais “delicado”, cuja compreensão é facilitada através da Figura 1.3. Para um conjunto de n amostras com estrutura comum, S_1, S_2, \dots, S_n , o processo de análise dá origem a uma generalização indutiva \mathcal{G} [16]. O processo subsequente de síntese deriva os elementos S'_1, S'_2, \dots, S'_m e a generalização é classificada como “satisfatória” caso estes novos elementos sejam identificados como pertencentes à classe original [16]. A primeira das dificuldades indicadas pelos pesquisadores diz respeito à complexidade da análise efetuada — uma análise muito simples pode resultar em uma generalização muito abrangente, de forma que os elementos sintetizados não se assemelhem às estruturas da classe [16]. Similarmente, o tamanho do *corpus* também pode inviabilizar a produção de uma generalização satisfatória, i.e., quando há poucas amostras. Essa é a segunda dificuldade [16]. O terceiro e último problema fala da diversidade do *corpus*: se as amostras forem muito semelhantes não há informação suficiente sobre a classe para permitir novidade nas deduções da síntese [16].

1.1.5 Teste de Turing

Em “*Computing Machinery and Intelligence*” [17], o matemático Alan Turing propõe um experimento intitulado “o jogo da imitação” (“*the imitation game*”), que posteriormente ficaria conhecido como “teste de Turing”. A pergunta por detrás do experimento — “uma máquina pode pensar?” [17] — se insere na fundação da área

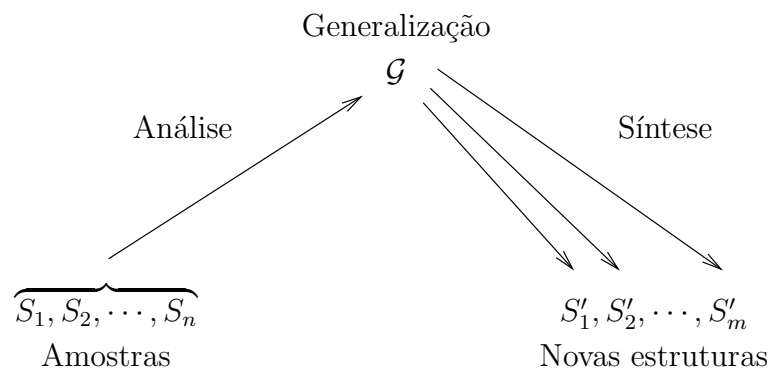


Figura 1.3: Modelo de análise-síntese. Adaptado de [16].

de Inteligência Computacional e tangencia, dessa forma, o conceito de criatividade, central para o presente trabalho. O jogo da imitação é jogado por três pessoas [17]: (A) um homem; (B) uma mulher; e (C) um interrogador. O objetivo do interrogador, que se encontra em local separado dos outros dois jogadores, é determinar qual dos dois é o homem e qual é a mulher [17]. Posto de outra forma, o interrogador deve determinar, ao final do jogo, através de perguntas⁹ quem são indivíduos X e Y que estão jogando com ele, i.e., “X é A e Y é B” ou “X é B e Y é A” [17]. O objetivo de A é fazer com que o interrogador erre essa identificação, enquanto B deve ajudá-lo respondendo verdadeiramente a cada pergunta [17].

Apresentado com a dificuldade de definir o que significam os termos “máquina” e “pensar”, Turing sugere que a pergunta original seja substituída por “pode ser concebido um computador digital que teria sucesso no jogo da imitação?”, i.e., um computador que, ocupando o lugar de A, atrapalhe a decisão do interrogador tantas vezes quanto o homem da configuração inicial [17]? Dessa maneira, o teste de Turing não tem como propósito determinar se computadores são capazes de pensar, mas se o desempenho de um computador pode ser comparável ou indistinguível daquele de um ser pensante. Não é uma análise dos processos internos do computador, mas uma avaliação qualitativa de sua “inteligência”, a partir de eventos externos e observáveis.

Em Composição Algorítmica, se a tarefa de um programa é a imitação de estilo, então este processo de simulação de criatividade pode ser avaliado em uma variante musical do teste de Turing. Nesse caso, o objetivo do ouvinte é determinar, de um conjunto de peças a ele apresentadas, quais foram compostas por um ser humano e quais são produções de um algoritmo compositor. De fato, alguns programas de imitação de estilo musical foram submetidos a esse tipo de avaliação subjetiva (cf. [7, 11, 18, 19]). A opção por esse tipo de análise do desempenho dos algoritmos

⁹A interface entre o interrogador e cada um dos jogadores deve ser tal que não forneça mais informações ao primeiro do que aquelas contidas nas respostas. Dessa forma, a comunicação não pode se dar oralmente ou ainda por escrita. Turing sugere que as respostas sejam datilografadas e exibidas em teleimpressores [17].

surge da inexistência de uma definição formal efetiva para a criatividade [11], que vai ao encontro de uma das motivações da Composição Algorítmica — a busca pela compreensão dos processos internos e subconscientes dos músicos.

1.2 Problema da Composição

No estudo da “forma”, reconhece-se o trabalho do compositor como sendo o de organizar a “matéria” musical, i.e., o que compõe a substância da composição — *pitch*, duração e tonalidade, entre outros. Tal arranjo dos elementos da música deve resultar em uma peça que soe agradável ou que projete um significado emocional no ouvinte. O problema da composição é, em sua forma mais básica, o problema da escolha da articulação do material musical. Cada compositor tem uma forma específica para a manifestação desse arranjo — as características da “fonte” no modelo proposto pela teoria da informação. Também estilos “bem formados” podem ser aglutinados em “fontes” particulares.

O problema da composição é atacado neste trabalho com auxílio de algoritmos estabelecidos no campo da Inteligência Artificial. O objetivo é, após o estudo cuidadoso de alguns desses algoritmos, ilustrar a imitação de estilo. Essa é, possivelmente, a tarefa da Composição Algorítmica que recebe mais atenção dos pesquisadores [7]. O caminho percorrido até o término deste projeto pode ser segmentado em diferentes etapas, a saber: (1) uma análise teórica e técnica de diferentes formalismos da Composição Algorítmica; (2) a escolha de um *corpus* de composições nacionais em estilo simples — as cantigas de roda; (3) a utilização dessa base como referência para a execução de um algoritmo desenvolvido pelo autor que combina algoritmos genéticos, métodos estocásticos e gramáticas gerativas; (4) a variação de alguns parâmetros dos algoritmos; (5) uma análise crítica das composições geradas e a análise da influência dos parâmetros sobre este conjunto.

1.3 Organização deste Trabalho

No restante dessa dissertação, primeiramente os Capítulos 2–4 compõem a Parte I, que revisita a história e teoria por trás de alguns formalismos. O Capítulo 2 trata das cadeias de Markov, utilizadas nos trabalhos pioneiros de Hiller e Isaacson [1, 20] e de Iannis Xenakis[5]. O Capítulo 3 apresenta uma revisão das gramáticas gerativas, um poderoso formalismo de composição que tem origem nas análises sintáticas da linguística. Por fim, as abordagens evolucionárias são discutidas no Capítulo 4, com especial atenção aos algoritmos genéticos.

A Parte II dá seqüência ao texto com a descrição do experimento em Composição Algorítmica que objetiva a imitação do estilo de cantigas de roda brasileiras. A base

de dados e o arranjo experimental são delineados no Capítulo 5 e os resultados são discutidos no capítulo seguinte. O Capítulo 7 expõe as considerações conclusivas e projeções de trabalhos futuros.

Finalmente, o Apêndice A apresenta um conjunto de medidas musicais desenvolvidas por Towsey *et al.* utilizadas no experimento do Capítulo 6 e o Apêndice B explica como pode ser efetuado o cálculo dos parâmetros de interpolação de modelos de Markov.

Parte I

Dos Aspectos Teóricos e Técnicos

Capítulo 2

Modelos de Markov

Os Modelos de Markov recebem seu nome do matemático russo Andrei Andreevich Markov¹ (1856–1922), aluno de Pafnuty Lvovich Chebyshev (1821–1894). Embora seja conhecido por seu trabalho em teoria dos números e análise, foi às áreas de probabilidade e estatística que Andrei Markov mais se dedicou. Instantaneamente publicou artigos, desenvolvendo um novo campo de pesquisa que se tornaria conhecido, 20 anos após suas descobertas iniciais, como “cadeia de Markov” [21].

Neste capítulo, a história e a teoria por trás dos modelos de Markov são brevemente descritas, e sua aplicação no campo da Composição Algorítmica é analisada do ponto de vista histórico. Posteriormente, são apresentadas outras ferramentas importantes que se inspiram nos conceitos de Markov e são discutidas as limitações das técnicas baseadas nesses modelos.

2.1 Motivação

Durante 20 anos, Jacob Bernoulli (1655–1705) se debruçou sobre um conceito probabilístico fundamental conhecido como a lei dos grandes números. Ele mostra em seu trabalho “*Ars Conjectandi*” (publicado postumamente) que, se um evento qualquer tem probabilidade desconhecida *a priori*, essa probabilidade pode ser estimada *a posteriori*, deduzida a partir de observações independentes dos diferentes eventos possíveis. Bernoulli argumenta que, aumentando-se o número de observações, a proporção das ocorrências do evento cuja probabilidade se deseja estimar converge para a probabilidade real desse evento. Tal conceito, Bernoulli descreve como sendo um instinto natural e que “o mais estúpido dos homens, tão só e sem instrução prévia (o que é um verdadeiro milagre)” poderia chegar à mesma conclusão [22]. Bernoulli ilustra seus argumentos com o seguinte exemplo, apresentado também na Figura 2.1.

¹O nome de Markov não é notado de forma única na literatura. Em alguns livros e artigos mais antigos, é possível encontrar sua referência como “Markoff”, outra possível transliteração fonética de seu nome em russo (Марков) [11].

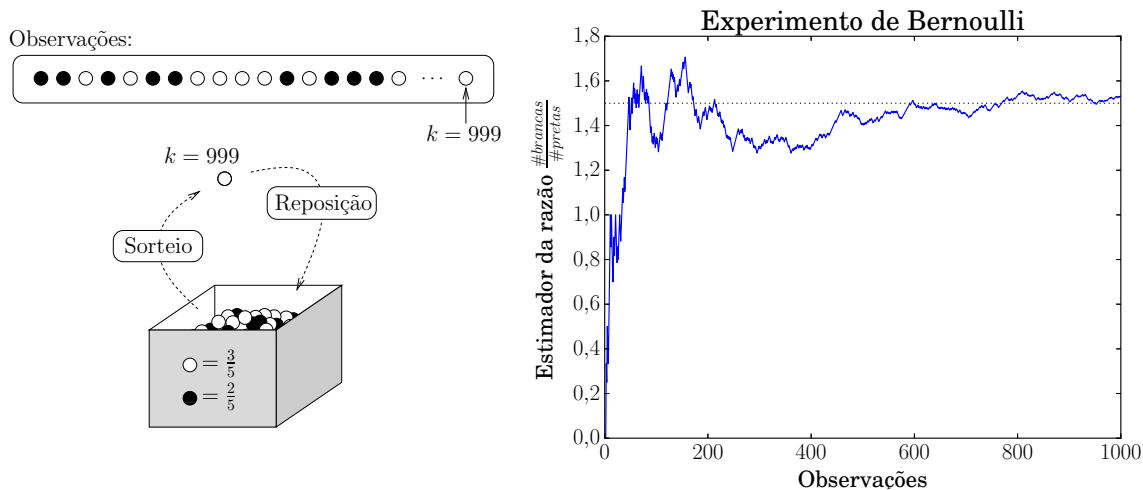


Figura 2.1: Experimento de Bernoulli.

Exemplo 2.1. São colocadas em uma mesma caixa 3 000 pedras brancas e 2 000 pedras pretas. Tais quantidades são desconhecidas por um observador que deseja determinar a razão entre as pedras brancas e pretas através do seguinte experimento: ele retira uma pedra, anota sua cor e devolve a pedra para a caixa antes de repetir todo o procedimento (de forma que o número de pedras na caixa não diminua). À medida que se aumenta o número de observações, torna-se mais provável que a razão encontrada seja a mesma que a do número de pedras, $3/2$.

Bernoulli nota que a razão que se quer determinar experimentalmente entre os dois casos não é precisa, mas admite um intervalo de confiança, i.e., está contida entre dois limites que podem ser feitos tão próximos quanto se queira, quando o número de observações cresce. Sejam pedras dentro de uma caixa ou doenças carregadas pelo ar, Bernoulli discute que é possível determinar por observação a probabilidade de ocorrência de determinado evento. Ele termina dizendo que “decorre de tudo isso que, se as observações de todos os eventos fossem tomadas por toda a eternidade (e, portanto, as probabilidades estimadas se tornariam certezas), tudo no mundo seria percebido como se governado por razões fixas e uma lei constante de mudança” [22].

Siméon Poisson (1781–1840) contribuiu para a lei fraca dos grandes números, mostrando sua aplicação não mais para variáveis i.i.d. (independentes e identicamente distribuídas), mas para observações independentes com distribuição variante no tempo [21]. Posteriormente, Chebyshev generaliza a lei para variáveis independentes e de variância limitada [21, 23].

O conceito de “predestinação estatística” que decorre de trabalhos como o de Bernoulli é, filosoficamente, impactante. Em um artigo de 1902, Pavel Alekseevich Nekrasov (1853–1924), teólogo e matemático russo, e membro da Igreja Ortodoxa Russa, inferiu que a independência era uma “condição necessária à lei dos grandes números”, i.e., a lei só podia ser aplicada em situações nas quais a ocorrência de

determinado evento não altera a distribuição de probabilidade para eventos futuros. É evidente que, no mundo físico, a maior parte dos eventos admite dependência de acontecimentos anteriores, e.g., o clima de uma cidade ou a expectativa de vida de um indivíduo. Assim, a afirmação de Nekrasov restringia tal “predestinação” a casos simples e não práticos, e.g., jogos de dados e experimentos de caixa como o de Bernoulli (também conhecidos como “cadeias de Bernoulli”). Conforme afirma Eugene Seneta em [23], por trás do trabalho de Nekrasov estavam suas próprias convicções religiosas, principalmente, a doutrina do livre arbítrio, que, ele acreditava, a necessidade da independência das variáveis aleatórias estava endossando.

Sua precisão e minuciosidade (além da inimizade que nutria por Nekrasov) levaram Andrei Markov a reagir contra as afirmações pouco fundamentadas daquele artigo [21]. Markov iniciou, então, um estudo detalhado sobre sequências de variáveis dependentes, visando a resolver a restrição dos trabalhos de Bernoulli, Poisson e Chebyshev. É assim, motivado pela vontade de demonstrar o erro de seu adversário Nekrasov, que Markov criou um novo campo de pesquisa — a “teoria das cadeias de Markov” [21].

2.1.1 Leis dos grandes números

A seguir, são descritas brevemente as leis fraca e forte dos grandes números para variáveis independentes. Recordar-se que o objetivo de Markov ao iniciar seu estudo com as cadeias era mostrar a aplicabilidade das leis dos grandes números a algumas sequências de variáveis aleatórias dependentes entre si. Nesse sentido, deseja-se, com a formulação apresentada abaixo, apenas ilustrar de que tipo de convergência tratam estes conceitos probabilísticos.

Definição 2.1. Lei fraca dos grandes números (WLLN — *Weak Law of Large Numbers*).

Seja a sequência de variáveis aleatórias X_1, X_2, \dots, X_k independentes com distribuições quaisquer. Se as médias $\mu_i = E\{X_i\}$ e variâncias $\sigma_i^2 = E\{(X_i - \mu_i)^2\}$ existem, então é possível mostrar que a soma $S_k = X_1 + X_2 + \dots + X_k$ tem média e variância dadas por

$$m_k = \sum_{i=1}^k \mu_i \quad \text{e} \quad s_k = \sum_{i=1}^k \sigma_i^2. \quad (2.1)$$

É dito que a sequência atende à lei fraca dos grandes números se,

$$\Pr \left\{ \frac{|S_k - m_k|}{k} > \epsilon \right\} \rightarrow 0, \forall \epsilon > 0, \quad (2.2)$$

quando $k \rightarrow \infty$ [24]. Se as variáveis são i.i.d., a Equação 2.2 pode ser escrita como $\Pr\{|\frac{S_k}{k} - \mu| > \epsilon\} \rightarrow 0$. Analogamente, a convergência em probabilidade da lei fraca

dos grandes números também pode ser notada como

$$\lim_{k \rightarrow \infty} \Pr \left\{ \frac{|S_k - m_k|}{k} \leq \epsilon \right\} = 1. \quad (2.3)$$

A interpretação da WLLN é que, para uma margem não nula que pode ser escolhida tão pequena quanto se queira, a média amostral $\frac{S_k}{k}$ converge em probabilidade para a média dos valores esperados das variáveis X_i , para k suficientemente grande.

A WLLN vale sempre que os valores de X_i forem limitados, i.e., $|X_i| < A, \forall i$. Contudo, é condição suficiente para sua aplicabilidade que

$$\lim_{k \rightarrow \infty} \frac{S_k}{k} = 0, \quad (2.4)$$

conforme mostrado por Chebyshev [24].

Definição 2.2. Lei forte dos grandes números (SLLN — *Strong Law of Large Numbers*).

A lei fraca dos grandes números dispõe sobre a convergência em probabilidade da média amostral $\frac{S_k}{k}$, isto é, afirma que é provável que, para cada k particular e suficientemente grande, o desvio $|S_k - m_k|$ seja pequeno se comparado ao próprio k [24]. Contudo, a lei fraca admite que valores grandes de $\frac{|S_k - m_k|}{k}$ ocorram em alguns raros instantes e de forma infrequente. Pode existir, por exemplo, uma sequência de variáveis que obedeça à lei fraca, mas para a qual $\frac{|S_k - m_k|}{k}$ supera o limite ϵ uma infinidade de vezes. A lei forte dos grandes números trata de um tipo de convergência conhecido, em probabilidade, como “convergência quase certa”.

Uma sequência X_1, X_2, \dots, X_k atende à lei forte dos grandes números se, “para cada par $\epsilon, \delta > 0$, corresponde-se um K tal que há probabilidade $(1 - \delta)$ ou maior de que, para cada $r > 0$, todas as desigualdades

$$\frac{|S_k - m_k|}{k} < \epsilon, \quad k = K, K + 1, \dots, K + r \quad (2.5)$$

são satisfeitas” [24].

É condição suficiente para a aplicação da SLLN que a sequência de variáveis aleatórias mutuamente independentes atenda ao critério de Kolmogorov [24], isto é, que convirja a sequência

$$\sum_{i=1}^k \frac{\sigma_i^2}{i^2}. \quad (2.6)$$

Evidentemente, obedecer a lei forte dos grandes números implica obedecer a lei fraca, embora a recíproca não seja sempre verdadeira. Assim, sequências de variáveis aleatórias que convirjam fortemente também convergirão em probabilidade, mas algumas sequências, por sua vez, só atendem à lei fraca.

2.2 Cadeias de Markov

Dentre os processos estocásticos, há aqueles que são discretos no tempo e nos estados — as cadeias —, de forma que a cada instante de tempo k inteiro associa-se uma variável aleatória X_k de domínio discreto². Em seu artigo de 1906, Markov descreve uma cadeia simples e específica como sendo “uma sequência infinita $X_1, X_2, \dots, X_k, X_{k+1}, \dots$, de variáveis conectada de forma que X_{k+1} , para todo k , é independente de X_1, X_2, \dots, X_{k-1} caso se conheça X_k ”.

O princípio de Markov estabelece que é possível fazer uma previsão acerca do evento futuro da sequência, X_{k+1} , conhecendo-se somente o valor da variável atual X_k (esta será classificada como de primeira ordem, como se verá adiante). Essa previsão ou probabilidade de transição pode, então, ser descrita por

$$\Pr\{X_{k+1} = q_j | X_k = q_i\} = p_{ij}(k), \quad (2.7)$$

onde X_m representa o valor da variável aleatória no instante m , e q_i e q_j são, respectivamente, os estados de origem e de destino nessa transição. A probabilidade de no instante k ocorrer uma transição do estado q_i para o estado q_j é dada pelo valor de $p_{ij}(k)$. Ademais, uma cadeia de Markov é dita homogênea se esta probabilidade é independente do instante k . Nesse caso, a probabilidade condicional pode ser escrita simplesmente como

$$\Pr\{q_j | q_i\} = p_{ij}, \quad (2.8)$$

onde se lê “a probabilidade de atingir o estado q_j dado o estado atual q_i é igual a p_{ij} ”.

Ao condicionar as transições entre as diferentes observações obtidas de seu espaço de estados, Markov se distancia das cadeias de Bernoulli — a observação X_{k+1} passa a ser dependente da variável aleatória X_k . O exemplo a seguir serve de ilustração do tipo de problemas que Markov tratou em seus artigos:

Exemplo 2.2. Ao observador do Exemplo 2.1 são apresentadas não uma, mas duas caixas. A caixa “1” contém o mesmo número de pedras brancas e pretas. Já a caixa “2” é a mesma do exemplo anterior e, portanto, possui mais pedras brancas que pedras pretas. O observador é instado a, começando pela caixa “2”, retirar uma pedra³. Caso a pedra seja branca, ele a repõe e realiza um novo sorteio na mesma caixa. Caso seja preta, ele a repõe e realiza um novo sorteio na caixa “1”. O raciocínio é análogo na caixa “1”: a retirada de uma pedra preta faz com que o novo sorteio seja realizado na mesma caixa, enquanto uma pedra branca obriga um

²Em outras palavras, o universo de eventos possíveis (espaço de estados) é contável. Em seu trabalho inicial, por exemplo, Markov considerou apenas cadeias binárias, i.e., sequências aleatórias dos estados “0” e “1”.

³O observador poderia, similarmente, começar pela caixa “1”.

retorno à caixa “2”, sempre após a reposição. Pode-se representar as caixas “1” e “2” através dos estados q_1 e q_2 , que significam, respectivamente, que a última pedra retirada é da cor preta ou da cor branca.

No Exemplo 2.2, existem, portanto, quatro transições possíveis:

- Estando na caixa “1”, retira-se uma pedra preta e continua-se no mesmo estado. A probabilidade real dessa transição é $p(q_1|q_1) = p_{11} = 0,5$;
- Estando na caixa “1”, retira-se uma pedra branca e o estado de destino é q_2 . A probabilidade real dessa transição é $p(q_2|q_1) = p_{12} = 0,5$;
- Estando na caixa “2”, retira-se uma pedra preta e o estado de destino é q_1 . A probabilidade real dessa transição é $p(q_1|q_2) = p_{21} = \frac{1}{3}$;
- Estando na caixa “2”, retira-se uma pedra branca e continua-se no mesmo estado. A probabilidade real dessa transição é $p(q_2|q_2) = p_{22} = \frac{2}{3}$;

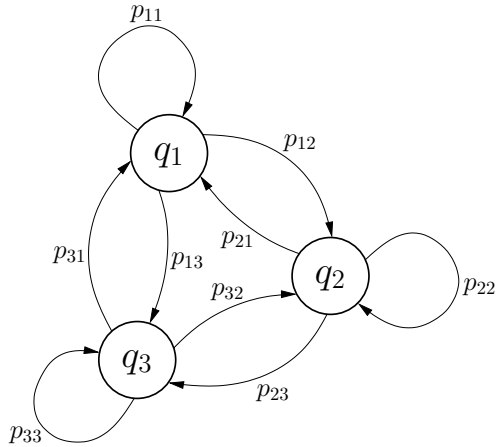
Na cadeia gerada por esse processo, a probabilidade de sortear uma pedra branca ou preta depende, claramente, apenas da última observação. Nesse exemplo, portanto, o observador está lidando com uma cadeia de Markov simples e também homogênea, uma vez que, dada uma transição $i \rightarrow j$, sua probabilidade p_{ij} é fixa (não depende do número da observação, k).

Caso o espaço de estados seja finito, i.e., $Q = \{q_1, q_2, \dots, q_M\}$ ⁴, uma cadeia de Markov simples e homogênea é dita também limitada e pode, então, ser realizada através de uma máquina de estados finitos (FSM — *Finite State Machine*)⁵. Sua representação pode ser feita através de um grafo orientado $G = (Q, T)$, onde novamente Q representa o espaço de estados — cada estado correspondendo a um vértice do grafo — enquanto $T : Q \rightarrow Q$ representa o conjunto de transições $\{q_i \rightarrow q_j\}$ com peso $p_{ij} \in [0, 1]$. Este peso p_{ij} é a mesma probabilidade de transição definida anteriormente. Uma cadeia também é unicamente definida por meio de sua matriz de transição $\mathbf{P}_{M \times M} = \{p_{ij} | q_i, q_j \in Q\}$. Percebe-se que cada linha i da matriz \mathbf{P} está associada a um estado de origem q_i , enquanto cada coluna j está diretamente relacionada a um estado de destino q_j . A Figura 2.2 apresenta um exemplo das duas representações de uma cadeia de Markov com $M = 3$.

Em trabalhos posteriores, Markov considera também cadeias complexas, nas quais a dependência de uma dada variável não está limitada a apenas uma variável precedente [21]. Assim, surgem os modelos de ordem N , nos quais N indica o número de eventos precedentes que contribuem para a condição, e.g., para a ordem $N = 2$

⁴No Exemplo 2.2, o espaço de estados é $Q = \{q_1, q_2\}$.

⁵No caso das cadeias de Markov, tal máquina de estados ou autômato finito é, usualmente, não determinístico e apresenta pesos para cada possível transição.



(a) Grafo orientado

$$\begin{array}{c}
 X_{k+1} \\
 \begin{array}{ccc}
 & q_1 & q_2 & q_3 \\
 \begin{array}{l}
 q_1 \\
 q_2 \\
 q_3
 \end{array} & \begin{bmatrix}
 p_{11} & p_{12} & p_{13} \\
 p_{21} & p_{22} & p_{23} \\
 p_{31} & p_{32} & p_{33}
 \end{bmatrix}
 \end{array}
 \end{array}$$

(b) Matriz de transição

Figura 2.2: Representações possíveis para uma cadeia de Markov.

a probabilidade de a cadeia atingir um estado qualquer no instante $k + 1$ depende do valor X_k e também do valor de X_{k-1} . Também é possível considerar cadeias não homogêneas, para as quais é necessário definir uma sequência de matrizes de transição $\mathbf{P}(k)$, uma vez que as probabilidades de transição entre os estados passam a depender do tempo.

A seguir, o comportamento no tempo de cadeias simples e homogêneas será descrito em maior detalhe.

2.2.1 Convergência de uma cadeia de Markov

Um processo aleatório é compreendido como uma extensão do conceito de variável aleatória que inclui o tempo [25]. Se uma variável aleatória está associada à distribuição de probabilidade dos resultados possíveis de uma observação, um processo aleatório descreve como essa distribuição se comporta ao longo do tempo. Uma cadeia de Markov pode modelar um processo aleatório discreto. Assim, cada grafo ou matriz de transição não representam uma observação sozinha, mas um conjunto de diferentes realizações (*ensemble*).

A cadeia de Markov $\mathbf{X}(k, r)$ descreve, portanto, o conjunto de realizações de um experimento como função no tempo. Escolher uma única realização significa fixar o valor de r e variar o tempo k . Similarmente, uma variável aleatória X_k representa um “retrato” da cadeia no instante k , i.e., é função da distribuição de estados em que se encontra o processo naquele instante.

Para uma realização de uma cadeia de Markov, pode-se definir $\boldsymbol{\pi}_k$ como o vetor linha que indica a distribuição de probabilidade dos estados em um instante k qualquer. Cada elemento desse vetor está associado a um estado e representa a probabilidade de a realização se encontrar nesse estado no instante k (e.g., no Exem-

Exemplo 2.2, a distribuição inicial é $\pi_1 = [0 \ 1]$ se o observador começa no estado q_2 . Similarmente à transição entre os estados em uma cadeia simples, dado o vetor π_k , é possível descobrir a distribuição dos estados no instante $k + 1$, π_{k+1} , com auxílio da matriz de transição da cadeia:

$$\pi_{k+1} = \pi_k \mathbf{P}. \quad (2.9)$$

Retornando ao Exemplo 2.2, é fácil verificar que a matriz de transição é

$$\mathbf{P} = \begin{bmatrix} 0,5 & 0,5 \\ 0,3 & 0,6 \end{bmatrix}. \quad (2.10)$$

Além disso, dada $\pi_1 = [0 \ 1]$, temos que a distribuição dos estados no instante $k = 2$ é $\pi_2 = [0,3 \ 0,6]$, ou seja, a probabilidade de a realização permanecer no estado q_2 nesse instante é o dobro da probabilidade de uma transição para q_1 acontecer. Pode-se aplicar novamente a Equação 2.9 para descobrir qual o estado mais provável no instante $k = 3$:

$$\begin{aligned} \pi_3 &= \pi_2 \mathbf{P} \\ &= \pi_1 \mathbf{P}^2 \\ &= [0,38 \ 0,61]. \end{aligned} \quad (2.11)$$

De maneira geral, o vetor da distribuição de probabilidades dos estados em qualquer instante k pode ser descrito a partir do vetor inicial como

$$\pi_k = \pi_1 \mathbf{P}^{k-1}. \quad (2.12)$$

Ainda no artigo de 1906, Markov mostrou que, à medida que se efetuam observações de uma dada cadeia, o vetor de distribuição dos estados atinge um equilíbrio, i.e.,

$$\lim_{k \rightarrow \infty} \pi_k = \pi \quad (2.13)$$

ou, equivalentemente, $E\{X_{k+j}|X_k = q_i\}, \forall q_i \in Q$, e $E\{X_{k+j}\}$ têm o mesmo limite à medida que k cresce [23]. Este vetor estacionário π é único e independente de π_1 quando a cadeia atende ao princípio da ergodicidade⁶. Dessa forma, Markov refutava o argumento de Nekrasov, de que a independência era uma condição necessária à lei fraca dos grandes números.

Em 1913, Markov apresenta uma análise do poema Eugene Onegin (Евгений Онегин) de Aleksandr Sergeevich Pushkin, descobrindo que, na sequência de 20 000

⁶Para tanto, é necessário que, a cada instante, todos os estados do modelo sejam atingíveis, não necessariamente em uma iteração.

letras, a probabilidade estacionária das vogais é 0,432, a probabilidade de uma vogal ser seguida de uma vogal é 0,128 e a probabilidade de uma consoante ser seguida de uma vogal é 0,663. Este foi o primeiro uso famoso de cadeias de Markov [7].

2.2.2 Estimadores de máxima verossimilhança

O trabalho de Markov com o poema de Pushkin apresenta de maneira clara uma abordagem que toca o objetivo principal dos modelos de linguagem, determinar o comportamento de sequências de letras ou palavras. O que Markov faz é estimar as probabilidades das sequências de símbolos através da contagem das ocorrências de tais sequências. Assim, tomando o processo como homogêneo, o matemático russo utiliza o que é conhecido como estimador de máxima verossimilhança (MLE — *Maximum Likelihood Estimator*). Por exemplo:

$$\Pr_{\text{ML}}\{X_k = [\text{vogal}] | X_{k-1} = [\text{vogal}]\} = \frac{\mathcal{C}(X_{k-1}X_k)}{\mathcal{C}(X_{k-1})} \Bigg|_{X_{k-1}, X_k = [\text{vogal}]} = 0,128, \quad (2.14)$$

onde $\mathcal{C}(\cdot)$ indica a contagem de ocorrências. A probabilidade de uma vogal ser seguida por uma vogal é, portanto, calculada como o número de vezes que uma sequência vogal-vogal ocorre dividido pelo número de ocorrências de vogais ao longo do texto.

Via de regra, uma matriz de transição que modela o processo de Markov pode ser inteiramente estimada dessa forma, i.e., a partir da análise de um *corpus* de dados de treinamento, da contagem das sequências que ocorrem e das subsequências que as precedem [16, 26–28].

2.2.3 Geração de sequências

O modelo de Markov de um processo aleatório discreto não somente descreve a distribuição dos estados ao longo do tempo como também permite gerar amostras de tal processo. Para tanto, é necessário, a cada instante k , sortear um estado em Q conforme a distribuição proposta em $\boldsymbol{\pi}_k$ ou, equivalentemente, de acordo com as linhas da matriz de transição.

Por exemplo, iniciando-se uma cadeia homogênea de primeira ordem com o estado q_i no instante $k = 1$ (que se nota, por simplicidade, $q_i^{(1)}$), sorteia-se o estado da cadeia no instante seguinte, $q_j^{(2)}$, a partir da distribuição discreta $p(\cdot | q_i)$, que equivale à i -ésima linha da matriz de transição \mathbf{P} . A partir de $k = 2$, pode-se repetir o procedimento, selecionando-se agora a j -ésima linha de \mathbf{P} .

Para modelos de ordem 0, utiliza-se uma distribuição de probabilidades unidimensional na descrição do processo. Em modelos de primeira ordem, como acima, a matriz de transição é única e suficiente. Então, é razoável admitir que, para modelos

de ordem N , a matriz de transição que descreve o processo tem $N + 1$ dimensões [27]. Nesta matriz multidimensional, as primeiras N dimensões descrevem cada um dos N estados anteriores que o modelo absorve. A $(N + 1)$ -ésima dimensão representa o estado futuro, que se deseja prever. Tal dimensionalidade do modelo pode significar um problema complicado de se tratar matematicamente, à medida em que a ordem cresce. Nesses casos, contudo, é possível aplicar um raciocínio análogo ao do modelo de primeira ordem [27].

Por exemplo, para ordem $N = 2$, a matriz de transição tridimensional carrega as probabilidades de ocorrência de todas as transições do tipo $q_i^{(k-1)} \rightarrow q_j^{(k)} \rightarrow q_z^{(k+1)}$. Pode-se reduzir sua dimensionalidade expressando-se o contexto no qual o estado do instante $k + 1$ está inserido como um único símbolo “ $q_i^{(k-1)}q_j^{(k)}$ ” (ou “ q_iq_j ” se o processo é homogêneo). Assim, utilizando-se uma matriz bidimensional ($M^2 \times M$), o mesmo procedimento utilizado no caso de primeira ordem torna-se aplicável, i.e., deve-se procurar as distribuições do tipo $p(\cdot|q_iq_j)$ ou a linha equivalente da nova matriz de transição⁷. De maneira genérica, qualquer processo de ordem N pode ser formulado assim como os de primeira ordem, bastando-se tratar cada sequência de estados precedentes como um “novo estado” [27].

Conforme observado no exemplo acima, embora reduza a dimensionalidade da matriz de transição \mathbf{P} e, por isso, seja crucial para a manipulação matemática da mesma, a representação de primeira ordem de modelos com ordem mais alta aumenta consideravelmente a cardinalidade do espaço de estados.

Genericamente, assumindo-se que todas as combinações dos M estados são possíveis no modelo original, de ordem N , a representação de primeira ordem apresentará M^N novos símbolos em seu espaço de estados, um para cada possível histórico.

2.2.4 Início e término de uma cadeia

Um modelo de Markov que atenda ao princípio da ergodicidade permite a geração de uma sequência infinita, dados um estado de início $q_i^{(1)}$ e a seleção ininterrupta de estados futuros para cada instante k . A probabilidade $p(s)$ de a sequência gerada ocorrer pode ser calculada através do teorema de Bayes:

$$\Pr\{X_1X_2X_3X_4 \dots\} = \Pr\{X_1\}\Pr\{X_2|X_1\}\Pr\{X_3|X_1X_2\}\Pr\{X_4|X_1X_2X_3\} \dots \quad (2.15)$$

Assumindo o princípio de Markov com $N = 2$, por exemplo, essa probabilidade

⁷O mesmo artifício pode ser utilizado para estimar as probabilidades de transição do modelo, como visto na subseção anterior. Dessa forma, o estimador de máxima verossimilhança $p(w_k|w_{k-2}w_{k-1})$ para um conjunto de três símbolos $w_{k-2}w_{k-1}w_k$ (modelo de segunda ordem) consiste na razão entre as contagens $\mathcal{C}(w_{k-2}w_{k-1}w_k)$ e $\mathcal{C}(w_{k-2}w_{k-1})$.

pode ser escrita como:

$$\begin{aligned}
p(s) &= \underbrace{\Pr\{X_1\}\Pr\{X_2|X_1\}}_{\Pr\{X_1X_2\}} \Pr\{X_3|X_1X_2\}\Pr\{X_4|X_2X_3\}\dots \\
&= \Pr\{X_1X_2\} \prod_{k=3}^{\infty} \Pr\{X_k|X_{k-2}X_{k-1}\}, \tag{2.16}
\end{aligned}$$

através da qual se observa a necessidade de se calcular também a probabilidade de eventos com menor complexidade (menor ordem). De fato, a distribuição marginal $\Pr\{X_1\}$ é a distribuição dos estados no instante $k = 1$, enquanto $\Pr\{X_2|X_1\}$ representa a probabilidade de transição entre os instantes $k = 1$ e $k = 2$.

É prática comum adicionar um conjunto de N estados iniciais ao modelo, com o objetivo de simplificar o cálculo das probabilidades de uma sequência e também a geração aleatória de novas sequências. O conjunto de símbolos unários que definem tais estados é $A = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$ e os estados estão encadeados de forma que apenas transições $\alpha_1\alpha_2\dots\alpha_i \rightarrow \alpha_1\alpha_2\dots\alpha_i\alpha_{i+1}$, $\forall i < N$, são possíveis (probabilidade unitária). O estado α_1 é o primeiro destes estados iniciais, e $\alpha_1\alpha_2\dots\alpha_N$ representa o último. Nesses casos, para a própria criação do modelo (contagem de ocorrências e cálculo das probabilidades) deve-se considerar a inserção dos símbolos de A no início das sequências de treinamento [16, 28]. Assim, para um modelo “aumentado” de segunda ordem, nota-se a probabilidade de uma sequência s simplesmente como:

$$\begin{aligned}
p^+(s) &= \Pr\{X_{-1}\}\Pr\{X_0|X_{-1}\}\Pr\{X_1|X_{-1}X_0\}\Pr\{X_2|X_0X_1\} \prod_{k=3}^{\infty} \Pr\{X_k|X_{k-2}X_{k-1}\} \\
&\triangleq 1 \times 1 \times \Pr\{X_1|X_{-1}X_0\}\Pr\{X_2|X_0X_1\} \prod_{k=3}^{\infty} \Pr\{X_k|X_{k-2}X_{k-1}\} \\
&= \prod_{k=1}^{\infty} \Pr\{X_k|X_{k-2}X_{k-1}\}, \tag{2.17}
\end{aligned}$$

onde $X_{-1} = \alpha_1$ e $X_0 = \alpha_2$.

Ademais, ao adicionar um estado final, ω , permite-se ao modelo representar a distribuição sobre sequências de diferentes comprimentos e a terminação característica do *corpus* analisado. Nota-se que ω é um estado absorvor — uma vez que seja atingido, não se pode retornar aos outros estados do modelo, i.e., a transição $\omega \rightarrow \omega$ ocorre com probabilidade igual a 1.

2.3 Aplicação à Composição

O potencial de geração das cadeias de Markov foi aplicado em diversos trabalhos de composição musical. O conceito “simples” e estocástico desta abordagem foi,

eventualmente, tratado como “periférico” [27] dentro do campo da Composição Algorítmica e criticado por suas limitações. Ao longo do tempo, novos resultados surgiram e algumas dificuldades foram sanadas, fazendo com que seu uso na música fosse redescoberto.

Os diferentes trabalhos que utilizam esse paradigma podem ser categorizados, contudo, com relação ao que observam como sendo estado (evento) em seus modelos e, mais especificamente, à abordagem com a qual os próprios modelos são criados, i.e., se as probabilidades de transição são formuladas de acordo com parâmetros estruturais específicos ou “aprendidas” através da análise de um *corpus*.

2.3.1 O que é um estado?

Conforme Charles Ames diz em [27], “formalmente, os estados que uma cadeia de Markov pode assumir podem ser qualquer coisa”. Quando aplicados à geração de estruturas musicais, a natureza dos eventos que constituem os estados de um modelo de Markov se restringe, não raro, às notas com suas alturas (*pitches*).

Assim, ao se debruçar sobre as notas do *corpus*, o modelo de Markov absorve, geralmente, um escalar — a duração da nota ou seu *pitch* — ou um vetor — uma combinação de duração, *pitch*, intensidade e articulação, entre outros [18, 27]. Alguns trabalhos se preocupam em modelar não as alturas específicas, mas a sucessão de intervalos entre as notas [1].

Com o desenvolvimento da técnica dentro da área de composição, observou-se que o que era considerado inicialmente uma das fraquezas do modelo — o que Ames chama de “sensibilidade vertical limitada”, i.e., incapacidade de trabalhar com mais de uma nota ao mesmo tempo —, acabou sendo revisitado e melhor estudado. Dessa forma, os modelos de Markov também se mostram capazes de operar sobre elementos de mais alto nível (e.g., acordes, motivos e frases polifônicas) [27].

2.3.2 Revisão bibliográfica

Nesta seção são apresentados, em ordem cronológica, alguns dos principais trabalhos envolvendo cadeias de Markov no campo da Composição Algorítmica.

A Electronic Music Synthesizer

É somente no início da década de 50 que os músicos e pesquisadores começam a reconhecer o potencial de composição que os modelos de Markov apresentam [27]. As primeiras aplicações desses modelos à geração de estruturas musicais são, possivelmente, as de Harry F. Olson e Herbert Belar [26]. Os dois pesquisadores desenvolveram a primeira máquina a levar o nome “sintetizador” [7], a *Electronic Music*

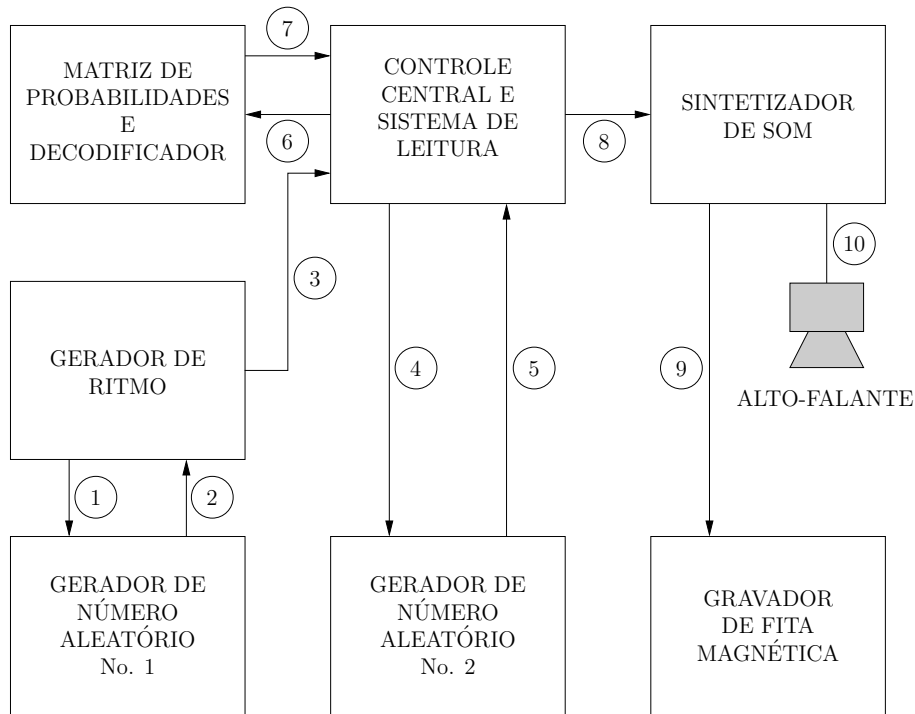


Figura 2.3: Diagrama de blocos do *Electronic Music Synthesizer*. Adaptado de [26].

Synthesizer. Controlada por cartões perfurados, essa máquina era, em sua essência, uma máquina de composição. Através do estudo de uma seleção de 11 melodias de Stephen Foster, transpostas para a tonalidade de Ré maior, Olson e Belar desenvolveram modelos de Markov representativos de tal *corpus* nas ordens 0, 1 e 2, i.e., os pesquisadores levantaram estatísticas das ocorrências das notas sem contexto, no contexto de uma nota e no contexto de duas notas. Também realizaram, separadamente, uma análise rítmica da seleção, computando as probabilidades de diferentes padrões de notas nos compassos $\frac{3}{4}$ e $\frac{4}{4}$ em um modelo de ordem 0. Publicaram esse estudo em 1961, após mais de 10 anos do desenvolvimento da máquina de composição. O objetivo era o de “produzir músicas que serão novas, mas que soarão como canções de Stephen Foster” [26], utilizando a máquina de composição.

A máquina operava a partir das estatísticas obtidas do *corpus*, combinando as saídas de dois geradores aleatórios independentes (um para a seleção do ritmo e outro para a seleção das alturas das notas) e fazendo soar as notas geradas através de um alto-falante⁸, conforme pode ser observado na Figura 2.3. Os resultados finais do experimento foram tomados selecionando-se 29 dos 44 compassos produzidos pela máquina, o que mostra a dificuldade da geração independente de uma estrutura musical coerente. Isto posiciona o trabalho de Olson e Belar “além da fronteira”, i.e., no campo de CAAC. Por fim, os pesquisadores descrevem o resultado como “muito bom” e “conforme o esperado, [...] soa como Stephen Foster”.

⁸As notas também eram guardadas em fita magnética.

A *Illiatic Suite*, para quarteto de cordas

Outra pesquisa igualmente seminal e que aplica a técnica de Markov é a de Lejaren Hiller e Leonard Isaacson [1, 20], ambos da Universidade de Illinois. Trabalhando, a partir de 1955, no computador ILLIAC, os dois desenvolveram diferentes experimentos musicais, unificando seus resultados naquela que é conhecida como a primeira composição gerada por um computador — a *Illiatic Suite*. Nos três primeiros experimentos (“movimentos”) da *Illiatic Suite*, Hiller e Isaacson aplicaram regras e restrições para geração de *cantus firmi*⁹, de contrapontos¹⁰ e harmonias a quatro vozes, e também de ritmos, dinâmicas e instruções de execução mais contemporâneos. Foi então, no 4º movimento da *Illiatic Suite*, que os pesquisadores se debruçaram sobre modelos matemáticos e sua capacidade de abstrair regras essenciais da estrutura musical [1]. Nesse sentido, definiram conjuntos de probabilidades condicionais (de transição) a partir de duas funções específicas: a função harmônica e a função de proximidade. Combinadas, estas funções permitiram a produção de melodias nas quais “a proporção e características de saltos e movimentos graduais, a proporção de intervalos consonantes e dissonantes, e a solução de texturas dissonantes para consonantes, ou vice-versa, foram controladas de maneira bem simples” [1].

Para a geração do que chamaram “música de cadeia de Markov” (*Markoff chain music*), Hiller e Isaacson não trabalharam sobre a duração ou a altura das notas, mas sobre o intervalo que existe entre pares de notas — isto veio de um entendimento de que é “esta sequência, ao invés da de *pitches*, que dá à melodia seu perfil característico” [1, p. 19]. Assim, uma nova melodia poderia ser obtida através de uma sequência de intervalos, adicionados sucessivamente a uma nota inicial (e.g., a nota Dó). As funções harmônica e de proximidade foram utilizadas para classificar os intervalos (do uníssono à oitava) quanto à consonância e à proximidade das notas envolvidas, respectivamente, e.g., um intervalo de quinta justa tem maior peso na função harmônica do que um intervalo de segunda menor, que, por sua vez, apresenta um peso superior na função de proximidade. A função harmônica avalia a sensação de tonalidade, de clareza e ambiguidade harmônicas, enquanto a função de

⁹*Cantus firmus* designa uma melodia preexistente que serve de base temática para uma nova composição polifônica. Convencionalmente, o termo remete ao modo de composição polifônica predominante nos séculos XIV e XV, de forma particular, para corais de música sacra [29].

¹⁰Contraponto refere-se à combinação de duas ou mais linhas melódicas de acordo com princípios técnicos que regulam como essas linhas soam simultaneamente. Embora seja uma “propriedade da polifonia” [29], o contraponto não pode ser separado da “natureza da melodia” [29], isto é, as linhas melódicas devem possuir coerência interna e produzir um sentimento de continuidade. A relação linear entre as melodias que surge através do contraponto deve atingir um “equilíbrio entre a individualidade das linhas e sua combinação” [29], ou seja, a estrutura particular de cada linha melódica deve reforçar a relação polifônica, e vice-versa. Johann Joseph Fux (1660–1741), em sua obra “*Gradus ad Parnassum*”, descreve e divulga um conjunto de regras de contraponto, ao estilo de Giovanni Pierluigi da Palestrina (1525–1594). Uma derivação desse conjunto pedagógico de regras é utilizado ainda hoje para o ensino de contraponto nas escolas de música.

proximidade trata dos movimentos graduais e dos saltos intervalares. Munidos destas duas funções¹¹ e de sua combinação simples (através da adição), os pesquisadores montaram cadeias de Markov nas ordens 0, 1 e também com ordem variável.

Nas cadeias de ordem 0, aplicaram diretamente as distribuições determinadas pelas funções vistas acima, observando que algumas regras de contraponto estrito de primeira espécie podem ser modeladas de forma similar. Notaram que foi gerado, através dessa estratégia, todo o espectro de música — da mais fortemente tonal até a atonal¹² (além de músicas classificadas como “antitonais” [1]). Embora o controle da proporção de intervalos consonantes e dissonantes se desse através da função harmônica, uma cadeia de ordem 0 só foi capaz de imprimir uma tonalidade nas músicas geradas “por sorte [...] e não como resultado do projeto” [1].

Continuaram, pois, com cadeias de primeira ordem, atrelando a escolha do intervalo I_k ao intervalo anterior I_{k-1} a partir de modificações correlatas nas funções-base. Tais modificações atuavam de forma a reduzir a probabilidade de repetição em I_k do intervalo escolhido para I_{k-1} , generalizando, dessa forma, outras regras contrapontuais. Com esse controle, as novas peças apresentaram um perfil médio de alternância entre intervalos consonantes e dissonantes (como efeito direto do uso da função harmônica) e entre movimentos graduais e grandes saltos (em razão da aplicação da função de proximidade).

Em vez de continuar aumentando a ordem do modelo, Hiller e Isaacson optaram por aplicar este conceito com algumas considerações estruturais da importância das notas na música. À vista disso, produziram músicas de cadeia de Markov “nas quais as notas que ocorressem em tempos fortes da música, escolhendo arbitrariamente um compasso $\frac{6}{8}$, eram tomadas como dependentes de uma das funções geradoras, enquanto as notas que ocorressem em tempos fracos eram consideradas como dependentes da função geradora restante” [1]. Dessa maneira, novas peças eram geradas a partir da sobreposição de duas estruturas — uma de longo alcance e outra de curto alcance — e tal interação podia ser obtida de quatro formas diferentes, fazendo com que as funções harmônica e de proximidade (nas ordens 0 ou 1) assumissem, cada uma, a responsabilidade por uma dessas estruturas.

¹¹Os pesos atribuídos pelas funções a cada intervalo podiam ser alterados de acordo com operações de adição ou subtração aplicadas sobre todos os valores simultaneamente (com saturação), de modo a reforçar a presença de um tipo específico de intervalo ou uniformizar a contribuição dos intervalos na geração das sequências. Também era possível inverter os pesos no espaço dos intervalos, e.g., fazendo com que o trítono fosse o mais provável pela função harmônica, em vez de o uníssono.

¹²Hiller e Isaacson definem música tonal como sendo aquela que utiliza um *pitch* fixo como referência, enquanto na música atonal essa referência não pode ser discernida. Assim, a coerência e a significância das notas emanam de sua relação com um centro tonal específico durante certo intervalo de tempo [1]. De maneira geral, uma peça tonal é “unificada harmonicamente pela força de suas funções tônicas, que, por sua vez, são estabelecidas, restabelecidas ou reforçadas principalmente pela relação dominante-tônica (V-I)” [29]. Já a música atonal é aquela caracterizada pelo “enfraquecimento ou supressão das condições que definem a tonalidade” [29].

Por fim, Hiller e Isaacson realizaram mais algumas gerações com o que chamaram de “processo de i -ésima ordem” [1]. Os pesquisadores simplificaram o conceito de tonalidade da música de forma a entendê-lo como uma memória associada ao primeiro *pitch* em determinada linha melódica. O processo de i -ésima ordem espelhou esse conceito, relacionando os tempos fortes diretamente com a primeira nota da música através da função harmônica, enquanto os tempos fracos mantinham uma relação de primeira ordem com o tempo forte imediatamente anterior através da função de proximidade. Essa nova estrutura permitiria a modulação de maneira quase automática — seria necessário apenas descrever um conjunto de regras para a mudança do centro tonal, i.e., para uma nova nota de referência —, mas Hiller e Isaacson fizeram isso manualmente e no final de todo o processamento, através da transposição das notas a cada seis compassos, de acordo com uma cadência pré-determinada. A estrutura também possibilitou um movimento tonal em direção à última nota (novamente, escolhida como a nota Dó). Para tanto, os pesquisadores fizeram descontos sucessivos sobre a função harmônica a cada dois compassos, de modo que, ao chegar na nota final, o único intervalo possível, com relação à primeira nota, fosse o uníssono.

Para consolidação do 4º movimento da *Illiad Suite*, Hiller e Isaacson juntaram os resultados dos diferentes experimentos anteriormente descritos, selecionando e combinando as várias monofonias geradas em duas ou quatro vozes¹³, após a devida transposição por oitavas, para adequar a melodia à faixa do instrumento correspondente àquela voz. Alteraram também algumas estruturas internas, mantendo apenas a coda¹⁴ desse movimento, correspondente ao experimento com os processos de i -ésima ordem, intocada. O andamento rápido e o compasso $\frac{6}{8}$ foram escolhidos arbitrariamente para contrastar com os movimentos anteriores. Foi atribuída ao computador apenas a tarefa de criação da melodia e não foram feitas mais considerações rítmicas, ao que se supõe que foi repetido um padrão rítmico simples com colcheias (*ostinato*) e que *legati* foram utilizados na transcrição quando a saída calculada pelo ILLIAC era um uníssono. Algumas indicações de dinâmica foram escritas posteriormente pelos pesquisadores.

Hiller e Isaacson observaram que o código necessário para realização desses experimentos foi consideravelmente mais simples do que o utilizado na programação do contraponto. De maneira geral, os resultados obtidos, especialmente a coda do movimento, parecem “expor, de forma satisfatória, todas as características pretendidas” [1]. A falta de preocupação com a combinação harmônica das diferentes vozes, que, de fato, o modelo proposto não comportava, é talvez sua maior fraqueza. Con-

¹³A essa realização chamaram pseudopolifonia [1].

¹⁴Em música instrumental, coda é a seção conclusiva de uma composição, que ocorre após a conclusão estrutural desta [29].

tudo, este trabalho é um dos pioneiros no uso de métodos de Markov no campo da CAAC, lançando fundações e servindo de base para muitos outros que o sucederam.

Um estudo sobre três efeitos adversos

Também no final da década de 50, F. P. Brooks Jr., A. L. Hopkins Jr., P. G. Neumann e W. V. Wright [16] se debruçaram sobre as técnicas de síntese dedutiva de novas composições, a partir da análise de um conjunto aceitável de amostras.

Um dos objetivos dos pesquisadores era estudar, através do método das cadeias de Markov e de um corpus de 37 hinos para coral, a existência e o comportamento de três efeitos que dificultam o processo de síntese, conforme o descrito na Seção 1.1.4. Segundo Brooks *et al.* [16], esses três efeitos se dão, de maneira específica no caso do processo de análise-síntese de Markov, da seguinte forma:

1. quando a ordem de análise é baixa, o modelo de Markov “aprendido” é por demais simples e as estruturas geradas pelo processo dedutivo não se assemelham aos elementos da “classe” original (o estilo que se quer imitar);
2. quando a base amostral (*corpus*) é pequena, não há informação suficiente para extrair e formular uma generalização adequada, tornando-se despropositada qualquer tentativa de sofisticação do modelo (aumento de ordem);
3. quando o *corpus* é pouco diversificado, a generalização induzida não consegue senão replicar os mesmos elementos originais e, novamente, qualquer aumento de ordem revela-se infrutífero (o modelo já “absorveu” toda a informação das amostras)¹⁵.

Para que o modelo proposto fosse mais simples, os pesquisadores restringiram a análise do corpus às linhas melódicas, escolhendo amostras musicais com estrutura métrica similar: hinos tradicionais, compostos por duas frases principais em compasso quaternário simples (tendo a semínima como unidade de tempo), com duração total de oito compassos e iniciando em anacruse, no último tempo do compasso. Como nenhum dos hinos apresentasse notas com duração mais curta do que a da colcheia, escolheram esta figura como unidade rítmica básica para representação no computador, i.e., o tempo foi quantizado, em função dessa unidade, em 64 células com duração de uma colcheia. Por conseguinte, cada célula temporal continha uma representação equivalente ao *pitch* da nota que ali se iniciara ou a indicação de que uma nota anterior continuara a soar.

As amostras foram transpostas para a tonalidade de Dó maior e a extensão resultante de valores de *pitch* em quatro oitavas foi codificada nos números inteiros

¹⁵O efeito da pouca diversidade apresenta-se anterior ou simultaneamente ao efeito do tamanho.



Sequência de *pitches*: $G\bar{G} C\bar{C} A\bar{A} C D$
 Representação interna: 36 37 26 27 32 33 26 22

Figura 2.4: Representação interna do algoritmo de Brooks *et al.* [16] para um octograma. A barra sobre o *pitch* indica que a nota foi sustentada.

de 2 até 99. Os números pares simbolizavam os ataques das notas e os ímpares indicavam suas sustentações, de acordo com a quantização temporal anteriormente descrita. A Figura 2.4 apresenta um exemplo do funcionamento dessa notação.

As análises de Brooks *et al.* foram realizadas até a sétima ordem¹⁶ do modelo de Markov. Primeiramente, todas as sequências de oito notas (“octogramas”) presentes no *corpus* foram determinadas e ordenadas em uma matriz de dados, de acordo com a representação numérica interna do programa. Essa etapa organizou todos os octogramas de acordo com seus “heptagramas” iniciais, todos os heptagramas de acordo com seus “hexagramas” iniciais, e assim por diante. Nessas circunstâncias, uma tabela organizada de sequências nas ordens inferiores poderia ser obtida a partir da matriz ordenada “realizando-se *shifts*, em um número apropriado de células para a direita, de todos os octogramas, assim eliminando as notas que não são de interesse” [16].

De posse dessas tabelas, os pesquisadores foram capazes de encontrar, para cada ordem, a probabilidade de uma nova nota ou a continuação de uma nota prévia acontecer após certo contexto executando a estimação de máxima verossimilhança sobre as contagens observadas (conforme visto na Seção 2.2.2). Os resultados foram armazenados em novas tabelas. A Tabela 2.1 ilustra o processo de formação das tabelas de frequência relativa e também de probabilidade acumulada. Nessas circunstâncias, para cada ordem N , o número de tabelas contendo as probabilidades calculadas era equivalente ao número de N -gramas distintos no *corpus*.

Brooks *et al.* inseriram, no início de cada hino, um bloco de “pausas iniciais” representadas pelo símbolo “00”. Com esse artifício (que remete imediatamente ao exposto na Seção 2.2.4), para uma dada ordem N , o primeiro $(N + 1)$ -grama observado em um hino continha N pausas, o segundo N -grama continha $(N - 1)$ pausas, e assim por diante. Então, cada hino apresentava 64 octogramas, um por célula de nota presente no hino. Os pesquisadores reportaram que o conjunto de 37 hinos forneceu um total de 2 368 octogramas, dos quais 1 701 eram distintos.

¹⁶Brooks *et al.* indicaram a ordem do modelo de análise através da letra “ m ” [16]. O valor de m está relacionado ao comprimento da sequência analisada, que chamaram “ m -grama” (“ m -gram” [16]). Dessa forma, a predição de Markov foi feita utilizando-se apenas $(m - 1)$ símbolos e, quando os pesquisadores afirmam estar indo da primeira à oitava ordem, deve-se, por coerência com a teoria exposta até aqui, entender ordens $N = 0$ até $N = 7$.

Tabela 2.1: Octogramas ordenados e um exemplo de formação das tabelas de probabilidade. A tabela também ilustra contagens para a ordem inferior, necessária para a estimação de máxima verossimilhança. Adaptado de [16].

Célula								Contagem ($m = 8$)	Contagem ($m = 7$)	Frequência relativa	Prob. acumulada
1	2	3	4	5	6	7	8				
36	37	26	27	32	33	26	22	1	1		
36	37	26	27	32	33	26	22	2	2	2/8	2/8
36	37	26	27	32	33	26	27	1	3		
36	37	26	27	32	33	26	27	2	4		
36	37	26	27	32	33	26	27	3	5		
36	37	26	27	32	33	26	27	4	6		
36	37	26	27	32	33	26	27	5	7	5/8	7/8
36	37	26	27	32	33	26	28	1	8	1/8	8/8
36	37	26	27	32	33	32	33	1	1		
36	37	26	27	32	33	32	33	2	2	2/2	2/2
36	37	32	33	32	33	32	33	1	1	1/1	1/1
36	37	32	33	32	33	36	37	1	1		
36	37	32	33	32	33	36	37	2	2	2/3	2/3
36	37	32	33	32	33	36	42	1	3	1/3	3/3

Para síntese de melodias originais, Brooks *et al.* seguiram o processo de produção de maneira similar ao que está descrito na Seção 2.2.3. Assim, selecionaram o $(N+1)$ -grama inicial com o auxílio da tabela de probabilidades obtida a partir do N -grama formado apenas por pausas. O $(N + 1)$ -grama seguinte foi escolhido através da tabela referente ao N -grama que contivesse $(N - 1)$ pausas e a nota selecionada anteriormente. O processo continuou até que todas as 64 notas já tivessem sido determinadas. Os pesquisadores atentaram para o fato que “a codificação de pausas iniciais precedendo cada hino [...] permitiu ao processo de síntese operar de maneira uniforme, desde seu princípio” [16].

Além das restrições implícitas absorvidas pelo modelo de Markov, Brooks *et al.* submeteram a seleção de cada nota a algumas restrições explícitas que diziam respeito: (1) à estrutura rítmica; e (2) ao *pitch* da nota terminal.

Para a primeira, elaboraram diferentes padrões de ataques, notas sustentadas e “tanto-faz”. Algumas considerações eram comuns a todos os padrões, e.g., “a primeira nota de cada compasso deveria ser um ataque e não uma continuação” e, também, “as duas frases principais deveriam terminar em uma mínima pontuada” [16]. A segunda limitação se deveu à observação de que, após a transposição, todos os hinos terminavam na mesma nota, “o Dó acima do Dó central” [16]. Na aplicação dessas restrições, os pesquisadores utilizaram o que na literatura é conhe-

Tabela 2.2: Exemplos de padrões de extensão dos N -gramas através da análise de uma sequência de notas. Os padrões estão separados de acordo com o tipo de célula inicial (ataque ou sustentação) e classificadas quanto ao número de notas diferentes que contemplam (extensão máxima). Adaptado de [16].

Sequência: ... $C\bar{C} D\bar{D} E\bar{E} F\bar{F}$...

Ordem N	$(N + 1)$ -grama		Extensão máxima
	Ataque	Sustentação	
3	$C \bar{C} D \boxed{\bar{D}}$	$\bar{C} D \bar{D} \boxed{E}$	3
4	$C \bar{C} D \bar{D} \boxed{E}$	$\bar{C} D \bar{D} E \boxed{\bar{E}}$	3
5	$C \bar{C} D \bar{D} E \boxed{\bar{E}}$	$\bar{C} D \bar{D} E \bar{E} \boxed{F}$	4
6	$C \bar{C} D \bar{D} E \bar{E} \boxed{F}$	$\bar{C} D \bar{D} E \bar{E} F \boxed{\bar{F}}$	4

cido como “algoritmo de geração-e-teste” (*generate-and-test algorithm*), i.e., se após certo número de tentativas a nota gerada não satisfizesse as restrições impostas, o processo de geração era reiniciado.

Variando-se a ordem de análise-síntese e os padrões rítmicos, os pesquisadores geraram, de um total de 6 000 tentativas, cerca de 600 hinos completos e que atendiam às restrições. Observaram que a taxa de sucesso, i.e., o percentual de hinos válidos gerados, apresentou similaridades entre as ordens 3 e 4, e entre as ordens 5 e 6, que podem ser entendidas como consequências diretas do processo de quantização temporal e da restrição a padrões rítmicos repletos de semínimas. Porque cada semínima ocupa, na representação interna do computador, duas células temporais, há um maior número de restrições implícitas sendo introduzidas no modelo ao se passar de uma ordem N par para a próxima ordem (ímpar) do que ao mudar de uma ordem ímpar para a próxima (par). Esse processo pode ser compreendido com o auxílio da Tabela 2.2, observando-se o número de notas diferentes incorporadas em cada padrão para alguns valores de ordem N .

Os pesquisadores analisaram, por fim, composições geradas com diferentes ordens. Destaca-se que:

- Para $N = 0$, a síntese é aleatória¹⁷. Os hinos gerados não são simples de cantar e apresentam diversos intervalos antinaturais.
- Para $N = 1$, foram observadas diferentes sequências de notas que não ocorrem no *corpus*, mas permitidas pelo uso de uma ordem baixa.
- Para $N = 3$ e $N = 4$, as composições sintetizadas são mais regulares. Contudo, ainda foi possível verificar um resquício do primeiro efeito adverso — a

¹⁷Para as cadeias de ordem 0, foi necessário considerar cada célula gerada que contivesse um valor ímpar (sustentação de nota) como um indicativo de continuação da nota imediatamente anterior.

ocorrência de longas sequências ascendentes e descendentes de notas, que não existem dessa forma nas amostras da base.

- Para $N = 5$ e $N = 6$, que apresentaram as menores taxas de sucesso (maior dificuldade no atendimento das restrições), atinge-se o que os pesquisadores denominaram “meio termo”, e uma das novas composições dada como exemplo não imita nenhum hino do *corpus* para mais do que quatro semínimas consecutivas, embora apresente ainda uma sequência descendente de notas de tamanho intermediário.
- Para $N = 7$, o modelo degenera (terceiro efeito). Alguns dos novos hinos são cópias integrais das amostras do *corpus*, enquanto muitos outros copiam cada uma de suas duas frases de um hino diferente. Brooks *et al.* mostram um hino que foi formado a partir da junção de pedaços de 3 dos hinos originais. A transição entre os dois primeiros pedaços acontece no meio da primeira frase, em uma seção de 7 células que era comum entre dois desses hinos, e a transição para o último pedaço acontece no final da primeira frase.

Para os pesquisadores, seria interessante que se estabelecesse uma “medida de complexidade estrutural em termos da informação contida [no *corpus*]” [16]. Esse desejo alude, essencialmente, ao conceito de entropia visto na Seção 1.1.3. Uma medida como essa permitiria determinar a quantidade de informação presente nas amostras e qual o teor da limitação imposta pelo conjunto de restrições. Analisando esses valores, poder-se-ia calcular para que ordens os efeitos previstos pela teoria de análise-síntese se manifestariam e escolher o valor adequado para que a produção da cadeia de Markov fosse identificável como pertencente ao estilo definido pela base.

Um modelo “explodido”

Em “*The Markov Process as a Compositional Model: A Survey and Tutorial*” [27], o pesquisador Charles Ames discursou sobre a aplicação musical das cadeias de Markov e apresentou a *Demonstration 4*, um dos resultados de seus experimentos conduzidos entre 1983 e 1984 para clarinete solo [30].

Ao abordar as características desse paradigma composicional, Ames destacou que, embora os modelos de Markov sejam reconhecidos principalmente por suas capacidades preditivas, algo que está relacionado ao sentido de expectativa na música, outro fator muito importante para o “juízo estético” empreendido pelo ouvinte é o equilíbrio entre os elementos da composição. Considerando-se esse conjunto de elementos como o “espaço de estados”, a própria expressão de convergência dos modelos de Markov para uma distribuição estacionária (cf. Seção 2.2.1) se mostra capaz de comportar a satisfação dessa “necessidade estética” levantada por Ames.

Ao algoritmo descrito nesse artigo foi atribuída a responsabilidade de gerar notas e pausas de forma a preencher 60 compassos, no compasso $\frac{2}{4}$. Para tanto, Ames projetou um conjunto de modelos de Markov, com cada tabela de probabilidade descrevendo o comportamento temporal das seguintes características musicais:

1. a “duração média” dos eventos (pausas ou notas), relativa à duração de uma semicolcheia;
2. a “articulação” entre pausas e notas, expressa na probabilidade de o novo evento ser uma pausa;
3. o “registro” em que as notas se encontram, correspondendo à divisão dos *pitches* em regiões de comprimento de uma oitava (com superposição);
4. o “grau cromático” das notas, restrito ao registro em que estão localizadas.

O algoritmo de Ames operava esses modelos a cada iteração, executando primeiro a seleção da “duração média”. Após decidir se o evento era uma nota ou uma pausa (através da probabilidade expressa no modelo para a “articulação”), perturbava-se de maneira aleatória o valor de duração selecionado. Finalmente, caso o novo evento fosse uma nota, o algoritmo cerceava seu valor a um determinado “registro” (escolhido com auxílio do terceiro modelo) e escolhia o “grau cromático” dentro dessa oitava por meio de uma cadeia de primeira ordem sobre os intervalos.

Cada tabela foi cuidadosamente projetada por Ames. Por exemplo, para a “duração média”, a probabilidade de permanência no estado atual foi feita maior para os estados correspondentes às menores durações, e as transições para os demais estados seguiam uma distribuição uniforme. No caso do “registro”, uma mudança “brusca”, e.g., da região “C \sharp 4–C5” para “D5–C \sharp 6”, foi considerada como pouco provável, contrastando com as mudanças para regiões imediatamente vizinhas. A tabela utilizada na determinação dos graus cromáticos foi descrita de forma que sequências de intervalos que apresentassem “identidades cromáticas”¹⁸ ou correspondessem a “tríades maiores, tríades menores, tríades aumentadas e tríades com duas consonâncias perfeitas [...] em qualquer inversão ou abertura” [27] possuísem probabilidade nula. O objetivo de Ames com essa distribuição de probabilidades era o de “encorajar um estilo consistentemente dissonante” [27], que valorizasse intervalos de segunda, de sétima e trítonos.

Por essas razões, o propósito da *Demonstration 4* não era o da imitação de um estilo específico qualquer; tampouco houve a preocupação de se gerar uma composição com início e fim definidos musicalmente — o resultado obtido foi apenas “a fotografia de um processo que poderia continuar indefinidamente” [27]. Nesse

¹⁸Notas com o mesmo nome, i.e., pertencentes à mesma classe de *pitch*.

trabalho, Ames deu nova interpretação ao “estado” de uma cadeia. Ao tratar cada característica musical separadamente, se distanciou ainda mais do processo de composição desempenhado por um ser humano. Além disso, introduziu, em seu modelo “explodido”, aleatoriedades que extrapolam o previsto por cada modelo de Markov, dando maior capacidade ao algoritmo de “surpreender”.

“Anotando” o *corpus*

Em 1999, Dan Ponsford, Geraint Wiggins e Chris Mellish [28] conduziram um experimento sobre o movimento harmônico em um *corpus* formado por sarabandas francesas do século XVII¹⁹. Dentre os requisitos iniciais dos pesquisadores, destacam-se [28]:

- a extensibilidade do algoritmo a outros *corpora*;
- um uso mínimo de conhecimento musical;
- e a fidelidade à forma de composição analisada, quando algum conhecimento musical for aplicado.

Embora ele seja regularmente citado como possuindo uma abordagem gramatical e, em verdade, os próprios pesquisadores definam o trabalho como fundado em “gramáticas probabilísticas” [28], a metodologia apresentada nesse trabalho é “markoviana” por natureza [7, 11].

O objeto de estudo de Ponsford *et al.* era a estrutura harmônica, mais especificamente sua progressão no decorrer das composições. Segundo os pesquisadores, “modelar somente a harmonia geraria peças musicais levemente mais satisfatórias do que modelar somente a melodia” [28]. A definição do *corpus* foi tomada de modo a refletir essa vontade — o conjunto de sarabandas escolhido é “fortemente harmônico” [28]. O *corpus* formado representa, de fato, uma “escola” ou estilo específico de composição e há muitas similaridades entre as peças dos diferentes compositores, e.g., o comprimento das frases (4 compassos) e das próprias composições (em sua maioria, entre 20 e 32 compassos), as mudanças de tonalidade ao longo das mesmas etc. Ademais, a base foi dividida em 9 diferentes *subcorpora*, variando-se os compositores e os modos (maior ou menor) das peças neles contidas, visando a permitir uma análise mais detalhada em termos de estilo e forma. As

¹⁹A sarabanda surgiu no México e na Espanha, no século XVI, como uma dança rápida e sensual acompanhada por castanholas e violão. Foi banida na Espanha no final desse século, mas lá sobreviveu até o fim do Barroco, espalhando-se para a Itália. Em sua origem, é caracterizada por seu esquema harmônico, mas a partir da segunda metade do século XVII passou a assumir características rítmicas marcantes. Na França, a sarabanda assume um caráter expressivo mais suave e majestoso, transformando-se em uma dança lenta, usualmente acompanhada por alaúde ou cravo [29].

peças foram, então, gravadas em MIDI com o auxílio de um piano; notas incorretas foram posteriormente corrigidas e os valores de duração foram arredondados. Para a representação interna do programa, todas as notas foram substituídas pelos respectivos graus de escala (considerando acidentes) e tornadas “independentes” da tonalidade em que se encontravam.

Ponsford *et al.* amostraram as harmonias das composições no período da colcheia (“segmentação”), de tal forma que todas as notas contidas em um intervalo de tempo formassem uma harmonia²⁰ e converteram-nas no estado fundamental (“*root form*” [28]) — preenchendo tríades incompletas, simplificando algumas harmonias e resolvendo dissonâncias. Ao término da conversão, observou-se a ocorrência de pares de harmonias nos períodos da colcheia. Essas harmonias foram, então, unidas em símbolos que compreendiam o dobro do tempo, i.e., a duração de uma semínima, e esta representação levou à correspondência um-para-um das harmonias com os *beats* das composições.

Após tal processamento, de modo similar ao que está descrito na Seção 2.2.2, as regularidades e os padrões musicais foram absorvidos por um modelo de Markov (nas ordens²¹ $N = 2$ e $N = 3$) de modo a formular um conjunto de regras para descrever aquela “linguagem” particular, i.e., o estilo das sarabandas de Jacques Champion de Chambonnières (1601–1672), de Louis (1626–1661) e de François Couperin (1668–1733), de Jean-Baptiste Lully (1632–1687) e outros. Além de incluírem, em cada composição, os símbolos de início e término (conforme apresentado na Seção 2.2.4), os pesquisadores também obrigaram o modelo a contemplar as divisas entre diferentes frases e entre os compassos. Isso foi feito adicionando-se automaticamente símbolos que representassem o início de cada um desses eventos (uma nova frase ou um novo compasso). As frases foram posteriormente numeradas e relacionadas ao comprimento (em número de frases) da composição à qual pertenciam, para que houvesse mais informação acerca da posição daquela frase nessa composição²². Dessa forma, Ponsford *et al.* esperavam que seu programa fosse capaz de “aprender” o contexto em que as frases e os compassos se inseriam, modelando a progressão harmônica de forma mais fiel ao que se encontrava no *corpus*. A Figura 2.5 exemplifica a estrutura dos elementos do *corpus* devidamente anotado.

O modelo estimado a partir das contagens dos $(N + 1)$ -gramas pôde, então, ser utilizado para a geração aleatória de novas composições, de forma similar ao que está

²⁰Os pesquisadores ressaltam o uso do termo “harmonia” ao invés de “acorde” porque a própria representação interna do programa abdica da precisão dos *pitches* ao tomar apenas os graus de escala.

²¹Aqui aplicam-se as mesmas considerações feitas para o trabalho de Brooks *et al.* com relação à ordem do modelo.

²²O mesmo processo poderia ter sido empreendido com as barras de compasso, mas Ponsford *et al.* notam que, de acordo com o esperado, o modelo se torna mais esparsa a cada símbolo de controle que é adicionado.

```

(início frase_1_de_A ... compasso ... compasso ... compasso
    frase_2_de_A ... compasso ... compasso ... compasso
    ...
    frase_A_de_A ... compasso ... compasso ... compasso fim)
(início frase_1_de_B ... compasso ... compasso ... compasso
    frase_2_de_B ... compasso ... compasso ... compasso
    ...
    frase_B_de_B ... compasso ... compasso ... compasso fim)

```

Figura 2.5: Exemplo de anotação do *corpus*. Adaptado de [28].

descrito na Seção 2.2.3 e ao que foi utilizado no trabalho de Brooks *et al.* Para uma nova cadeia, a seleção de símbolos foi tomada até que o símbolo **fim** fosse selecionado. Alternativamente, cada frase da nova composição foi gerada individualmente e os pesquisadores utilizaram um algoritmo do tipo geração-e-teste para eliminar cadeias que não fossem compatíveis com padrões específicos (cf. Brooks *et al.* [16]). Nesse trabalho, os pesquisadores utilizaram, a cada frase de 4 compassos que desejavam gerar, um padrão contendo um símbolo do tipo **frase_C** ou **frase_C_de_A** e espaços para as harmonias nos *beats* de cada compasso (encabeçado pelo respectivo marcador). No início de cada geração, foi passado um contexto, representado pelos N últimos símbolos da frase anterior ou N símbolos de **início**, no caso da primeira frase. Isso obrigou que as frases fossem geradas na ordem em que estavam dispostas na composição. O número de frases a serem geradas era dado como parâmetro.

Para os testes, os pesquisadores variaram a ordem do modelo de análise-síntese ($N = 2, 3$), o *subcorpus* de análise (diferentes compositores e modo — maior ou menor) e o tipo de anotação (somente **início** e **fim**; adicionando marcadores de frase sem numeração; adicionando também marcadores de compasso; e, por fim, numerando as frases). Os resultados foram avaliados de forma subjetiva, através da similaridade de cada nova composição com as peças do *subcorpus*. Ponsford *et al.* também realizaram uma análise objetiva, comparando a probabilidade de ocorrência de diferentes sequências geradas a partir de um mesmo *subcorpus*. Notaram que essa informação não pode ser estudada adequadamente quando as sequências analisadas apresentam tamanhos diferentes entre si. Isso acontece porque, em uma dada sequência, a probabilidade de uma transição é sempre um número entre 0 e 1. Dessa forma, sequências de maior comprimento tendem a apresentar menores probabilidades de ocorrência, o que pode não refletir as características dos *corpora*. Assim, alterando a Equação 2.17, sugeriram o uso da seguinte medida, na tentativa de remover a influência do comprimento L de uma sequência s :

$$M(s) = \sqrt[L]{\prod_{k=1}^L \Pr\{X_k | X_{k-2} X_{k-1}\}}. \quad (2.18)$$

Mesmo utilizando essa medida, ainda resta a necessidade de se dar maior probabilidade total a composições com o número de compassos múltiplo de 4, uma vez que apenas peças com essa características são encontradas no *corpus*. Na ausência de um modelo melhor para o comprimento das sequências, optaram por gerar, nos casos com anotação de frases e compassos, peças com duração de 24 compassos, o tamanho mais comum observado.

Cerca de 200 composições foram sintetizadas, e os pesquisadores perceberam que tanto o aumento da ordem quanto um maior número de anotações parecem levar o programa a produzir progressões harmônicas mais convincentes. Utilizando-se somente os símbolos de início e final, peças não usuais podem ser geradas, e.g., peças contendo duas harmonias apenas ou com duração que não é múltipla de 4 compassos. Ao adicionar marcadores de frase e de compasso, o algoritmo foi capaz de gerar frases mais plausíveis e de agrupar as harmonias de acordo com o observado no *corpus* (usualmente na forma \mathcal{AAA} ou \mathcal{AAB} [28]). Posteriormente, a análise e geração a partir das frases numeradas pareceu permitir, de acordo com o esperado, o aparecimento de modulações adequadas. Essa interpretação das frases como dependentes da posição em que se encontram na peça é importante e evita que a mesma se torne trivial [28]. Por fim, os pesquisadores destacaram que o uso de ordem mais alta de análise foi um passo necessário para aumentar a compreensão das cadências, apesar de tornar o modelo mais esparsos e, conseqüentemente, exigir também um aumento do tamanho do *corpus*. Apesar de tudo, algumas peças geradas apresentaram alterações súbitas de tonalidade, o que os pesquisadores atribuíram ao tamanho ainda pequeno do contexto. Ademais, Ponsford *et al.* observaram que “peças às quais foram atribuídas pelo modelo baixas probabilidades [de ocorrência] parecem, ao ouvinte, melhores do que algumas das peças de maior probabilidade” [28], o que torna a medida objetiva inadequada para avaliação da qualidade das composições.

The Continuator

A abordagem de François Pachet no trabalho “*The Continuator: Music Composition with Style*” [18], de 2002, foi inovadora no sentido de buscar permitir a interatividade no âmbito dos modelos de Markov, o que o pesquisador descreveu como sendo uma tentativa de “conectar duas classes tradicionalmente incompatíveis de sistemas musicais” [18]. O propósito do *Continuator* era um novo tratamento da relação músico-máquina, que permitisse uma interação consistente com o estilo definido pelo músico — ao contrário dos sistemas interativos usuais, que “não têm a capacidade de aprender” [18].

O sistema de Pachet utilizava como entrada para as notas um controlador MIDI (e.g., um teclado ou uma guitarra), que registrava, entre outros parâmetros, *pitch*,

*velocity*²³, início e duração de cada nota. Assim, permitia-se que um fluxo qualquer de notas tocado no instrumento fosse segmentado em frases e, posteriormente, analisado. Os padrões estilísticos das frases eram aprendidos por um modelo de Markov, capaz de “responder” com uma nova sequência a cada frase do compositor, no seu próprio estilo. Pachet notou que esse sistema era capaz de mascarar a incapacidade que os modelos de Markov geralmente possuem de absorver grandes contextos: “o sistema apenas ‘preenche as lacunas’, e assim o poder das cadeias de Markov pode ser explorado ao máximo” [18].

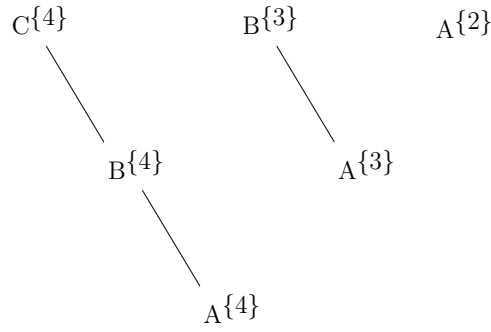
Para operar adequadamente e produzir sequências compatíveis com diferentes estilos, Pachet expôs três requisitos necessários ao algoritmo do *Continuator* [18]:

- Eficiência e capacidade de aprendizado em tempo real;
- Condução das continuidades de maneira “realista”;
- Tratamento de outras características musicais, como o ritmo e a polifonia.

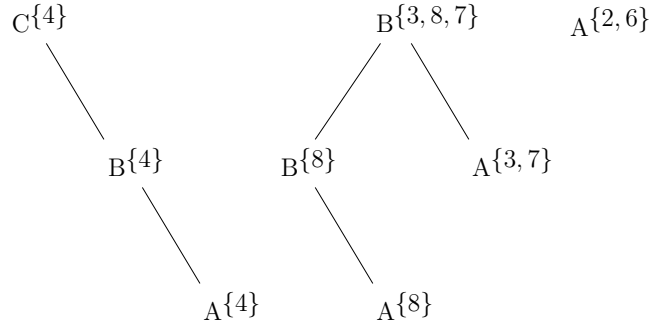
Para realizar a análise das frases, Pachet não utilizou o esquema de matrizes de transição, como visto até agora. O sistema do *Continuator* funcionava através do preenchimento do que é conhecido na literatura como “árvore de prefixos” (*prefix tree* [18] ou, simplesmente, *trie* [31]). Assim, para dada sequência apresentada como entrada, o algoritmo executava uma “varredura”, da direita para a esquerda, enquanto atualizava seu modelo interno com todos os prefixos e continuações possíveis nesta sequência. Essa implementação pode ser mostrada equivalente a um modelo de Markov completo com ordem variável e, a princípio, é mais “poderosa” que uma representação através da matriz de transição. O exemplo apresentado por Pachet em [18] pode auxiliar na compreensão desse procedimento.

Seja a primeira sequência de símbolos de qualquer natureza, {A B C D}. A posição do símbolo “A” na sequência é 1, o símbolo “B” assume a 2ª posição e assim por diante. O algoritmo de preenchimento do modelo era iniciado a partir da montagem de uma árvore com a raiz indexada pelo símbolo em “C”. Esse elemento era, então, marcado pelo identificador da posição do último elemento da sequência, no caso, 4. Em seguida, um nó indexado por “B” era adicionado como “filho” da raiz, sendo igualmente anotado com a posição do elemento “D”. E, por último, um nó “A” era colocado sob “B” e anotado. O processo era repetido para a subsequência {A B C} (cada nó sendo marcado pela posição do elemento “C”) e, finalmente, para a subsequência {A B} (para a qual o marcador era a posição 2). Cada nó era criado apenas na primeira vez que aparecesse e a floresta crescia à medida que novas sequências fossem analisadas. A Figura 2.6a apresenta o estado da floresta de prefixos ao final dessa análise. O processo podia ser reiniciado, agora para uma segunda

²³O parâmetro *velocity* descreve quão rapidamente/fortemente determinada nota foi tocada. Seu valor é um número inteiro, variando entre 0 e 127.



(a) Após a análise da sequência $\{A B C D\}$



(b) Floresta completa após a análise das sequências $\{A B C D\}$ e $\{A B B C\}$

Figura 2.6: Exemplos de árvores de prefixos no treinamento do *Continuator*. Os índices associados a cada nó representam as posições dos elementos de continuação nas diferentes sequências de entrada. Adaptada de [18].

sequência, e.g., $\{A B B C\}$. Novamente da direita para a esquerda, para a continuação “C”, de índice 8, a árvore com raiz “B” era aumentada para representar a subsequência $\{A B B\}$ e os nós envolvidos eram devidamente marcados. As iterações continuavam até que toda a sequência fosse analisada. A Figura 2.6b apresenta a floresta de prefixos completa, após a análise das duas sequências.

Após o treinamento do modelo em tempo real, o algoritmo era chamado a “responder” com uma nova sequência. Tal procedimento de geração era executado por meio de percursos nas árvores obtidas na etapa imediatamente anterior e pela seleção aleatória de elementos nas listas de continuações (índices) em um determinado nó. Assim, suponha-se que a sequência $\{A B\}$ foi tocada pelo músico e que o modelo neste instante era descrito pela floresta contida na Figura 2.6b (cf. Pachet [18]). O algoritmo assumia essa sequência como um prefixo e vasculhava a floresta de treinamento em busca de uma árvore cuja raiz fosse igual ao último elemento deste prefixo (no exemplo, “B”). Uma vez descoberta essa árvore, o resto da sequência de entrada era processado (como sempre, da direita para a esquerda) o tanto quanto possível, i.e., enquanto elementos equivalentes fossem encontrados no percurso descrito na mesma árvore²⁴. A lista de continuações permitidas pelo último nó do percurso

²⁴O percurso era realizado, a partir da raiz, da seguinte forma: uma vez que determinado elemento da sequência fosse identificado como equivalente a um nó da árvore, o próximo elemento

era retornada e o algoritmo sorteava, de maneira uniforme, uma das possibilidades apresentadas. É evidente que quanto maior o número de ocorrências de determinada sequência de eventos, maior a probabilidade de essa transição ser sorteada, já que o último elemento da sequência figurará na lista de continuações da árvore corresponde com diferentes índices. No exemplo, a lista de continuações retornada é “{3, 7}”, correspondendo aos elementos “C” e “B” das sequências de treinamento, respectivamente, com igual probabilidade de servirem como continuação. O processo de geração podia ser repetido conforme o desejado e uma rotina era acionada para lidar com descontinuidades, i.e., quando nenhuma continuação fosse encontrada no modelo (não há árvore com a raiz dada pelo último elemento do prefixo).

Até aqui nada se discutiu sobre a natureza dos elementos das sequências analisadas pelo *Continuator*, i.e., os elementos musicais observados pelo modelo de Markov. Conforme visto na Seção 2.3.1, um estado no modelo pode ser uma nota, um acorde, um motivo etc. Nessa compreensão, Pachet não restringiu a representação do *Continuator*; pelo contrário, permitiu que qualquer atributo MIDI fosse considerado, através do que denominou “função de redução” (*reduction function* [18, 31, 32]). Assim, o tratamento que Pachet deu ao modelo de Markov não depende de um único “ponto de vista” (*viewpoint* [18, 31, 32]), pois os estados do modelo podem ser representados por qualquer um dos elementos musicais (os “tipos básicos” como *pitch*, duração, *velocity* etc.), bem como por combinações dos mesmos em “tipos derivados” através do produto cartesiano (e.g., *pitch* \otimes duração). Além das funções de redução fomentadas em [31] e [32], Pachet também introduziu outras musicalmente significativas, e.g., regiões estreita e larga de *pitch*. Todo esse conjunto de funções de redução possibilitaram ao *Continuator* lidar de forma simples com as descontinuidades por meio da mudança de ponto de vista e, subsequentemente, uma nova busca por possíveis continuações. Pachet estabeleceu, então, uma hierarquia das funções de redução, da mais “refinada” à mais abrangente:

1. *pitch* \otimes duração \otimes *velocity*;
2. região estreita de *pitch* \otimes *velocity*;
3. região estreita de *pitch*;
4. região larga de *pitch*;

O *Continuator* também apresenta uma maneira própria de tratar as polifonias e o ritmo da música [18]. Além disso, Pachet observou a necessidade de se “polarizar” as probabilidades das transições durante o processo de geração para considerar as mudanças locais de tonalidade, intensidade etc., que podem ocorrer entre uma frase

analisado deveria ser rastreado no conjunto de nós-filhos deste nó. O número de níveis da árvore vasculhados era, no máximo, igual ao tamanho da sequência de entrada.

e outra e são críticas no contexto da interatividade musical. Para tanto, Pachet definiu uma função de adequação, que descreve quão bem um dado elemento (w_k) se adequa a um contexto imediato, e.g., as últimas 8 notas tocadas pelo músico (C):

$$f(w_k, C) = \frac{|\{w_k\} \cap C|}{|C|}, \quad (2.19)$$

onde $|\cdot|$ indica a cardinalidade do conjunto e o operador \cap representa a interseção ou, em última instância, o número de ocorrências de w_k no conjunto²⁵ C .

Dessa forma, em vez de considerar somente a probabilidade markoviana das continuações propostas pelo algoritmo, Pachet passou a ponderar também a adequação de cada continuação ao contexto imediato fornecido pelo músico. Por meio de uma nova função, a probabilidade de o elemento w_k ser selecionado como continuação passa a depender de uma combinação linear entre a probabilidade dada pelo modelo (através conjunto de continuações possíveis retornado após um percurso descrito na árvore) e a função de adequação de w_k no contexto C :

$$p_{\text{seleção}}(w_k) = \delta \text{Pr}_{\text{ML}}\{w_k | s_{k-i-1}^{k-1}\} + (1 - \delta)f(w_k, C), \quad (2.20)$$

com o parâmetro δ limitado ao intervalo $[0, 1]$ e s_{k-i-1}^{k-1} representando a subsequência que define o percurso na árvore até uma profundidade i ²⁶. Evidentemente, os valores intermediários de δ eram interessantes para Pachet, quando o sistema “gera material musical que é, ao mesmo tempo, consistente em termos de estilo e sensível às entradas [do músico]” [18].

Foram realizados alguns testes com o objetivo de verificar a operação do *Continuator*. Notando a dificuldade de se avaliar qualidade musical, Pachet sugeriu um tipo de teste de Turing (cf. Seção 1.1.5) — o objetivo do ouvinte é identificar quando a máquina está em funcionamento e quando é um músico que está tocando. Pachet descreve que “na maior parte dos casos, senão em todos, a música produzida é indistinguível da entrada fornecida pelo músico” [18].

2.3.3 Outros trabalhos

A Tabela 2.3 apresenta um resumo dos trabalhos revisitados na seção anterior, destacando suas principais contribuições.

É evidente que há muitos outros trabalhos de igual relevância que não foram avaliados aqui. Por exemplo, é digna de nota a contribuição do músico e pesquisador Iannis Xenakis na área da música markoviana, em especial, através de suas composições “*Analogique A et B*”, de 1958 [5]. Muitos trabalhos utilizam os mode-

²⁵Em verdade, C é um multiconjunto, i.e., permite-se multiplicidade não unitária dos elementos.

²⁶Para $i = 0$, o único elemento encontrado na árvore é a própria raiz, e $s_{k-i-1}^{k-1} = w_{k-1}$.

Tabela 2.3: Referências para a Seção 2.3.2.

Referência	Objetivo	Estado	Ordem
OLSON e BELAR [26]	imitação do estilo de Stephen Foster (base com 11 composições)	<i>pitch</i> e ritmo ²⁷	$N = 0, 1, 2$
HILLER e ISAACSON [20]	composição “genuína” a partir das funções harmônica e de proximidade	intervalo entre pares de notas	$N = 0, 1$ e variável ²⁸
BROOKS <i>et al.</i> [16]	imitação do estilo de uma base contendo 37 hinos para coral	<i>pitch</i> (ataque e sustentação de notas)	$N = 0, \dots, 7$
AMES [27]	composição “genuína” a partir de um modelo “explodido”	duração média, articulação, registro e grau	$N = 1$
PONSFORD <i>et al.</i> [28]	imitação do estilo de progressão harmônica de uma base contendo 84 sarabandas	“harmonia”	$N = 2, 3$
PACHET [18]	continuação de melodias em tempo real	índice e função de redução	variável ²⁹

los de Markov como parte de um processamento maior, e.g., para a inicialização de candidatos em um método de busca ou em conjunção com uma rede neural artificial. Destaca-se também o trabalho de Man Yat Lo e Simon Lucas [33], que combina a técnica dos algoritmos evolucionários com as cadeias de Markov, e será analisado em maior detalhe na Seção 4.3.3.

Os artigos de Ames [27] e de Fernández e Vico [11], e o livro de Nierhaus [7] apresentam pesquisas extensas e bastante completas sobre os modelos de Markov. Nesses trabalhos, é possível encontrar também informações acerca dos modelos ocultos de Markov (*HMM — Hidden Markov Models*). Uma HMM é uma generalização de um modelo de Markov na qual os estados e transições não são observáveis. O que se observa são sequências de “emissões”, cuja distribuição de probabilidade depende dos estados internos, desconhecidos. Esses modelos têm sido amplamente utilizados no campo da Composição Algorítmica [11], especialmente nos casos de criação de contrapontos, harmonização de melodias e o contrário, i.e., criação de melodias dada uma harmonia subjacente. Trata-se de uma técnica custosa computacionalmente, de modo particular na fase de treinamento. Destaca-se aqui o trabalho pioneiro de aplicação de HMMs à composição desenvolvido por Mary Farbood e Bernd Schoner [34], que objetivava a composição em contraponto (primeira espécie) ao estilo de Palestrina.

2.4 Limitações

A seguir, estão listados os principais pontos que devem ser observados ao se trabalhar com modelos de Markov:

- Dimensionalidade e ordem N ;
- Verticalidade;
- Conformação do *corpus*;
- Estrutura e dependências entre elementos distantes.

O problema da dimensionalidade está intimamente relacionado ao valor da ordem N . Para exemplificar, suponha-se a formulação de um espaço de estados contendo as 26 letras do alfabeto e que o objetivo do modelo é capturar as regras implícitas para formação das palavras da língua portuguesa. É evidente que, se for utilizado um modelo de primeira ordem, há 26×26 bigramas possíveis. De maneira geral, o algoritmo deverá estar apto a trabalhar com $26^{(N+1)}$ combinações de sequências (cf. Seção 2.2.3). Embora nem todas as sequências ocorram, e aliás, usualmente, a maior parte das sequências não ocorra ou ocorra raramente, a complexidade do problema aumenta de forma exponencial com o aumento da ordem [28]. Além disso, como apresentado em [16], a ordem de análise não pode ser muito pequena, pois será incapaz de fornecer uma generalização válida, i.e., que represente adequadamente o *corpus* estudado. A ordem tampouco pode ser muito grande, ou o modelo se restringe a reconhecer e reproduzir somente as sequências já existentes na base — o modelo “degenera” [16].

A verticalidade diz respeito à definição do espaço de estados. Se, em sua origem, os modelos de Markov só operavam sobre estados “simples”, posteriormente observou-se sua capacidade de também absorver cadeias formadas a partir de estruturas mais complexas, e.g., acordes e polifonias [27], conforme visto na Seção 2.3.1.

Outro ponto importante é a conformação da base de dados. É essencial que o *corpus* seja coerente, isto é, represente verdadeiramente uma classe na qual os elementos compartilham características comuns. Caso contrário, nenhum treinamento será capaz de induzir uma generalização válida e representativa. Em última instância, isso significa dizer que o modelo abstraído só poderá ser tão bom quanto o *corpus* que ele absorve. É importante, contudo, que o *corpus* seja diversificado, caso contrário o modelo não conseguirá absorver informação — há muita redundância.

²⁷Analisados e sintetizados separadamente.

²⁸Relacionando tempos fortes e fracos em estruturas separadas.

²⁹O modelo proposto por Pachet é “completo” e permite investigar o passado das sequências de treinamento tão profundamente quanto possível.

Conforme exposto na Seção 2.3.2, tanto o aumento da ordem como o aumento do espaço de estados (com a adição de símbolos de início e de término ou de símbolos para o controle da estrutura da composição) acarretam maior esparsidade no modelo. Essa esparsidade pode ser insatisfatória em diversos casos. Por exemplo, determinadas transições podem ser caracterizadas por probabilidade nula ou muito baixa não como resultado de uma regra implícita da classe estudada, mas em função da falta de exemplos (possivelmente, indicando má formação do *corpus*). É evidente que, pela própria caracterização e definição matemática de Markov, subsequências inexistentes nos dados de entrada acarretarão alta perplexidade nos processos de reconhecimento de frases e não constarão nas cadeias criadas através dos processos de geração. O que se observa, em geral, é que elementos de baixa probabilidade são subestimados pelos modelos probabilísticos e sua ponderação carece de algum tipo de ajuste [35]. É esse o campo de ação das técnicas de suavização (*smoothing*) (cf. Seção 2.4.1).

Além disso, os modelos de Markov capturam apenas a *taxis* de uma linguagem, i.e., a ordenação local dos elementos [28]. Não são projetados visando a capturar a estrutura em sua totalidade ou as relações entre elementos que estejam demasiadamente separados. Conforme destacado acima, aumentar a ordem N de análise não soluciona esse problema.

2.4.1 Suavização (*Smoothing*)

Os modelos de Markov são capazes de reconhecer e também de gerar sequências de acordo com uma dada base de treinamento. Um problema muito comum enfrentado, por exemplo, na área de processamento de linguagem natural (NLP — *Natural Language Processing*) é a esparsidade desses modelos.

Há mais de uma forma de se lidar com a esparsidade de um modelo de Markov. Uma delas diz respeito ao próprio tamanho do *corpus* analisado, uma vez que aumentando-se o número de sequências de treinamento há mais “oportunidades” de ocorrência de transições pouco prováveis.

Outra possibilidade é realizada *a posteriori* do processo de treinamento — trata-se do conjunto de técnicas de suavização, cujo objetivo geral é “redistribuir” as probabilidades calculadas pelo modelo, tirando de elementos muito prováveis para aumentar o peso de elementos menos prováveis [35]. Logo se percebe que tais técnicas são aplicáveis não somente às transições com probabilidade nula, mas também aos elementos com pequena probabilidade. Dependendo do problema com o qual se está lidando isso pode ser necessário, porque transições pouco prováveis são usualmente “subestimadas” nos cálculos dos modelos — costumam aparecer menos que a frequência relativa estimada — e, por isso, têm associadas a elas probabilidades

menores do que a generalização da classe realmente deveria apresentar.

A literatura destaca dois tipos de métodos de suavização — os métodos de *back-off* e os métodos de interpolação [28, 31, 35–37] —, cada um sendo definido pela forma de uso das probabilidades estimadas na ordem do modelo e em ordens inferiores. Conforme visto na Seção 2.2.2, um modelo de Markov de qualquer ordem $N - 1$ estima a probabilidade de uma transição de maneira similar ao exposto na Equação 2.14, reescrita aqui de forma mais genérica por facilidade [35]:

$$p_{\text{ML}}(w_k | s_{k-N+1}^{k-1}) = \frac{\mathcal{C}(s_{k-N+1}^k)}{\mathcal{C}(s_{k-N+1}^{k-1})} = \frac{\mathcal{C}(s_{k-N+1}^k)}{\sum_{w_k} \mathcal{C}(s_{k-N+1}^k)}. \quad (2.21)$$

Nos métodos de *backoff*, usualmente as probabilidades de todas as transições em ordem $N - 1$ que apresentem contagem nula são aproximadas pelas probabilidades de um modelo com ordem $N - 2$, conforme:

$$p_{\text{backoff}}(w_k | s_{k-N+1}^{k-1}) = \begin{cases} \alpha(w_k | s_{k-N+1}^{k-1}) & \text{se } \mathcal{C}(s_{k-N+1}^k) > 0 \\ \gamma(s_{k-N+1}^{k-1}) p_{\text{backoff}}(w_k | s_{k-N+2}^{k-1}) & \text{se } \mathcal{C}(s_{k-N+1}^k) = 0 \end{cases}, \quad (2.22)$$

onde, em cada diferente método de *backoff*, $\alpha(w_k | s_{k-N+1}^{k-1})$ representa uma alteração no estimador de máxima verossimilhança padrão, enquanto $\gamma(s_{k-N+1}^{k-1})$ é um fator de escala para garantir que as probabilidades somem 1 [35]. É um exemplo de método de *backoff* famoso na literatura o método de suavização de Katz [35].

Já os métodos de interpolação combinam as probabilidades estimadas nas ordens $N - 1$, $N - 2$ etc. com diferentes pesos [31] como pode ser observado em:

$$p_{\text{interpolação}}(w_k | s_{k-N+1}^{k-1}) = \lambda_{s_{k-N+1}^{k-1}} p_{\text{ML}}(w_k | s_{k-N+1}^{k-1}) + (1 - \lambda_{s_{k-N+1}^{k-1}}) p_{\text{interp}}(w_k | s_{k-N+2}^{k-1}), \quad (2.23)$$

onde o fator de interpolação linear $\lambda_{s_{k-N+1}^{k-1}}$ pode ser calculado de forma a maximizar alguma probabilidade ou escolhido igual para toda subsequência s_{k-N+1}^{k-1} [35]. Uma regra prática consiste na segmentação das diferentes subsequências s_{k-N+1}^{k-1} e posterior seleção dos valores de $\lambda_{s_{k-N+1}^{k-1}}$ de acordo com as contagens observadas [35]. Os métodos de Witten-Bell e Jelinek-Mercer (visto na Equação 2.23) são métodos de interpolação linear típicos [35]. O Apêndice B apresenta o algoritmo de maximização de expectativa que permite calcular os valores ótimos dos parâmetros de interpolação para um conjunto de dados.

O método aditivo também é de particular interesse os pesquisadores da área de processamento de linguagem natural. Para evitar o problema das contagens nulas, esse método de suavização considera que cada transição “acontece com um pouco mais de frequência do que ela realmente ocorre” [35], adicionando um fator $\xi \in (0, 1]$

a sua contagem, de forma que:

$$p_{\text{aditivo}}(w_k | s_{k-N+1}^{k-1}) = \frac{\xi + \mathcal{C}(s_{k-N+1}^k)}{\xi |T| + \sum_{w_k} \mathcal{C}(s_{k-N+1}^k)}, \quad (2.24)$$

onde T representa o conjunto de transições possíveis. Tipicamente, a cardinalidade máxima que esse conjunto pode assumir para um modelo de ordem $N-1$ é $|T| = M^N$, onde M denota o número de símbolos diferentes existentes no *corpus*.

O relatório técnico de Stanley F. Chen e Joshua Goodman [35] apresenta uma análise bastante exaustiva dos principais métodos de suavização.

Capítulo 3

Gramáticas Gerativas

As gramáticas gerativas são um formalismo sintático desenvolvido pelo linguista norte-americano Avram Noam Chomsky (1928–). São ferramentas matemáticas poderosas, capazes de analisar e sintetizar sequências de símbolos através de um conjunto de regras de substituição. Uma boa gramática para uma linguagem é aquela capaz de validar (e gerar) todas e somente aquelas sequências pertencentes à linguagem em questão.

Neste capítulo, será apresentado um panorama geral sobre esta técnica de análise e de síntese, desde sua inspiração histórica. A teoria é brevemente apresentada, bem como os diferentes tipos de gramática, de acordo com a hierarquia de Chomsky. A partir da interpretação da música como um formalismo matemático, esclarece-se a aplicação das gramáticas e suas limitações no campo da Composição Algorítmica, através dos esforços de alguns pesquisadores. Por fim, é explicitada a relação desta técnica com as cadeias de Markov, já vistas em detalhe no Capítulo 2.

3.1 Motivação

“A sintaxe é o estudo dos princípios e dos processos pelos quais as sentenças são contruídas em linguagens particulares. Uma investigação sintática de determinada linguagem tem como objetivo a construção de uma gramática que pode ser vista como uma ferramenta para produção de sentenças da linguagem sob análise” [38, p. 11].

Por sobre essa pequena definição da relação entre sintaxe e gramática, presente no início do livro “*Syntactic Structures*” [38] (1957), Noam Chomsky baseou todo seu trabalho em linguística, posteriormente consagrando-se como um dos maiores pensadores dessa área, que se dedica ao estudo da língua, da fala e da linguagem. Chomsky aborda a questão da formação da linguagem de maneira diversa ao paradigma em voga em toda a Europa até o final da primeira metade do século XX

— o estruturalismo linguístico, cuja origem se deve ao suíço Ferdinand de Saussure (1857–1913), e que encontrou no linguista Leonard Bloomfield (1887–1949) um dos seus principais difusores nos Estados Unidos.

3.1.1 O estruturalismo de Saussure

No trabalho póstumo, “*Cours de linguistique générale*” [39], de 1916, Saussure apresenta o objeto da linguística como sendo “um aglomerado confuso de coisas heteróclitas sem ligação entre si” [39]. Para Saussure, a relação que o linguista traça com seu objeto de estudo é diferente do observado em qualquer outra ciência, porque o fenômeno linguístico apresenta um conjunto de dualidades que o tornam demasiadamente complexo, a saber:

1. A palavra falada é uma impressão sonora, fruto do sistema vocal. A língua, contudo, não pode ser reduzida à fala. Da mesma forma, não se pode dissociar o som do aparelho fonador que o produz e as articulações vocálicas, por sua vez, não fazem sentido sem a impressão sonora.
2. O som é instrumento — forma de propagação — do pensamento e não existe por si mesmo. Som e ideia, juntos, formam uma unidade “complexa, fisiológica e mental” [39].
3. A linguagem tem uma faceta individual e outra social, e uma não pode ser concebida sem a outra.
4. O fenômeno linguístico implica, a cada instante, um sistema estável e uma evolução; a cada momento, ele é “uma instituição atual e um produto do passado” [39].

Diante desse terreno nebuloso, Saussure afirma que é necessário estudar a língua (“*langue*”) e tomá-la por norma de todas as outras manifestações da linguagem (“*language*”). O pesquisador destaca que a língua não é “uma função de um indivíduo” [39], embora seja assimilada passivamente (e não integralmente) por esse mesmo indivíduo, que é capaz da fala (“*parole*”). Para Saussure, a língua seria uma construção social, fruto da “faculdade da linguagem” [39] e, ao mesmo tempo, de um conjunto de convenções necessárias adotadas porque permitem “o exercício dessa faculdade nos indivíduos” [39]. Ou seja, porque ela é convencional, a língua permite a comunicação.

Na visão de Saussure, a expressão e o compartilhamento de ideias se dá através de um sistema de signos linguísticos arbitrários, mais especificamente através de cadeias desses signos. Cada signo linguístico compreende duas dimensões, sempre indissociáveis, que ele irá chamar “significado” e “significante” [39]. Essas dimensões

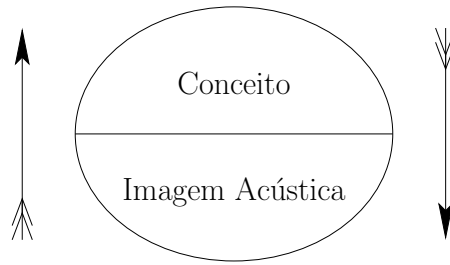


Figura 3.1: Modelo básico de um signo de Saussure, com significado (“Conceito”) e significante (“Imagem Acústica”). As setas indicam a relação de proporcionalidade e interdependência entre as duas dimensões do signo. Adaptado de [39].

se relacionam, respectivamente, com o “conceito” (ideia) do signo e sua “imagem acústica” (som¹). Por exemplo, o conceito de “árvore” na mente de um ser humano que fale latim encontrará correspondência em uma imagem sonora — a palavra latina “*arbor*” [39] —, e vice-versa. Em cada signo, a relação entre significado e significante é, portanto, indivisível, mas também arbitrária, i.e., poderia-se associar um dado conceito a qualquer outro conjunto de sons além daquele que lhe serve de significante [39]. A Figura 3.1 apresenta um diagrama para os signos saussurianos.

Outra novidade que Saussure introduz no campo da semiótica está na sua compreensão do valor de um signo linguístico. Para o estruturalista, o valor de um signo não pode ser entendido observando-se tão somente a relação arbitrária interna entre significado e significante. Pelo contrário: porque existe em um sistema linguístico, um signo deve ser analisado em conjunto com os outros elementos da linguagem — sejam aqueles signos que podem ser agrupados porque se assemelham a ele fonética ou morfologicamente (nas relações ditas associativas ou paradigmáticas²), ou aqueles que estão dispostos proximamente nas construções de frases (nas relações sintagmáticas³) [39].

Saussure mostra que, nos diferentes sistemas semiológicos, se estabelece, inexoravelmente, uma estrutura de diferenças que permite valorar os significados e os significantes dos signos. Os valores desses elementos emanam do próprio sistema, uma vez que os conceitos e as imagens acústicas não são definidos “positivamente”, pelo seu conteúdo, mas “negativamente”, pelo seu relacionamento com os outros termos da linguagem. Assim, um conceito ou uma imagem acústica é distinguível,

¹Mais do que apenas o som, por “imagem acústica” Saussure quer indicar toda a impressão psíquica que um dado conceito evoca em um indivíduo. É sensorial, uma imagem interna da palavra e não uma ação vocal. Diferencia-se do outro termo do signo — o conceito —, “geralmente mais abstrato” [39].

²Os elementos em um conjunto paradigmático estão unidos por associação mental. São exemplos os conjuntos que a palavra “ensinamento” faz com o conjunto {ensino, ensinar}, ou com {armamento, tratamento}, ou ainda com {educação, aprendizado} [39].

³As relações sintagmáticas estão “fundadas no caráter linear da língua” [39], i.e., se estabelecem entre dois ou mais elementos no decorrer do discurso. Tais conjuntos abrangem, portanto, desde palavras compostas até frases e parágrafos inteiros. São exemplos apresentados em [39]: “re-ler”; “contra todos”; “a vida humana”; “Deus é bom”; “se o clima estiver bom, nós sairemos” etc.

precisamente, por ser “o que os outros [conceitos e imagens acústicas] não são” [39]. Para Saussure, o sistema linguístico consiste, portanto, na série de diferenças das entidades separadas; é isto que cria “o caráter e, da mesma forma, faz o valor e a unidade” [39] dos signos linguísticos.

3.1.2 Bloomfield e a teoria behaviorista

Não se sabe ao certo quanto os trabalhos iniciais de Chomsky na área da linguística foram influenciados pelo pensamento de Saussure, mas alguns pesquisadores parecem concordar que o tom de seus escritos toma uma nova forma a partir da publicação, em 1959, de uma versão traduzida para o inglês do “*Cours de linguistique générale*” [40]. Então, a área da linguística já contava com as contribuições do Círculo Linguístico de Praga e de alguns norte-americanos, entre os quais destaca-se, especialmente, Leonard Bloomfield [40, 41].

Em seu principal trabalho, o livro “*Language*” [42] (1933), Bloomfield apresenta o que ele chama de análise dos constituintes imediatos (“*immediate constituent analysis*” [42]), através da qual o linguista segmenta hierarquicamente uma dada sentença em diferentes níveis. Esse processo visa à formalização da linguagem por meio da análise das diferentes classes constituintes.

Na definição apresentada por Bloomfield, um constituinte é “a parte comum de quaisquer (duas ou mais) formas [linguísticas] complexas” [42] e está, portanto, contido nessas formas e acompanhado de outros constituintes. Os constituintes imediatos de uma sentença podem ser, eles mesmos, formas complexas ou formas simples (morfemas). Ao final da análise de constituintes, contudo, os “constituintes últimos” [42] da sentença são sempre morfemas, que Bloomfield define como sendo aquelas formas que “não guardam semelhança fonético-semântica parcial com outras formas” [42]. A seguir, é apresentado um exemplo de como se sucedem os passos da análise de constituintes imediatos proposta por Bloomfield.

Exemplo 3.1. A análise dos constituintes imediatos permite destrinchar a sentença “o menino adora batata-doce” nos elementos “o menino” e “adora batata-doce”. Como cada um desses constituintes ainda é uma forma complexa, um segundo passo de análise pode ser realizado — do primeiro, obtém-se os elementos “o” e “menino”; e “adora batata-doce” é dividido em “adora” e “batata-doce”. O elemento “batata-doce” pode também, por sua vez, ser separado em “batata” e “doce”. A Figura 3.2 apresenta a representação em árvore dessa análise.

É também em “*Language*” que Bloomfield alia o estruturalismo de Saussure ao behaviorismo de Burrhus F. Skinner. Sob a ótica behaviorista, o homem seria uma “tábula rasa” e o aprendizado da linguagem se daria através de reforços a certas

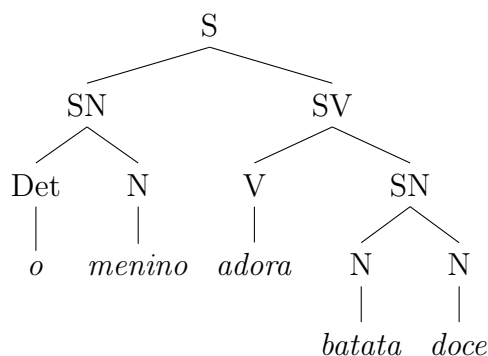


Figura 3.2: Exemplo de análise de constituintes imediatos da sentença “o menino adora batata-doce”. Os nós intermediários foram nomeados com o vocabulário usual da sintaxe — sintagmas nominal (SN) e verbal (SV), determinante (Det), nome (N) e verbo (V).

respostas aos estímulos do meio e da cultura nos quais esse homem se insere, i.e., através da formação de hábitos. Em [42], é apresentado o seguinte exemplo:

Exemplo 3.2. Assuma que Jack e Jill estão caminhando na rua. Jill está com fome e vê uma maçã em uma árvore. Ela emite um grunhido com seu sistema fonador. Jack sobe na árvore, colhe a maçã e a entrega para Jill. Jill come a maçã.

Segundo Bloomfield, a sucessão de eventos do Exemplo 3.2 pode ser segmentada da seguinte maneira [42]:

- (A) Eventos práticos precedentes ao ato da fala (“estímulo do falante” [42]);
- (B) Ato da fala;
- (C) Eventos práticos subsequentes ao ato da fala (“resposta do ouvinte” [42]).

Generalizando a situação apresentada, Bloomfield argumenta que, uma vez que um estímulo (A) seja apresentado a um indivíduo, sua reação na forma da fala⁴ (B) evoca no ouvinte uma certa reação (C) [43]. Dessa forma, “por hábito social que cada pessoa adquire dos mais velhos na infância, A–B–C estão estreitamente correlacionados” [43].

Bloomfield nega qualquer tipo de fenômeno mental e também o próprio conceito de significado conforme visto no estruturalismo saussuriano. Para ele, assim deve ser o proceder do linguista: registrar os fatos — as ocorrências da linguagem e as situações em que acontecem — sem se preocupar com o que não pode ser observado, i.e., os processos mentais, o que o falante “sente”. Bloomfield apresenta uma visão materialista (ou mecânica) da conduta humana, apoiando-se no paradigma estímulo–resposta. Dessa forma, ele acredita que é possível prever as ações (e aqui inclui-se

⁴De maneira mais genérica, pode-se considerar uma reação em qualquer tipo de linguagem.

também a fala) de um ser humano, tais quais as sequências de reações físicas ou químicas. Para tanto, bastaria conhecer “a estrutura exata de seu corpo nesse momento” [42] ou, equivalentemente, saber “a constituição do seu organismo em algum estágio inicial [...] e, então, guardar um registro de cada modificação naquele organismo, incluindo cada estímulo que o tenha afetado” [42].

3.1.3 O gerativismo de Chomsky

Chomsky observa o funcionamento da linguística estruturalista sobre os conjuntos restritos de fonemas e morfemas, bem como as limitações de sua aplicação na caracterização das sentenças de uma linguagem [44]. Ele critica a visão da *langue* saussuriana como um “armazém de signos” [41, 45] e argumenta que o sistema de diferenças de Saussure não funciona para um conjunto de sentenças em uma linguagem qualquer, que usualmente podem existir em número infinito. Segundo Chomsky, falta no trabalho de Saussure uma definição formal do conhecimento gramatical que possa explicar justamente essa criatividade de um falante, i.e., sua capacidade de produzir elocuições “bem formadas” e nunca antes proferidas. Bloomfield explica esse fenômeno utilizando-se de “padrões gramaticais” por meio dos quais uma nova frase nunca antes ouvida é construída com a substituição, por analogia, das formas similares que foram ouvidas [42]. Chomsky, por sua vez, vai além ao pensar essa capacidade da linguagem como sendo inata ao ser humano⁵.

Segundo o gerativista, o homem já apresentaria, em seu cérebro, o conjunto das ferramentas necessárias para exercer a faculdade da linguagem em todo seu potencial criativo, i.e., o homem não seria capaz somente da interpretação e da repetição (sob a presença ou na ausência dos estímulos behavioristas), mas também conseguiria, a partir de estruturas gramaticais internas, gerar novas sentenças. Dessa forma, diferente de Saussure e de Bloomfield, o pai do gerativismo não considera a língua um constructo social ou comportamental, pelo contrário: ela seria, sobretudo, a expressão do pensamento⁶. Percebe-se, assim, o caráter racionalista da teoria gerativa que ele irá propor, e é por isso que Chomsky vai atrás de uma análise de aspecto sintático, superando os níveis fonológico e morfológico de interesse do estruturalismo [46].

Para Chomsky, o desenvolvimento da linguagem nos seres humanos é um traço genético e programado do próprio desenvolvimento humano, assim como o que determina que desenvolvamos braços e pernas, não asas ou patas. A linguagem e o conhecimento gramatical não são “aprendidos” especialmente por meio de estímulos

⁵Chomsky também identifica outros elementos sem tratamento adequado no estruturalismo. Em [46], esses elementos são resumidos em: (1) interdependência entre constituintes descontínuos; (2) estruturas ambíguas; (3) estruturas formalmente distintas, mas semanticamente similares.

⁶Esse conceito está expresso no estruturalismo, mas ali admite um caráter secundário e não central como na teoria gerativista.

do ambiente em que um indivíduo esteja inserido. São aspectos inatos, aos quais o ambiente só contribui como um guia. É por causa desse mecanismo que se observam regras gramaticais tão intuitivas que um falante nativo não precisa “pensar” para diferenciar construções corretas das incorretas gramaticalmente.

Chomsky irá propor a existência de uma “gramática universal” [45], presente em todo ser humano e que dá origem, “superficialmente”, às gramáticas particulares dos diferentes idiomas nativos. Diz o gerativista: “a gramática de uma linguagem particular, então, é complementada por uma gramática universal que acomoda o aspecto criativo do uso da linguagem e expressa as regularidades profundas que, sendo universais, são suprimidas da gramática [particular]” [45]. Esse mecanismo “pré-programado” guarda consigo os princípios comuns a todas as linguagens e é, portanto, a base para a aquisição de um idioma.

A capacidade das crianças de aprender qualquer linguagem à qual forem expostas (isto é, de adquirir qualquer língua materna) é um exemplo que fortalece essa teoria proposta por Chomsky, i.e., uma criança não está predisposta a aprender um linguagem em detrimento de outra [45]. Segundo Chomsky, os processos internos em uma criança são acionados de forma a desenvolver nela uma linguagem à medida que os dados dessa mesma linguagem — exemplos de sentenças — forem apresentados a ela, sempre em acordo com a sociedade na qual ela se encontra. Para ele, “a criança aborda os dados assumindo que eles foram tirados de uma linguagem de um tipo previamente bem definido, e seu problema é determinar qual das linguagens (humanamente) possíveis é essa da comunidade na qual ela está inserida” [45], tomando por base as regras universais. Somente dessa forma, o conhecimento armazenado no que Chomsky denomina “órgão da linguagem” poderia extrapolar os dados linguísticos observados e ser mais do que apenas uma generalização indutiva sobre esses dados.

Em seus escritos, Chomsky introduz o conceito de “competência” linguística, que representa o conhecimento que o falante tem de sua própria língua e que precede seu emprego em situações concretas (o que ele irá denominar “desempenho”)⁷.

Regras de substituição

Em “*Aspects of the Theory of Syntax*” [45], Chomsky escreve: “o conhecimento de uma linguagem envolve a habilidade implícita de compreender continuamente muitas sentenças”. Uma gramática, portanto, precisa ao mesmo tempo conter um sistema finito de regras e ser capaz de atender ao aspecto criativo da linguagem ou, nas palavras de Chomsky: “um sistema de regras que possa iterar para gerar um número indefinidamente grande de estruturas” [45].

⁷Os conceitos de competência e desempenho admitem um paralelo com os conceitos de *langue* e *parole* em Saussure [41].

Chomsky lida com essa aparente dificuldade representando as regras gramaticais como funções recursivas ou ainda regras de substituição⁸. Assim, no paradigma gerativo–transformacional, é o conjunto limitado de regras recursivas que permite descrever a infinidade de sentenças de uma linguagem, de maneira similar a um falante da língua que apresenta a capacidade de compreendê-las mesmo que nunca as tenha formulado ou ouvido previamente [38]. O poder das regras recursivas faz reduzir enormemente a complexidade das gramáticas na teoria de Chomsky [38].

A criação de sequências em uma gramática gerativa se dá através da derivação das regras de substituição, assim chamadas porque, em cada regra, símbolos do lado esquerdo são substituídos por outros no lado direito [7, 38]. Chomsky diferencia três grandes partes do sistema de regras de uma gramática gerativa: a componente sintática, a componente semântica e a componente fonológica. A componente sintática, em um nível abstrato, descreve a estrutura, as propriedades e a relação entre os diferentes símbolos formativos da linguagem. As componentes semântica e fonológica são interpretativas e tratam, respectivamente, da representação semântica e da forma fonética que as sentenças assumem a partir das formulações das estruturas sintáticas. Segundo Chomsky, portanto, a componente sintática de uma gramática específica, para cada sentença, “uma estrutura profunda (*deep structure*) que determina sua interpretação semântica e uma estrutura superficial (*surface structure*) que determina sua interpretação fonética” [45]. Por exemplo, Chomsky avalia a diferença de orações na voz ativa (e.g., “o professor reprovou o aluno”) e na voz passiva (e.g., “o aluno foi reprovado pelo professor”) como sendo resultante, principalmente, de uma diferença na derivação das regras superficiais; as estruturas profundas das duas sentenças seriam similares [45].

Ademais, Chomsky divide as próprias regras de substituição em dois grupos: (1) a base e (2) as regras de transformação. As regras da base definem as sentenças em suas estruturas profundas enquanto as regras de transformação levam à derivação das estruturas superficiais, i.e., às elocuições que são observadas nas suas diferentes formas (e.g., voz passiva ou ativa, imperativo). Dessa forma, uma gramática gerativa utiliza as regras de substituição para transitar das estruturas mais profundas em direção às estruturas superficiais [7]. É evidente que uma análise de constituintes imediatos de uma sentença e seus formantes é capaz de fornecer informações importantes sobre a estrutura superficial desta sentença. Contudo, essa mesma análise não contempla as especificidades da estrutura profunda, e.g., em uma análise de constituintes, palavras de mesmo campo semântico ou de igual valor sintático podem ser separadas em classes diferentes. Assim, uma análise de constituintes imediatos apenas não permite a comparação das regras sintáticas mais basais aplicadas na derivação da sentença. O estudo das gramáticas gerativas possibilita perscrutar o que

⁸As regras gramaticais da teoria gerativa também são chamadas regras de reescrita.

Chomsky denomina “universais linguísticos” e lança luz sobre as estruturas internas e a competência humana da linguagem. Por essas e por outras razões, o trabalho de Chomsky ficou conhecido por revolucionar a área da linguística⁹. Para ele, o objetivo da linguística deve ser justamente esse: mais do que se ater à análise do desempenho do falante, descrever as estruturas internas de competência, responsáveis por gerar as diferentes elocuições por meio do conjunto de regras recursivas [45].

3.2 Linguagem e Gramática

Chomsky define “linguagem” como sendo “um conjunto (finito ou infinito) de sentenças, cada qual limitada em comprimento e construída a partir de um conjunto finito de elementos” [38]. Assim, com um número finito de fonemas ou letras de um dado alfabeto é possível construir diferentes sentenças, faladas ou escritas, a partir da combinação destes símbolos em sequências limitadas. O conjunto dessas sentenças pode muito bem ser infinito, por exemplo. Da mesma forma, nota Chomsky, “um conjunto de ‘sentenças’ de um dado sistema matemático formal pode ser considerado uma linguagem” [38].

O conjunto V de elementos que compõem as diferentes sentenças é chamado de “vocabulário” ou “alfabeto” [48]. São exemplos, o conjunto de dígitos binários, $\{0, 1\}$, e o próprio alfabeto latino, $\{a, b, \dots, z\}$. Dessa maneira, as sentenças de uma determinada linguagem \mathcal{L} em V só serão consideradas válidas se forem compostas somente dos elementos de V . Além disso, é importante definir uma sentença (ou cadeia) que não possua nenhum elemento; esta é dita “nula” ou “vazia” e pode ser representada pelo símbolo ϵ .

Definição 3.1. (Fecho de Kleene) O operador de Kleene sobre um vocabulário V qualquer (nota-se V^*) indica o conjunto infinito de todas as sequências possíveis que podem ser representadas pelos elementos de V , incluindo a cadeia nula [48]. Por exemplo, definindo-se um alfabeto $V = \{\square, \triangle\}$, o fecho de Kleene nesse alfabeto fornece o conjunto infinito $V^* = \{\epsilon, \square, \triangle, \square\square, \square\triangle, \triangle\triangle, \triangle\square, \square\square\square, \dots\}$. Assim, posto de outra maneira, diz-se que \mathcal{L} é uma linguagem em V se e somente se $\mathcal{L} \subseteq V^*$.

Denota-se V^+ o conjunto $V^* - \{\epsilon\}$, i.e., o conjunto de Kleene sobre V excetuando-se a cadeia vazia.

A análise linguística de uma dada linguagem \mathcal{L} consiste na classificação de diferentes sequências em “gramaticais” ou “não gramaticais” [38], “corretas” ou “in-

⁹É de interessante estudo a pesquisa de Geoffrey K. Pullum [47], que visa a desmistificar alguns exageros acerca das contribuições de Chomsky e seus impactos na literatura nesse tema. Konrad Koerner também discute sobre isso, i.e., se acaso os avanços de Chomsky não seriam apenas uma releitura do paradigma estruturalista de Saussure [40]. Destaca-se, contudo, a grande diferença de visão entre Chomsky e Saussure no que concerne ao aspecto inato ou coletivo da linguagem.

corretas” [39], i.e., pertencentes à linguagem ou não, respectivamente. Conforme visto anteriormente, uma gramática \mathcal{G} da linguagem \mathcal{L} deve ser capaz de reconhecer e também de gerar todas (e tão somente) as sequências gramaticais de \mathcal{L} . Essa gramática é, portanto, avaliada em função das sentenças que ela gera — quanto mais sequências gramaticais e menos sequências não gramaticais (idealmente, zero) uma gramática da linguagem \mathcal{L} gerar, mais adequada enquanto formalismo representativo desta linguagem ela é.

Chomsky destaca que não se pode associar o conjunto de sequências gramaticais a um *corpus* qualquer de elocuições que seja observado por um linguista, como é feito na perspectiva estruturalista, por exemplo, na qual o funcionamento do sistema linguístico é descrito a partir da segmentação, identificação e classificação dos constituintes de um *corpus* [46]. O gerativista recusa esse tratamento, porque, da mesma forma que um falante nativo da língua, as gramáticas projetam esse *corpus* finito e “acidental” no conjunto infinito das sequências gramaticais [38], e tal *corpus* — o conjunto finito de sentenças produzidas por uma linguagem — não é representativo de todo seu potencial criativo. Para Chomsky, uma análise sintática real, mais do que uma descrição de um *corpus*, só é possível ao adentrar no domínio das estruturas sintáticas profundas, construindo assim um “modelo da competência dos falantes” [46]. Além disso, Chomsky discute que para determinar se uma dada sequência é gramatical ou não gramatical não é possível se ater ao seu significado, e.g., em português, as sentenças “uma ideia verde incolor dormiu” e “dormiu incolor verde ideia uma” não fazem sentido, mas pode-se dizer que apenas a primeira é gramatical.

Em [45], Chomsky classifica os símbolos de um vocabulário V no qual uma gramática se baseia em símbolos formativos (e.g., “o”, “menino”) e símbolos de categoria ou classe (e.g., “S”, “SN”, “V”)¹⁰. Os símbolos formativos também são chamados “constantes” ou “terminais” e representam as palavras em uma sentença [48]. Os símbolos de categoria são as diferentes classes gramaticais e, porque estão sujeitos aos passos de derivação das regras sintáticas, são ditos “variáveis” ou “não terminais”. Dessa forma, duas coisas podem ser afirmadas sobre o vocabulário V e sua composição:

$$V = V_N \cup V_T \quad \text{e} \quad V_N \cap V_T = \emptyset; \quad (3.1)$$

esta última é equivalente a dizer que os conjuntos de símbolos terminais (V_T) e não terminais (V_N) são disjuntos.

De maneira geral, uma gramática \mathcal{G} é denotada por uma quádrupla do tipo $\langle V_N, V_T, R, S \rangle$ [48], onde V_T e V_N são os conjuntos de símbolos definidos acima, R

¹⁰Chomsky subdivide os símbolos formativos em lexicais e gramaticais, que aparecem em “profundidades” diferentes das regras gramaticais. Para mais detalhes sobre essa divisão dos formantes, a referência [45] é um bom ponto de partida.

é o conjunto de regras de reescrita e $S \in V_N$ é o símbolo de partida ou inicial. O conjunto R é um conjunto finito de relações $r : \alpha \rightarrow \beta$, nas quais $\alpha \in V^+$ e β é uma cadeia em V^* [48].

Cada aplicação de uma regra r corresponde a um passo de derivação. Por exemplo, se s_1 e s_2 são cadeias em V^* , interpreta-se

$$s_1\alpha s_2 \xRightarrow{\mathcal{G}} s_1\beta s_2 \quad (3.2)$$

como sendo a derivação direta da cadeia $s_1\beta s_2$ por meio da aplicação da regra $\alpha \rightarrow \beta$ sobre a cadeia $s_1\alpha s_2$, em uma gramática \mathcal{G} . Genericamente, diz-se que $s_a \xRightarrow[\mathcal{G}]{*} s_b$ (lê-se: “uma cadeia s_a é derivada em uma cadeia s_b em um número qualquer de passos de derivação de uma gramática \mathcal{G} ”), se existirem regras de reescrita em \mathcal{G} que possibilitem

$$s_1 \xRightarrow{\mathcal{G}} s_2, s_2 \xRightarrow{\mathcal{G}} s_3, \dots, s_{m-1} \xRightarrow{\mathcal{G}} s_m, \quad (3.3)$$

com $s_1, s_2, s_3, \dots, s_{m-1}, s_m \in V^*$ [48], se $s_1 = s_a$ e $s_m = s_b$.

Finalmente, define-se a linguagem \mathcal{L} gerada pela gramática \mathcal{G} como sendo [48]:

$$\mathcal{L}(\mathcal{G}) = \{s \mid s \in V_T^* \text{ e } S \xRightarrow[\mathcal{G}]{*} s\}, \quad (3.4)$$

ou seja, uma sentença s pertence à linguagem $\mathcal{L}(\mathcal{G})$ e, portanto, é gramatical, se é composta somente dos formantes terminais do vocabulário e se pode ser produzida, partindo-se de S , através da aplicação das regras de reescrita. Uma cadeia de \mathcal{L} é também dita “terminal”, porque nenhum de seus símbolos pode ser substituído [45]. A seguir, são apresentados alguns exemplos de gramáticas.

Exemplo 3.3. Uma gramática $\mathcal{G} = \langle V_N, V_T, R, S \rangle$ é definida a partir dos alfabetos $V_N = \{S\}$ e $V_T = \{0, 1\}$, e das regras de produção:

$$\left\{ \begin{array}{l} r_1 : S \rightarrow 0S0; \\ r_2 : S \rightarrow 1S1; \\ r_3 : S \rightarrow \epsilon; \\ r_4 : S \rightarrow 0; \\ r_5 : S \rightarrow 1. \end{array} \right. \quad (3.5)$$

Aplicando-se as regras de reescrita algumas vezes, as sequências produzidas podem ser colocadas em um conjunto $\{\epsilon, 0, 1, 00, 000, 010, 11, 101, 111, 0000, \dots\}$. É fácil perceber que a gramática \mathcal{G} gerará uma linguagem de sequências dos dígitos binários com uma característica particular — são todas palíndromas. Dessa forma, verifica-se que 0010100 é uma sentença gramatical, gerada a partir da aplicação sucessiva das regras r_1, r_1, r_2 e r_4 .

Exemplo 3.4. Em [48], é apresentado o exemplo de uma gramática \mathcal{G} definida sobre o mesmo vocabulário do Exemplo 3.3, mas com regras de produção do tipo $R = \{S \rightarrow 0S1; S \rightarrow 01\}$. Pode ser verificado que a aplicação das regras de reescrita dará origem a uma linguagem na qual as sentenças são do tipo 0^n1^n , com $n \geq 1$. A sequência 0010100, embora definida com os símbolos formativos, não é gramaticalmente correta nesse caso.

Duas gramáticas \mathcal{G}_1 e \mathcal{G}_2 são ditas equivalentes se $\mathcal{L}(\mathcal{G}_1) = \mathcal{L}(\mathcal{G}_2)$. Esse não é o caso das gramáticas apresentadas nos Exemplos 3.3 e 3.4, porque geram linguagens distintas.

Retomando o Exemplo 3.1, por meio da árvore gerada pela análise dos constituintes imediatos é possível identificar as seguintes regras de reescrita¹¹:

$$\begin{aligned}
 (i) \quad & S \rightarrow SN \frown SV; \\
 & SV \rightarrow V \frown SN; \\
 & SN \rightarrow \text{Det} \frown N; \\
 & SN \rightarrow N \frown N; \\
 & \text{Det} \rightarrow o; \\
 (ii) \quad & V \rightarrow \text{adora}; \\
 & N \rightarrow \text{menino}; \\
 & N \rightarrow \text{batata}; \\
 & N \rightarrow \text{doce}.
 \end{aligned} \tag{3.6}$$

As regras apresentadas acima para uma possível gramática do português, embora sejam suficientes para regerar a sentença “o menino adora batata-doce”, também são capazes de produzir cadeias atípicas como, por exemplo, “o doce adora o menino” ou ainda “batata doce adora batata doce”.

3.2.1 Hierarquia de Chomsky

Os diferentes tipos de gramáticas podem ser distinguidos pela natureza das restrições às quais elas estão sujeitas [7, 48], i.e., ao formato das regras de substituição que as descrevem. Cada tipo de gramática gera um tipo diferente de linguagem formal e pode ser associado a um reconhecedor específico¹². Esses tipos são denominados,

¹¹O arco “ \frown ” indica a operação de concatenação dos símbolos terminais ou não terminais. Chomsky utiliza essa notação em [45, 49] e o sinal de adição “+” em [38]. Chomsky faz notar uma distinção entre as regras que substituem símbolos não terminais por formantes léxicos (indicadas em (ii)) e as demais regras.

¹²Um reconhecedor ou aceitador é uma máquina abstrata (autômato) capaz de processar uma sequência de entrada e produzir uma saída binária, que indica se essa sequência foi aceita ou rejeitada. É fácil perceber a similaridade que há entre esse formalismo e as gramáticas, con-

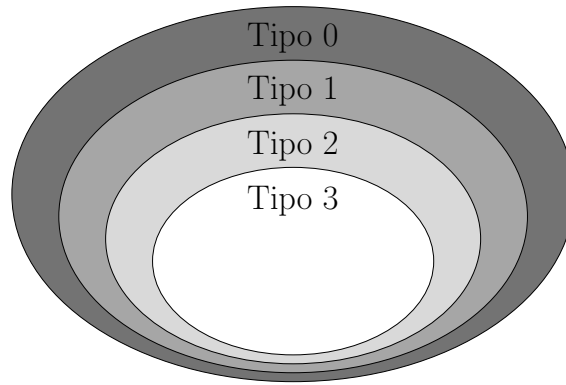


Figura 3.3: Hierarquia de Chomsky.

usualmente, na hierarquia de Chomsky, de tipos 0, 1, 2 e 3, e podem ser diferenciados de acordo com seu potencial gerativo — quanto maior é a sua “ordem”, mais restritivas são as regras de produção e menor é o seu potencial gerativo [7]. O valor da ordem também está relacionado à complexidade computacional que ele apresenta para derivação de suas expressões [7], i.e., quanto menor a ordem, maior o tempo para computação de suas derivações (podendo, eventualmente, ser infinito).

Sobre os tipos de gramáticas, tem-se que (Figura 3.3):

$$\text{Tipo 3} \subset \text{Tipo 2} \subset \text{Tipo 1} \subset \text{Tipo 0}, \quad (3.7)$$

o que deverá ficar claro a partir das definições que se seguem.

Gramática do tipo 0

É uma gramática sem restrições, i.e., cujas regras de substituição não apresentam restrições, responsáveis por gerar linguagens do tipo 0.

Seu reconhecedor é a máquina de Turing não determinística (NTM — *Non-deterministic Turing Machine*) [48].

Gramática do tipo 1

É uma gramática cujas regras de produção estão sujeitas a restrições do tipo:

$$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2, \quad (3.8)$$

com $\alpha_1, \alpha_2, \beta \in V^*$, $\beta \neq \epsilon$ e $A \in V_N$. Esse tipo de gramática também é dito sensível ao contexto¹³, indicando que A pode ser substituído por β no contexto de α_1 e α_2 [48].

forme definidas por Chomsky. Um exemplo de reconhecedor é uma máquina de estados finitos ou autômato finito, cuja versão não determinística pode ser vista na Seção 2.2.

¹³Em [45], Chomsky usa a notação $A \rightarrow \beta/\alpha_1-\alpha_2$.

Sem perda de generalidade, as regras de produção de uma gramática sensível ao contexto também podem ser escritas como sendo na forma:

$$\alpha \rightarrow \beta, \tag{3.9}$$

com $|\beta| \geq |\alpha|$, onde $|\cdot|$ indica o comprimento da sequência (número de símbolos).

Seu reconhecedor é um autômato linearmente limitado (LBA — *Linear Bounded Automaton*) [48].

Gramática do tipo 2

As gramáticas do tipo 2 (livres de contexto) são aquelas que só admitem regras do tipo:

$$A \rightarrow \beta, \tag{3.10}$$

onde $|A| = 1$ e $\beta \neq \epsilon$. Esse tipo de regra permite que, na derivação das sentenças das linguagens livres de contexto, um símbolo não terminal A seja substituído por β qualquer que seja o contexto em que se encontre.

Seu reconhecedor é um autômato com pilha não determinístico (NPA — *Nondeterministic Pushdown Automaton*) [48].

Gramática do tipo 3

Gramáticas regulares são aquelas que apresentam regras de substituição do tipo:

$$A \rightarrow a \quad \text{e} \quad A \rightarrow aB, \tag{3.11}$$

onde A e B são símbolos não terminais e a é um símbolo terminal. Alternativamente, as regras para as gramáticas do tipo 3 podem ser escritas como: $A \rightarrow a$ e $A \rightarrow Ba$.

Seu reconhecedor é uma máquina de estados finitos não determinística (NFSM — *Nondeterministic Finite State Machine*), também conhecida como autômato finito não determinístico (NFA — *Nondeterministic Finite Automaton*) [48].

3.3 Aplicações à Composição

A aplicação do conceito expresso pelas gramáticas de Chomsky à Composição Algorítmica está diretamente relacionada a uma visão semiótica da música, i.e., da música como uma linguagem, uma forma de expressão e comunicação do pensamento, um sistema de signos etc. Pode-se identificar, nos diversos estilos musicais, estruturas hierárquicas que são a base desse formalismo [11], mas, conforme aponta Nierhaus [7], a aplicação desses conceitos à música e, mais especificamente, à Com-

posição Algorítmica não pode ser realizada de forma direta e descuidada, i.e., sem um conjunto de considerações prévias.

Por exemplo, em uma análise linguística, tem-se que estruturas profundas similares permitem a geração, por meio de transformações sucessivas, de estruturas superficiais e de “elocuições” que diferem no arranjo dos seus símbolos terminais. A mesma análise em um âmbito musical acabaria por revelar a dificuldade de se estabelecer uma “semântica” da música, uma vez que cada sentença gerada a partir de uma estrutura profunda comum seria, ela mesma, uma composição nova e essencialmente distinta. Desconsidera-se aqui, por simplicidade, a diferença que há entre a música notada na partitura e a sua interpretação por um músico, que “nunca são, diretamente, uma e a mesma coisa” [50]. Essa última pode ser entendida como fruto de uma análise interna do intérprete e, com efeito, sua própria expressividade contribuirá para a compreensão que o ouvinte final terá do significado de um motivo ou, eventualmente, de uma peça completa.

Fernández e Vico [11] destacam que um ponto fundamental da abordagem gramatical de uma análise musicológica é a formalização do conjunto de regras recursivas, que dão origem aos processos gerativos. Essas regras têm, também aqui, um caráter hierárquico e podem ser agrupadas de acordo com a “fase” do processo de composição em que serão aplicadas: “[desde a definição] dos temas gerais da composição até o arranjo de notas individuais” [11]. As pesquisas na área usualmente se concentram no uso de gramáticas regulares ou livres de contexto (tipos 3 e 2, respectivamente), uma vez que é grande a dificuldade de implementação de gramáticas sensíveis ao contexto e gramáticas sem restrições [11]. Além disso, Fernández e Vico apontam que no início das pesquisas com gramáticas os musicólogos determinavam as regras gramaticais manualmente a partir de princípios da teoria da música (abordagem baseada no conhecimento — “*knowledge-based approach*” [7]), mas que outras abordagens são possíveis, como por exemplo a análise automática de um *corpus* para extração destas regras (“inferência gramatical” [7]) e até mesmo abordagens mistas com algoritmos evolucionários, nas quais o objetivo é otimizar a árvore de derivação das estruturas gramaticais [11].

Uma composição gerada por meio de uma gramática, por definição, satisfará o critério de “boa formação” da mesma [7]. Contudo, segundo Nierhaus [7], isso não significa que a composição estará de acordo com todas as regras musicais implícitas em cada caso, o que é ainda mais verdadeiro no caso da imitação de um estilo a partir da análise de inferência de um determinado *corpus*. Trata-se do mesmo problema descrito no Capítulo 2 para as cadeias de Markov: as regras de reescrita inferidas pela gramática sobre as transições das unidades musicais no *corpus* podem, por vezes, ignorar algumas limitações musicais implícitas que só seriam observadas ao se tratar de um contexto mais abrangente [7].

3.3.1 O que é um símbolo?

A definição dos símbolos em uma gramática para a linguagem musical é tão ampla quanto o observado na Seção 2.3.1 sobre os estados de uma cadeia de Markov. Gramaticalmente falando, as unidades da gramática podem assumir qualquer forma, desde notas e acordes (que atuam como símbolos terminais) até elementos mais complexos como movimentos harmônicos, motivos e frases inteiras (que podem também assumir um caráter não terminal). Em verdade, parte essencial da análise gramatical (e subsequente processo de síntese) é a segmentação da música em diferentes trechos, agrupamentos maiores das suas unidades. Diversos pesquisadores se debruçaram sobre esse problema, atacando-o com o uso de regras explícitas ou através de métodos bayesianos, conforme o descrito na seção a seguir.

3.3.2 Revisão bibliográfica

Nesta seção, são revisitados alguns trabalhos importantes da aplicação dos conceitos de Chomsky em composição musical dos quais se destacam, particularmente, as abordagens musicológicas de Steedman e de Johnson-Laird.

Análise schenkeriana

Ao se tratar de análise gramatical da música, de suas estruturas profundas e superficiais, tem-se como referência na literatura o pioneiro trabalho de Heinrich Schenker (1868–1935), “*Der Freie Satz*” [51]. Em seu método de análise da música tonal, antes mesmo do desenvolvimento por Chomsky do paradigma gerativo–transformacional, Schenker estabelece uma forma de interpretar uma composição e suas estruturas subjacentes por meio da descrição de relações hierárquicas entre as notas da partitura [7]. Através de sua análise, Schenker reforça a noção de tonalidade na música — em especial o conceito de espaço tonal —, mostrando como cada nota participa da estrutura global da composição, i.e., qual sua função no todo. Como bem coloca Temperley, “a análise schenkeriana se preocupa, em parte, em desemaranhar os fios [...] que formam o tecido da maior parte das peças tonais” [10], contemplando assim “melodia, contraponto e harmonia” [52].

Schenker define o que chama de “*Ursatz*”, a estrutura fundamental abstrata que se encontra no plano mais profundo da composição — o “*Hintergrund*” (“nível fundamental”) [1, 7]. A *Ursatz* é composta de dois elementos básicos, que formam uma estrutura em duas vozes. O primeiro é o movimento harmônico essencial [1], o conjunto de tons básicos da progressão harmônica, chamado “*Bassbrechung*” (“arpejamento do baixo”) [7]. A partir de diversos exemplos de grandes compositores dos séculos XVIII e XIX, Schenker determinou que todas as composições tonais teriam como suporte uma progressão harmônica do tipo I–V–I, ou ainda formas mais

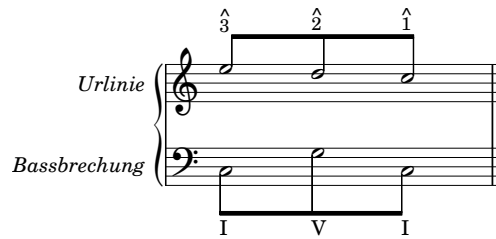


Figura 3.4: Exemplo de *Ursatz* — *Urlinie* e *Bassbrechung* — de uma análise schenkeriana. Adaptado de [51].

elaboradas como I-III-V-I e I-IV-V-I [1]. A linha melódica fundamental, a “*Urlinie*”, completa então a *Ursatz*. Essa é, basicamente, uma sequência simples e linear que descreve um movimento em direção à nota fundamental da tônica (indicada por “ $\hat{1}$ ”), partindo, usualmente, de um grau “ $\hat{3}$ ”, “ $\hat{5}$ ” ou “ $\hat{8}$ ”, i.e., de uma nota qualquer do acorde da tônica. A Figura 3.4 mostra um exemplo possível de *Ursatz* em uma análise schenkeriana.

A partir de elaborações, ornamentos e prolongamentos na *Ursatz* no processo conhecido por “*Ausfaltung*” (do alemão, “desdobramento”), obtém-se o conjunto de níveis do “*Mittelgrund*” (“nível intermediário”) e, finalmente, o “*Vordergrund*” (“nível externo”) [1, 7], que corresponde a uma simplificação da composição notada na partitura¹⁴. O processo de *Ausfaltung* se desenvolve de forma similar ao contraponto de um *cantus firmus*. Conforme destaca Nierhaus [7], algumas das técnicas básicas utilizadas nesse processo são o arpejamento completo ou incompleto, a inserção de notas vizinhas e também a conexão de trechos inteiros usando-se do preenchimento com tons pertencentes a um mesmo acorde, em pequenos intervalos. Através da diferenciação dos níveis da composição, Schenker estabelece uma maneira de interpretar uma obra musical por meio da “percepção da coerência orgânica desta obra” [51], relacionando o que é tocado com o que fora intencionado. Contudo, Schenker não visa a descrever os “desígnios do autor [compositor]”, mas a preencher de significado suas “palavras”, referindo-as a uma “unidade possível de sentido” [53].

Segundo alguns pesquisadores, a análise schenkeriana representa o paralelo musical aos conceitos das estruturas profunda e superficial propostas por Chomsky [7]. O objetivo de Schenker, muito criticado [7], não era o de reduzir toda a música tonal a uma estrutura simples como a *Ursatz*, mas o de mostrar as muitas maneiras através das quais o prolongamento da tônica poderia se dar para gerar as diferentes composições. Isso está expresso no lema que, segundo ele próprio, personifica todo seu trabalho: “*semper idem, sed non eodem modo*” (“sempre a mesma coisa, mas não da mesma maneira”) [51], referindo-se às diferentes formas que as notas de pas-

¹⁴É evidente que a teoria de análise de Schenker traça o caminho contrário ao descrito aqui, i.e., a partir de reduções do “*Vordergrund*” até o “*Hintergrund*”. A teoria foi aqui exposta segundo seu caminho gerativo, o caminho do compositor, para facilitar a comparação com os conceitos de Chomsky.

sagem e ornamentações são incorporadas a partir do *Hintergrund* até finalmente o *Vordergrund*.

Teoria gerativa da música tonal

Em “*A Generative Theory of Tonal Music*” [54], Fred Lerdahl e Ray Jackendoff, respectivamente, musicólogo e linguista americanos, apresentam uma “descrição formal das intuições musicais de um ouvinte que é experiente em um idioma musical” [54], tendo como objetivo estabelecer um conjunto de regras gerais e compreender a própria “cognição da música, um fenômeno psicológico” [54]. Nessa tentativa de elaboração de uma teoria para a “competência musical” [55], nos mesmos termos da teoria de Chomsky, Lerdahl e Jackendoff traçam uma ponte entre as estruturas da música tonal e os processos perceptivo–cognitivos que levam à sua interpretação, através de elementos próprios da linguística [55], combinando, pois, aspectos da teoria de análise de Schenker (e.g., a redução/prolongamentos na duração) e princípios da teoria psicológica da *Gestalt*¹⁵ (a saber, a composição de um conjunto de regras de formação para condução das análises [55, 56]).

Para Lerdahl e Jackendoff, o ouvinte “perfeito”, versado na linguagem musical, “naturalmente organiza os sinais sonoros em unidades como motivos, temas, frases, períodos, grupos temáticos, seções e a própria peça” [54], unidades estas sem sobreposição que são denominadas “agrupamentos” [54]. Ao mesmo tempo, o ouvinte percebe uma sucessão regular de pulsos fortes e fracos, que ele associa aos eventos musicais — a este padrão dá-se o nome de “métrica” [54]. Segundo os pesquisadores, quatro tipos de estruturas hierárquicas podem ser associadas às composições tonais (duas das quais expressam diretamente as noções de agrupamentos e métrica previamente apresentadas); são elas [54, 57]:

- (1) a estrutura de agrupamento (“*grouping structure*”), que descreve a segmentação da música em unidades de vários tamanhos (e.g., motivos, frases, seções);
- (2) a estrutura métrica (“*metrical structure*”), que descreve a hierarquia dos pulsos fortes e fracos na música, nos diferentes níveis métricos;
- (3) a redução temporal–duracional (“*time-span reduction*”), que estabelece a importância relativa de cada evento musical, de acordo com sua posição nas

¹⁵A teoria da *Gestalt*, também conhecida como “psicologia da forma”, é uma teoria psicológica da percepção humana. Essa teoria advoga que a construção da percepção do “todo” não pode ser tida como, simplesmente, a soma das suas “partes” — o “todo” é percebido como uma entidade por si só, e de maneira diversa às suas componentes, sendo estruturado e organizado de acordo com certas “leis de formação” [55]. No âmbito visual, essas leis se baseiam em fatores de “proximidade, similaridade, boa continuidade e de destino comum” [56], entre outros.

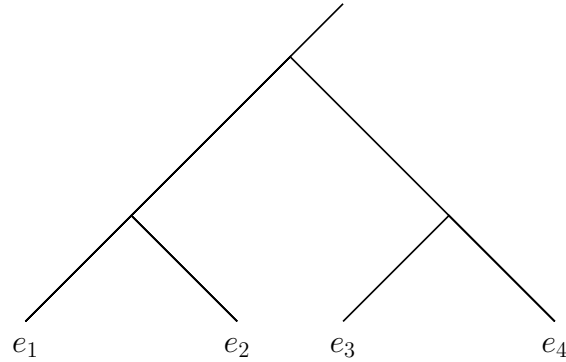


Figura 3.5: Exemplo de árvore reducional. Os eventos e_2 e e_3 estão subordinados, respectivamente, aos eventos e_1 e e_4 . Em um nível mais profundo, pode-se verificar que e_4 também é elaboração de e_1 . Adaptado de [57].

estruturas de agrupamento e métrica, i.e., determina os eventos mais significativos da composição, em termos da sua “estabilidade”, por meio de uma série de simplificações recursivas da estrutura rítmica. Os níveis mais superficiais são determinados pela estrutura métrica, enquanto a estrutura de agrupamentos se faz notar nos níveis mais profundos;

- (4) a redução prolongacional¹⁶ (“*prolongational reduction*”), que estabelece uma hierarquia dos *itches* na música de acordo com padrões de tensão e relaxamento melódicos e harmônicos, determinados pelo grau de continuidade (prolongação) ou progressão dos eventos melódicos e harmônicos, e.g., a repetição de um evento não traz uma sensação de tensão ou relaxamento como a percebida entre dois eventos similares (com inversão de acorde) ou, mais ainda, entre eventos completamente diferentes.

As estruturas reducionais admitem uma representação similar à das gramáticas de Chomsky, i.e., estruturas em árvore, nas quais os elementos musicais da superfície são sucessivamente reduzidos, sendo interpretados dos eventos estruturalmente mais importantes, que permanecem nos níveis mais profundos. Conforme notam os autores, aqui há uma fundamental diferença em relação às árvores sintáticas da linguística, porque as árvores reducionais “não expressam categorias gramaticais”, mas “relações de elaboração” [57]. Nas árvores musicais de Lerdahl e Jackendoff, há que se notar a maneira que as ramificações se dão: uma ramificação à direita indica uma elaboração do evento precedente, enquanto uma ramificação à esquerda representa um evento subordinado ao evento que o sucede. A Figura 3.5 exemplifica esse tipo de relação.

Para cada uma dessas componentes hierárquicas, os pesquisadores descrevem, então, um conjunto de regras, divididas em [54, 57]: “regras de boa formação” (“*well-formedness rules*”), que especificam as “possíveis descrições estruturais” [54]; e “re-

¹⁶Essa última componente, notam os autores, aproxima seu trabalho da teoria proposta por Schenker [57].

gras de preferência”, que indicam, dentre as possíveis estruturas, quais correspondem às intuições de um ouvinte experiente¹⁷. Lerdahl e Jackendoff acenam às diferentes possibilidades de interpretação de uma dada “superfície” musical, por isso mesmo trabalham em termos das regras de preferência. Essas regras, que fazem “a maior parte do trabalho de elaboração das análises” [54], apontam para as estruturas profundas preferíveis, i.e., mais coerentes em termos da cognição musical. Sua teoria não pretende destacar a “gramaticalidade” das superfícies musicais, mas indicar a viabilidade de uma dada interpretação estrutural [57] — nisso o trabalho de Lerdahl e Jackendoff se distancia das gramáticas da linguística, tendo como ícone desta distinção as regras de preferência, sem paralelo na teoria linguística padrão [54].

Para exemplificar, as seguintes regras constituem o conjunto de regras de boa formação para a estrutura de agrupamentos (GWFRs — *Grouping Well-Formedness Rules*) [54]:

GWFR1 Qualquer sequência contígua de eventos sonoros (*pitches*, elementos percussivos ou outros) pode constituir um grupo, e somente sequências contíguas podem constituir um grupo.

GWFR2 Uma peça completa constitui um grupo.

GWFR3 Um grupo pode conter grupos menores.

GWFR4 Se um grupo G_1 contém parte de um grupo G_2 , então G_1 deve conter G_2 em sua totalidade.

GWFR5 Se um grupo G_1 contém um grupo menor G_2 , então G_1 deve ser exaustivamente dividido em grupos menores.

Um exemplo de regra de preferência para a estrutura de agrupamentos (GPR — *Grouping Preference Rule*) é a GPR2, sobre a proximidade dos elementos, cuja inspiração pode ser traçada até a teoria da *Gestalt*. Essa regra postula que, dada uma sequência de quatro notas, $n_1n_2n_3n_4$, a transição n_2 – n_3 pode ser ouvida como um limite entre dois grupos se [54]:

- (a) (Pausa) o intervalo de tempo entre o fim de n_2 até ao início de n_3 é maior que aquele entre o fim de n_1 até o início de n_2 e que aquele entre o fim de n_3 até o início de n_4 , ou se
- (b) (Ataque) o intervalo de tempo entre os ataques de n_2 e de n_3 é maior que aquele entre os ataques de n_1 e de n_2 , e que aquele entre os ataques de n_3 e de n_4 .

¹⁷Há também um terceiro conjunto de regras, ditas “regras de transformação”. Essas regras permitem alterações bem comportadas dos elementos musicais na superfície de forma a permitir o tratamento de eventuais más-formações, e.g., sobreposição dos agrupamentos, como eventos bem formados [57].

A regra seguinte, GP3, prossegue na definição do limite entre dois agrupamentos, adicionando as seguintes condições, que tomam por base mudanças nos elementos da composição [54]:

- (a) (Registro) a transição n_2-n_3 envolve uma maior distância intervalar que n_1-n_2 e n_3-n_4 , ou
- (b) (Dinâmica) a transição n_2-n_3 envolve mudança na dinâmica, enquanto n_1-n_2 e n_3-n_4 não, ou
- (c) (Articulação) a transição n_2-n_3 envolve mudança de articulação, enquanto n_1-n_2 e n_3-n_4 não, ou
- (d) (Duração) n_2 e n_3 têm durações diferentes, enquanto os pares n_1n_2 e n_3n_4 não diferem em duração.

Lerdahl e Jackendoff admitem inspiração na análise de Schenker; inspiração esta que está presente, por exemplo, na preocupação com a estrutura hierárquica da composição, bem como em alguns processos e transformações adotados pelos autores. Destacam, contudo, algumas importantes diferenças entre as duas teorias. Segundo os pesquisadores, o trabalho de Schenker pode ser entendido como uma “proto-teoria gerativa da música tonal —, isto é, [...] um conjunto finito de princípios capazes de gerar recursivamente um conjunto potencialmente infinito de peças tonais” [54]. Contudo, Schenker não teria desenvolvido uma gramática formal *per se* [54] e, além disso, sua preocupação principal teria um caráter majoritariamente estético, artístico. Isso contrasta com a abordagem psicológica de Lerdahl e Jackendoff, que busca sistematizar o próprio processo da cognição musical a partir da formulação de um “sistema de regras que gera descrições estruturais” [57]. É a gramática de Lerdahl e Jackendoff que determina as estruturas preferenciais de interpretação, e não um analista, como é o caso da abordagem schenkeriana. Enquanto Schenker objetiva “esclarecer as obras-primas” [57], Lerdahl e Jackendoff desejam alcançar os princípios da percepção da linguagem musical¹⁸ e, para tanto, visitam a estrutura hierárquica das composições em maior detalhe. Para os pesquisadores, uma composição musical seria, antes de tudo, uma entidade mentalmente construída, mais do que a partitura ou a sua interpretação, que dela são apenas representações parciais [54].

Embora os autores afirmem, em [54], que sua teoria “não é capaz de fornecer um procedimento computável para determinar as análises musicais”, mais recentemente outros pesquisadores investiram esforços neste sentido [58, 59].

¹⁸Lerdahl e Jackendoff afirmam que Schenker estava “ao menos implicitamente preocupado com a cognição musical” [57].

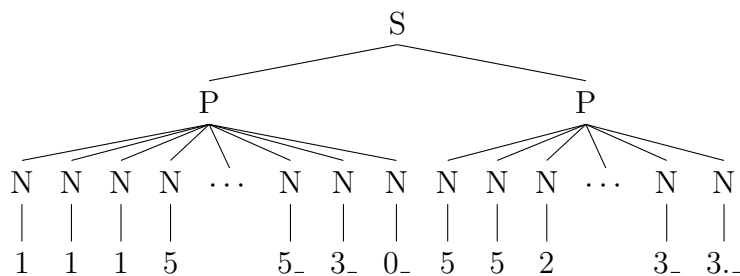


Figura 3.6: Exemplo de melodia e estrutura de frases em concordância com o sistema de codificação empregado por Bod [61].

Verossimilhança e simplicidade

Outro exemplo de tratamento da música em estruturas superficiais e profundas é o trabalho desenvolvido por Rens Bod [60]. O modelo de Bod é uma abordagem probabilística (bayesiana) para detecção e segmentação de frases melódicas. Assim como o visto na análise de constituintes de Bloomfield e nas estruturas gramaticais de Chomsky, Bod representa as estruturas frasais com o auxílio de árvores, e uma dada canção é analisada em diferentes níveis: a raiz “S”; o nível das frases, com nós do tipo “P”; o nível das notas, indicadas pelos nós “N”; e, por fim, folhas terminais que apresentam *pitch* e duração de cada nota [61]. Os *itches* são representados em graus de escala, enquanto as durações são tomadas em relação ao menor valor rítmico presente na canção, cujo valor pode ser dobrado (“_”) ou pontuado¹⁹ conforme a notação musical convencional (“.”) [60]. Por exemplo, o símbolo “b2_” representa um grau de escala $b\hat{2}$ com duração equivalente ao dobro da menor unidade rítmica presente na música. A Figura 3.6 apresenta um exemplo de melodia segmentada em frases (indicadas, na partitura, através das ligaduras) de acordo com o sistema de codificação proposto por Bod.

Bod ataca o problema da ambiguidade, ou seja, da dificuldade de determinação da organização estrutural por trás de um texto, uma composição musical ou uma imagem. Para inferir corretamente as estruturas frasais, e.g., agrupar as notas sob os nós “P” [60], o modelo de Bod atua com base em dois princípios: verossimilhança e simplicidade [60, 61]. Segundo Bod, certas subárvores²⁰ (e, conseqüentemente, as estruturas locais que estas determinam) serão mais prováveis, de acordo com o primeiro critério, por serem mais frequentes em um dado *corpus* previamente analisado. Por esse critério, a estrutura global mais apropriada para modelar a percepção de

¹⁹Uma nota pontuada tem duração 1,5 vez a duração da figura original, i.e., tem a ela adicionada metade da duração de sua figura.

²⁰Bod define, de maneira ampla, uma subárvore como sendo um subconjunto qualquer de nós conectados de uma árvore e suas expansões [60].

um texto ou uma canção, por exemplo, seria justamente a estrutura mais provável, em função da probabilidade de suas partes — as subárvores. Alternativamente, uma árvore pode ser preferível, enquanto modelo da percepção humana para uma determinada entrada, se for mais simples, i.e., se possuir o menor número de subárvores presentes no mesmo *corpus* [60, 61]. Grosso modo, no âmbito musical os dois critérios preferenciais da análise de Bod equivalem, respectivamente, às análises estatísticas dos padrões de *pitch* em frases e do número de frases em uma dada melodia.

Os experimentos desenvolvidos por Bod o levaram a concluir que a combinação dos dois critérios — verossimilhança e simplicidade²¹ — fornece os melhores resultados em seu modelo da percepção das estruturas frasais. Além disso, a aplicação dos mesmos modelo e parâmetros em duas bases de dados de caráter distinto (um subconjunto do *Penn Treebank* [62], contendo frases anotadas do *Wall Street Journal*, e um subconjunto da *Essen Folksong Collection*, de canções populares europeias também anotadas com suas estruturas percebidas [63]) apresentou altas taxas de acerto, o que pode indicar um território comum para percepção das estruturas frasais na língua e na música [61]. Contudo, justamente por se basear em estatísticas extraídas dos diferentes *corpora*, a abordagem de Bod pode ser questionada [60]. Chomsky já falara sobre a impossibilidade de esse tipo de análise e a teoria da sintaxe coadunarem [38], uma vez que não permitem a construção de uma generalização para a linguagem estudada. Tal problema não ocorre com outros modelos de estruturas frasais, e.g., Schenker [51], Lerdahl e Jackendoff [54], que trabalham em cima de regras gerais.

Os modelos propostos por Schenker [51], Lerdahl e Jackendoff [54], e Bod [61] são modelos de análise que permitem profundas e importantes intuições acerca da percepção das estruturas musicais. São de grande importância na área da Composição Algorítmica, pois, conforme nota Nierhaus, “um modelo de análise válido pode também, na maioria das vezes, ser utilizado para geração de material musical” [7].

Uma gramática para *blues* de doze compassos

Em “*A Generative Grammar for Jazz Chord Sequences*” [64], Mark Steedman descreve uma gramática para geração de sequências de acordes, aproveitando-se de um “caráter recursivo” [64] que estas sequências usualmente apresentam. O sistema opera a partir de um pequeno número de regras de reescrita e permite gerar progressões harmônicas na forma específica do *blues* de doze compassos. Steedman justifica o uso desse formalismo ao intuir que, da mesma forma que o falante de uma língua distingue uma sentença qualquer das sentenças de sua língua nativa,

²¹O critério da simplicidade de Bod comunga, conforme o próprio autor aponta, dos princípios da teoria da *Gestalt* [61].

uma pessoa é capaz de diferenciar arranjos de notas aleatórios de arranjos “culturalmente típicos” [64]. O *blues* de doze compassos, uma forma musical comum no *jazz* e na música popular, evoluiu e se modificou rapidamente ao longo do tempo. Contudo, os músicos de *jazz* parecem concordar que todas as suas variantes derivam de um “esqueleto” simples e comum, uma sequência de acordes originária [64]. Essa sequência é composta de três frases de quatro compassos cada (que podem ser subdivididos até os *beats*, usualmente, quatro por compasso²²), sobre as quais se desenvolvem os acordes de tônica, subdominante e dominante [64]: a primeira frase é composta inteiramente de acordes de tônica (I); a segunda frase consiste de uma “cadência plagal”, formada por dois compassos com acordes de subdominante (IV) e dois compassos da tônica; e a terceira frase apresenta uma “cadência autêntica”, formada por dois compassos com acordes de dominante (V) que se “resolvem” em dois compassos da tônica. Tal sequência pode, então, ser representada por [64]:

$$I/I/I/I7||IV/IV/I/I||V7/V7/I/I, \quad (3.12)$$

onde as barras indicam a separação dos compassos e || é a transição entre frases. Alguns acordes são realizados como acordes de sétima “dominante” e criam a expectativa de um acorde subsequente com fundamental uma quinta abaixo ou uma quarta acima [64]. É o caso dos acordes de tônica no final da primeira frase e de dominante no início da terceira frase.

Steedman organiza sua gramática com base em observações da prática do *blues* de doze compassos e institui suas fundações em dois importantes princípios. O primeiro é a base das gramáticas conforme descritas por Chomsky: o conjunto de regras deve ser capaz de gerar todas e somente aquelas sequências de acordes reconhecíveis como possíveis no gênero do *blues* de doze compassos. Ao mesmo tempo, cada regra deve apresentar uma semântica definida, nos termos da harmonia subjacente, i.e., as regras devem carregar a noção de cadência e operar de acordo com os aspectos do sentido musical que são expressados pelos músicos e que estes reconhecem [64]. Na gramática de Steedman, a “Regra 0” determina que o símbolo inicial “S12”, representando uma sequência genérica que abrange doze compassos, seja substituído da seguinte forma:

$$\text{Regra 0 : S12} \longrightarrow I I7 IV I V7 I, \quad (3.13)$$

e é assumido, como convenção, que os lados direito e esquerdo das regras de produção possuem as mesmas durações — neste caso, cada um dos acordes ocupa um total de dois compassos [64]. As regras seguintes, apresentadas aqui apenas a mérito de ilustração, destacam pontos mais interessantes da prática do *blues* de doze compassos,

²²Steedman destaca que essa forma musical também aparece notada em $\frac{3}{4}$ ou $\frac{6}{8}$, sempre com um total de doze compassos [64].

Tabela 3.1: Regras da gramática de Steedman para geração de frases de 12 compassos [64].

0:		S12(m)	→	I(m)	I7	IV(m)	I(m)	V7	I(m)
1:		x(m)(7)	→	x(m)		x(m)(7)			
2:		x(m)(7)	→	x(m)(7)		Sd _x			
3a:	w	x7	→	D _x (m)7		x7			
3b:	w	xm7	→	D _x 7		xm7			
4:		D _x 7	x(m)(7)	→	bSt _x (m)(7)	x(m)(7)			
5:	x	x	x	→	x	St _x	M _x m		
6:	x(m)	x(m)	$\left\{ \begin{array}{c} D_x \\ St_x m 7 \\ L_x m 7 \end{array} \right\}$	→	x(m)	‡x ^o	$\left\{ \begin{array}{c} D_x \\ St_x m \\ L_x m 7 \end{array} \right\}$		

tratam da extensão das cadências, da substituição de acordes etc. O pesquisador constata, por exemplo, que, por vezes, ocorre a substituição de um acorde de tônica por um acorde de subdominante. Essa substituição não ocorre em qualquer lugar, o que faz com que das duas variantes

$$I/IV/I/I7||IV/IV/I/I||V7/V7/I/I \quad \text{e} \quad IV/I/I/I7||IV/IV/I/I||V7/V7/I/I, \quad (3.14)$$

apenas a primeira seja aceita enquanto *blues* de doze compassos. Steedman insere esse conceito na gramática através das seguintes regras de substituição:

$$\text{Regra 1 : } x(7) \rightarrow x \ x(7) \quad (3.15)$$

$$\text{Regra 2 : } x(7) \rightarrow x(7) \ Sd_x, \quad (3.16)$$

onde x é uma variável do conjunto de acordes, a indicação (7) representa que a regra se aplica tanto aos acordes simples quanto aos de sétima, e Sd_x significa o acorde que tem a subdominante de x como sua fundamental. A Tabela 3.1 apresenta a gramática completa, vista em [64], na qual é considerado também o caso dos acordes menores (indicados pela letra “m”). Além disso: w denota um acorde que sofrerá substituição (nas regras 3a e 3b); D_x , St_x e M_x representam, respectivamente, os acordes da dominante, da supertônica (II) e da medianta (III) de x ; e o uso de chaves na última regra indica algumas opções de sua aplicação.

Em [64], Steedman mostra como seu conjunto de regras é capaz de produzir um conjunto padrão de sequências de *blues* de doze compassos²³. Sua preocupação não é formar “boas” sequências, mas “especificar o conjunto de possíveis [*blues* de] doze compassos — bons, ruins e indiferentes” [64], por mais que alguns pareçam por de-

²³Steedman também sugere um conjunto de regras opcionais para a especificação de acordes como “acordes menores e acordes de sexta, sétima maior e menor, nona [...] e assim por diante” [64].

mais complexos para serem tocados ou compreendidos. O pesquisador discute que o conjunto compacto e simples de regras de reescrita parece descrever adequadamente esse subconjunto de sequências de acordes harmonicamente “significativas” [64]. Steedman não se estende no assunto da geração das sequências, i.e., se são geradas mais sequências ou menos do que as musicalmente aceitáveis, argumentando em favor da relevância semântica das regras de produção.

Improvisações de *jazz*

Em um trabalho similar, “*Jazz Improvisation — A Theory at the Computational Level*” [65], Philip Johnson-Laird busca delinear uma gramática para a descrição de improvisações do *jazz* moderno. De forma simplificada, tais improvisações consistem de uma melodia “extemporizada” combinada com uma sequência de acordes tonais, que é repetida enquanto for necessária [65]. São compostas por um solista — como trompete ou saxofone — acompanhado de uma seção rítmica — piano, contrabaixo e bateria. Além de fornecer o ritmo para a improvisação, o piano e o contrabaixo também desenvolvem, respectivamente, uma sequência de acordes e uma linha de baixo [65].

Ao introduzir a caracterização das sequências de improvisação, Johnson-Laird descreve uma gramática regular para a estrutura rítmica de canções de Natal e discute sobre as dificuldades na geração de sequências gramaticalmente corretas, em particular, sobre os problemas de subgeração²⁴ e supergeração²⁵. O pesquisador chega à conclusão, anteriormente levantada por Chomsky, que a elaboração de gramáticas com base em um *corpus* usualmente incorre no problema do primeiro tipo, uma vez que o material disponível no *corpus* pode não representar todas as possibilidades para aquela linguagem específica. Ao extrapolar um sistema de regras a partir dos dados, contudo, o linguista ainda pode se deparar com essas deficiências da gramática [65]. Johnson-Laird prossegue, apresentando uma gramática regular para a estrutura rítmica nas improvisações do músico Charles Parker e argumentando que, como não são observados no *corpus* quaisquer tipos de restrições de uma frase sobre outra ou uma estrutura interna complexa, este tipo de gramática (com pouca memória de trabalho) é suficiente para a caracterização dos ritmos das improvisações de *jazz*. As regras dessa gramática, em geral, assumem a forma:

$$S_0 \longrightarrow \text{♪} S_1, \quad (3.17)$$

e todas as transições da máquina de estados finitos são feitas equiprováveis²⁶.

²⁴Do orig., “*undergenerate*”, i.e., quando a gramática não é capaz de gerar algumas sequências que são caracterizadas como válidas.

²⁵Do orig., “*overgenerate*”, i.e., quando a gramática gera sequências gramaticalmente incorretas.

²⁶Johnson-Laird destaca que, se as transições apresentassem pesos diferentes, essa máquina de

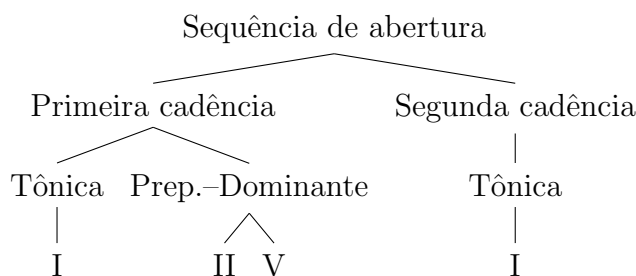


Figura 3.7: Representação da estrutura de uma progressão de acordes. Adaptado de [65].

Tabela 3.2: Regras da gramática de Johnson-Laird para geração sequências de acordes em oito compassos [65].

Oito-compassos	→	Quatro-primeiros	Quatro-últimos
Quatro-primeiros	→	Cadência-abertura	Cadência-abertura
	→	Cadência-abertura'	Cadência-abertura
Quatro-últimos	→	Cadência-meio	Cadência-abertura
Cadência-abertura	→	I	I
	→	I	V
Cadência-abertura'	→	I	III
	→	I	IV
Cadência-meio	→	I	IV
	→	I	V
	→	IV	I

Inspirado pelo trabalho de Steedman [64], Johnson-Laird investiga um *corpus* de improvisações de *jazz* moderno para desenvolver uma gramática de suas progressões de acordes. A estrutura hierárquica desse tipo de progressão, cujo exemplo pode ser visto na Figura 3.7, sugere o uso de um tipo mais poderoso de gramática — gramática livre de contexto. A Tabela 3.2 apresenta algumas das regras mais simples estipuladas por Johnson-Laird em [65]. O pesquisador sugere a necessidade de um conjunto de regras que atuem como “transdutores” [65], transformando as sequências produzidas pelas regras básicas através da interpolação de acordes (em função do ciclo das quintas), de maneira similar ao mecanismo de extensão das cadências autênticas de Steedman [64], por exemplo²⁷.

Por último, Johnson-Laird descreve a geração das linhas de baixo e de melodias. Para a linha de baixo, é uma característica do *jazz* moderno o “baixo caminhante” (“*walking bass*” [65]), que estabelece um padrão rítmico constante com uma nota

estados geraria sequências com características de uma cadeia de Markov (cf. Seção 3.4.1).

²⁷Em verdade, as restrições musicais necessárias para esse tipo de regra “transdutora” exigiria o uso de gramáticas sensíveis ao contexto, com maior complexidade e uso de memória. Johnson-Laird apresenta uma solução computacionalmente menos custosa em [65].

para cada pulso do compasso²⁸. Assim, sem considerar a complexidade rítmica das frases, Jonhson-Laird se atém às escolhas das notas, que devem obedecer à sequência harmônica do tema. Ele sugere que essas escolhas estão fundamentadas em alguns conjuntos de critérios. O primeiro conjunto trata da sequência de acordes que serve como base para essa melodia, definindo as notas “concordantes” e também as notas de passagem. Os critérios desse conjunto são, em grande parte, “acessíveis à introspecção”, podendo ser “ensinados e aprendidos explicitamente” [65]. Outro conjunto seria, nas palavras do pesquisador, “menos tangível” [65] e abordaria o conceito de “contorno melódico”, i.e., como se dá, de forma abrangente, a sucessão de movimentos graduais e saltos intervalares²⁹.

Johnson-Laird sugere que a “computação” desempenhada pelo músico para composição das linhas de baixo poderia seguir, portanto, um procedimento em duas etapas. Para cada nota: primeiro, estabelece-se o contorno melódico, e.g., se o intervalo é grande ou não (maior que um intervalo de segunda); em seguida, define-se seu *pitch* de maneira precisa, a partir da progressão de acordes subjacente. O contorno melódico serve, dessa forma, para restringir o conjunto de possíveis notas harmonicamente aceitáveis e de passagem já determinadas para cada acorde. Outras restrições sobre o uso de notas de passagem também são aplicadas [65]. A Figura 3.8 apresenta parte de uma máquina de estados para a produção de contornos melódicos, que equivale a uma gramática regular. O procedimento final para a formulação de uma melodia para a improvisação pode ser feito similar à computação da linha de baixo, adicionando-se, contudo, uma gramática para o ritmo (para a determinação de pausas e sequências rítmicas) [65].

3.3.3 Outros trabalhos

Os trabalhos revisitados na seção anterior correspondem a uma pequena parcela da pesquisa musicológica e de composição inspirada pelo paradigma gramatical. Outros trabalhos também são de grande interesse como, por exemplo, a ferramenta desenvolvida por Curtis Roads [66], que permite descrever e compilar uma gramática livre de contexto, assim gerando composições de acordo com as regras formuladas pelo usuário desde os níveis hierárquicos mais profundos até os terminais dessa linguagem — os “objetos sonoros” [66]. Em [67], Johan Sundberg e Björn Lindblom desenvolvem uma gramática para a descrição de cantigas de roda suecas do século XIX, levantando informações sobre ritmo, harmonia e melodia em cada um de seus oito

²⁸Como Jonhson-Laird observa, as durações das notas variam de uma forma complexa, em função dos desvios expressivos do próprio músico. Para simplificar, o pesquisador sugere que a realização das durações das notas seja entendida como resultado de um “filtro” pelo qual passam os padrões rítmicos concebidos pelo músico originalmente [65].

²⁹Esses dois critérios possuem uma grande similaridade com as funções harmônica e de proximidade apresentadas por Hiller e Isaacson na *Illiad Suite* [1, 20] (cf. Seção 2.3.2).

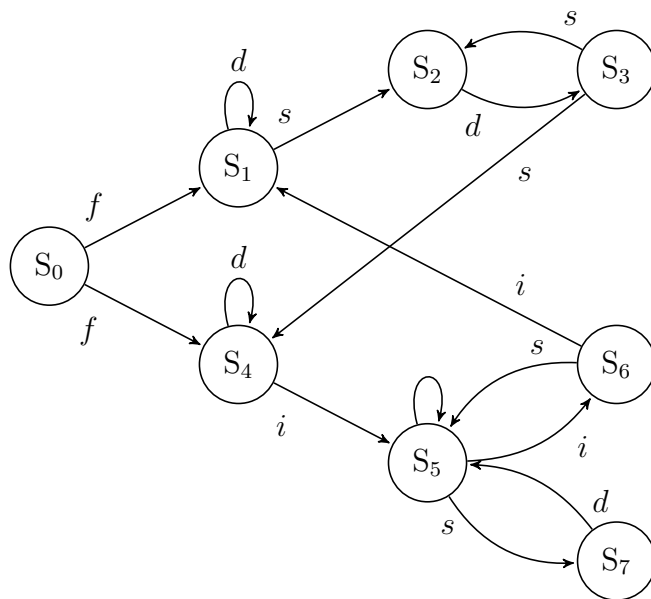


Figura 3.8: Parte de uma máquina de estados para geração de contornos melódicos. As excitações f , d , s e i representam, respectivamente, a primeira nota, a repetição da nota anterior, um pequeno intervalo e um grande intervalo na produção do contorno melódico. Adaptado de [65].

compassos. A partir da interpretação de “graus de proeminência” — inferidos da estrutura hierárquica (período, frases, subfrases, compassos e *beats*) —, que determinam a aplicação de regras de produção para o ritmo e para a harmonia, e, com a combinação destes, também a estrutura melódica, a gramática é capaz de gerar novas cantigas ao estilo das cantigas de roda suecas.

Alguns pesquisadores também se dedicaram à confecção automática das regras gramaticais, no paradigma conhecido como “inferência gramatical”. Um exemplo é o trabalho de Teuvo Kohonen na aplicação à composição de seu “contexto dinamicamente expansível” [68] (DEC — *Dynamically Expanding Context*). O modelo de Kohonen se assemelha a um modelo de Markov com ordem variável, configurando-se em uma espécie de “gramática estocástica” sensível ao contexto. O funcionamento do contexto dinamicamente expansível permite determinar de maneira única e regerar completamente um dado *corpus*. Por exemplo, com um contexto pequeno, dada a sequência de símbolos (e.g., notas),

$$ABCDEFG \dots IKFH \dots LEFJ \dots, \quad (3.18)$$

é impossível determinar unicamente qual o sucessor do símbolo “F”. Aumentando-se o contexto, contudo, é evidente que “H” é precedido por “KF”, enquanto ainda há ambiguidade com as regras de produção “EF \rightarrow G” e “EF \rightarrow J”. Essas regras só poderiam ser unicamente definidas em um contexto mais amplo, equivalente a uma cadeia de Markov de terceira ordem. Para geração de novas sequências, Kohonen su-

gere fixar um “parâmetro de profundidade” [68] e, então, escolher aleatoriamente as possibilidades sugeridas por cada regra nos passos de produção. Pedro Cruz-Alcázar e Enrique Vidal-Ruiz implementam, em [69], diferentes métodos de inferência gramatical para análise de *corpora* formados por cantos gregorianos, por peças sacras de Bach e por trechos de *ragtimes* de Scott Joplin. Os pesquisadores testam diferentes representações para as peças e analisam as bases com gramáticas estocásticas regulares. As composições geradas pelos modelos inferidos admitem um estilo similar ao dos *corpora* originais.

Por fim, o EMI — “*Experiments in Musical Intelligence*”, de David Cope [19], também merece citação. A aplicação de Cope permite a análise de composições musicais de um estilo específico e a formulação de uma rede de transição aumentada (ATN — *Augmented Transition Network*), um formalismo similar ao das gramáticas gerativas. Através de algoritmos de reconhecimento de padrões, o sistema extrai as características estilísticas do *corpus* em questão, recombina estas componentes em novas composições [7].

Novamente, o artigo de Fernández e Vico [11] e o livro de Nierhaus [7] trazem uma pesquisa aprofundada sobre o uso de gramáticas gerativas no campo da Composição Algorítmica. O trabalho de Roads [66] também traz uma breve descrição de alguns estudos desenvolvidos, por exemplo, pelos linguistas Nicholas Ruwet, Jean-Jacques Nattiez e Terry Winograd.

3.4 Limitações

O formalismo gramatical permite a organização hierárquica e estrutural dos dados sob análise, mas, assim como as cadeias de Markov (cf. Capítulo 2), está sujeito a algumas limitações. Algumas dessas fronteiras são apresentadas a seguir. O final deste capítulo expõe brevemente, na Seção 3.4.1, a relação entre as gramáticas e as cadeias de Markov.

Em primeiro lugar, quanto menos restritivas as regras de reescrita de uma gramática e, portanto, menor o seu tipo na hierarquia de Chomsky, maior a complexidade computacional e o empenho necessário para derivação de suas sentenças. De fato, em uma gramática do tipo 0 o problema da gramaticalidade de uma sentença pode ser, no pior caso, indecidível e o processo de análise não terminar [7].

Embora Schenker [51] e Lerdahl e Jackendoff [54, 57] destrinchem as linhas contrapontuais ou lidem com melodia e harmonia quase simultaneamente, seus trabalhos não fogem ao paradigma das gramáticas, que é, essencialmente, sequencial e opera em uma única dimensão. A maior parte dos trabalhos que buscam descrever as regras da sintaxe de uma dada “linguagem” musical se concentram, portanto, em monodias e sequências de acordes [61, 64, 65].

Quando aplicados em um contexto musical, alguns elementos do arcabouço desenvolvido por Chomsky não encontram paralelo. É o caso, por exemplo, das regras de transformação gramatical (e.g., mudança de voz ativa para voz passiva) que operam sobre as estruturas superficiais das sentenças, mantendo intocadas as estruturas profundas. O próprio conceito de estrutura profunda exige uma teoria da semântica musical, que ainda não está definida — uma transformação gramatical como a citada acima poderia alterar completamente uma composição, transformando também seu sentido. Para Schenker [51] e Lerdahl e Jackendoff [54, 57], a estrutura profunda representa um conceito abstrato da composição, um passo inicial ou intermediário do seu processo de construção. Por fim, as classes de estrutura de frases (e.g., “SN”, “V”), ainda que não tenham um equivalente “universal” na música, por vezes aparecem em diferentes formas (e.g., cadência [65], subfrase [67]).

De qualquer forma, a aplicação dos conceitos de Chomsky nos estudos musicológicos é importante porque auxilia a compreensão da competência e desempenho do “falante”. Em um ambiente tipicamente criativo como o da música, as gramáticas gerativas permitem a introspecção na produção de significado e a conexão entre “mensagem” e “código” neste meio.

3.4.1 Gramáticas e Cadeias de Markov

As cadeias de Markov, vistas em detalhe no Capítulo 2, podem ser descritas de forma equivalente às gramáticas do tipo 3 na hierarquia de Chomsky, o tipo mais estrito de gramática gerativa, se a cada regra gramatical for atribuída uma probabilidade. Isso não é uma surpresa, uma vez que aquelas podem ser representadas pelos autômatos reconhecedores dessas últimas — as máquinas de estados finitos.

Em [49] e também em [38], Chomsky discute que a simplicidade das cadeias de Markov não permite a descrição de uma gramática para a língua inglesa. Além disso, em [38], o gerativista argumenta contra a análise estatística de um *corpus* de uma linguagem — tal análise não permitiria a verdadeira compreensão da competência do falante ou, no caso em questão, do compositor. Assim como apontado na Seção 3.2, o tamanho finito do *corpus* não seria representativo do infinito poder gramatical de geração de novas sentenças. Por isso, conforme Chomsky observa, a probabilidade de ocorrência de uma sentença nada tem a ver com sua gramaticalidade, isto porque uma sentença gramaticalmente correta nunca antes proferida pode ser enunciada pelo falante, que conhece as regras gramaticais e tem competência, sem grande dificuldade. A probabilidade de a sentença ocorrer em um discurso é, nessa situação, próxima de zero, mas isso não faz dela incorreta ou não gramatical. Entenda-se: Chomsky não descarta a importância dos estudos estatísticos, mas atenua sua relevância no contexto da caracterização das sentenças gramaticais.

Capítulo 4

Algoritmos Evolucionários

A teoria da evolução do naturalista britânico Charles Robert Darwin (1809–1882) serviu de inspiração a toda uma classe de algoritmos de otimização — os algoritmos evolucionários. Os algoritmos genéticos e a programação genética fazem parte dessa gama de modelos de busca heurística. As analogias com os processos biológicos são encontradas, por exemplo, nos operadores de seleção e de mutação que atuam sobre os candidatos à solução do problema que se deseja resolver.

Neste capítulo, a motivação para esse paradigma é apresentada, desde as observações de Darwin. O modelo algorítmico é visto, então, em detalhe, e um conjunto básico de técnicas de seleção, recombinação e mutação é apresentado. Similarmente, dedica-se atenção à função de aptidão, que avalia a “capacidade” dos candidatos à solução e está diretamente relacionada à “pressão seletiva” do meio, i.e., o problema em mãos. Em seguida, a aplicação do formalismo ao campo da Composição Algorítmica é revisitada, exemplificada pelos trabalhos de alguns pesquisadores. Finalmente, são apontadas algumas das limitações dos algoritmos evolucionários.

4.1 Motivação

Em “*The Origin of Species*”¹ [70] (1859), o naturalista Charles Darwin reúne o conjunto de observações e introspecções feitos durante os cinco anos em que participou de uma expedição ao redor do mundo a bordo do *HMS Beagle*. Nesse livro, considerado um divisor de águas na história da ciência, Darwin descreve os mecanismos através dos quais uma população de seres vivos é lentamente modificada no processo conhecido como evolução, que tem como resultado a diversidade biológica observada hodiernamente.

¹Em sua edição original de 1859, o título completo do livro é “*On the Origin of Species by Means of Natural Selection, or The Preservation of Favoured Races in the Struggle for Life*”. Contudo, a partir da sexta edição, tida por muitos como definitiva, foi abreviado para, simplesmente, “*The Origin of Species*”.

Até a aceitação dos conceitos propostos por Darwin, a maior parte dos naturalistas defendia que as diferentes espécies fossem “produções imutáveis e que foram criadas separadamente” [70]. Contudo, como o próprio Darwin destaca, a ideia de que as espécies sofrem modificações e de que “as formas de vida existentes são descendentes, por geração, de formas preexistentes” [70] não surge com “*The Origin of Species*”. Aristóteles (384–322 a.C.), por exemplo, em “*Physicæ Auscultationes*”, discute de forma muito incipiente sobre a formação dos dentes, aludindo aos princípios da seleção natural [70]: “como os dentes, por exemplo, crescem por necessidade, os da frente afiados, adaptados para dividir, e os molares achatados, úteis para para mastigar os alimentos; uma vez que não foram feitos para este fim, mas são resultado de acidente. [...] Portanto, quando quer que todas as coisas reunidas (isto é, todas as partes de um todo) aconteceram como se fossem feitas para alguma finalidade, estas foram preservadas, constituídas por uma espontaneidade interna; e tudo quanto não foi assim constituído, desapareceu e ainda desaparece”.

Jean-Baptiste de Monet (1744–1829), cavaleiro de Lamarck, é o primeiro a receber atenção por uma teoria evolucionista para explicar a diversidade biológica — o lamarckismo — exposta em detalhe em “*Philosophie Zoologique*” [71], de 1809. De forma similar a Aristóteles, Lamarck apresenta uma visão teleológica em sua teoria para a evolução dos seres vivos — o caráter das mudanças observadas nas espécies tem explicação em sua finalidade. Para Lamarck, todas as espécies, incluindo o ser humano, surgiram a partir de transformações graduais em outras espécies [70, 71]. Essas transformações propostas por Lamarck são resultado de leis naturais e ocorrem pela ação direta das condições do ambiente e pelos efeitos do hábito [71]. As necessidades que surgem das transformações gradativas do ambiente e os hábitos adquiridos pelos seres vivos apontam para a diversidade biológica segundo, portanto [71]:

(Lei do uso e do desuso) Em qualquer animal que não tenha concluído seu desenvolvimento, o uso mais frequente e contínuo de um órgão qualquer fortifica este órgão pouco a pouco, desenvolve-o, aumenta-o e lhe dá um poder proporcional à duração de seu uso; enquanto a falta constante de uso de tal órgão o enfraquece imperceptivelmente, deteriora-o, diminui progressivamente suas capacidades e, por fim, faz com que desapareça.

(Lei da transmissão de caracteres adquiridos) Todas as aquisições ou perdas acarretadas nos indivíduos pela natureza através da influência das circunstâncias do ambiente em que sua raça se encontra após longo tempo e, conseqüentemente, pela influência do uso predominante de dado órgão ou pela falta constante de uso desta parte são preservadas pela reprodução para os novos indivíduos, desde que as mudanças adquiridas sejam comuns aos dois sexos, ou àqueles que produzem os novos indivíduos.

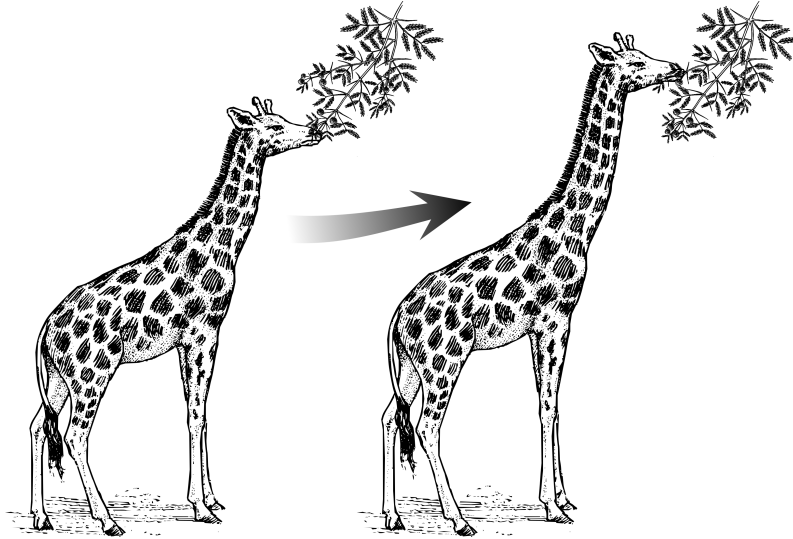


Figura 4.1: Explicação de Lamarck para as características físicas das girafas. Os caracteres são desenvolvidos e transmitidos geração após geração.

Dessa forma, Lamarck atribui as adaptações dos diferentes organismos a um fenômeno gradual, no qual: a natureza cria a necessidade; a função do órgão o modifica, segundo a lógica do uso e do desuso; e a hereditariedade faz perpetuar essas características adquiridas à geração futura. Um exemplo apresentado por Lamarck e repetido na literatura se encontra a seguir e é ilustrado na Figura 4.1.

Exemplo 4.1. “Relativamente aos hábitos, é curioso de se observar o resultado na forma particular e no tamanho da girafa (*camelopardalis*): sabe-se que este animal, o maior [em altura] dos mamíferos [terrestres], habita o interior da África e que vive em lugares onde a terra, quase sempre árida e sem pasto, obriga-o a buscar as folhagens das árvores, e se esforçar continuamente para alcançá-las. Resulta desse hábito mantido por muito tempo, em todos os indivíduos de sua raça, que suas patas dianteiras se tornem mais compridas que as traseiras, e que seu pescoço torne-se tão alongado que a girafa, sem se apoiar nas patas traseiras, eleva sua cabeça e atinge seis metros de altura” [71].

O mecanismo proposto por Lamarck não convenceu os naturalistas [72]. Contudo, não obstante a falta de evidências que comprovassem a hereditariedade de características adquiridas pelo uso e desuso, as ideias de Lamarck ressurgiram, no final do século XIX, apoiadas por aqueles que tentavam desacreditar o darwinismo [72]. Em [70], Darwin faz uso e admite certa importância das leis de Lamarck, mas propõe uma alternativa diante dos obstáculos enfrentados pelo mecanismo lamarkista — a seleção natural [72].

A hipótese darwinista para as mudanças evolutivas se apoia nas ideias de seu conterrâneo, o economista Thomas Robert Malthus (1766–1834). Em “*An Essay on the Principle of Population*” [73], Malthus argumenta que “o poder da população [de

crescer] é infinitamente maior que o poder na terra de produzir meios de subsistência para o homem. População, quando não é controlada, cresce de acordo com uma progressão geométrica. Os meios de subsistência crescem, apenas, de acordo com uma progressão aritmética. Uma pequena familiaridade com números fará notar a imensidade do primeiro poder se comparado ao segundo”. Darwin se apropria dessa noção de desbalanço entre uma população e o meio de subsistência, que inspira a competição por alimento, e a aplica nos reinos animal e vegetal [70].

Assim opera o mecanismo de seleção natural: o grande potencial de crescimento das diferentes espécies na natureza faz com que suas populações superem a quantidade de recursos disponíveis para a sobrevivência de seus indivíduos, o que acarreta a competição pelos recursos do ambiente, e.g., espaço e alimento; segue-se que, uma vez que as espécies apresentam variações, um indivíduo que varie levemente de maneira proveitosa, sob as “condições complexas e por vezes variáveis da vida” [70], tem mais chance de sobreviver, sendo, portanto, selecionado naturalmente. Considerando o princípio da hereditariedade, Darwin aponta que uma “variedade selecionada tenderá a propagar sua forma nova e modificada” [70]. O naturalista faz um paralelo desse mecanismo com o da seleção artificial exercida pelo homem sobre espécies de plantas e animais domésticos, que se dá de forma gradativa [70] — assim também a seleção natural favorece, ao longo das gerações, a manutenção e também a melhoria de características vantajosas à adaptação. Além disso, Darwin indica que a seleção natural, invariavelmente, leva à extinção formas de vida menos “aperfeiçoadas” [70] e causa o que se conhece como divergência de caracteres, i.e., a forma como ocorre a formação de novas espécies a partir de uma espécie ancestral. À luz de sua hipótese da seleção natural, Darwin sugere, em [70], uma explicação para comprimento do pescoço da girafa, de modo diverso ao proposto por Lamarck e reproduzido no Exemplo 4.1.

Exemplo 4.2. “A girafa, com sua estatura elevada, pescoço alongado, patas dianteiras, cabeça e língua, tem toda sua estrutura bem adaptada para buscar [alimento] nos galhos mais altos das árvores. [...] Assim, para a girafa em estado natural, os indivíduos que tinham maior estatura e eram capazes, durante os tempos de escassez, de alcançar uma polegada ou duas acima dos demais muitas vezes foram preservados; [...] esses indivíduos que tinham uma ou mais partes de seus corpos um pouco mais alongadas que o normal, geralmente, teriam sobrevivido. Seu cruzamento deixou uma descendência que, ou herdou as mesmas particularidades corporais, ou apresentava a tendência a variar da mesma maneira, enquanto os indivíduos menos favorecidos nesse aspecto, foram mais suscetíveis a perecer” [70].

As evidências observadas em diferentes áreas da biologia (e.g., a adaptação dos seres vivos a um ambiente, os fósseis e as semelhanças anatômicas entre diferentes

espécies) dão grande suporte à teoria da evolução de Darwin, que tem embasado, constantemente, as discussões e especulações sobre a origem e as modificações da vida na Terra. Atualmente, fala-se em uma teoria sintética da evolução, que alia os conceitos de Darwin aos avanços na área da genética. A próxima seção apresenta um panorama geral da teoria evolucionista moderna, com ênfase nos aspectos da genética, que não foram tratados até este ponto.

4.1.1 Um pouco de biologia

Em um nível “microscópico” [74], a teoria da evolução é complementada pelos estudos da genética. Essa área da biologia permite a compreensão, em um nível molecular, dos mecanismos de reprodução, de hereditariedade e de variação das estruturas dos seres vivos.

É essencial na genética a observação de que cada indivíduo manifesta duas dimensões em estreita relação: o fenótipo e o genótipo [74]. O fenótipo² designa as características externas e observáveis desse indivíduo, e.g., aspectos morfológicos e de desenvolvimento, enquanto o genótipo³, por sua vez, refere-se à sua composição genética interna, que é responsável, em última instância, por determinar a expressão de suas características fenotípicas [74].

Nos organismos vivos, a informação completa contida no genótipo — o genoma — está inscrita na molécula de ácido desoxirribonucleico (DNA — *Deoxyribonucleic Acid*) e os segmentos do DNA — os genes — são as suas unidades de codificação [74]. A estrutura de nucleotídeos que se encontra na dupla hélice do DNA contém toda a informação necessária ao “plano de construção” [74] do ser vivo, de acordo com a sequência de bases nitrogenadas ali disposta. O processo de determinação do fenótipo a partir do código genético se dá, então, através dos mecanismos de transcrição, no qual a informação do DNA é “copiada” em uma molécula de ácido ribonucleico (RNA — *Ribonucleic Acid*), e de tradução, no qual a cadeia simples do RNA é “lida” para síntese de proteínas [74], que são as responsáveis pela construção das características fenotípicas⁴. De maneira simplificada, portanto, cada gene do DNA está relacionado a um conjunto de códons, e representa uma unidade funcional que codifica um traço fenotípico⁵. Por isso pode-se dizer que “o genótipo codifica o fenótipo” [74].

²Do grego, φαίνω (*phaínō*), “aparente”/“brilhante”, e τύπος (*typos*), “marca”/“característica”.

³Do grego, γένος/γενεα (*genos/genea*), “origem” ou “raça”, e τύπος (*typos*).

⁴Cada tripla de bases nitrogenadas determina um “códon” [74], que codifica um aminoácido específico — uma unidade de composição das proteínas. Embora existam apenas quatro tipos de bases, que permitiriam $4^3 = 64$ combinações diferentes, há apenas 20 aminoácidos codificados [74] e mais um de um códon pode indicar um mesmo tipo de aminoácido.

⁵Em verdade, essa codificação não é “um para um”, pois mais de um gene pode estar associado a um dado traço fenotípico e, ao mesmo tempo, mais de uma característica do indivíduo pode ser determinada pelo mesmo gene [74].

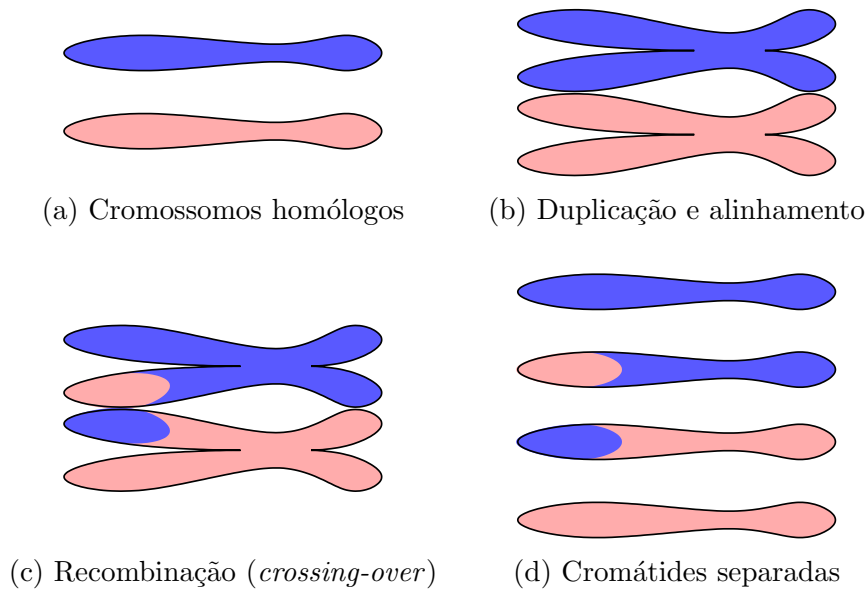


Figura 4.2: Cromossomos homólogos nas fases da meiose.

Nas células, o material genético do indivíduo se encontra, geralmente, organizado em formas condensadas — os cromossomos [74]. Em muitos animais e plantas, a maior parte das células carrega os cromossomos aos pares, os chamados cromossomos homólogos, que apresentam os genes para os mesmos traços característicos, e.g., os 46 cromossomos nos seres humanos podem ser agrupados em 23 pares homólogos [74]. Essas células são chamadas diploides, assim como as suas espécies [74], e, para cada par de cromossomos homólogos, um deles tem origem paterna, enquanto o outro tem origem materna. As células responsáveis pela reprodução sexuada — os gametas — são haploides, i.e., carregam apenas um cromossomo de cada tipo [74]. Durante a fecundação, os gametas paterno e materno se fundem em um zigoto, que dá origem ao organismo diploide combinando, portanto, o material genético dos pais [74]. As versões materna e paterna de um gene nos cromossomos homólogos — os alelos [75] — não são necessariamente idênticas. Um alelo pode ser “dominante” sobre o outro (conhecido como “recessivo”), sendo então responsável por determinar as características que se tornarão aparentes [75].

Os gametas são formados por um processo de divisão celular particular, conhecido por “meiose” [74], uma vez que divide o número de cromossomos por dois. No início do processo de meiose, cada cromossomo é duplicado e suas fitas — chamadas cromátides [74] — permanecem unidas em uma posição específica — o centrômero [74]. Os pares de cromossomos homólogos então se alinham, podendo ocorrer a formação de um quiasma, i.e., um ponto de cruzamento das cromátides dos cromossomos homólogos, e subsequente recombinação (“*crossing-over*” [75]), i.e., a troca dos segmentos unidos no quiasma [74]. A Figura 4.2 ilustra esse processo de recombinação durante a formação dos gametas. É interessante observar que, ao final

do processo de meiose, são produzidas quatro células haploides [75] e, para um dado cromossomo específico, duas carregam as informações recebidas dos progenitores; e duas apresentam estruturas recombinadas [74].

Na teoria sintética da evolução, são três os principais fatores que determinam a variação e adaptação nos seres vivos — a mutação e a recombinação gênicas, e a seleção natural. Conforme visto na Seção 4.1, a seleção natural atua sobre as populações determinando “sucesso ou fracasso” das variações nas espécies. Essas variações, no entanto, resultam diretamente dos processos de mutação e recombinação [75].

As mutações gênicas são mudanças nas sequências de nucleotídeos do DNA [75]. Elas alteram o material genético de uma célula e por isso, dependendo da localização desta alteração, a proteína codificada será diferente, o que poderia determinar um fenótipo diferente para o indivíduo [75]. Esse tipo de mutação ocorre normalmente em taxas baixas e pode ser benéfico ou maléfico [75]. São as mutações, contudo, essenciais ao processo de evolução, porque representam a única maneira de surgirem novas versões de um gene nas populações [75].

A recombinação gênica, por sua vez, altera a organização dos genes nos cromossomos [75]. No processo de formação dos gametas, os cromossomos são segregados independentemente [75], e.g., nos seres humanos, desconsiderando-se o processo de recombinação, há 2^{23} tipos de gametas diferentes que um indivíduo pode produzir a partir das combinações de seus cromossomos. Com a recombinação, portanto, a diversidade genética aumenta ainda mais.

4.2 Algoritmos Evolucionários

Darwin apresenta uma “visão macroscópica da evolução” [74], que é fundamentada por sua hipótese da seleção natural. Em um ambiente com capacidade limitada, a competição por recursos surge naturalmente e vai de encontro ao instinto de reprodução dos indivíduos. Assim, a seleção age para limitar o potencial exponencial de crescimento da população [74]. Darwin também faz referência a esse fenômeno como sendo a “sobrevivência do mais apto” [70, 74], uma vez que são os indivíduos mais adaptados às condições do ambiente que terão mais chance de sobreviver e perpetuar suas características para as gerações seguintes.

As adaptações são, precisamente, as variações observadas nos indivíduos de uma mesma espécie, i.e., as diferenças nas características fenotípicas que compõem os indivíduos e que, em última instância, são determinadas por seus genes. Quanto mais bem “avaliadas pelo meio” [74] essas características forem, maiores as chances de serem propagadas para os descendentes daquela população. À medida que os indivíduos se reproduzem, novas variantes dos traços fenotípicos podem surgir. Essas

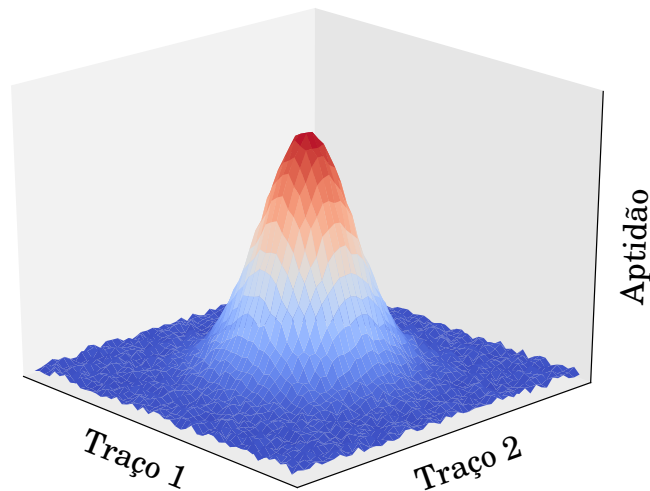


Figura 4.3: Superfície de aptidão.

se devem, em última instância, às mutações. Os novos indivíduos de uma população também estarão sujeitos à avaliação do meio e, novamente, indivíduos que apresentarem um fenótipo desfavorável estarão mais vulneráveis e passíveis de eliminação. Nesse processo contínuo e gradativo de seleção, reprodução e mutação, os indivíduos representam as unidades sobre as quais a seleção opera [74]. A população é, por sua vez, a “unidade de evolução” [74]. Isso significa que o que se entende por “evolução” é a mudança nas características de uma população, e não de um indivíduo.

É possível compreender o processo da evolução através da metáfora da “paisagem adaptativa” ou “superfície adaptativa” [74], como representa a Figura 4.3. Nessas superfícies, uma dimensão representa a aptidão enquanto as outras correspondem aos diferentes traços fenotípicos dos indivíduos de uma população [74] — características similares sendo dispostas proximamente em cada eixo. Observar uma população com suas diferentes características seria equivalente a estudar a distribuição de pontos nessa superfície — cada ponto correspondendo a um indivíduo. Conforme indivíduos “bem avaliados” pelo meio propagam sua herança genética, a distribuição da população pode “subir um monte”, por exemplo. As variações genéticas e a seleção natural vão, portanto, alterar a distribuição da população sobre essa superfície [74].

A área da computação evolucionária engloba o conjunto de técnicas inspiradas nas teorias evolutivas [76]. Os algoritmos evolucionários imitam a capacidade da natureza de, através do processo de evolução, gerar diferentes espécies adaptadas às mais variadas condições. Em última análise, a abordagem dessa classe de algoritmos para a resolução de problemas é a de “tentativa e erro”, ou ainda, “geração-e-teste” [74]. Como os indivíduos em uma população na natureza competem pela sobrevivência e têm vantagens ou desvantagens em suas características fenotípicas, assim também, em um algoritmo evolucionário, os diferentes candidatos à solução de um problema são avaliados de acordo com sua qualidade, i.e., quão bem “aten-

dem” ao problema em questão, o que determina suas chances de serem mantidos e servirem de base para novos candidatos [74]. A expectativa é que, com o passar das “gerações”, a população seja composta por indivíduos com altos valores de aptidão. O paralelo com a área da otimização⁶ torna-se, então, evidente e o objetivo dos algoritmos evolucionários, enquanto métodos de busca, pode ser entendido como sendo encontrar o indivíduo que maximiza o valor de uma função — a superfície de aptidão. Esse indivíduo é o ótimo global dessa superfície, i.e., corresponde ao valor de maior aptidão [74].

Um dos primeiros pesquisadores a sugerir a aplicação dos conceitos da evolução na área da computação teria sido o matemático britânico Alan Turing [74, 76]. Em 1948, Turing sugeriu o uso de um procedimento de evolução artificial para o desenvolvimento de máquinas adaptativas que, codificadas por genes artificiais, estariam sujeitas aos processos de mutação e seleção [76]. Desde então, numerosos foram os trabalhos a utilizar os conceitos evolutivos em diferentes implementações⁷, reconhecidas hoje como “dialetos” dos algoritmos evolucionários [74]; são eles:

- algoritmo genético (GA — *Genetic Algorithm*);
- estratégias de evolução (ES — *Evolution Strategies*);
- programação evolutiva (EP — *Evolutionary Programming*);
- programação genética (GP — *Genetic Programming*).

Grosso modo, essas variantes da computação evolucionária diferem nos usos dos operadores de seleção, recombinação e mutação, mas principalmente quanto ao tipo dos indivíduos que representam os candidatos à solução. Cada “dialeto” sugere uma forma de representação que pode se adequar melhor à solução de um problema específico. Assim, os GAs operam sobre cadeias de um alfabeto específico, e.g., *strings* binárias, cadeias de números inteiros ou reais [74]. Nas ESs, cada indivíduo é representado por um vetor de números reais que, além de apresentar suas características para solução do problema, também possui um conjunto de valores para “autoadaptação”, i.e., cada indivíduo carrega a informação de como o operador de mutação atua sobre ele (sobre cada um dos elementos do vetor), que também está sujeita a variação [74]. A EP clássica descreve a evolução de uma população de máquinas de estado finito, sobre as quais a única fonte de variação é o operador de mutação [74]. Por fim, os algoritmos de GP têm como unidades de seleção programas de computador, descritos na forma de árvores [74].

⁶Há que se notar, contudo, que o paralelo traçado entre o processo biológico da evolução das espécies e o conceito de otimização não é perfeito, como será discutido brevemente ao final desta seção (cf. Seção 4.2.9).

⁷Para uma lista abrangente de tais trabalhos, ver [74].

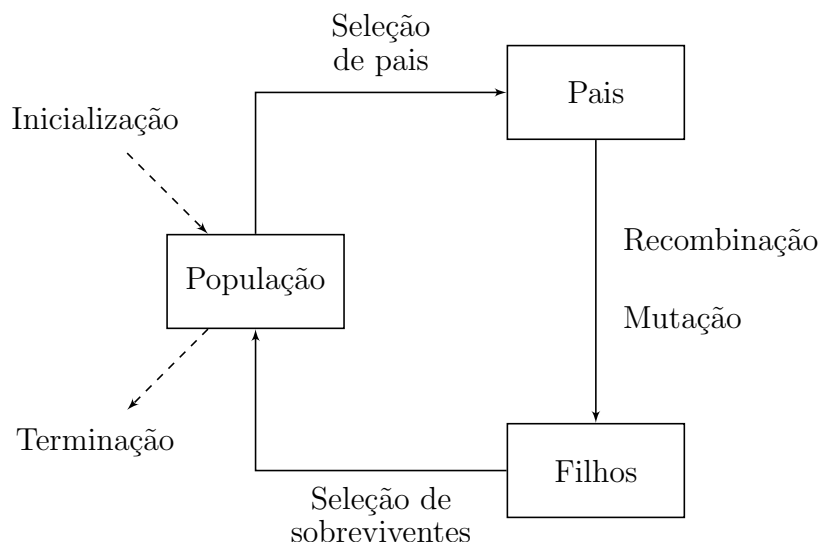


Figura 4.4: Funcionamento de um algoritmo evolucionário. Adaptado de [74].

A despeito das diferenças específicas de cada método, todos os algoritmos evolucionários possuem um mesmo mecanismo [74], que se descreve a seguir. O problema que se deseja resolver determina o “meio”, descrito através de uma “função de aptidão” [74]. Essa função corresponde à pressão seletiva que intenciona o aumento da aptidão média da população através da sobrevivência do mais apto [74]. Cada indivíduo da população é um candidato à solução do problema. Um conjunto de indivíduos mais aptos é selecionado para reprodução — são os “pais”, as “sementes” para a próxima geração [74]. Sobre esse conjunto de pais, são aplicados os operadores de recombinação, que atuam sobre dois ou mais indivíduos pais gerando um ou mais indivíduos “filhos” [74], e/ou de mutação, que geram um novo candidato a partir de um único candidato [74]. Os operadores de recombinação e mutação são responsáveis por aumentar a variedade da população, e agem em sentido diferente dos operadores de seleção. Os indivíduos filhos, então, competem com os pais pela sobrevivência e composição da nova geração, com base em sua aptidão e, eventualmente, idade [74]. O processo de evolução da população é repetido até que se atinja um critério de terminação previamente estabelecido, e.g., número máximo de gerações, pouca variação na qualidade média da população entre uma geração e outra etc. O ciclo básico das gerações de um algoritmo evolucionário pode ser observado na Figura 4.4.

Nas diferentes etapas de uma geração, os algoritmos evolucionários se apoiam em probabilidades, e.g., na seleção de pais, na escolha do ponto de aplicação do operador de mutação ou, até mesmo, na ocorrência ou não de mutação. Esses algoritmos não são, contudo, métodos de busca completamente aleatória, uma vez que os mecanismos de seleção “direcionam” o movimento da população sobre a superfície adaptativa. Os algoritmos evolucionários permitem buscar solução para problemas

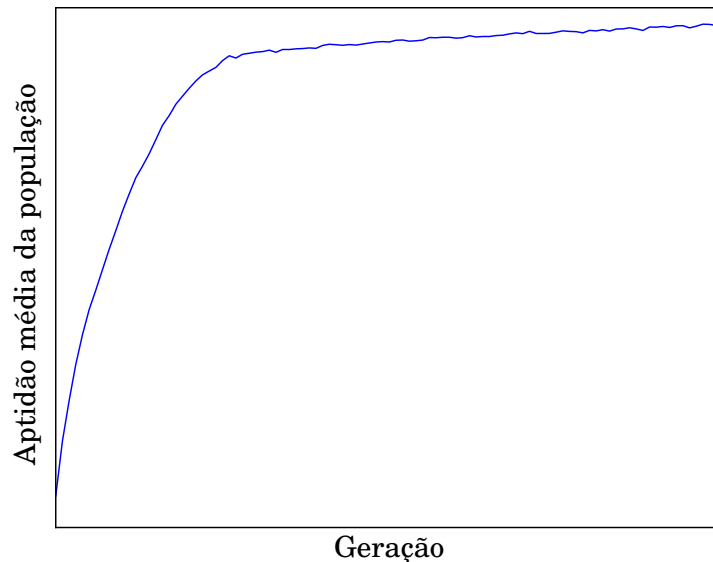


Figura 4.5: Curva de desempenho típica de um algoritmo evolucionário com base na aptidão média da população.

sem requerer informação específica do meio, i.e., sem conhecimento da superfície do problema. Eles fornecem boas e aceitáveis soluções para um amplo conjunto de problemas, sendo superiores aos métodos de busca aleatória, mas inferiores aos algoritmos especialistas nos problemas para os quais foram desenvolvidos⁸. A Figura 4.5 apresenta a curva de desempenho típica em função das gerações de um algoritmo evolucionário.

Em concordância com a Figura 4.4, a especificação completa de um algoritmo evolucionário requer a definição dos seguintes elementos:

- representação dos indivíduos;
- função de aptidão;
- população;
- método de inicialização da população;
- mecanismo de seleção de pais;
- operadores de variação (recombinação e mutação);
- mecanismo de seleção de sobreviventes;
- critério de parada.

⁸Em [74], discute-se sobre desempenho e aplicabilidade dos algoritmos evolucionários e o teorema da “inexistência do almoço grátis” (NFL — *No Free Lunch*) [77]. Segundo esse teorema, se satisfeitas algumas condições, todos os métodos de otimização são equivalentes em desempenho, tomada sua média sobre o espaço de todos os problemas existentes [74].

Cada um desses itens será analisado em detalhe nas próximas seções. O método de inicialização e o critério de parada não apresentam variações específicas entre os diferentes dialetos dos algoritmos evolucionários [74].

4.2.1 Representação

Um aspecto fundamental dos algoritmos evolucionários é a representação dos indivíduos candidatos à solução do problema. No contexto dessa classe de algoritmos, “representação” carrega um duplo significado. O primeiro está relacionado ao mapeamento do “espaço do fenótipo” (o contexto original do problema) para o “espaço do genótipo” (o escopo interno do algoritmo, o domínio dos operadores de variação — mutação e recombinação) [74]. Por exemplo, dado um problema com números inteiros, o conjunto de elementos do fenótipo equivale ao conjunto de números inteiros fornecido pelo problema [74] e o mapeamento escolhido poderia ser a representação de cada número inteiro como uma cadeia binária (e.g., o número “12” é um elemento do fenótipo e seu genótipo correspondente seria “1100”). Observe-se que os espaços de fenótipo e de genótipo podem ser diferentes, de acordo com o problema, e requerem um processo de codificação/decodificação [74]: para operar o algoritmo, cada fenótipo deve ser codificado para o espaço do genótipo; ao término do algoritmo, é necessário decodificar o genótipo de melhor aptidão, por exemplo, e retornar ao domínio do problema original antes de interpretar seu significado.

O outro significado para “representação” é, precisamente, a estrutura de dados genotípica, que separa as diferentes classes de algoritmos evolucionários, conforme apontado na seção anterior. A representação determina diretamente os tipos de operadores de recombinação e de mutação que serão utilizados, e.g., no caso de cadeias binárias é comum o uso do operador de mutação conhecido como “*bit flipping*”, enquanto em uma aplicação musical, a mutação pode ser a transposição de um trecho inteiro do cromossomo.

Embora o termo “indivíduo” seja utilizado indiscriminadamente nos espaços de genótipo ou do problema, alguns termos são aplicados ou em um ou no outro ambiente. Assim, no espaço do algoritmo, os indivíduos podem ser chamados de “cromossomos” ou, simplesmente “genótipos” [74]. Cada posição em um genótipo é um “gene” e o elemento que o ocupa (geralmente, um valor numérico ou um objeto) é um “alelo” [74]. No espaço de fenótipos, os indivíduos também são conhecidos como “candidatos à solução” ou “fenótipos” [74].

É importante que a representação dos indivíduos seja adequada ao problema. Uma representação desnecessariamente complexa pode inutilizar o algoritmo evolucionário [76]. No caso dos algoritmos genéticos, por exemplo, a representação dos genótipos é, usualmente, uma cadeia de um vocabulário bem definido, e.g., dígitos

binários, números inteiros ou números em ponto flutuante [74]. Os cromossomos podem representar também elementos de permutação, e.g., no clássico problema do caixeiro viajante, uma das possíveis representações adequadas é a ordem em que as cidades são visitadas [74].

4.2.2 Função de Aptidão

A função de aptidão descreve o meio ao qual os indivíduos estão sujeitos. Em última instância, ela é que determina o que significa uma adaptação [74] nesta classe de algoritmos de “geração-e-teste” [76]. No espaço de fenótipos, ela é o problema que se deseja resolver e deve atribuir, a cada candidato à solução, um valor correspondente à sua qualidade [74]. Com o auxílio da função de aptidão atuam os operadores de seleção, e.g., selecionando para ser pai um indivíduo com boa aptidão ou eliminando um genótipo mal avaliado.

No contexto de otimização, a função de aptidão é a função objetivo a ser maximizada ou minimizada. No exemplo anterior de otimização sobre um conjunto de números reais, se a função objetivo que se deseja maximizar é $f(x) = 10x - x^2$, a aptidão do genótipo “100” corresponde à aptidão de seu fenótipo equivalente, “4”, $f(4) = 40 - 16 = 24$. O indivíduo com maior aptidão, nesse caso, seria o fenótipo “5”, que maximiza a função.

Funções de aptidão podem se apoiar em restrições ou sistemas de penalização, por exemplo, e podem se basear em funções reais, em sistemas de regras, em preferências humanas nos chamados “métodos iterativos” (“evolução interativa” ou “evolução colaborativa” [74]) e até mesmo serem “multiobjetivo” [74]. Essa classe de funções de aptidão, de particular interesse no presente trabalho, é detalhada a seguir.

Funções de Aptidão Multiobjetivo

Os problemas de otimização que utilizam algoritmos evolucionários geralmente se concentram em buscar uma solução ótima, i.e., que apresente o melhor valor de qualidade (objetivo) para o problema específico. Eventualmente, torna-se interessante encontrar não uma, mas um conjunto de soluções com boa aptidão em diferentes “nichos” [74], i.e., regiões da superfície adaptativa (cf. Seção 4.2.8). Ainda assim, a medida de aptidão ainda é um valor único [74]. Em outros casos, a qualidade de um indivíduo pode ser medida por vários objetivos, possivelmente conflitantes, e encontrar uma solução equivale a otimizar seu valor nessas diferentes dimensões [74]. Nesses problemas, fala-se em otimização multiobjetivo, ou ainda, em algoritmo evolucionário multiobjetivo (MOEA — *Multiobjective Evolutionary Algorithm*).

Os problemas com um único objetivo são vistos como um caso particular dos problemas multiobjetivo e, em verdade, muitos problemas multiobjetivo são simpli-

ficados, por exemplo, através da soma ponderada das várias dimensões em uma única função aptidão [74]. A dificuldade desse tipo de estratégia é a determinação dos pesos ideais, que podem não ser conhecidos *a priori* e não são necessariamente fixos entre as diversas execuções do algoritmo [74]. Uma abordagem multiobjetivo pode fornecer diferentes soluções, cada uma melhor avaliada em um objetivo específico, de acordo com o compromisso implícito entre as diferentes dimensões [74].

A literatura apresenta alguns algoritmos específicos para tratamento de problemas multiobjetivos (e.g., MOGA, NSGA etc. [74]). Na maior parte dessas abordagens, cada indivíduo é avaliado segundo o conceito de dominância sobre os demais [74]. Diz-se que um indivíduo domina sobre o outro caso seus valores de aptidão sejam pelo menos igualmente altos nos diferentes objetivos e, em ao menos um objetivo, sua aptidão seja estritamente superior. Equivalentemente, se os valores de aptidão de um indivíduo A nas diferentes funções $f_1(x), \dots, f_n(x)$ compõem um vetor $\mathbf{a} = [a_1, \dots, a_n]$, em que $a_i = f_i(x)$, com $i = 1, \dots, n$, então:

$$A \succeq B \iff \begin{cases} \forall i \in \{1, \dots, n\}, a_i \geq b_i \\ \exists i \in \{1, \dots, n\} \text{ tal que } a_i > b_i \end{cases}, \quad (4.1)$$

onde $A \succeq B$ significa “ A domina sobre B ” [74]. Quando há conflito entre os diferentes objetivos, não existe uma única solução que domine sobre todas as demais e esses algoritmos permitem aproximar o chamado “conjunto de Pareto”, que contém somente soluções não dominadas por nenhuma outra [74]. Esse conjunto ótimo auxilia o estudo do compromisso entre os diferentes objetivos, uma vez que, para cada solução proposta, aumentar um valor de aptidão implica diminuir os outros [74]. A Figura 4.6 apresenta um exemplo de conjunto de Pareto, no qual são destacados os elementos da fronteira ótima, bem como alguns dos elementos dominados. Nesse exemplo, há pontos da fronteira com maior densidade de soluções (e.g., a vizinhança do ponto A) e regiões de menor concentração (e.g., a vizinhança do ponto C).

4.2.3 População

A população é a unidade da evolução, o multiconjunto dos genótipos que representam os candidatos à solução [74]. Geralmente, esse conjunto tem tamanho constante, i.e., a população não cresce ou diminui com o passar do tempo. Isso acontece porque, enquanto os operadores de recombinação e de mutação agem sobre os indivíduos, os operadores de seleção atuam sobre a população [74]. Contudo, nos algoritmos evolucionários, todos os indivíduos de uma população podem ser substituídos entre uma geração e outra, i.e., a cada ciclo de seleção de pais, variação e seleção de sobreviventes. O número de diferentes soluções que uma população apresenta é a sua diversidade [74], uma medida que pode ser tomada equivalente ao número de valores

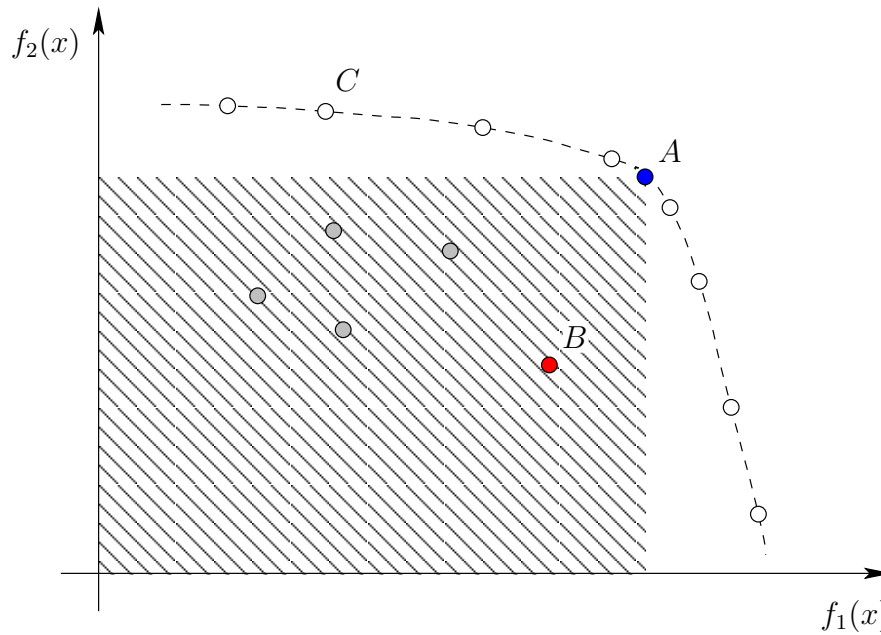


Figura 4.6: Exemplo de conjunto de Pareto. Os elementos não dominados formam a chamada “fronteira de Pareto” [74]. A área hachurada indica o conjunto de soluções dominadas por A , como por exemplo o elemento B .

de aptidão diferentes, ou à quantidade de genótipos ou fenótipos distintos [74].

Em algumas situações, os algoritmos apresentam estruturas mais complexas para as populações, e.g., estruturas de vizinhança [74], que influenciam diretamente no modo de atuação dos operadores.

Inicialização da população

O processo de inicialização de uma população em um algoritmo genético é feito o mais simples possível; e a população inicial, composta por μ indivíduos, usualmente é escolhida de maneira aleatória sobre o suporte da função de aptidão [74] — essa escolha favorece a busca por “morros” da superfície adaptativa.

Algumas heurísticas podem ser utilizadas para que a população inicial apresente maior aptidão média que uma população gerada aleatoriamente; contudo, na maior parte das aplicações, esse esforço computacional não é necessário, porque a curva característica de desempenho de um algoritmo evolucionário possui tipicamente uma derivada alta no decorrer das gerações iniciais [74] (cf. Figura 4.5). Dessa forma, a aptidão média ou máxima em uma inicialização “esperta” é geralmente alcançada após poucas gerações do algoritmo evolucionário inicializado aleatoriamente [74].

4.2.4 Seleção de pais

Em computação evolucionária, o objetivo da seleção de pais é, geralmente, com base na função aptidão, selecionar, com maior probabilidade, os melhores genótipos

da população para gerar novos indivíduos [74]. Um indivíduo é, então, chamado de “pai” caso passe por um ou mais processos de variação [74]. Junto com o mecanismo de seleção de sobreviventes, a seleção de pais direciona o algoritmo para melhoria da qualidade da população, mas permite que mesmo os indivíduos com baixa aptidão tenham chance (pequena) de gerar novos indivíduos [74]. Isso auxilia o algoritmo a não “se perder” em um ótimo local da função de aptidão [74].

Um método simples e amplamente aplicado de seleção de pais é a “seleção proporcional à aptidão” (FPS — *Fitness Proportional Selection*) [74]. Como o próprio nome indica, a probabilidade de cada indivíduo ser selecionado como pai é escolhida diretamente proporcional à sua aptidão, i.e.,

$$p_{\text{FPS}}(x) = \frac{f(x)}{\sum_{i=1}^{\mu} f(x_i)}. \quad (4.2)$$

Essa distribuição é usualmente implementada pelo método da roleta, no qual cada indivíduo recebe uma seção da roleta e a área deste setor circular é proporcional à sua aptidão [74]. A seleção de um pai é executada “rodando-se a roleta” e, evidentemente, os indivíduos com maior área serão selecionados com maior probabilidade. Na literatura, destaca-se que tal tipo de seleção apresenta algumas deficiências, a saber [74]:

- Esse tipo de pressão seletiva pode se tornar muito grande e favorecer indivíduos com boa aptidão, que dominam a população rapidamente (“convergência prematura”);
- Quando os indivíduos apresentam valores de aptidão similares, não há quase pressão seletiva e a seleção é praticamente uniforme, o que retarda a convergência do algoritmo;
- O funcionamento do algoritmo é fortemente alterado se uma transposição for aplicada à função de aptidão.

Essas duas últimas dificuldades podem ser minimizadas se à função aptidão $f(x)$ for subtraído um valor β , que representa, tipicamente, o menor valor de qualidade de um indivíduo naquela geração, i.e., $\beta = \min_{x_i} f(x_i)$, ou uma média desses últimos valores mínimos nas diferentes gerações — por isso o método é denominado “janelamento” [74]. Se o aspecto da superfície de aptidão é conhecido, sua média e variância também podem ser utilizadas para atacar esses problemas.

Outro método que visa a atender às dificuldades da seleção proporcional à aptidão é o “método do *ranking*” [74], no qual cada indivíduo é classificado de acordo com sua qualidade e recebe um “posto” (em inglês, “*rank*” [74]). A relação entre posto e seleção é monotonicamente decrescente e pode se dar de maneira linear ou

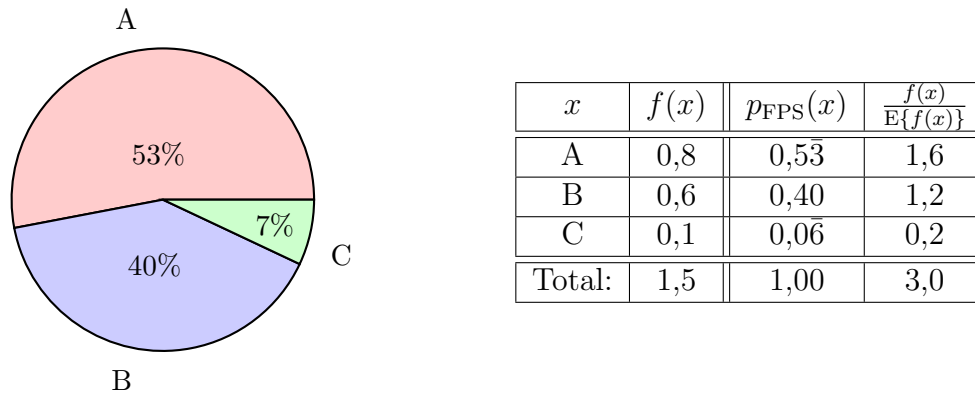


Figura 4.7: Exemplo de seleção de pais pelo método da roleta.

exponencial, desde que a soma das probabilidades de seleção de todos os indivíduos seja normalizada para 1 [74]. O método de seleção por torneio é uma alternativa aos métodos de seleção proporcional e de *ranking*, e é particularmente necessário em aplicações nas quais calcular a aptidão de todos os indivíduos da população é muito custoso computacionalmente ou impraticável [74]. A seleção por torneio também pode ser utilizada nos casos em que não há uma medida objetiva de aptidão [74]. Assim como a seleção por *ranking*, a seleção por torneio utiliza os valores de qualidade não de forma absoluta, mas relativa [74]. Enquanto for necessário selecionar pais, o algoritmo sorteia aleatoriamente κ indivíduos, compara seus valores de qualidade⁹ e retorna para ser pai o melhor indivíduo dentre os escolhidos [74]. A escolha dos κ indivíduos pode ser feita sem ou com reposição e, neste último caso, até mesmo o pior indivíduo de uma população tem chance de ser selecionado [74]. O valor de κ permite controlar a pressão seletiva [74].

É importante salientar que a seleção de pais não depende da representação escolhida para o problema, mas, em geral, somente da aptidão dos indivíduos. Por vezes, o valor de aptidão não é utilizado e os algoritmos utilizam métodos de seleção uniforme ou selecionam todos os μ indivíduos da geração atual para gerar $\lambda = \mu$ filhos (através de mutação somente) [74].

A Figura 4.7 exemplifica o processo de seleção através do método da roleta. Para um dado indivíduo x , o valor de $\frac{f(x)}{\mathbb{E}\{f(x)\}} = \mu p_{\text{FPS}}(x)$ é um estimador do número de vezes em que x será selecionado como pai, supondo que μ pais sejam selecionados.

4.2.5 Operadores de Variação

Pode-se separar os operadores de variação nos algoritmos evolucionários em dois tipos diferentes, de acordo com sua aridade, i.e., o número de operandos que utilizam como entrada. Em geral, esses operadores são identificados respectivamente

⁹Essa etapa pode ser realizada subjetivamente, nos métodos iterativos, por exemplo.

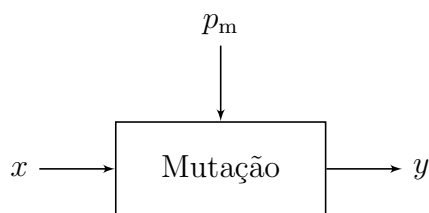


Figura 4.8: Operador de mutação básico.

como operador de recombinação e operador de mutação. O papel desses operadores é criar novos indivíduos a partir dos antigos, o que equivale, no espaço de fenótipo, a novos candidatos à solução (eles respondem pela parte da “geração”, na definição dos algoritmos evolucionários como algoritmos de “geração-e-teste” [74]). Destaca-se a diferença do termo “recombinação” em suas aplicações algorítmica e biológica. Enquanto nos algoritmos evolucionários a recombinação ocorre em um contexto reprodutivo (i.e., entre os pais) no qual o fruto da recombinação é o próprio indivíduo, na biologia, a recombinação está atrelada ao processo de formação dos gametas de cada um dos pais separadamente, conforme visto na Seção 4.1.1.

Dada a natureza estocástica dos algoritmos evolucionários, atribui-se aos operadores de variação uma ou mais probabilidades que indicam, por exemplo, a taxa com que é utilizado na geração dos novos indivíduos (no caso da recombinação) ou a escolha de um ponto para sua aplicação. Conforme notado na Seção 4.2.1, os operadores de variação são muito dependentes da representação escolhida para os genótipos [74].

Operadores de Mutaçãõ

Um exemplo clássico de operador unário é o operador de mutação, aplicado sobre um genótipo para gerar um novo indivíduo ligeiramente diferente — um mutante [74]. O operador de mutação é, tipicamente, a única forma de serem introduzidos novos valores para os genes em um algoritmo evolucionário¹⁰. Esse operador causa sempre uma “alteração aleatória, não polarizada” [74] e, em geral, a probabilidade de mutação em cada indivíduo é mantida baixa. O operador de mutação é importante porque garante que o espaço do problema está “conectado” [74], i.e., a cada geração, todos os indivíduos podem ser alcançados, com pequena probabilidade. O modelo básico de um operador de mutação pode ser visto na Figura 4.8.

Dos diferentes tipos de operadores de mutação, alguns são observados mais comumente na literatura. Em representações de cadeias de números binários ou inteiros, um tipo de mutação típico é o “*random resetting*”, na qual um outro valor do conjunto de valores permitidos para cada gene é escolhido aleatoriamente com probabilidade p_m , e que, no caso binário, recebe o nome de “*bit-flipping*” [74]. Na

¹⁰Excetuando-se, evidentemente, o *crossover* aritmético, detalhado mais adiante.

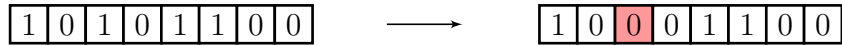


Figura 4.9: Exemplo de mutação (*bit-flipping*) sobre uma cadeia binária.

mutação *creep*, o valor de cada gene é substituído por um valor próximo, sempre dentro do conjunto de possibilidades e com probabilidade p_m [74]. Se a representação utiliza números reais, pode-se, por exemplo, realizar uma escolha uniforme sobre o domínio de cada gene (caso os domínios sejam limitados) ou empregar uma mutação adicionando-se o valor de uma variável aleatória (tipicamente) gaussiana com média zero e variância pequena [74]. Nesse último caso, normalmente, a mutação ocorre com probabilidade unitária por gene, e o parâmetro do operador é utilizado para controlar a variância da gaussiana ou é dado pelo próprio cromossomo, como parte das estratégias de evolução [74]. A atualização dos valores de cada gene, portanto, é dada por [74]:

$$x'_i = x_i + N(0, \sigma), \quad (4.3)$$

onde $N(\xi, \sigma)$ é uma variável aleatória gaussiana com função de densidade de probabilidade [74]:

$$p(\Delta x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\Delta x_i - \xi)^2}{2\sigma^2}}. \quad (4.4)$$

Em programação genética, é comum que a mutação se dê pela substituição de um nó em um indivíduo por uma subárvore gerada aleatoriamente [74]. É importante observar que o tipo de operador de mutação sempre dependerá da representação dada ao problema. A Figura 4.9 exemplifica o funcionamento de um operador de mutação sobre uma cadeia binária.

Operadores de Recombinação

O operador de recombinação (por vezes, chamado também de operador de *crossover* [74]) é um operador de variação binário, i.e., que requer dois operandos. Esse operador é o “motor” dos algoritmos evolucionários e os distingue de outros tipos de algoritmos de otimização [74]. Na maior parte dos problemas, o operador de recombinação gera um ou dois filhos a partir de dois pais. Modelos com mais pais também são possíveis e permitem experimentar com situações que não acontecem na biologia, mas são pouco utilizados na literatura [74]. O princípio por trás do processo de recombinação é simples: a união das características diferentes de indivíduos bem adaptados em novos indivíduos [74]. Os filhos podem apresentar uma qualidade inferior ou igual à dos pais, mas a expectativa é de que, através da combinação de características vantajosas, o valor médio da aptidão da população aumente com o passar das gerações. Geralmente, nos algoritmos evolucionários, após a escolha dos pais há uma chance de que o processo de recombinação não seja aplicado através

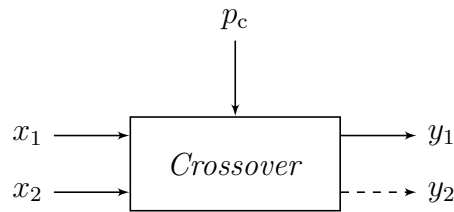


Figura 4.10: Operador de recombinação básico.

da associação de uma probabilidade de ocorrência a esse operador [74]. O modelo básico de um operador de recombinação está representado na Figura 4.10, onde se destacam a probabilidade de se realizar a recombinação, p_c , e a saída y_2 , que aparece tracejada uma vez que nem sempre é utilizada.

Existem alguns tipos clássicos de operadores de combinação aplicados amplamente na literatura. Um deles é o *crossover* de um ponto, com o qual, através da escolha aleatória de um *locus* gênico, um segmento inteiro é trocado entre os cromossomos pais [74]. De maneira genérica, no *crossover* de n pontos, os pais são divididos em diferentes segmentos (através de n pontos) e cada filho é gerado alternando-se a proveniência dos segmentos herdados [74]. No *crossover* uniforme, cada gene dos filhos é tratado independentemente e pode ser herdado de um pai ou de outro, na maior parte dos casos com igual probabilidade [74]. Todos esses métodos compõem a chamada “recombinação discreta” [74]. Se a representação trabalha com cadeias de números reais, por exemplo, a “recombinação aritmética” pode ser utilizada e, em vez de serem trocados, os genes são combinados a partir de um fator de interpolação α . Nesse caso, se y_i é um gene do indivíduo filho e x_{1i} e x_{2i} são os genes correspondentes em cada um dos pais, então a recombinação aritmética, em geral, dispõe que, para um trecho ou em todo o cromossomo filho [74]:

$$y_i = \alpha x_{1i} + (1 - \alpha)x_{2i}, \quad (4.5)$$

com $0 \leq \alpha \leq 1$.

Novamente, o tipo de recombinação utilizado em um algoritmo evolucionário está relacionado à representação escolhida para o problema, e.g., quando os cromossomos são árvores, costuma-se sortear um nó de cada árvore–pai e um *crossover* de subárvores é aplicado [74]. A Figura 4.11 exemplifica a operação de *crossover* de um ponto sobre duas cadeias binárias.

4.2.6 Seleção de sobreviventes

Ao final de cada geração do algoritmo evolucionário, i.e., após os filhos serem criados, realiza-se o processo de seleção de sobreviventes, por vezes também chamado de processo de substituição [74]. Esse procedimento é importante para manter a po-

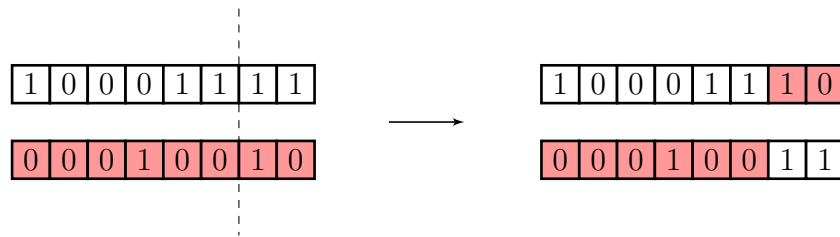


Figura 4.11: Exemplo de recombinação com cadeias binárias: *crossover* de um ponto.

pulação com um tamanho constante e eliminar os indivíduos menos aptos ou mais velhos [74]. Embora possa ser efetuada de forma estocástica, simulando mais perfeitamente o observado na biologia, em regra a seleção de sobreviventes é feita de modo determinístico.

Há duas modelagens distintas para a população comumente empregadas nos algoritmos evolucionários. A primeira é a abordagem geracional [74], representada por “ (μ, λ) ” e assim chamada porque, a cada geração, o conjunto de μ indivíduos (pais) é integralmente substituído pelos novos indivíduos (se $\lambda = \mu$) ou pelos μ melhores indivíduos dentre os λ filhos. No modelo de estado estacionário, “ $(\mu + \lambda)$ ”, os piores indivíduos são eliminados com base em sua aptidão, para que a nova geração mantenha apenas μ indivíduos [74]. Nesse último caso, a nova população pode ser uma composição de indivíduos antigos e novos. Em modelos generacionais, pode-se optar por manter o melhor ou alguns dos melhores cromossomos da população, em vez de substituir todos os antigos indivíduos por novos, muitos dos quais podem exibir aptidões inferiores. Essa técnica é conhecida como “elitismo” [74].

4.2.7 Critério de parada

O ciclo de avaliação de aptidão, seleção de pais, recombinação e mutação, e seleção de sobreviventes pode ser repetido até que determinado critério seja satisfeito. Esse critério de terminação ou critério de parada pode estar atrelado ao ótimo global da função aptidão, caso seja conhecido, a menos de um erro pequeno [74]. Contudo, dada a natureza estocástica dos algoritmos evolucionários, esse ponto da superfície adaptativa pode não ser atingido e o algoritmo nunca terminar. Em [74], são sugeridos alguns critérios de parada alternativos, a saber:

- atingir o tempo máximo de uso de CPU;
- superar um número limite estipulado de avaliações da função de aptidão;
- melhoria de aptidão inferior a um limiar previamente estipulado (permanecendo assim após certo tempo, medido em número de gerações ou em avaliações da função de aptidão);

- superar um limite inferior determinado para a diversidade populacional.

De forma geral, após certo número de gerações, um prolongamento do esforço computacional não significa um aumento proporcional na qualidade das soluções [74]. Esse efeito se traduz no pequeno valor da derivada da curva de desempenho de um algoritmo evolucionário após um grande número de gerações (cf. Figura 4.5).

4.2.8 Manutenção de Diversidade

Sobretudo em problemas multimodais, i.e., problemas que apresentam diferentes ótimos locais além do ótimo global, a manutenção da diversidade no espaço do genótipo pode ser importante porque fornece soluções nos diferentes nichos populacionais [74]. As técnicas de manutenção de diversidade atuam sobre os mecanismos de seleção de pais ou de sobreviventes, e.g., restringindo as opções de “acasalamento” de um indivíduo aos “vizinhos” no mesmo nicho [74], e até mesmo sobre a função de aptidão, como é o caso das técnicas de “compartilhamento de aptidão” ou de “*crowding*” [74]. Um panorama geral sobre as técnicas de manutenção explícita ou implícita de diversidade é encontrado em [74].

4.2.9 A Evolução é um Processo de Otimização?

Conforme apresentado na Seção 4.1, as teorias da evolução propostas por Lamarck e por Darwin compartilham algumas noções. Ambos acreditam que as espécies de seres vivos em estado natural não surgiram na forma em que são encontrados hoje em dia, mas sofreram modificações gradativas em suas características fenotípicas, que foram transmitidas geração após geração [70, 71]. Lamarck contraria a ideia de que “a natureza (ou seu Autor), ao criar os animais, previu todos os tipos possíveis de circunstâncias nas quais eles viveriam e deu a cada espécie uma organização constante, além da forma determinada e invariável de suas partes, forçando cada espécie a viver nos lugares e nos climas onde são encontradas e a conservar os hábitos conhecidos” [71]. Para Lamarck, a natureza teria criado, independente e sucessivamente, formas primitivas distintas através da “geração espontânea” [71]. Começando pelos “mais imperfeitos ou os mais simples” [71] e objetivando “terminar sua obra com os mais perfeitos” [71], ela teria complicado gradualmente a organização das espécies, através das circunstâncias que cada espécie enfrentou ao se espalhar ao redor do planeta e dos hábitos que foram adquiridos com o tempo [71]. A adaptação dos seres vivos às condições sempre mutáveis do ambiente seria suscitada por uma força interna advinda da própria vida [71]. Conquanto carregue traços teleológicos, i.e., posicione a motivação das alterações dos organismos em sua finalidade última — um estado de perfeição —, a visão de Lamarck é também materialista, porque ele

acredita que as adaptações dos seres vivos podem ser determinadas pelas interações com o ambiente e a força do hábito na lei do uso e do desuso [71].

Descreditado até 1859, o mecanismo de transmissão dos caracteres adquiridos logo se tornou a única alternativa à aleatoriedade subjacente ao processo de “tentativa e erro” da seleção natural [72]. Darwin propõe uma ancestralidade comum a todos os seres vivos, libertando-se da ideia de Lamarck de gerações espontâneas ocorrendo em paralelo. O mecanismo da evolução é explicado de forma convincente através da competição entre os seres vivos¹¹ e a noção da hereditariedade das características é consolidada com auxílio da genética de Gregor Mendel [72]. Inumeráveis foram as tentativas de interpretar e conciliar a teoria darwiniana com o pensamento teleológico, atribuindo à variação a causa primária e motor da evolução [78]. Para Darwin, contudo, a variação nas espécies não apresenta um propósito de desenvolvimento, i.e., uma nova característica não é gerada com a prospecção de seu uso futuro, e é a seleção natural que direciona o processo evolutivo [78]. Paralelamente, o valor da aptidão de um indivíduo com relação às condições impostas pelo meio no qual está inserido não pode ser entendido como uma medida absoluta *a priori*, mas é determinado *a posteriori*, em função de seu sucesso reprodutivo [79]. Na teoria da evolução de Darwin, a adaptação evolutiva é interpretada como sendo um processo de “especialização constante” [78], que não implica “melhoria contínua”, visto que, muitas vezes, o processo evolutivo “leva a becos sem saída fatais” [80]. Adicionando-se a isso o caráter variável das condições do meio, a evolução pode muitas vezes se traduzir em “um movimento irregular em zigue-zague” [80]. Assim, dificilmente se pode creditar qualquer traço teleológico ao processo evolutivo de Darwin.

Nas palavras de Ernst Mayr, biólogo que fez grandes contribuições para a teoria sintética da evolução: “Para clarificar, a seleção natural é um processo de otimização, mas não tem meta definida e, considerando o número de restrições e a frequência dos eventos aleatórios, seria muito equivocada chamá-la de teleológica. Tampouco é qualquer melhoria por adaptação um processo teleológico, porque a decisão se dada mudança evolucionária se qualifica como uma contribuição à adaptabilidade é estritamente *post hoc*” [80]. É preciso, portanto, tomar certo cuidado com a linguagem empregada ao abordar o tema. Quando se diz que “as girafas evoluíram alongando seus pescoços para alcançar os ramos mais altos das árvores”, deve-se compreender tão somente que os indivíduos que apresentavam maior alcance obtiveram maior sucesso reprodutivo que os de menor alcance. A seleção natural não possui um “objetivo de longo prazo” [80], embora pareça quando analisada considerando-se uma série de gerações passadas. O processo de evolução deve ser pensado, então, menos

¹¹Darwin nunca abandonou completamente o lamarckismo, que está presente no próprio “*Origin of the Species*”. Para ele, a lei de transmissão dos caracteres adquiridos admitia um papel secundário no processo de evolução [78].

como um método de otimização e mais “em termos de uma estratégia de exploração e adaptação a uma paisagem de aptidão complexa e variante no tempo” [81].

Em um problema de otimização global, objetiva-se encontrar o vetor-solução $\mathbf{x}^* = [x_1^*, \dots, x_n^*]$ que, dentre um conjunto de soluções possíveis, maximiza¹² uma função $f(\mathbf{x})$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Ou seja, \mathbf{x}^* é tal que, para todo \mathbf{x} no espaço de busca, $f(\mathbf{x}^*) \geq f(\mathbf{x})$. Exceto em casos muito simples, não existe garantia de convergência dos algoritmos evolucionários para o máximo global da função e, assim como os métodos de busca local, esses algoritmos podem ficar presos em ótimos locais [74]. Contudo, ao contrário de diversos tipos de algoritmos de otimização, os algoritmos evolucionários contam com um conjunto diverso de pontos candidatos à solução — sua população — e com uma função de densidade de probabilidade não uniforme para geração de novos pontos, mantida através de seus operadores de recombinação e mutação. Essa característica dos algoritmos evolucionários permite “tatear” na superfície adaptativa em busca de um ótimo global e também, eventualmente, escapar de ótimos locais [74].

Teorema dos *Schemata*

John Henry Holland, pioneiro dos algoritmos genéticos, tentou formalizar o processo de evolução em seu livro “*Adaptation in Natural and Artificial Systems*” [82]. Conhecido como “Teorema do *Schema*”, esse trabalho possui muitas falhas e foi posteriormente criticado e expandido por Holland e outros pesquisadores [74]. Fornece, contudo, um parecer do funcionamento dos GAs [74].

Holland desenvolve o teorema dos *schemata* sobre um GA “canônico”, i.e., de representação genotípica binária e generacional, com mutação simples, *crossover* de um ponto e seleção de pais proporcional à aptidão. Um *schema* \mathcal{H} é um hiperplano no espaço de busca, representado com os dígitos binários e também com # — “tanto faz”¹³ [74]. Assim, o *schema* $\mathcal{H} = 1\#0\#\#$ é o hiperplano que contém todos os indivíduos cujos genes da primeira e terceira posições contém os alelos “1” e “0”, respectivamente. Os indivíduos de uma população são entendidos como “instâncias” de um ou mais *schemata* aos quais podem ser associados [74]. No exemplo anterior, há 2^3 diferentes instâncias para o *schema* \mathcal{H} . Sob essa ótica, o processo de otimização é entendido como a busca por um *schema* sem símbolos de “tanto faz” e com a maior aptidão possível [74].

Definindo [74]:

- $o(\mathcal{H})$ como a ordem do *schema* \mathcal{H} , i.e., o número de posições definidas no *schema*,

¹²Para problemas de minimização, o raciocínio é análogo.

¹³Em inglês, “*don't care*”.

- $d(\mathcal{H})$ como o comprimento de definição do *schema* \mathcal{H} , i.e., a distância entre as posições definidas em cada extremidade do *schema*,
- $f(\mathcal{H}, t)$ como a aptidão média do *schema* \mathcal{H} , i.e., a média dos valores de aptidão das diferentes instâncias de \mathcal{H} , na geração t ,
- $\langle f(t) \rangle$ como a aptidão média da população na geração t
- e l como o comprimento dos genótipos,

Holland calcula a probabilidade de destruição de instâncias de um *schema* \mathcal{H} pelos operadores de variação. Para a recombinação, essa probabilidade é função do comprimento do genótipo e do comprimento de definição do *schema*, podendo ser definida por:

$$P_c^d(\mathcal{H}) = \frac{d(\mathcal{H})}{l-1}. \quad (4.6)$$

Similarmente, a destruição do *schema* como resultado de mutação é dada por:

$$P_m^d(\mathcal{H}) = 1 - (1 - p_m)^{o(\mathcal{H})}. \quad (4.7)$$

Além disso, Holland determina a probabilidade de o *schema* \mathcal{H} ser selecionado para reprodução na geração t^{14} . Se nesse momento a proporção de instâncias do *schema* \mathcal{H} na população é $m(\mathcal{H}, t)$, então é possível mostrar que, na geração seguinte [74]:

$$m(\mathcal{H}, t+1) \geq m(\mathcal{H}, t) \frac{f(\mathcal{H}, t)}{\langle f(t) \rangle} \left[1 - \left(p_c \frac{d(\mathcal{H})}{l-1} \right) \right] (1 - p_m o(\mathcal{H})). \quad (4.8)$$

A Equação 4.8 é o resultado fundamental do teorema, e sua interpretação é que a proporção de instâncias na população aumenta com o passar das gerações para *schemata* com aptidão “acima da média” de forma que:

$$\frac{f(\mathcal{H}, t)}{\langle f(t) \rangle} \left[1 - \left(p_c \frac{d(\mathcal{H})}{l-1} \right) \right] (1 - p_m o(\mathcal{H})) > 1. \quad (4.9)$$

4.3 Aplicações à Composição

O problema da composição pode ser tratado com o auxílio dos algoritmos evolucionários, cuja interpretação no contexto musical possui interessantes implicações. Segundo nota Nierhaus, a característica de geração e avaliação contínua dessa classe de algoritmos se assemelha ao próprio conceito de composição musical [7]. Nas aplicações musicais, os operadores de variação podem assumir novas e diferentes formas, conforme já mencionado neste capítulo, mas o elemento fundamental em

¹⁴Porque o método de seleção de sobreviventes é generacional, a seleção para reprodução é a única forma de um *schema* ser perpetuado para as próximas gerações, através de suas instâncias.

um algoritmo evolucionário continua sendo a função de aptidão. É essa função que carrega a informação subjetiva do que se define como “boa música” ou, eventualmente, como um estilo específico de composição. Para enfrentar o problema da composição com algoritmos evolucionários de forma “não supervisionada” por um ser humano, é necessário definir uma codificação da estética, do processo criativo de um compositor [76]. As abordagens evolucionárias em composição algorítmica podem ser distinguidas em três grupos, considerando, principalmente, o uso que fazem da função de aptidão:

1. Métodos automáticos;
2. Métodos interativos;
3. Métodos sem aptidão.

Nos métodos automáticos, as composições musicais são usualmente analisadas com relação à sua similaridade com alguma peça (e.g., medida em desvios de *pitch*) ou ainda por meio de regras definidas por alguma teoria musicológica e que assumem a forma de restrições ou penalizações da função de aptidão [76]. Métodos heurísticos também são incluídos nesse grupo. Nesses métodos, a avaliação é realizada tomando por base propriedades funcionais e estéticas das composições, tipicamente características melódicas e harmônicas [11], e o valor de aptidão de uma composição é ou uma soma ponderada das características extraídas ou um vetor numérico que se deseja otimizar, nas abordagens multiobjetivo [11].

Uma alternativa ao estabelecimento formal de uma função de aptidão que codifique a estética musical, dada a dificuldade dessa tarefa, é a utilização de métodos interativos, i.e., com avaliadores humanos [11]. Um problema comum a esse grupo de métodos, contudo, é conhecido como “gargalo de aptidão” (“*fitness bottleneck*” [7]) e decorre do grande número de avaliações da função aptidão que são necessárias em cada geração. O critério de avaliação do usuário pode variar no tempo ou mesmo ser influenciado pelos sintomas do cansaço [11]. Algumas soluções foram propostas para atacar essa dificuldade, por exemplo [7, 11]:

- manter pequenos o tamanho da população e o número de gerações;
- utilizar avaliações em diferentes estágios, alternando o uso de avaliações algorítmicas e subjetivas;
- diminuir o número de avaliações humanas de aptidão, tomando como referência de um grande grupo de indivíduos o centroide mais próximo, que foi determinado por uma técnica de *clusterização*;
- usar redes neurais artificiais treinadas para substituir as preferências do usuário.

Essa última solução indicaria um retorno aos métodos automáticos.

Finalmente, nos métodos sem aptidão não há pressão seletiva, e a aplicação dos operadores de variação geralmente é determinada por um usuário [7].

4.3.1 O que é um indivíduo?

Além da definição da função de qualidade, outro importante aspecto dos algoritmos evolucionários é a escolha da codificação dos indivíduos — a representação [7]. As abordagens evolucionárias, quando aplicadas à música, usualmente assumem a forma do dialeto dos algoritmos genéticos, i.e., cada indivíduo é uma peça musical (ou uma frase, ou um compasso) representada através de uma cadeia de valores, e.g., sequência de *pitches*, de durações ou combinações dessas duas dimensões [76]. Os genes da representação genotípica também podem carregar informações de intensidade, articulação ou timbre, se necessário [76].

De acordo com os resultados de diferentes trabalhos, a escolha de uma codificação absoluta ou relativa (intervalar) para os tons pode fazer diferença no desempenho do algoritmo [7]. Observando, por exemplo, o funcionamento de um operador de mutação: em uma representação absoluta, este operador tem um efeito pontual, mas sua aplicação afeta o fenótipo inteiro caso a representação seja relativa [7]. Uma alternativa que combina algumas características das representações absoluta e relativa é a representação em graus de escala [76].

Nas aplicações da programação genética, os indivíduos são árvores contendo chamadas de função e variáveis que formam estruturas de controle para criação musical [7].

4.3.2 Revisão bibliográfica

Embora existam abordagens de composição envolvendo outros dialetos dos algoritmos evolucionários, principalmente programação genética, nesta seção apenas o uso de algoritmos genéticos no campo da Composição Algorítmica é revisado, através de alguns trabalhos com características diferentes. Especial atenção é dedicada à forma da função de aptidão, elemento fundamental nessa classe de algoritmos.

Pontes temáticas

Um dos trabalhos pioneiros em evolução musical é o de Andrew Horner e David Goldberg [83], no qual os pesquisadores desenvolvem um algoritmo para geração de sequências melódicas com base na técnica de “pontes temáticas” (“*thematic bridging*” [83]). Nessa técnica, um padrão musical inicial é transformado em outro padrão de referência, por uma determinada duração, através da aplicação sequencial

Tabela 4.1: Exemplo de ponte temática. Padrão inicial: Gb Bb F Ab Db. Adaptado de [83].

Descrição da operação	Padrão resultante
Deletar última nota do padrão inicial	Gb Bb F Ab
Rotacionar o padrão	Bb F Ab Gb
Deletar última nota	Bb F Ab
Alterar a nota inicial	Eb F Ab
Rotacionar o padrão	F Ab Eb

Resultado: Gb Bb F Ab Bb F Ab Gb Bb F Ab Eb F Ab F Ab Eb

de um conjunto de operações, e.g., alteração e reordenação de notas ou durações [83]. O resultado é obtido através da concatenação dos padrões intermediários [83]. Um exemplo de ponte pode ser observado na Tabela 4.1, dado um padrão inicial (Gb, Bb, F, Ab, Db), um padrão de referência (F, Ab, Eb) e a duração de 17 notas.

Na representação indireta [74] proposta por Horner e Goldberg, cada gene nos cromossomos representa um operador [83]. Uma vez que os padrões inicial e final e a duração das pontes estejam determinados, a avaliação dos indivíduos é feita executando-se a sequência codificada nos cromossomos, da esquerda para direita [74].

Com relação ao funcionamento do algoritmo genético, os operadores de recombinação (*crossover* de um ponto) e mutação foram utilizados [83]. O método do torneio foi escolhido para seleção de pais e o comprimento dos genótipos foi feito fixo, mas diferentes comprimentos (até 40 operadores) foram testados [83]. A aptidão dos indivíduos foi calculada por um sistema em duas etapas: a primeira verifica que as notas do padrão final estavam corretas (inclusive, corretamente dispostas), e a segunda etapa compara a duração da ponte temática com a duração desejada [83].

O aplicação simples de Horner e Goldberg introduz o conceito de computação evolucionária na composição musical e descreve alguns operadores que podem, inclusive, ser utilizados como alternativas aos operadores de variação.

GenJam: compondo solos de jazz

O programa *GenJam*, criação de John Al Biles, tem como objetivo modelar, por meio de algoritmos evolucionários, “um músico novato de *jazz* aprendendo a improvisar” [84].

Para gerar uma melodia, o algoritmo recebe um arquivo que contém, entre outras coisas, a progressão de acordes sobre a qual realizar a improvisação [84]. O *GenJam* também lê sequências de MIDI da seção rítmica, formada por piano, baixo e bateria geradas previamente pelo usuário [84]. O algoritmo de Biles opera sobre duas populações diferentes, uma de frases e outra de compassos. Na população de com-

passos, cada cromossomo é uma cadeia binária representando 8 eventos MIDI, cada um equivalente em duração a uma colcheia em compasso quaternário simples [84]. Os eventos do genótipo de compasso podem ser [84]:

- uma nova nota (representadas pelos números 1–14), evento MIDI que interrompe o evento imediatamente anterior;
- uma pausa (representada pelo número 0);
- uma sustentação (representada pelo número 15), indica a continuação do evento anterior por mais um *quantum* temporal.

Cada evento pode ser codificado com 4 bits; um compasso inteiro, portanto, ocupa 32 bits, fazendo com que o espaço de genótipo compreenda 2^{32} indivíduos¹⁵ [84]. Na população de frases, cada cromossomo carrega quatro números, ponteiros para os indivíduos da população de compassos [84], com possibilidade de repetição de compassos na mesma frase. Na primeira versão do *GenJam*, a população de frases contava com 48 indivíduos, enquanto a população de compassos continha 64 indivíduos [84]. Assim, cada posição no cromossomo de uma frase corresponde, em binário, a 6 bits, e o indivíduo todo é representado com 24 bits [84]. Como cada frase compreende quatro compassos, as duas populações formam uma “hierarquia de estruturas melódicas mutuamente dependentes” [84].

Os símbolos de nova nota são mapeados para *pitches* absolutos de acordo com a progressão de acordes subjacentes (máximo de dois acordes por compasso) [84]. Isso significa que cada repetição de um compasso em uma dada frase poderia soar diferente, caso a progressão de acordes indicasse uma mudança no mapeamento das notas [84]. O mapeamento indica quais notas são permitidas em cada escala, geralmente, compreendendo duas oitavas. Por exemplo, um acorde C7 indica uma escala de Dó mixolídio sem o 4º grau (C D E G A Bb) e o mapeamento resultante para as 14 diferentes notas seria: C3 D E G A Bb C4 D E G A Bb C5 D, onde os números indicam as oitavas do piano.

Essas limitações de representação posicional (em que a duração dos eventos é codificada pela distribuição dos mesmos nos cromossomos e cada espaço corresponde a um *quantum* temporal) e de número de notas, que certamente seriam sentidas por um músico profissional, auxiliam na redução do espaço de busca [84]. Por sua vez, a limitação de escala imposta às notas impede que o *GenJam* sugira uma nota “errada” (o que não garante que as notas sugeridas sejam necessariamente as “corretas”) [84].

Para determinar a aptidão dos indivíduos (frases e compassos), o *GenJam* utiliza um avaliador humano [84]. A avaliação de aptidão interativa é o grande “gargalo”

¹⁵O espaço do fenótipo contém menos de 2^{32} indivíduos diferentes, uma vez que uma pausa seguida por uma sustentação ou um par de pausas soam ao ouvinte da mesma forma [84].

do algoritmo, o que limita tanto o espaço de busca como também o tamanho das populações e o número de gerações [84]. A avaliação é feita de modo binário, com o usuário indicando através de duas teclas o quanto gostou ou desgostou de um compasso ou de uma frase [84]. Cada apertar da tecla implica um incremento ou decremento do valor da aptidão do indivíduo, limitado superior e inferiormente pelos valores 30 e -30 , respectivamente [84]. O valor da aptidão é guardado junto ao cromossomo do indivíduo.

A inicialização do algoritmo é feita de modo aleatório, conforme é a prática comum dos algoritmos genéticos. A distribuição é uniforme para a população de frases, mas na população de compassos apresenta diferentes probabilidades para notas ($\frac{1}{24}$ cada), pausas e sustentações ($\frac{5}{24}$ cada) [84]. Essas probabilidades foram determinadas empiricamente e, segundo Biles, dão um “mínimo de ritmo” [84] às populações iniciais.

A seleção de pais e a seleção de sobreviventes são unidas em um mesmo procedimento, que se assemelha ao método do torneio. Para a população de compassos, quatro indivíduos são selecionados aleatoriamente para formar uma “família”: os dois melhores geram filhos que substituem os dois piores e que não podem ser selecionados para participar de uma nova “família” na mesma geração em que foram criados [84]. A cada geração, o processo é repetido até que metade da população seja substituída [84]. Esses 32 novos compassos são combinados em 8 frases, que substituem as piores de um esquema de torneio a quatro frases [84]. As 16 frases restantes passam por um procedimento similar ao dos compassos, sendo agrupadas em famílias para seleção e reprodução [84].

O primeiro operador de variação aplicado é o operador de recombinação (*crossover* de um ponto) [84]. Um dos filhos gerados (frase ou compasso) é mantido intacto enquanto o outro é alterado por um dentre os diferentes operadores de mutação [84]. Para a população de compassos, os operadores de mutação possíveis podem ser vistos na Tabela 4.2. O operador de retrogradação reverte a ordem dos valores dos genes no cromossomo e o operador de rotação “gira” esses valores por um valor aleatório (entre 1 e 7) de genes para a direita. O operador de inversão transforma pausas em sustentações (e vice-versa) e reflete os *pitches* em relação ao valor central [84]. Os operadores de ordenação e transposição preservam a estrutura rítmica dos cromossomos [84]. O número de passos para transposição é escolhido uniformemente entre 1 e 4, sempre na direção com “maior espaço” de cada cromossomo [84]. Se uma nota é transposta para além dos limites superior ou inferior, ela é “refletida” de volta para este domínio [84]. Os operadores de mutação para as frases¹⁶ podem ser vistos na Tabela 4.3. A variabilidade genética é garantida por meio dos operadores

¹⁶O leitor interessado pode referir-se a [84] para uma análise detalhada do funcionamento de cada operador de mutação de frase.

Tabela 4.2: Operadores de mutação para compassos do *GenJam*. Adaptado de [84].

Operação de mutação	Compasso mutante
Nenhum (compasso original)	9 7 0 5 7 15 15 0
Retrogradação	0 15 15 7 5 0 7 9
Rotação à direita (e.g., 3)	15 15 0 9 7 0 5 7
Inversão (15 – alelo)	6 8 15 10 8 0 0 15
Ordenação ascendente	5 7 0 7 9 15 15 0
Ordenação descendente	9 7 0 7 5 15 15 0
Transposição (e.g., +3)	12 10 0 8 10 15 15 0

Tabela 4.3: Operadores de mutação para frases do *GenJam*. Adaptado de [84].

Operação de mutação	Frase mutante
Nenhum (compasso original)	57 57 11 38
Retrogradação	38 11 57 57
Rotação à direita (e.g., 3)	57 11 38 57
Reparação genética	57 57 11 29
Super frase	41 16 57 62
Diluição de padrão	31 57 11 38
Frase órfã	17 59 43 22

de mutação de frases e compassos, que forçam leves variações estruturais nas frases da população, evitando assim a produção de “material musical monótono” [7].

Biles reporta os resultados dos testes, apontando como a distribuição de *pitches*, uniforme na inicialização, gradualmente atinge um estado “normal” para o ambiente de improvisações de *jazz* [84]. Um movimento similar é observado para as distribuições de durações e de intervalos, que, de acordo com as preferências do usuário nos testes, foram encaminhadas para valores e passos pequenos, respectivamente [84]. Em uma implementação posterior do *GenJam*, Biles *et al.* substituem a avaliação de aptidão por uma rede neural artificial. Os resultados não são tão satisfatórios quanto no método iterativo [7, 11].

Um conjunto de parâmetros

Em “*Towards Melodic Extension Using Genetic Algorithms*” [85], Michael Towsey e coautores apresentam uma alternativa às funções de aptidão interativas, com base em regras ou em redes neurais artificiais para avaliação de material musical em um algoritmo genético. Definem, portanto, um conjunto não exaustivo, mas abrangente, de 21 medidas estatísticas, heurísticas desenvolvidas com base nas “melhores práticas” de composição melódica [85]. Segundo os autores, uma análise que use essas medi-

das, tendo em vista sua fonte de inspiração, deve se limitar à música ocidental (que emprega, tradicionalmente, uma escala diatônica), às composições monofônicas, sem modulação de tonalidade, com ritmos regulares e fórmula de compasso simples [85].

Cada uma das medidas estatísticas é uma razão adimensional, entre 0 e 1, que determina o número de ocorrências de uma característica particular em relação ao número máximo possível para uma dada melodia. Por exemplo, a variedade de *pitches* é tomada como a razão entre o número de *pitches* distintos e o número de notas da melodia [85]. As medidas, que se encontram detalhadas no Apêndice A, são divididas em cinco grupos, em função do tipo de característica que avaliam:

- Medidas de *pitch*;
- Medidas de tonalidade;
- Medidas de contorno (melódico);
- Medidas rítmicas;
- Padrões.

Para um experimento, Towsey *et al.* preparam um *corpus* contendo 36 composições: 18 melodias “clássicas”, 10 melodias “pré-clássicas”, 6 cantigas de roda e 2 canções populares [85]. Todas as melodias foram transpostas para a tonalidade de Dó maior ou de Lá menor e as durações foram quantizadas [85]. As 21 medidas foram, então, calculadas para cada melodia; média e variância foram determinadas para o *corpus* completo e para os diferentes *subcorpora* [85]. As seguintes dimensões apresentaram as menores variâncias (absolutas), sendo identificadas como mais “restritivas” na caracterização de uma melodia: variedade de *pitches*; intervalos dissonantes; direção e estabilidade do contorno; variedade e extensão rítmicas [85].

Os pesquisadores discutem sobre como um subconjunto das 21 medidas permite distinguir os diferentes estilos musicais e propõem, em seguida, a classificação das melodias segundo técnicas de *clusterização* [85]. Antes, contudo, as dimensões do espaço de medidas são reduzidas através da técnica de análise de componentes principais¹⁷ (PCA — *Principal Component Analysis*). Embora, por exemplo, todas as melodias do *subcorpus* “Clássico” apareçam bem separadas em 3 classes diferentes, o número de melodias por estilo é insuficiente para que os resultados sejam ditos conclusivos [7].

É possível, dessa forma, que os valores médio e de variação permitida para cada medida representem critérios de qualidade gerais, sobretudo quando as variâncias são

¹⁷Uma análise de componentes principais objetiva a redução da dimensionalidade do espaço de variáveis observadas, a partir da transformação desse conjunto em um novo conjunto de variáveis descorrelacionadas — as componentes principais.

pequenas, podendo, então, ser utilizados na formulação de uma função de aptidão — a hipótese é que melodias geradas em concordância com esses valores serão “bem formadas”. Por exemplo, uma dada melodia pode ser penalizada caso seu valor para uma dada medida se distancie por mais que dois desvios padrões do valor médio esperado para aquela medida [85]. Segundo Nierhaus, esse conjunto de critérios pode permitir a composição de melodias vinculadas a um estilo específico [7].

4.3.3 Outros trabalhos

Outros trabalhos também merecem destaque pela utilização de algoritmos evolucionários no campo da Composição Algorítmica. Manuel Alfonseca, Manuel Cébrian e Alfonso Ortega utilizam como medida da aptidão de cada composição a soma das distâncias a um *corpus* previamente escolhido [86]. A função de aptidão é, então, dada por [86]:

$$f(x) = \frac{1}{\sum_{i=1}^N \hat{d}(x, t_i)}, \quad (4.10)$$

onde t_i , com $i = 1, \dots, N$, representa cada uma das composições de referência do *corpus* e $\hat{d}(x, y)$ é a estimativa computável da “distância normalizada de compressão” (“*normalized compression distance*” [86]), uma medida de quão diferentes duas sequências de símbolos são, tomando como referência a quantidade de bits necessária para representá-las (e também sua concatenação) após compressão, i.e.,

$$\hat{d}(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}. \quad (4.11)$$

Aqui, $C(\cdot)$ representa o compressor e xy é a concatenação das sequências x e y [86].

George Papadopoulos e Geraint Wiggins [87] descrevem um sistema para geração de melodias de *jazz* no qual, assim como em [84], a codificação de 21 *itches* (3 oitavas) está relacionada a uma sequência de acordes dada como parâmetro [87]. Cada indivíduo é uma sequência de pares (nota, duração) ou (pausa, duração) e inicialização da população é feita de forma aleatória, com as pausas ocorrendo com 12,5% de probabilidade. Papadopoulos e Wiggins apresentam uma série de operadores genéticos diferentes, incluindo operadores de “cópia” e “troca” de segmentos musicais, que favorecem o desenvolvimento de temas nas composições [87]. A função de aptidão é composta da análise de oito diferentes critérios, que dão pontos ou aplicam penalizações aos indivíduos, e.g., avaliando-se os intervalos, os tamanhos dos intervalos que quebram este critério são somados e multiplicados por um peso de -20 [87]. Também Matt Johnson *et al.* [88] propõem um método de composição com algoritmos evolucionários no qual a função custo é uma soma de pontuações, e.g., cada indivíduo recebe 50 pontos em sua aptidão caso se inicie ou termine na tônica, 15 para cada movimento gradual entre notas (intervalo de primeira ou se-

gunda), mas é penalizado em 20 pontos para cada mudança brusca de duração nas notas [88]. Como os indivíduos podem ter tamanhos variados, as pontuações de um indivíduo são somadas e divididas pelo seu comprimento (em número de notas) [88].

Conforme introduzido na Seção 2.3.3, em [33], Man Yat Lo e Simon Lucas [33] combinam algoritmos evolucionários com cadeias de Markov. Em sua implementação, as cadeias de Markov de primeira ordem são treinadas com um *corpus* de peças clássicas, suavizadas com um método aditivo e utilizadas como a função de aptidão do algoritmo [33]. Assim, quanto maior a probabilidade de a sequência ocorrer, que pode ser estimada pela cadeia de Markov, maior sua aptidão [33]. Destacam, contudo, que o “sucesso” do algoritmo em produzir “melodias interessantes” reside em sua incapacidade de convergir para o ótimo [33], uma vez que as melodias com maior probabilidade são muito simples e repetitivas. Sugerem, finalmente, a aplicação de conceitos musicais gerais em conjunção com o modelo markoviano [33].

4.4 Limitações

Os algoritmos evolucionários têm um papel único no campo da Composição Algorítmica [7]. Enquanto métodos de busca heurística, suas vantagens estão em sua população diversificada e nos operadores de variação, que permitem procurar ótimos locais e globais por todo o espaço do fenótipo.

Uma parcela importante da tarefa dos algoritmos evolucionários é a definição da função de aptidão, a superfície do problema que se deseja resolver. Tipicamente, existem duas maneiras diferentes para sua formulação: a partir de regras ou medidas de características (automática) ou por meio da interação com um ser humano. Formalizar um conjunto de regras para servir de função de aptidão exige conhecimento musical específico e, dependendo do estilo que se deseja representar, pode ser muito trabalhoso. Métodos interativos, por sua vez, sofrem com o chamado “gargalo de aptidão” [7] e, por seu caráter subjetivo, não lançam luz no processo de composição, i.e., não auxiliam a compreensão das escolhas por trás das estruturas musicais [87]. Métodos que se baseiam em estatísticas/extração de características de um *corpus* preestabelecido correspondem a uma “terceira via”, mas não descrevem o conhecimento musical da maneira adequada, apresentando um desempenho inferior ao dos métodos interativos.

Outro ponto relevante nos métodos evolucionários é a representação dos indivíduos (genótipos). *Pitches* podem ser codificados de maneira absoluta ou relativa, ou ainda através de graus de escala. Já as durações podem ser representadas explicitamente ou de forma implícita nas notações posicionais. Os operadores de variação são formulados com grande liberdade, e operadores “musicalmente inteligentes” já são amplamente utilizados para composição.

O grande atrativo dessa classe de algoritmos é a sua capacidade de produzir não uma, mas uma variedade de soluções.

Parte II

Dos Aspectos Experimentais

Capítulo 5

Detalhes Experimentais

Os conceitos e técnicas revistos nos capítulos anteriores foram utilizados para um experimento em Composição Algorítmica, cujos detalhes são apresentados neste capítulo. Para tanto, cadeias de Markov e gramáticas gerativas são utilizadas como forma de determinar a aptidão dos indivíduos em um algoritmo genético. O objetivo desse experimento é a imitação de estilo, mais especificamente o estilo das cantigas de roda que fazem parte da prática comum do folclore brasileiro.

Uma base de dados foi montada com a finalidade de servir como referência estilística para a investigação experimental, e é descrita brevemente na Seção 5.1. Alguns detalhes experimentais são apresentados no final do capítulo.

5.1 Descrição da Base de Dados

A base de dados utilizada no experimento é um subconjunto da coletânea de canções folclóricas registradas e organizadas por Ermelinda Paz em “500 Canções Brasileiras” [89]. Um grupo de critérios foi arbitrado na seleção desse subconjunto, a saber: (1) a localidade de origem; e (2) a divisão dos tempos no compasso. Foram escolhidas apenas músicas nos modos¹ maiores e menores, em divisão binária simples e, conforme indicação da autora, originárias do bairro de Realengo, na cidade do Rio de Janeiro. As 29 canções encontradas que atendem a essas características podem ser subdivididas de acordo com a tonalidade em que foram notadas (além do próprio modo), com a unidade de tempo (U.T.) e com o apoio no compasso inicial (compasso acéfalo, tético ou anacrústico²).

¹Os diferentes tipos de escalas podem ser classificados, de acordo com a organização dos intervalos que as compõem, em diferentes “modos” [29]. Destacam-se as escalas maior e menor naturais (que compõem os modos maior e menor), e os demais modos gregos: dórico, frígio, lídio etc.

²Quando o compasso inicial e, conseqüentemente, o primeiro tempo forte da música são precedidos por uma ou mais notas (com duração total de, no máximo, metade de um compasso), diz-se que a música começa em anacruse ou com compasso anacrústico [29]. O compasso inicial é acéfalo quando seus tempos iniciais (até metade do compasso) não são marcados por notas ou tético se o primeiro tempo forte coincide com a primeira nota.

Tabela 5.1: *Corpus* transcrito para o experimento. As colunas representam, em ordem: o número da canção em [89], o nome da canção, a tonalidade e o modo, a unidade de tempo, o apoio do compasso inicial e o número de compassos. Todas as canções apresentam divisão binária simples nos tempos do compasso.

No.	Nome	Tonalidade	Modo	U.T.	Apoio	Comp.
20	Ciranda, cirandinha	F	Maior	Semínima	Anacrústico	16
21	Bam-ba-la-lão					4
36	Margarida	G				9
50	Tengo, tengo, tengo	B \flat			Tético	8
51	Sinházinha					17
52	Vamos maninha					12
65	Capelinha de Melão	D			Anacrústico	8
82	Carneirinho, carneirão	E \flat				16
83	Peixe vivo					20
84	Folhinha do coqueiro	A			Tético	16
97	Eu era assim					7
99	Sapo Jururú	A \flat			Anacrústico	8
112	Escravos de Job					17
126	Acordei de madrugada	E			Tético	8
127	Pai Francisco					24
140	Entrei na roda	D \flat			Acéfalo	25
141	Caranguejo					16
155	Pombinha rolinha	B			Acéfalo	24
168	Samba-le-lê	G \flat			Tético	16
183	A carrocinha	F \sharp			Acéfalo	17
196	Marcha soldado	C \flat			Tético	8
197	A canoa virou				Acéfalo	9
210	A pobre e a rica	C \sharp			Anacrústico	8
211	Passaraio					9
212	Relâmpago		Colcheia	Tético	76	
213	Florista				39	
216	Samaritana		Mínima	37		
218	Fada	Anacrústico	32			
270	Nesta rua		C	Menor	Semínima	17

Tabela 5.2: Diferentes *subcorpora* e análise do número de compassos.

No.	Descrição do <i>subcorpus</i>	Quantidade de canções	Total de compassos
0	Canções do <i>corpus</i>	29	523
1	Canções maiores	28	506
2	Maiores e semínima como unidade de tempo	24	322
3	Maiores e compasso anacrústico	13	179
4	Maiores e compasso tético	10	245
5	Maiores, semínima como U.T. e comp. anacrústico	12	147
6	Maiores, semínima como U.T. e comp. tético	7	93

Esse *corpus* foi transcrito para o computador utilizando a notação ABC. O processo de transcrição foi fiel às indicações da autora em [89], exceto no caso das cantigas com compasso inicial acéfalo, no qual pausas foram adicionadas (no início e no final da composição) de forma tornar a notação da base de dados a mais universal possível. A lista completa de canções pode ser conferida na Tabela 5.1, e a Tabela 5.2 apresenta os diferentes *subcorpora*, avaliados com relação ao número de compassos.

5.2 Descrição do Experimento

Os objetivos principais do experimento proposto neste trabalho são:

1. projetar uma técnica de composição de peças simples conformadas a um estilo específico que utilize uma abordagem evolucionária no estilo dos algoritmos genéticos (GAs), amparada pelos conceitos expressos nas cadeias de Markov e nas gramáticas gerativas;
2. testar o sistema desenvolvido com um conjunto de cantigas de roda do folclore brasileiro.

Para avaliar a capacidade do algoritmo genético e os benefícios de sua combinação com os outros formalismos, a investigação deste projeto foi conduzida em três partes. Primeiro, foi estudado o problema da composição de linhas melódicas com o estilo da base apresentada acima. Em seguida, a criação de ritmos adequados, i.e., característicos das cantigas de roda, foi avaliada com o mesmo algoritmo. Finalmente, o problema da composição de peças com melodia e ritmo foi atacado, com apoio no conhecimento adquirido pelos estudos iniciais. Dessa forma, apenas dois elementos das composições serão avaliados e simulados em computador — *pitch*

e duração. Outras características musicais, e.g., a articulação e a intensidade das notas, não serão abordadas neste trabalho.

A “fundação” do experimento constitui-se somente do algoritmo genético. Para esta abordagem inicial em Composição Algorítmica, um algoritmo genético foi desenvolvido de maneira similar à definição formal deste “dialeto” dos algoritmos evolucionários, i.e., com operadores de variação sem conhecimento específico do domínio do problema. A superfície de aptidão, que determina a qualidade dos candidatos à solução, foi elaborada, em um momento inicial, com uma adaptação das heurísticas de Towsey *et al.* (cf. Apêndice A). Os critérios de qualidade gerais expressos pelas heurísticas, insuficientes para geração de material melódico consistente com um estilo particular, foram, então, combinados com medidas obtidas ou por cadeias de Markov ou por gramáticas de Kohonen em um algoritmo evolucionário multiobjetivo.

A hipótese, então, é que o sistema proposto pela aplicação conjunta desses métodos da Composição Algorítmica não apresenta ou apresenta em menor escala algumas das deficiências de cada um dos métodos aplicados separadamente. Os modelos de Markov e gramatical permitem uma análise mais “fina” das estruturas musicais, que um algoritmo genético com aptidão fundamentada em heurísticas não parece ser capaz de conceber. Esse, por sua vez, fornece rapidamente um grande número de soluções, e suas heurísticas permitem um controle mais explícito de determinados elementos musicais.

Conforme exposto no Capítulo 5, a probabilidade das sequências, calculada através de cadeias de Markov suavizadas, serviu como função de aptidão no sistema de Lo e Lucas [33]. Similarmente, Sheikholharam e Teshnehlab investigaram o desempenho de um algoritmo genético que utilizava como função de avaliação uma gramática de Kohonen [90]. Nesse caso, a aptidão de uma sequência musical era calculada como função das regras estruturais que ela continha, i.e., o valor percentual de padrões com correspondência no modelo gramatical. Um peso maior era atribuído a regras com maior contexto, considerando que são mais importantes para a descrição da base [90].

A aplicação de algoritmos evolucionários em problemas multiobjetivo como o deste projeto, no qual uma dimensão é função das heurísticas e a outra depende do modelo markoviano ou gramatical, possibilita avaliar o compromisso entre estas duas funções. Os indivíduos ótimos gerados pelo algoritmo determinam, possivelmente, o conjunto de Pareto, através do qual pode-se descobrir o “quanto” de cada função é necessário para uma boa composição, i.e., que se adeque ao estilo da base de dados. Além disso, o experimento permite comparar e contrastar as contribuições feitas ao algoritmo genético pelo modelo de Markov e pela gramática de Kohonen na construção de novas peças ao estilo da cantigas de roda.

Toda a implementação foi desenvolvida em Python 2.7, incluindo os modelos de Markov e da gramática de Kohonen. As canções da base de dados foram processadas com o auxílio do pacote de ferramentas “music21” [91] e o algoritmo genético foi desenvolvido sobre o conjunto de funções do *framework* “DEAP” [92].

5.3 Detalhes Experimentais

O desenvolvimento de um algoritmo evolucionário pressupõe que, previamente, um conjunto de decisões seja arbitrado, e.g., a representação dos indivíduos no espaço do genótipo, os métodos de seleção de pais e sobreviventes, a probabilidade de ocorrência dos operadores de variação etc. Embora a maior parte dessas questões deva ser resolvida empiricamente, uma vez que os algoritmos evolucionários não exigem conhecimentos específicos da tarefa em mãos, algumas são facilitadas após uma análise simples no domínio do problema — em particular, este é o caso da representação fenotípica e de seu mapeamento para o genótipo. Uma análise do *corpus* descrito na Seção 5.1 é apresentada a seguir, com a finalidade de clarificar o caminho trilhado na construção do algoritmo genético proposto neste trabalho a partir da representação dos indivíduos.

5.3.1 Uma análise preliminar do *corpus*

Primeiramente, todas as composições foram transpostas para a tonalidade de Dó maior (ou de Lá menor), para melhor avaliação da diversidade de notas da base³. A extensão de valores de *pitch* encontrada foi de 2 oitavas, como pode ser verificado na Figura 5.1. Destaca-se que a grande maioria dos tons pertencem à escala de Dó maior, i.e., não possuem acidentes. De fato, em apenas 1,15% do tempo em que há notas soando (i.e., ignoram-se as pausas) estas notas não pertencem à escala. É importante que o sistema projetado seja capaz de replicar essa característica do *corpus*. Além disso, destaca-se que em todo *subcorpus* 2 (SC_2) não há ocorrência alguma de acidentes. Esse *subcorpus* é o que contém a maior parte das cantigas.

Além da análise da variedade de *itches*, pode-se avaliar também o desenvolvimento temporal dessas melodias, i.e., a variedade de figuras com diferentes durações empregadas em cada peça. Novamente ignorando as pausas, as diferentes figuras utilizadas ao longo de todo o *corpus* podem ser vistas na Figura 5.2, na qual o número “3” sobre a colcheia e a semínima indica quiáltera⁴. A única peça que apre-

³Em algumas situações, as composições também foram transpostas uma oitava para baixo, o que não altera a tonalidade.

⁴As durações de um grupo de notas em quiáltera não obedecem às divisões regulares do compasso, i.e., cada nota tem a duração maior ou menor do que uma subdivisão do pulso, dependendo do caso. Por exemplo, para as tercinas indicadas na Figura 5.2, a cada três notas em quiálteras tem-se a duração equivalente de apenas duas notas.

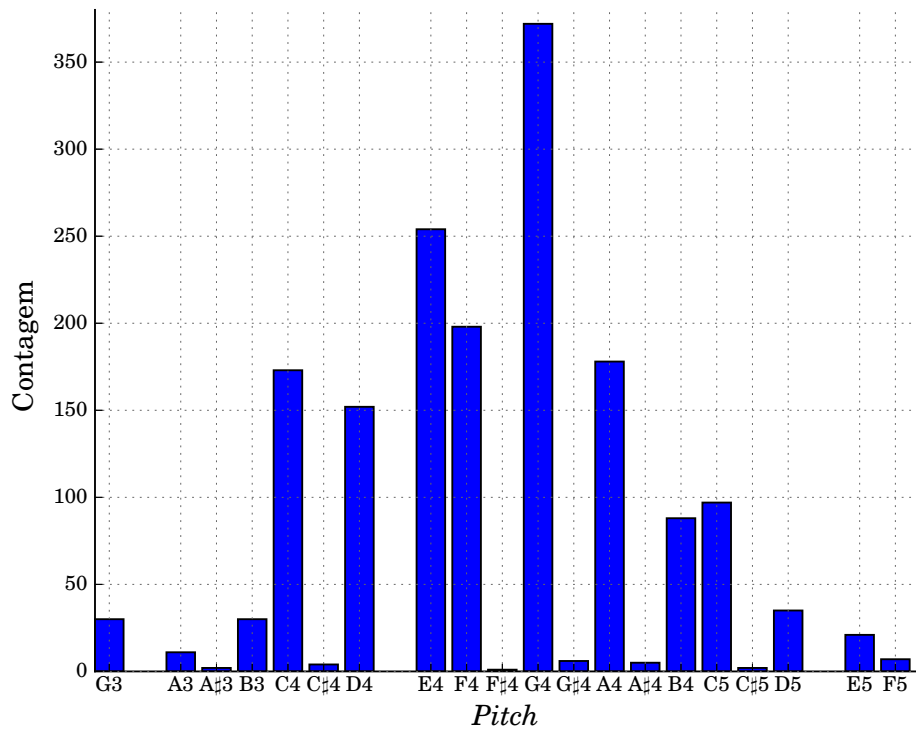


Figura 5.1: Perfil dos *pitches* da base. Todas as cantigas foram transpostas para a tonalidade de Dó maior (ou de Lá menor).

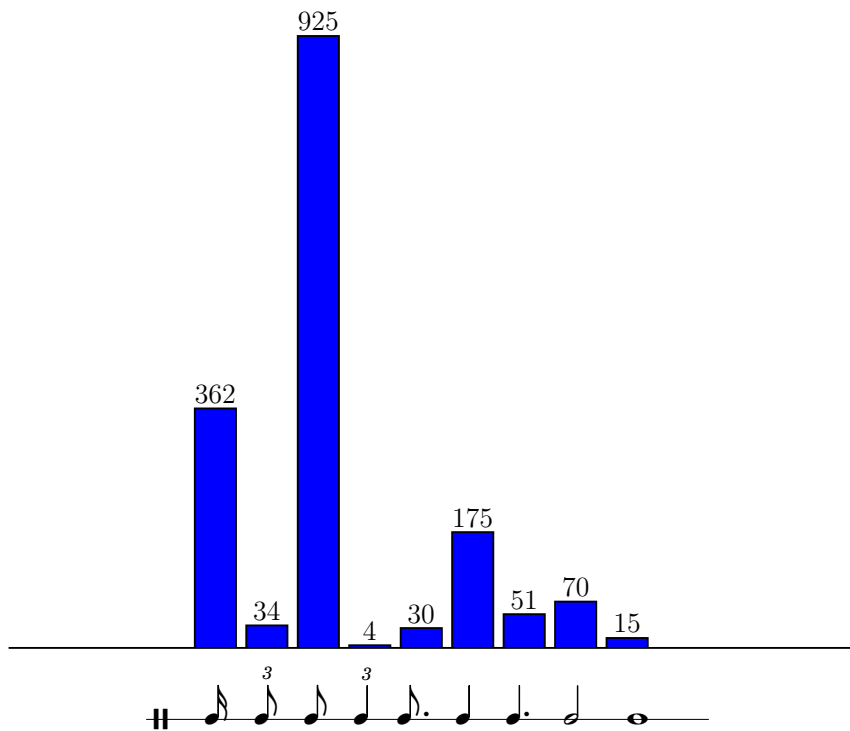


Figura 5.2: Histograma das diferentes figuras presentes no *corpus*. Para esta análise, a ligação entre as notas e também as pausas foram desconsideradas.

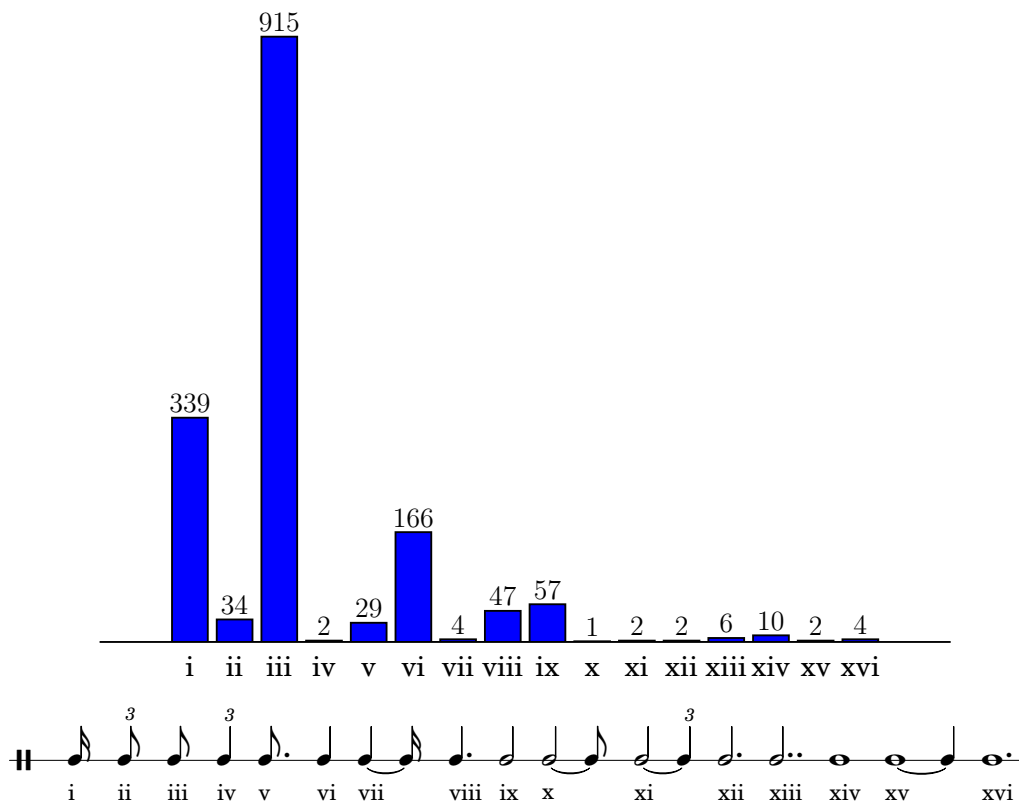


Figura 5.3: Histograma das diferentes durações presentes no *corpus*. Para esta análise, apenas a duração das pausas foi desconsiderada.

senta quiálteras e, de fato, é praticamente toda notada dessa forma, é a canção 155 (“Pombinha rolinha”). As demais composições apresentam uma extensão de figuras que vai da semicolcheia até a semibreve⁵. Essa análise, contudo, é enganosa, porque deve-se considerar, além das diferentes figuras, como as notas soam. Em uma partitura, duas ou mais notas com uma mesma altura podem aparecer ligadas, i.e., estas notas devem soar como uma cuja duração é a soma das durações das notas separadas [29]. A Figura 5.3 apresenta a análise completa das durações das notas na base de dados. Novamente, as pausas são ignoradas, já que uma ou duas pausas com metade de sua duração “soam” da mesma maneira, o que não é verdadeiro para as notas.

Percebe-se agora uma extensão muito maior de valores de duração: há 16 valores diferentes (em vez de 9), indo desde a semicolcheia até a semibreve pontuada. Conforme acima destacado, os valores em quiáltera (“ii” e “iv”) e de notas ligadas com notas em quiáltera (“xi”) só são vistos na canção 155. As durações equivalentes às da ligação entre semínima e semicolcheia (“vii”), da mínima duplamente pontuada (“xiii”), da semibreve (“xiv”), da ligação entre semibreve e semínima (“xv”) e semibreve pontuada (“xvi”) não aparecem no *subcorpus* 2, sendo encontradas apenas nas canções 212 (“Relâmpago”), 216 (“Samaritana”) e 218 (“Fada”).

⁵Uma semibreve tem duração equivalente a 16 semicolcheias.

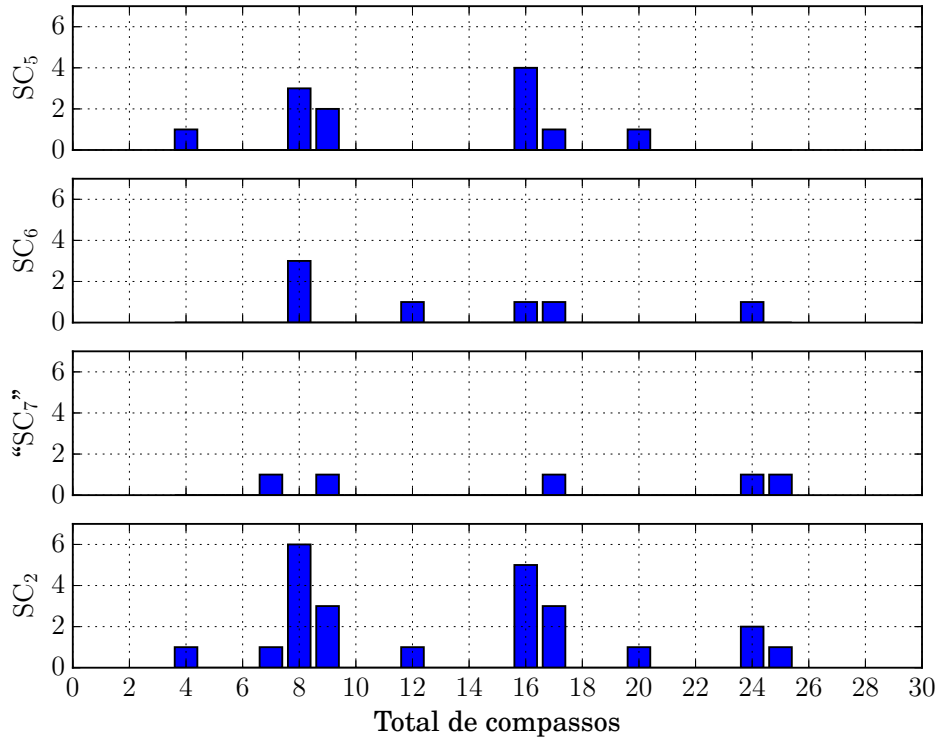


Figura 5.4: Histogramas dos comprimentos (em termo do número de compassos) das canções de diferentes *subcorpora*. Um *subcorpus* “SC₇” contém as cantigas maiores, tendo a semínima como U.T. e com compasso inicial acéfalo. Não se definiu SC₇ anteriormente porque pode ser obtido através da operação $SC_2 - \{SC_5 \cup SC_6\}$.

Outra pesquisa poderia ainda ser realizada, desta vez sobre o comprimento das canções, i.e., o número de compassos que cada uma possui. Essa medida dá uma indicação do comprimento esperado das canções que vão ser geradas pelo algoritmo. A Figura 5.4 apresenta o resultado dessa análise para todas as cantigas do *subcorpus* SC₂ e seus próprios *subcorpora*. As demais canções maiores (as cantigas indicadas por 212, 213, 216 e 218) possuem um grande número de compassos, como pode ser visto na Tabela 5.1.

5.3.2 O problema da representação

A representação relativa (intervalar), utilizada em [1] na *Illiad Suite* e em [65] para a produção de contornos melódicos das improvisações de *jazz*, é mais adequada à produção e interpretação de composições, pois considera que a estrutura das tonalidades é sempre a mesma [89]. Durações também podem ser representadas de maneira similar, indicando que há escalabilidade no âmbito dos padrões rítmicos. Essa representação também pode ser utilizada para descrição dos indivíduos em um algoritmo genético. Seja e_i um evento da composição; se e_i e e_{i+1} são notas, o gene na posição i de um cromossomo que represente essa composição é a dupla $(I_{i,i+1}, D_{i,i+1})$, formada, respectivamente, pelo intervalo e pela razão entre as durações de e_i e e_{i+1} .

Há que se tomar cuidado com as pausas, que precisam ser codificadas de maneira diferente — cada pausa é uma dupla formada por um símbolo indicativo de pausa e pela duração da pausa, em representação absoluta⁶.

Como já foi dito, deseja-se, no presente trabalho, utilizar uma adaptação das medidas heurísticas propostas por Towsey *et al.* [85] como parte de uma função de aptidão em um algoritmo genético. Da maneira que são calculadas, essas medidas não fazem distinção entre os tipos de representação que estão sendo utilizados para a composição. Uma exceção a esse fato são as medidas de tonalidade. Porque a representação relativa é invariante à tonalidade e, uma vez codificadas segundo esse esquema, as peças não carregam mais informação de em qual tom foram escritas, esta representação prejudica a aplicação desse conjunto de medidas. Além disso, conforme indicado no Capítulo 4, a notação por intervalos torna por demais complexa a atuação do operador de mutação, que ou altera toda a melodia a partir do gene em que a esta ocorre ou requer uma alteração de dois em dois genes (para corrigir os efeitos de alteração no valor absoluto das notas subsequentes). Dessa forma, uma representação absoluta dos *itches* e, possivelmente, das durações é mais adequada para os indivíduos da população do algoritmo genético descrito nesta dissertação.

De acordo com a análise preliminar da Seção 5.3.1, os *itches* de G3 a G5 (duas oitavas) podem ser codificados pelos números inteiros 0 até 24⁷, enquanto as pausas assumem o valor 25. Como neste projeto os elementos musicais analisados são a altura e a duração de cada composição, permanece indefinida a representação deste último. De maneira geral, dois tipos de representação absoluta para a duração são observados na literatura [76]: ordenada (ou explícita) e posicional (implícita). No primeiro, o valor da duração de cada nota ou pausa é colocado em uma dupla junto à representação do *itch* ou da indicação de silêncio. É um exemplo o sistema proposto por Papadopoulos e Wiggins [87]. No segundo caso, cada gene é um *quantum* temporal e os alelos indicam ataque de uma nota, sua sustentação ou pausa. O *GenJam* de Biles representa os cromossomos dessa maneira [84].

Neste trabalho, não se deseja empregar operadores de variação que sejam musicalmente “inteligentes” e que não estejam entre os tipos de operadores comumente utilizados nos algoritmos genéticos. Nesse caso, a principal diferença entre as duas formas de representação reside na propriedade das sequências, que é mantida fixa durante os ciclos de gerações. Na representação ordenada (também chamada representação em múltiplos “pontos de vista” — “*multiple-viewpoint*” [31]), o número de eventos é mantido constante. O operadores de mutação e recombinação podem alterar a duração total da composição, mas não o número de notas e pausas que

⁶Se a pausa se encontra na i -ésima posição, o gene na posição $i - 1$ é calculado a partir da relação entre as notas em $i - 1$ e $i + 1$.

⁷Os valores MIDI de 55 a 79 poderiam ter sido usados sem alteração no desempenho do algoritmo. Os números 0 a 24 foram escolhidos por simplicidade.

ela possui. Para a representação posicional, o número de *quanta* ou, mais ainda, o número de compassos, é que se determina antes do início da execução. Nesse caso, o operador de mutação é capaz de gerar novos eventos em qualquer ponto da peça. A notação posicional foi escolhida para este primeiro trabalho em Composição Algorítmica. Nessa representação, a indicação de sustentação de uma nota ou pausa é feita pelo número 26. O *quantum* temporal é a semicolcheia.

Representação e a função de aptidão

O algoritmo genético proposto é multiobjetivo, i.e., além da dimensão definida pelas heurísticas, há uma dimensão comandada por um modelo probabilístico ou gramatical. A aptidão de um indivíduo x , portanto, é um vetor como

$$\mathbf{f}(x) = [f_1(x), f_2(x)], \quad (5.1)$$

onde $f_1(x)$ é a soma ponderada do módulo das distâncias dos valores que x recebe em cada medida de um conjunto \mathbf{H} (que é uma adaptação do conjunto de heurísticas de Towsey *et al.* [85]) a um valor de referência dessa medida (calculado do *subcorpus* de referência, ou matematicamente

$$f_1(x) = \sum_{i \in \mathbf{H}} w_i |F_i(x) - \bar{F}_i|, \quad (5.2)$$

e $f_2(x)$ é a probabilidade de x no modelo de Markov do *corpus* ou uma medida das regras gramaticais que o indivíduo x segue.

A definição da representação dos cromossomos do algoritmo genético não exclui a possibilidade de o modelo de Markov e a gramática de Kohonen que serão utilizados (um de cada vez) como restante da função multiobjetivo trabalharem sobre uma representação diferente. Uma função de aptidão é função dos alelos de um indivíduo, bem como de sua disposição nos genes. Esses valores podem, dentro dessa função, ser combinados de maneira arbitrária, e a função de aptidão não precisa ser linear. Dessa forma, esses dois modelos podem, internamente, utilizar uma notação relativa nas sequências que avaliam e, assim, o sistema como um todo pode se beneficiar desta forma de representação, que, conforme indicado no início desta seção, é mais adequada para a análise de melodias e de ritmos. Se $g(x)$ é a função que mapeia a representação absoluta e posicional em uma representação intervalar, a aptidão de um indivíduo pode ser calculada como:

$$\mathbf{f}(x) = [f_1(x), f_2(g(x))] = [f_1(x), (f_2 \circ g)(x)], \quad (5.3)$$

que tem a mesma forma da Equação 5.1.

Capítulo 6

Resultados e Discussões

A técnica de composição projetada neste trabalho consiste, segundo visto no Capítulo 5, em uma combinação de alguns formalismos da Inteligência Artificial usualmente aplicados em Composição Algorítmica. Os testes da ferramenta desenvolvida foram separados em três grandes tarefas — a composição de melodias, a composição de ritmos e a composição simultânea de melodias e ritmos —, cujos resultados são apresentados e discutidos a seguir. Dentro de cada uma das tarefas, o algoritmo genético ou funciona sozinho ou é amparado por outra técnica (modelos de Markov e gramáticas de Kohonen).

Alguns detalhes da implementação de cada um desses formalismos também podem ser vistos neste capítulo.

6.1 Composição de Melodias

O sistema foi testado, em primeiro lugar, na tarefa de composição de melodias. As sequências de *pitch* das cantigas do *subcorpus* SC₅¹ foram selecionadas para servir como referência dos dois testes desse tipo. Conforme determinado anteriormente, os *itches* são codificados pelos números inteiros de 0 a 24, e os números 25 e 26 representam, respectivamente, as pausas e as sustentações de eventos. Os dois testes de composição melódica realizados com essas sequências diferem, basicamente, na forma da função de aptidão, que pode ser unidimensional ou bidimensional (problema multiobjetivo).

6.1.1 Caso 1: Função de aptidão unidimensional

No caso unidimensional, o algoritmo genético utiliza como função de aptidão uma adaptação² das heurísticas de Towsey *et al.* [85], cumuladas em uma soma ponderada

¹Excetuando-se a cantiga de número 155.

²A natureza das adaptações ao conjunto de heurísticas pode ser vista no Apêndice A.

Tabela 6.1: Valor médio e de desvio das cantigas do *subcorpus* SC₅ após análise pelo conjunto de heurísticas do teste melódico.

Medida	Nome	Média e Desvio	Desvio relativo
1	Variedade de <i>pitch</i>	0,171 ± 0,046	27,2%
2	Extensão de <i>pitch</i>	0,444 ± 0,084	18,9%
3	Centralidade tonal	0,487 ± 0,145	29,8%
4	Notas fora de escala	0,000 ± 0,000	0,0%
5	Intervalos dissonantes	0,005 ± 0,012	244,9%
6	Direção do contorno	0,502 ± 0,029	5,9%
7	Estabilidade do contorno	0,408 ± 0,117	28,8%
8	Movimento gradual	0,442 ± 0,170	38,5%
9	Retorno de salto	0,167 ± 0,373	223,6%
10	Intensidade do clímax	0,565 ± 0,335	59,3%
11	Densidade de notas	0,987 ± 0,025	2,6%
16	Repetição de <i>pitch</i>	0,266 ± 0,166	62,5%
18	Padrões de <i>pitch</i> (3 notas)	0,016 ± 0,036	228,0%
19	Padrões de <i>pitch</i> (4 notas)	0,023 ± 0,075	331,7%

similar à descrita na Equação 5.2. Assim, se \bar{F}_i representa o valor médio da avaliação de todo *subcorpus* em uma dada heurística $F_i(x)$, com $i \in \mathbf{H}$, então a aptidão de um indivíduo x é calculada através de

$$f_1(x) = \sum_{i \in \mathbf{H}} \frac{1}{|\mathbf{H}|} d_i(x), \quad (6.1)$$

onde

$$d_i(x) = \frac{|F_i(x) - \bar{F}_i|}{\sigma_i + \epsilon}, \quad (6.2)$$

σ_i é o desvio padrão da medida i observado no mesmo *subcorpus* e ϵ é um valor pequeno (e.g., $\sim 10^{-4}$) que permite incorporar medidas com variância nula à função de aptidão. O fator de ponderação de cada distância $d_i(x)$ pode ser estipulado de acordo com a heurística i , i.e., dando mais peso na soma a algumas medidas em detrimento de outras. A função $f_1(x)$ é o objetivo a ser minimizado.

Para o teste melódico, as heurísticas que dão informação sobre ritmo e valores rítmicos não têm aplicação. Isso significa que as heurísticas 13 a 15, 17, 20 e 21 não compõem a função de aptidão neste teste. Além dessas, a heurística 12 (“Densidade de pausas”) também foi desconsiderada, porque é complementar à heurística 11 (“Densidade de notas”), i.e., não traz informação nova para a função de aptidão. A Tabela 6.1 apresenta os valores médio e de desvio (além do desvio relativo) da análise do *subcorpus* SC₅ pelas heurísticas utilizadas no teste melódico.

Além da função de aptidão, é importante alinhar os demais parâmetros do algoritmo genético. Os indivíduos são inicializados de maneira aleatória, sendo cada gene escolhido uniformemente no intervalo $[0, 25]^3$. O comprimento de cada cromossomo também é escolhido quando da inicialização no intervalo $[20, 82]$, que são os limites inferior e superior do número de notas no *subcorpus*. A seleção de pais é feita com o método do torneio de tamanho 2. Isso significa que, cada vez que um pai é requisitado pelo algoritmo, dois indivíduos são selecionados aleatoriamente da população e o melhor dos dois, em termos da função de aptidão, é retornado para reprodução. Os operadores de variação utilizados são simples: a recombinação é feita com dois indivíduos através do *crossover* de um ponto e ocorre com 60% de probabilidade após a seleção de pais; a mutação é uniforme, e cada gene é alterado com 5% de probabilidade. Ao final de cada geração, os $\mu = 100$ melhores indivíduos da população e de sua descendência ($\lambda = 100$) são selecionados como sobreviventes, i.e., o modelo populacional é o de “estado estacionário” [74]. O número de gerações (e.g., 100) é utilizado como critério de parada.

O esquema do algoritmo genético desenvolvido nesta dissertação é apresentado no Pseudocódigo 1 e seus parâmetros para o teste melódico com função de aptidão unidimensional estão expressos na Tabela 6.2.

Pseudocódigo 1 Esquema do algoritmo genético deste trabalho.

- 1: Inicializa a população aleatoriamente.
 - 2: Avalia os candidatos a solução.
 - 3: **para** geração $\leftarrow 1$ **até** número de gerações **faça**
 - 4: Seleciona os pais com o método do torneio. $\triangleright \kappa = 2.$
 - 5: Aplica recombinação sobre os pares de pais. $\triangleright p_c = 60\%.$
 - 6: Aplica mutação sobre os filhos gerados. $\triangleright p_m = 5\%.$
 - 7: Avalia os novos candidatos.
 - 8: Seleciona os melhores indivíduos para a próxima geração.
 - 9: **fim para**
-

6.1.2 Caso 2: Função de aptidão bidimensional

Na abordagem multiobjetivo do problema de geração de melodias, um modelo de Markov foi treinado para formar a segunda dimensão da função de aptidão. A estrutura do algoritmo genético é, essencialmente, a mesma do caso unidimensional. Além da função de aptidão, a única diferença reside nos mecanismos de seleção de pais e sobreviventes utilizados.

Para a seleção de pais, o método do torneio ainda é utilizado, mas agora com base no critério de dominância. Conforme visto na Seção 4.2.2, para que se considere

³As sustentações são desconsideradas nessa análise puramente melódica.

Tabela 6.2: Configuração do teste de composição de melodias com função de aptidão unidimensional.

Algoritmo Genético	
Número de gerações:	100
Tamanho da população:	100 indivíduos, $\ell(x) \in [20, 82]$
Representação:	absoluta, somente <i>pitch</i>
Recombinação:	<i>crossover</i> de um ponto
Mutação:	uniforme no intervalo $[0, 25]$
Seleção de pais:	torneio de tamanho 2
Seleção de sobreviventes:	$(\mu + \lambda)$, “estado estacionário”

que um indivíduo A domina sobre outro indivíduo B , é necessário que o valor de sua aptidão em cada uma das diferentes dimensões seja pelo menos igual ao valor de aptidão de B na mesma dimensão e estritamente superior em pelo menos um destes casos (no sentido de maximização ou de minimização). No método do torneio com dominância, em vez de se analisar o valor de cada indivíduo em uma única função objetivo, os valores de aptidão em cada dimensão são comparados para que se estabeleça qual dos indivíduos domina sobre o outro e, portanto, será selecionado como pai. Caso a relação encontrada seja de não dominância, entende-se que os dois indivíduos fazem parte da mesma fronteira, i.e., o subconjunto da população em que todos os indivíduos expressam uma relação de não dominância entre si. Nessa situação, o indivíduo na região da fronteira que é menos densa⁴ é que será selecionado como pai. A esse mecanismo dá-se o nome de *crowding distance* (CD) [93].

A seleção de sobreviventes também utiliza o conceito de dominância. Basicamente, a nova população é preenchida até o tamanho desejado a partir das diferentes fronteiras. Isso significa que, uma vez que os filhos tenham sido gerados, as fronteiras de não dominância são calculadas e os indivíduos são selecionados como sobreviventes a começar da fronteira mais externa, i.e., aquela que domina sobre todas as demais fronteiras e não é dominada por nenhuma. Se na última fronteira nem todos os indivíduos poderão ser selecionados para sobreviver, os melhores indivíduos são escolhidos novamente com base na *crowding distance* [93].

Essas duas alterações no funcionamento do algoritmo genético são características da abordagem do NSGA-II [93], algoritmo que se tornou um “padrão” no tratamento de problemas multiobjetivo.

A outra diferença neste caso do teste de composição de melodias é a segunda dimensão da função objetivo, que é dada pela raiz L -ésima (cf. Equação 2.18) da probabilidade dos indivíduos em um modelo markoviano suavizado. O modelo foi

⁴Essa métrica é definida para cada indivíduo como o comprimento médio dos lados do cuboide definido por seus vizinhos imediatos na mesma fronteira.



- (a) Cromossomo: [25, 26, 9, 26, 9, 26, 14, 26, 12, 26, 26, 26, 26, 26, 26]
- (b) Intervalos: [25, 0, 5, -2]
- (c) Ritmo (relativa): [(25, 0,5), (0, 1,0), (0, 1,0), (0, 4,0)]
- (d) Completa (relativa): [(25, 0,5), (0, 1,0), (5, 1,0), (-2, 4,0)]

Figura 6.1: Diferentes tipos de representação no sistema: (a) representação manipulada pelos operadores do GA; (b) representação utilizada pelo modelo de Markov no teste de composição de melodias; (c) representação utilizada pela gramática de Kohonen no teste de composição de ritmos; e (d) representação utilizada pelos modelos no teste de composição simultânea de melodia e ritmo.

treinado nas ordens $N = 0, 1, 2$ sobre o mesmo *subcorpus* do qual as estatísticas das medidas heurísticas foram tiradas. Contudo, as sequências de *pitch* não foram consideradas em sua representação absoluta, mas na representação relativa, i.e., as sequências de *pitch* foram transformadas em sequências de intervalos antes de serem processadas⁵ (cf. Figura 6.1). Para lidar com a esparsidade e ao mesmo tempo acelerar a convergência do algoritmo genético nessa dimensão da função multiobjetivo, o modelo de segunda ordem foi suavizado com dois métodos: aditivo e de interpolação. Para o método aditivo (cf. Equação 2.24), foram utilizados $\xi = 1$ (método “soma um”) e $|T| = 50^3$ (tamanho do vocabulário⁶). No método de interpolação, os modelos de diferentes ordens foram combinados segundo a Equação B.3, com um peso por modelo (e não por contexto, como na suavização de Jelinek-Mercer). O vetor de pesos ótimo foi calculado assim como indicado no Apêndice B, separando um subconjunto das cantigas do *subcorpus* para treinamento do modelo (75% das cantigas) e o restante para as estimativas do algoritmo.

A função $f_2(x)$ representada pela probabilidade no modelo suavizado é, ao contrário de $f_1(x)$, um objetivo a ser maximizado. Se no caso da primeira dimensão a minimização da soma ponderada das distâncias $d_i(x)$ faz com que as medidas do indivíduo ótimo se aproximem dos valores médios do *subcorpus*, a maximização da segunda dimensão indica que a sequência é mais provável segundo o modelo matemático treinado, i.e., os intervalos musicais são mais prováveis e estão dispostos (até um contexto de duas notas) da maneira mais provável. As duas dimensões “competem” entre si, ou seja, o conjunto de soluções retornado pelo algoritmo representa o compromisso entre essas duas otimizações.

A Tabela 6.3 resume a configuração do teste de composição de melodias para uma função de aptidão com duas dimensões.

⁵Marcadores de início e de final foram colocados para avaliação pelo modelo de Markov assim como no trabalho de Ponsford *et al.* [28] (cf. Seção 2.3.2).

⁶Como a representação absoluta admite 26 valores para cada gene, há 49 diferentes intervalos (entre -24 a 24 semitons) mais a pausa, que é codificada independentemente.

Tabela 6.3: Configuração do teste de composição de melodias com função de aptidão bidimensional.

Algoritmo Genético		Modelo de Markov	
Gerações e população:	100, 100 indivíduos	Representação:	relativa, diferença de <i>pitch</i>
Representação:	absoluta, somente <i>pitch</i>		
Recombinação:	<i>crossover</i> de um ponto	Ordem:	2
Mutação:	uniforme	Suavização:	aditiva e de interpolação com modelos de menor ordem (0 e 1)
Seleção de pais e de sobreviventes:	assim como em NSGA-II [93] (dominância e <i>crowding distance</i>)		

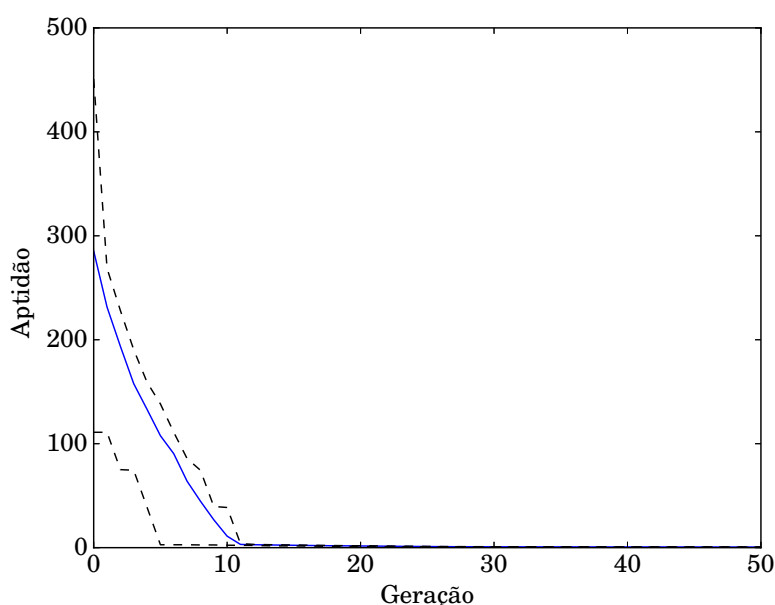
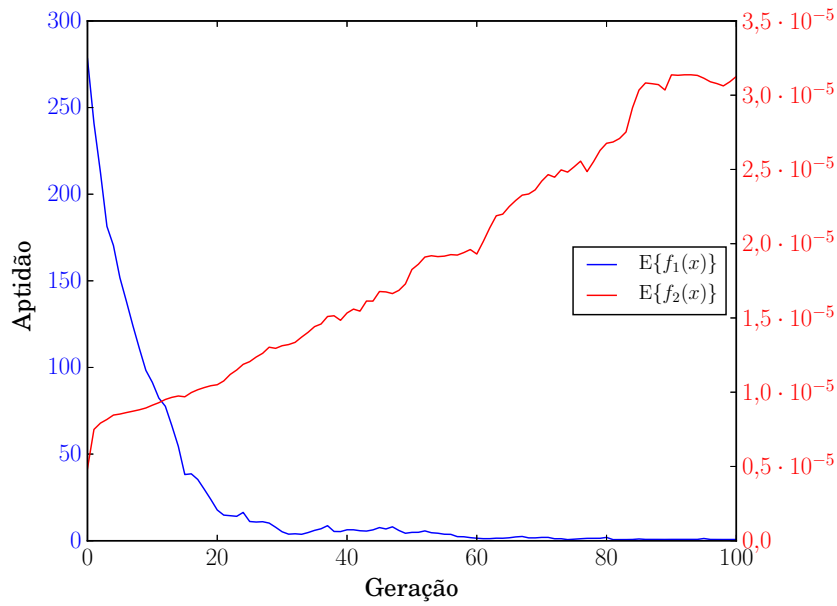


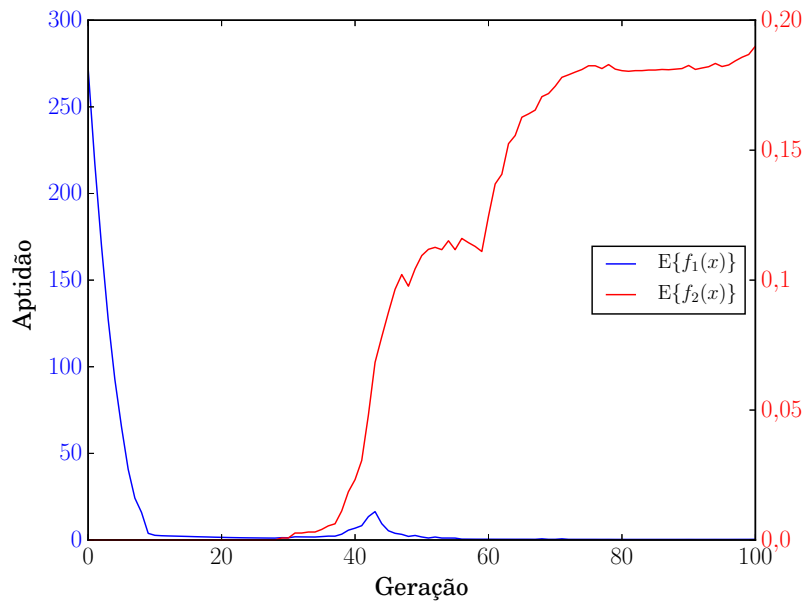
Figura 6.2: Curva de desempenho típica na composição de melodias com função de aptidão unidimensional. O gráfico apresenta o valor médio de aptidão em cada geração, além dos valores máximo e mínimo (tracejados).

6.1.3 Resultados

O algoritmo genético converge rapidamente no caso da função de aptidão unidimensional, atingindo em aproximadamente 10 gerações um vale de mínimo local. Na Figura 6.2 encontra-se a curva de desempenho do algoritmo genético nesse caso, em termos dos valores médio, máximo e mínimo da função de aptidão. Como não há nenhuma ferramenta explícita de manutenção de diversidade, tipicamente, as gerações finais apresentam um ou dois tipos diferentes de indivíduos, que dominaram a população. A população final no caso em que a função de aptidão tem duas dimensões apresenta maior diversidade devido à atuação do mecanismo de *crowding distance* e os indivíduos na última geração diferem uns dos outros em boa parte dos genes. As curvas de desempenho das duas dimensões da função de aptidão estão representadas



(a) Suavização com método aditivo

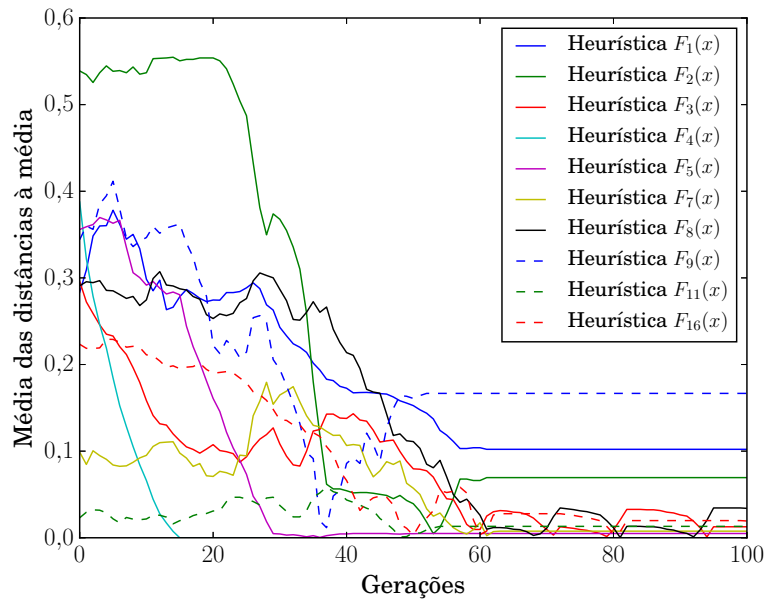


(b) Suavização por interpolação

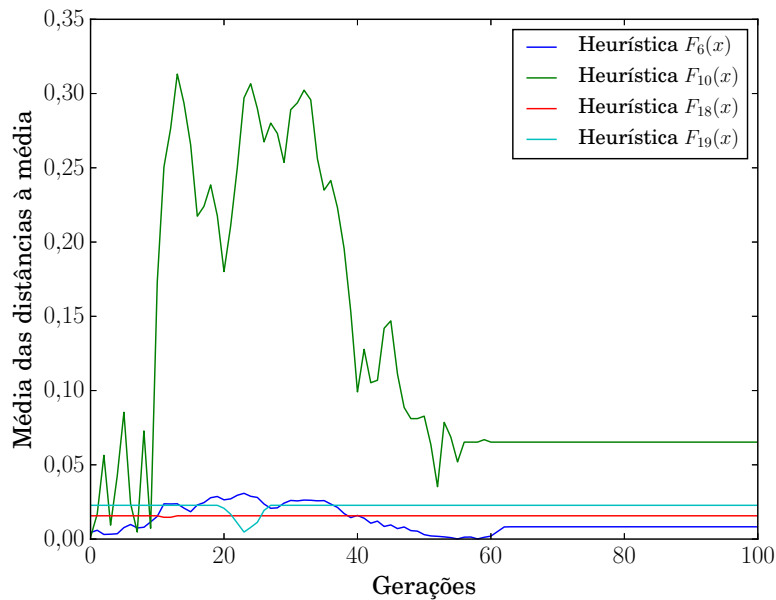
Figura 6.3: Curvas de desempenho típicas na composição de melodias com função de aptidão bidimensional.

na Figura 6.3 para os dois tipos de suavização do modelo de Markov utilizados neste trabalho. Destaca-se o número consideravelmente maior de gerações para que os valores de $f_2(x)$ no caso da suavização por interpolação comecem a crescer. Isso indica que nem mesmo o modelo de ordem 0 possui todas as amostras possíveis que surgem da inicialização aleatória dos cromossomos.

Uma curva de desempenho como a da Figura 6.2 pode ser enganosa, porque faz acreditar que todas as distâncias $d_i(x)$ que compõem esta dimensão da função de aptidão foram minimizadas adequadamente. Como pode ser verificado na Figura 6.4,



(a) Distâncias minimizadas



(b) Distâncias não minimizadas

Figura 6.4: Média das distâncias de cada medida à média esperada no teste melódico. Avalia-se dessa forma quais distâncias foram de fato minimizadas (em média).

que apresenta a evolução da média na população das distâncias de cada medida à média de referência (sem a normalização pelo desvio), algumas das heurísticas não são minimizadas ou até crescem com o tempo.

O que se observa no conjunto dessas medidas é, novamente, um compromisso como aquele expresso entre as duas dimensões de um problema multiobjetivo: internamente, as medidas “competem” entre si — algumas são minimizadas em detrimento de outras⁷. Uma alteração do peso dado a cada distância $d_i(x)$ pode

⁷Um efeito similar pode ser observado ao final das gerações na Figura 6.4a, no qual algumas

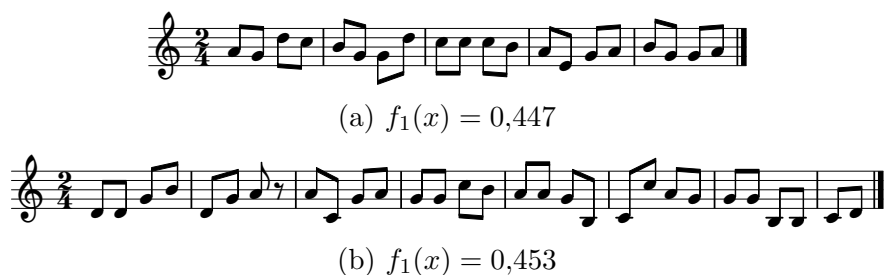


Figura 6.5: Indivíduos melódicos gerados somente por meio das heurísticas. Cada indivíduo é o melhor ao final de todas as gerações para duas rodadas diferentes do algoritmo.

“distribuir” melhor o padrão de minimização, efetivamente otimizando distâncias como as da Figura 6.4b. Contudo, isso se dará, possivelmente, à custa de valores finais maiores para as heurísticas da Figura 6.4a.

As linhas melódicas geradas pelo algoritmo nos dois casos são musicalmente interessantes. De maneira geral, as melodias construídas somente pelas heurísticas apresentam intervalos grandes e desordenados. A Figura 6.5 apresenta duas melodias geradas em rodadas diferentes do algoritmo, mas com “qualidade” similar. Apesar da inicialização aleatória dos comprimentos das sequências (uniformemente distribuída a partir da extensão de valores observados na base), as melhores soluções comumente se formam em sequências de menor comprimento (entre 20 e 30 notas), que rapidamente dominam a população. Talvez esse fato se deva ao pequeno desvio (e portanto grande peso) da heurística $F_4(x)$ (“Notas fora de escala”): os indivíduos de menor comprimento tem maior probabilidade de eliminarem as notas fora de escala e, assim, serem assinalados por bons valores de aptidão logo nas primeiras gerações; esses indivíduos, então, dominariam a população e o algoritmo forneceria soluções curtas. Note-se também que as melodias finais não apresentam acidentes, o que demonstra o funcionamento adequado dessa heurística.

A Figura 6.6 mostra uma fronteira de soluções não dominadas ao final de uma rodada do algoritmo com função de aptidão bidimensional (interpolação). As soluções nos extremos e no meio dessa fronteira são vistas na Figura 6.7. Aqui é possível observar o contorno melódico mais comportado da solução com melhor aptidão na dimensão de $f_2(x)$. Ao contrário das soluções no caso unidimensional, as melodias destacadas da fronteira ótima na Figura 6.6 se assemelham mais às cantigas de roda, se não como um todo, em determinados trechos.

Deve-se ainda levantar um último detalhe sobre esse teste: os pesos utilizados para interpolação dos modelos de Markov nas ordens 0 a 2. Como o algoritmo de maximização de expectativa descrito no Apêndice A não convergisse sempre para um mesmo valor a cada separação diferente dos conjuntos de treinamento e de estimação,

 medidas aumentam para que outras, com maior peso, possam diminuir, e.g., $F_1(x)$ e $F_9(x)$.

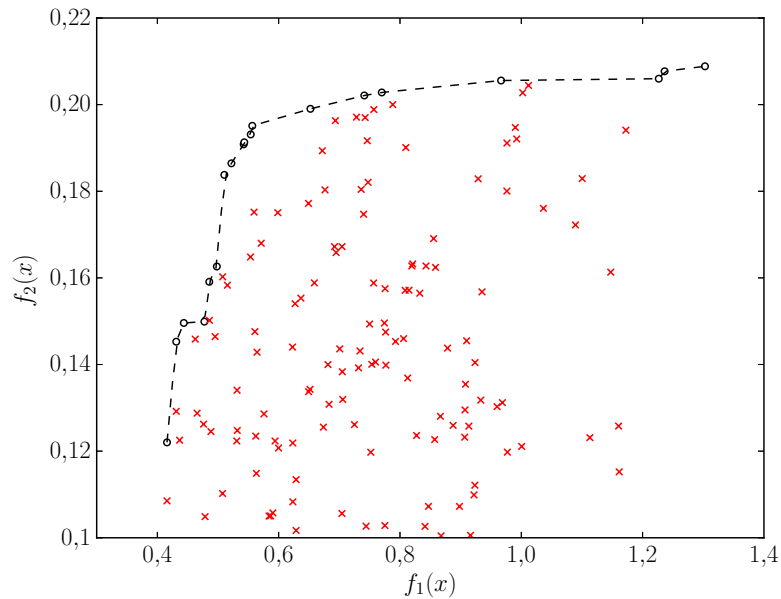


Figura 6.6: Fronteira de soluções não dominadas encontrado após a última geração. Os indivíduos marcados com “o” são não dominados e as soluções indicadas por “x” são dominadas por um ou mais indivíduos da fronteira.



(a) $f(x) = [0,416, 0,122]$



(b) $f(x) = [0,543, 0,191]$



(c) $f(x) = [1,303, 0,209]$

Figura 6.7: Indivíduos melódicos retirados de diferentes pontos da fronteira de soluções não dominadas. A fronteira pode ser vista na Figura 6.6 e cada indivíduo corresponde a um ponto: (a) no “extremo das heurísticas”; (b) no meio da fronteira; e (c) no “extremo markoviano”.

um conjunto de pesos foi escolhido arbitrariamente, levando-se em consideração os valores observados para diferentes configurações neste algoritmo. O vetor de pesos escolhido foi, para o caso melódico, $[0,5, 0,35, 0,15]$, da menor para a maior ordens. Dois fatos são curiosos nessa escolha e em sua necessidade: (1) o baixo peso atribuído ao modelo de ordem 2 indica a baixa confiabilidade no modelo interpolado; (2) o algoritmo de maximização de expectativa não conseguiu determinar unicamente a solução a esse problema de interpolação devido ao pequeno número de dados disponíveis. Possivelmente, se os pesos fossem calculados como função dos contextos ou de suas contagens, o menor número de graus de liberdade permitiria encontrar um

Tabela 6.4: Valor médio e de desvio das cantigas do *subcorpus* SC₆ após análise pelo conjunto de heurísticas do teste rítmico.

Medida	Nome	Média e Desvio	Desvio relativo
11	Densidade de notas	0,997 ± 0,007	0,73%
13	Variedade rítmica	0,518 ± 0,156	30,06%
14	Extensão de ritmos	0,452 ± 0,193	42,76%
15	Síncopes	0,003 ± 0,008	244,94%
17	Repetição de valor rítmico	0,564 ± 0,161	28,57%
20	Padrões rítmicos (3 notas)	0,335 ± 0,143	42,53%
21	Padrões rítmicos (4 notas)	0,107 ± 0,174	161,73%

conjunto único de pesos ou ainda um conjunto com menor variação que a observada nos cálculos efetuados para a base de dados.

6.2 Composição de Ritmos

A segunda tarefa atacada pelo algoritmo genético foi a composição de ritmos ao estilo das cantigas de roda. Para tanto, um conjunto de informações foi levantado a partir das sequências rítmicas do *subcorpus* SC₆, que servem como objetivo nessa segunda abordagem. As principais diferenças entre o teste de composição rítmica e o teste de composição melódica são: (1) a representação das sequências; (2) o modelo auxiliar para o problema multiobjetivo.

Neste teste, o ataque de uma nota, a sustentação da nota ou pausa anterior e o silêncio representam o conjunto de informações relevantes. Assim, codificação proposta para as sequências rítmicas não carrega mais a informação de *pitch*. Assume-se que todas as notas tem o mesmo *pitch* C4, que é codificado pelo número 5. O número 26 passa a ser utilizado para a criação de padrões rítmicos com duração maior que a do *quantum* e o número 25 ainda representa as pausas.

6.2.1 Caso 1: Função de aptidão unidimensional

Para o problema com um único objetivo, é preciso reescolher o conjunto de heurísticas que vão ser combinadas na função de aptidão. Agora, as heurísticas referentes ao *pitch*, à tonalidade ou ao contorno não são aplicáveis. Tampouco podem ser consideradas as medidas de padrões melódicos. O conjunto de heurísticas proposto para este teste, bem como os valores de média e desvio para o *subcorpus* em questão se encontram listados na Tabela 6.4.

Os indivíduos são novamente inicializados de maneira aleatória — pausas ($\frac{1}{30}$ de probabilidade), ataques ($\frac{10}{30}$) e sustentações ($\frac{19}{30}$) — e a mutação segue essa mesma

Tabela 6.5: Configuração do teste de composição de ritmos com função de aptidão bidimensional. Detalhe dos elementos da gramática gerativa.

Gramática de Kohonen	
Representação:	relativa
Contexto máximo:	3
Peso das regras:	[0,2, 0,3, 0,5]

distribuição. Os comprimentos de cada sequência estão limitados superior e inferiormente pelo número de compassos vistos no *subcorpus* (de 8 a 24 compassos). Os demais parâmetros do algoritmo genético são mantidos, com relação ao teste anterior no caso do problema com um objetivo a ser minimizado.

6.2.2 Caso 2: Função de aptidão bidimensional

Para o problema multiobjetivo de composição de ritmos, uma gramática de Kohonen foi escolhida para representar a função de aptidão $f_2(x)$. O restante dos parâmetros do algoritmo genético é feito igual ao caso 2 da composição de melodias, com os operadores de seleção que se baseiam em aptidão dando lugar a operadores calcados em dominância e *crowding distance*.

De modo similar ao que é feito no trabalho de Sheikholharam e Teshnehlab [90], o *subcorpus* é avaliado e suas regras determinísticas são extraídas pela gramática até um determinado nível do contexto⁸. Nesta pesquisa, um contexto de 3 símbolos foi utilizado como valor limite. A gramática pode, assim, ser utilizada para avaliar os indivíduos de uma população de acordo com a proporção de regras que este indivíduo segue corretamente. Regras com maior contexto devem receber um peso maior, porque teoricamente são mais significativas para o modelo. A representação utilizada para essa análise sintática é, mais uma vez, a representação relativa. Dessa vez, contudo, cada símbolo da gramática é uma dupla $(I_{i,i+1}, D_{i,i+1})$, conforme definido no Capítulo 5. A Figura 6.1 explica esse esquema de codificação das sequências rítmicas. Novamente, a segunda dimensão da função $\mathbf{f}(x)$ é um objetivo que deve ser maximizado, i.e., quanto maior a proporção das regras que um indivíduo cumpre, maior sua qualidade e, possivelmente, seu grau de pertencimento ao estilo da base.

A Tabela 6.5 sintetiza o conjunto de parâmetros da gramática de Kohonen utilizada no teste rítmico.

⁸Os pesquisadores em [90] definem como “contexto” o lado esquerdo das regras gramaticais e chamam “padrão” a regra completa. Já Kohonen, em [68], admite como básica uma regra $A \rightarrow B$, com $|A| = |B| = 1$, de forma que, para ele, o contexto é tudo aquilo adicionado a uma regra deste tipo. Então, para tornar a comparação entre a gramática e o modelo de Markov mais imediata, a primeira definição será utilizada nesse trabalho, i.e., o contexto é equivalente à ordem em uma cadeia de Markov.

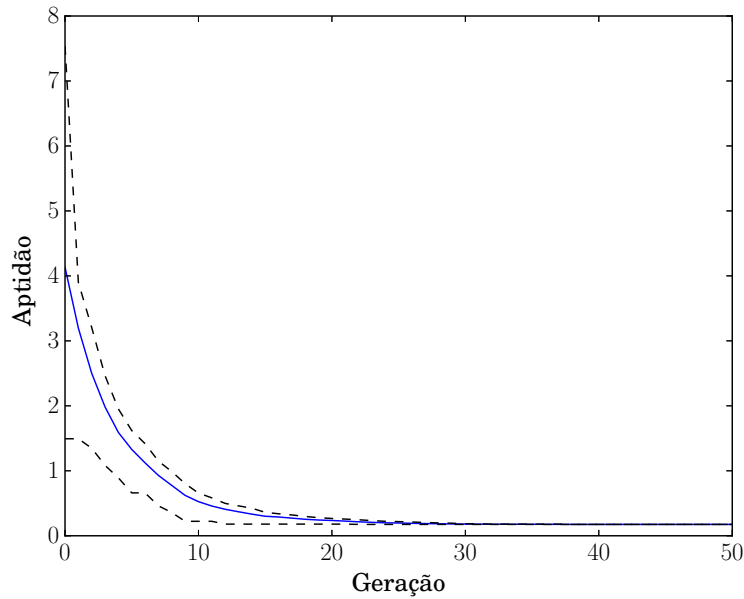


Figura 6.8: Curva de desempenho típica na composição de ritmos com função de aptidão unidimensional. O gráfico apresenta o valor médio de aptidão em cada geração, além dos valores máximo e mínimo (tracejados).

6.2.3 Resultados

As Figuras 6.8 e 6.9 apresentam as curvas de evolução da aptidão média da população e da média das distâncias de cada heurística ao objetivo (desconsiderando-se os desvios). A convergência da população no teste rítmico é mais lenta que no teste melódico, e percebe-se que todas as heurísticas tem seus valores iniciais reduzidos. A Figura 6.10 mostra a otimização dos dois objetivos da função de aptidão. Note-se que as curvas de desempenho se tornaram mais ruidosas por efeito da competição entre os dois objetivos.

De maneira geral, a produção rítmica do algoritmo, tanto no caso unidimensional como no caso bidimensional, é insatisfatória. As sequências são ritmicamente inconsistentes e muito “rápidas”, porque nelas figuram muitas notas cuja duração é a do *quantum*. É importante perceber que essa não é uma característica da base como um todo (cf. Figura 5.3) ou mesmo do *subcorpus* SC_6 , no qual as poucas aparições de semicolcheias geralmente compõem síncopes (e.g., cantiga 168 — “Samba-le-lê”). Em verdade, nenhuma das dimensões consegue agir direta e explicitamente no sentido da resolução desse problema, mas outras heurísticas poderiam ser implementadas para esse fim (cf. Seção 7.1). Exceto em alguns casos, as sequências rítmicas não apresentam nada que se assemelhe a uma estrutura de frases e também não terminam em notas mais longas, como é recorrente nesse *subcorpus*. As Figuras 6.11 e 6.12 apresentam algumas soluções retornadas pelo algoritmo nos casos das funções de aptidão unidimensional e bidimensional, respectivamente.

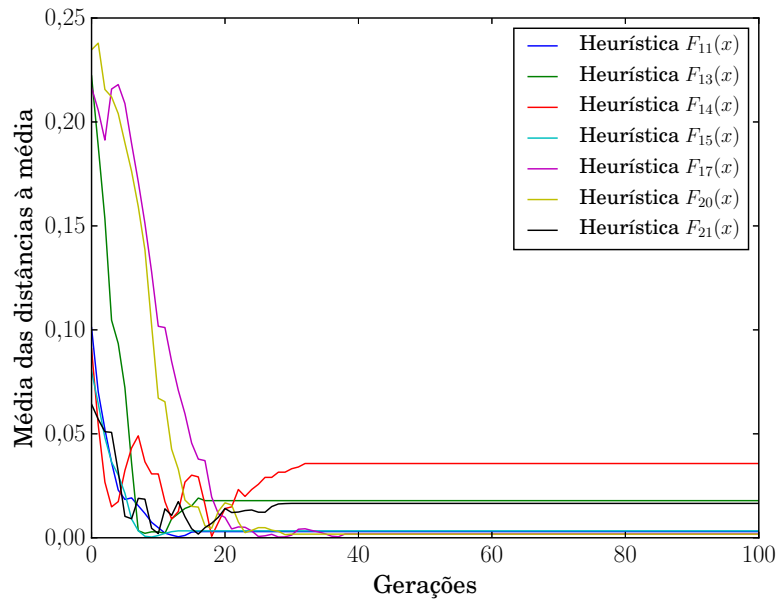


Figura 6.9: Média das distâncias de cada medida à média esperada no teste rítmico. As distâncias das heurísticas às respectivas médias passam por uma minimização, ainda que não atinjam o mínimo global.

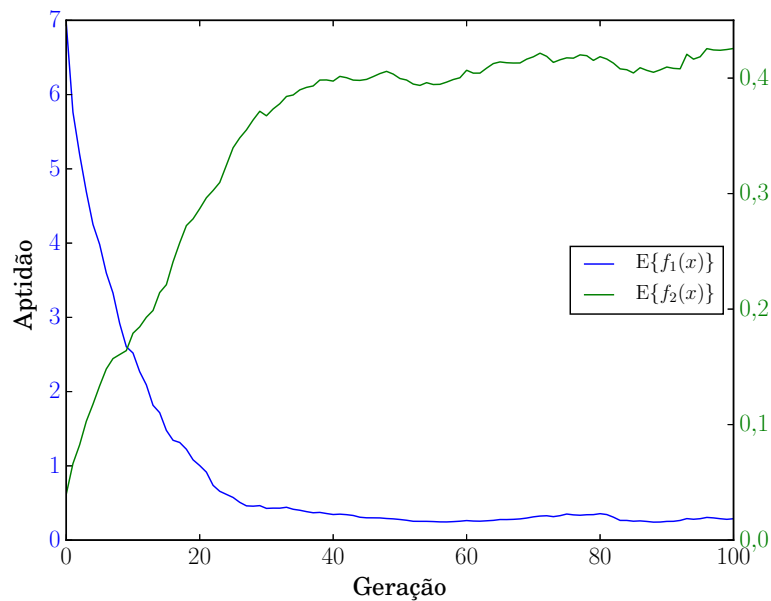


Figura 6.10: Curvas de desempenho típicas na composição de ritmos com função de aptidão bidimensional.

6.3 Composição de Melodias e Ritmos

A tarefa final do algoritmo compositor é o problema de composição completo, i.e., composição simultânea de melodias e ritmos. Esse problema representa a combinação dos dois problemas anteriores, principalmente no tocante à representação. Os genes dos cromossomos do algoritmo genético agora podem assumir qualquer valor em $[0, 26]$ e o comprimento das sequências pode variar entre 4 e 25 compassos,

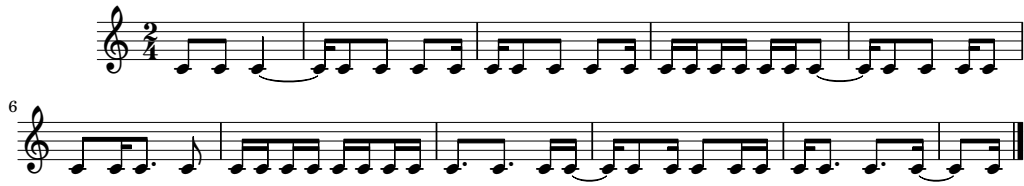
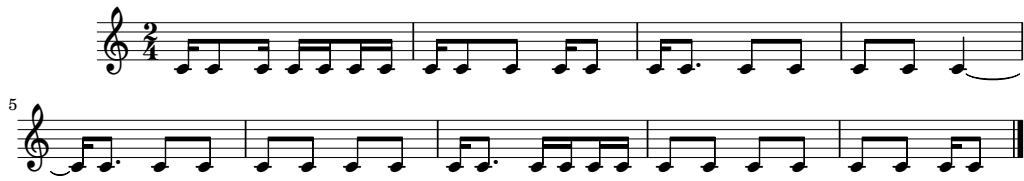
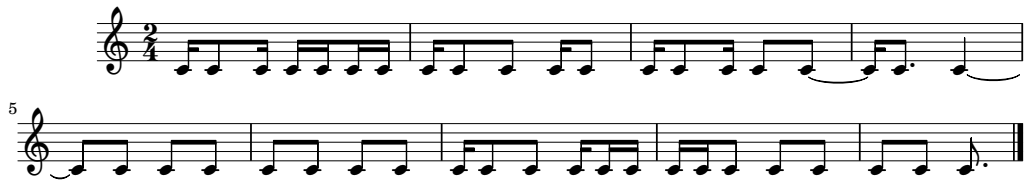


Figura 6.11: Sequência rítmica gerada somente com as heurísticas. Esse indivíduo foi o melhor avaliado ao final das gerações ($f_1(x) = 0,175$)



(a) $f(x) = [0,163, 0,303]$



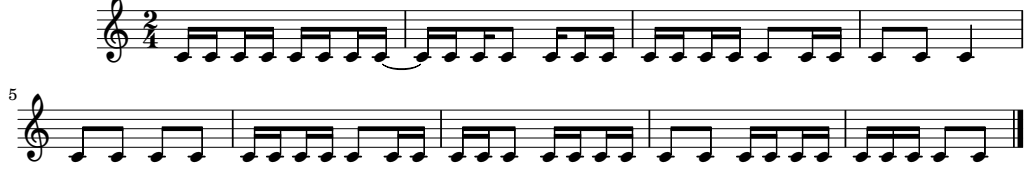
(b) $f(x) = [0,172, 0,378]$



(c) $f(x) = [0,206, 0,475]$



(d) $f(x) = [0,432, 0,496]$



(e) $f(x) = [0,490, 0,496]$

Figura 6.12: Indivíduos rítmicos retirados de diferentes pontos da fronteira de soluções não dominadas. Cada indivíduo corresponde a um ponto: (a) e (b) no “extremo das heurísticas”; (c) no meio da fronteira; e (d) e (e) no “extremo gramatical”.

Tabela 6.6: Valor médio e de desvio das cantigas do *subcorpus* SC₂ após análise pelo conjunto de heurísticas do teste completo.

Medida	Nome	Média e Desvio	Desvio relativo
1	Variedade de <i>pitch</i>	0,177 ± 0,076	42,7%
2	Extensão de <i>pitch</i>	0,449 ± 0,083	18,5%
3	Centralidade tonal	0,455 ± 0,124	27,3%
4	Notas fora de escala	0,000 ± 0,000	0,0%
5	Intervalos dissonantes	0,005 ± 0,015	287,8%
6	Direção do contorno	0,486 ± 0,039	8,0%
7	Estabilidade do contorno	0,380 ± 0,120	31,6%
8	Movimento gradual	0,412 ± 0,150	36,4%
9	Retorno de salto	0,152 ± 0,344	225,9%
10	Intensidade do clímax	0,597 ± 0,345	57,7%
11	Densidade de notas	0,977 ± 0,040	4,1%
13	Variedade rítmica	0,435 ± 0,132	30,4%
14	Extensão de ritmos	0,366 ± 0,164	44,9%
15	Síncopes	0,025 ± 0,052	210,0%
16	Repetição de <i>pitch</i>	0,288 ± 0,136	47,4%
17	Repetição de valor rítmico	0,654 ± 0,190	29,0%
18	Padrões de <i>pitch</i> (3 notas)	0,013 ± 0,030	234,6%
19	Padrões de <i>pitch</i> (4 notas)	0,016 ± 0,056	350,3%
20	Padrões rítmicos (3 notas)	0,471 ± 0,180	38,2%
21	Padrões rítmicos (4 notas)	0,300 ± 0,235	78,5%

conforme as cantigas do *subcorpus* SC₂, que serve como referência para este último teste. Não há muitas mudanças no funcionamento do algoritmo nos casos unidimensional e bidimensional da função de aptidão. A codificação utilizada para os indivíduos no cálculo da função $f_2(x)$ ainda é relativa, mas agora considera tanto a relação entre os *itches* como a relação entre as durações (cf. Figura 6.1). Os modelos de Markov e gramatical foram treinados e preparados para operar na segunda dimensão da função multiobjetivo. A Tabela 6.6 resume as estatísticas do *subcorpus* que servem como objetivo em $f_1(x)$.

6.3.1 Resultados

No problema de otimização com uma função de aptidão unidimensional, novamente observou-se um conjunto de resultados insatisfatórios. Embora musicalmente interessantes, as composições geradas pelo algoritmo por vezes apresentam grandes saltos ascendentes e descendentes. Além disso, conforme esperado, não foi obser-

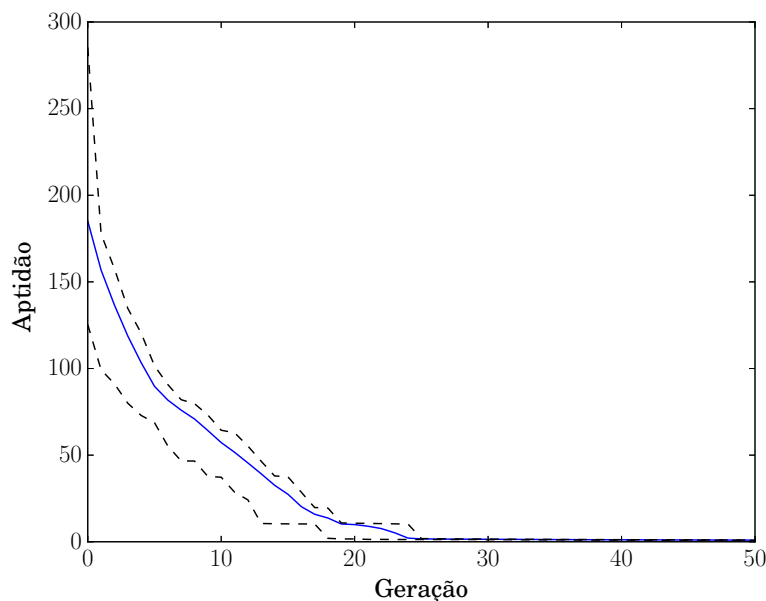


Figura 6.13: Curva de desempenho típica na composição de melodias ritmos com função de aptidão unidimensional. O gráfico apresenta o valor médio de aptidão em cada geração, além dos valores máximo e mínimo (tracejados).



Figura 6.14: Composição completa gerada somente com as heurísticas. Esse foi o melhor indivíduo ao final de 100 gerações ($f_1(x) = 0,652$)

vada uma regularidade na formação de frases ou na terminação das músicas geradas. A qualidade de alguns trechos e de algumas composições é admirável, dada a pequena quantidade de conhecimento musical que foi codificada no sistema como um todo. A Figura 6.13 apresenta a curva de desempenho do algoritmo neste teste, que é similar à dos outros testes realizados. Assim como no caso melódico, a evolução particular de cada uma das medidas é diferente da curva de desempenho total, i.e., nem todas puderam ser minimizadas. As heurísticas que perderam essa “competição” de minimização variaram um pouco entre as diferentes rodadas do algoritmo; mas as mais frequentes são as heurísticas $F_{18}(x)$, $F_{19}(x)$, $F_{20}(x)$ e $F_{21}(x)$ (padrões melódicos e rítmicos). A Figura 6.14 apresenta o melhor indivíduo em uma das rodadas do algoritmo, que se configurou como um interessante resultado.

No problema multiobjetivo, primeiramente o modelo de Markov suavizado por interpolação foi utilizado no lugar da função $f_2(x)$. Os pesos foram otimizados através do algoritmo de maximização de expectativa e convergiram para os valores $[0,65, 0,30, 0,05]$, que novamente indicam a baixa confiabilidade do modelo de ordem 2 para a representação relativa das sequências musicais. Nas muitas diferentes tentativas de se rodar o algoritmo nessa configuração, em algumas das quais o número de gerações foi quintuplicado, foram raras as vezes em que a dimensão de $f_2(x)$ saiu



(a) $f(x) = [0,446, 0,019]$



(b) $f(x) = [0,439, 0,110]$

Figura 6.15: Composições completas geradas pelo algoritmo multiobjetivo em duas rodadas diferentes. Cada um dos indivíduos foi retirado da fronteira de indivíduos não dominados fornecida pelo algoritmo ao final de 100 gerações.

do valor 0,0. Esse problema do início de otimização tardia do modelo suavizado por interpolação já havia sido observado no teste melódico, mas agora assume uma proporção maior, porque a combinação dos valores de intervalo e de razão entre as durações aumenta enormemente o número de símbolos possíveis, i.e., o modelo é relativamente mais esparsa. A cardinalidade do espaço de estados é muito maior que o valor de 50^3 utilizado no modelo “soma um”. O passo seguinte foi, portanto, a substituição do modelo de Markov por uma gramática de Kohonen. Alguns resultados de rodadas diferentes do processo de otimização podem ser visualizados na Figura 6.15. Alguns resultados da aplicação das heurísticas podem ser observados também nesse conjunto de soluções, e.g., a falta de notas fora de escala e o contorno melódico que começa e termina pela mesma nota, entre outros.

Novamente, as composições que o algoritmo avalia como sendo as melhores tendem a ser apresentadas em poucos compassos. Esse problema do comprimento acontece apesar de todos os passos de “independência” do comprimento do cromossomo que foram realizados — desde o caráter normalizado das heurísticas de Towsey *et al.* até às próprias funções expressas pelos modelos markoviano e gramatical, que empregam a raiz L -ésima ou a divisão por L em suas formulações — e só pode ser causado pela própria natureza dos operadores de variação do algoritmo genético. Comparativamente aos exemplos do caso unidimensional, as soluções fornecidas em conjunção com a gramática de Kohonen soam mais bem comportadas. Esse efeito é mais evidente aqui do que nos testes melódico e rítmico. Ainda assim, as composições apresentam muitas notas com duração de um *quantum*, uma característica que não se percebe no *corpus*. Essas duas “deficiências” — composições curtas e notas de pequena duração —, trabalham juntas para “enganar” as heurísticas e a gramática de Kohonen.

Por fim, a Figura 6.16 apresenta uma comparação entre indivíduos de uma mesma rodada do algoritmo, mas de gerações diferentes. Dois aspectos são destacados nesse exemplo: (1) o estabelecimento de um tema que está presente nas três composições; (2) o desenvolvimento desse mesmo tema com as heurísticas e a gramática de Koho-



Figura 6.16: Composições completas retiradas de diferentes gerações em uma mesma rodada do algoritmo. Cada composição representa o melhor indivíduo nas seguintes gerações: (a) 10^a; (b) 50^a; e (c) 100^a.

nen. É nítido o aperfeiçoamento da composição com o passar das gerações. Contudo, o problema do tamanho prejudica a composição em 6.16c: alguns compassos a mais seriam necessários para completar a composição.

6.4 Sobre a seleção de sobreviventes

Nos casos unidimensional e bidimensional, a seleção de sobreviventes é feita sobre o conjunto $(\mu + \lambda)$ de pais e filhos. Esse tipo de seleção mantém os melhores indivíduos da população, independente de sua idade, e é conhecido por acelerar o processo de convergência dos algoritmos evolucionários. Essa convergência, sobretudo em problemas multimodais, pode ser prematura, i.e., o algoritmo converge rapidamente para um ótimo local.

Alguns testes foram realizados utilizando somente o conjunto de filhos para formação da nova população (abordagem generacional) para a composição de melodias com função de aptidão bidimensional. Dado um mesmo número máximo de gerações (100) utilizado no caso anterior, essa configuração do algoritmo genético apresentou, em geral:

- Convergência da média de $f_1(x)$ para um valor 5 vezes maior;
- Não convergência na dimensão de $f_2(x)$, que atinge um valor médio 10 vezes menor;
- Melhor indivíduo no extremo das heurísticas da fronteira não dominada com aptidão pelo menos 2 vezes maior nessa dimensão;
- Melhor indivíduo no extremo gramatical da fronteira não dominada com aptidão cerca de 3 vezes menor nessa dimensão.

Portanto, a seleção $(\mu + \lambda)$ parece ser a mais adequada para o problema em questão, e o mecanismo de *crowding distance* do NSGA-II age de forma a manter a diversidade do espaço de soluções, impedindo o efeito de convergência prematura. Auditivamente, as soluções também apresentam grande variação de *pitch* e muitas notas com duração de um *quantum*.

6.5 Arquivos

Os arquivos MIDI referentes aos exemplos apresentados neste capítulo podem ser encontrados no site <http://www.smt.ufrj.br/~lucas.maia/mestrado/exemplos>.

Capítulo 7

Conclusões

Esta dissertação apresentou um apanhado histórico-técnico do campo da Composição Algorítmica. Alguns paradigmas emprestados da grande área de Inteligência Artificial foram estudados desde suas origens e seguiu-se o caminho que trilharam nas aplicações musicais. Três foram esses formalismos — os modelos de Markov, as gramáticas gerativas e os algoritmos evolucionários —, destacados dentre os principais comumente utilizados na literatura para solução de problemas em composição.

Posteriormente, com a finalidade de ilustrar o problema clássico de imitação de estilo e motivar as pesquisas brasileiras feitas nesse campo, um algoritmo foi desenvolvido e apresentado em detalhe. Esse algoritmo objetivava, por meio da combinação dos diferentes formalismos estudados, compor peças simples que pudessem ser identificadas como cantigas de roda do folclore nacional, mais especificamente, associadas ao estilo carioca do bairro do Realengo.

Algumas dificuldades foram encontradas a montante do processo de desenvolvimento dessa técnica de composição, notadamente: a descrição de um *corpus* fiel à tradição oral desse estilo de música; e a formatação notacional deste *corpus* de forma universal. O aspecto multidisciplinar deste trabalho, cujos traços são encontrados em cada página, foi um desafio à parte.

O sistema de composição implementado baseia-se em um algoritmo genético com operadores de variação simples. Observou-se que o uso de medidas heurísticas como única forma de avaliação dos indivíduos candidatos à solução é insuficiente para caracterizar o estilo da base de dados. A combinação desse formalismo com as cadeias de Markov e as gramáticas gerativas certamente trouxe melhorias na qualidade das composições geradas¹. Algumas deficiências permanecem, contudo, entre elas: a tendência do algoritmo de retornar sequências musicais curtas; a ausência de

¹Em testes preliminares com somente cadeias de Markov ou somente gramáticas gerativas, os efeitos adversos indicados por Brooks *et al.* [16] foram notados com muita facilidade. O aumento da ordem em uma unidade era suficiente para transformar composições completamente aleatórias em “colagens” de pedaços do *corpus*. Isso pode estar relacionado com o pequeno tamanho da base utilizada.

restrições estruturais, i.e., que formem frases, motivos e temas no decorrer das composições; a ausência de uma base harmônica sobre a qual a melodia se desenvolva, que auxilia muito o processo de composição. Além disso, o número de parâmetros que o sistema proposto possui é enorme. Eles se estendem desde o domínio do algoritmo genético (probabilidades de mutação e de *crossover*, comprimento das sequências, representação etc.) até os modelos de Markov (ordem, tipo de suavização e seus próprios parâmetros, novamente a representação etc.) e gramatical (comprimento do contexto, forma das regras sintáticas etc.). Pode-se dizer que o problema de seleção dos parâmetros do algoritmo é, em si mesmo, um problema de otimização.

As composições geradas pelo algoritmo são musicalmente interessantes e, embora não possam ser consideradas “cantigas de roda” estritas, trechos seletos de cada resultado evocam este estilo musical. No mais, o objetivo da Composição Algorítmica é ajudar o ser humano a alcançar conhecimento do próprio processo criativo. Não visa a substituir o seu trabalho, mas a inspirá-lo e a clarificar os movimentos que operam no subconsciente. É um movimento de mão dupla, como diz Moorer [94]: as tentativas de simular as capacidades composicionais de um ser humano só serão bem sucedidas quando os modelos musicais utilizados pelo homem forem descritos e modelados. O experimento proposto neste trabalho se insere nesse contexto e, dessa forma, se manifesta como uma ferramenta útil com diversas possibilidades de extensão.

7.1 Trabalhos Futuros

Alguns caminhos podem ser identificados como possibilidades de trabalhos no futuro e estão descritos a seguir:

- O estudo detalhado de outros paradigmas da Composição Algorítmica que não foram contemplados na presente dissertação, e.g., redes neurais artificiais e autômatos celulares, entre outros.
- Uma análise do problema do tamanho das sequências ao longo do processo de convergência do algoritmo genético.
- Uma inicialização “esperta” que ajude na questão do desenvolvimento tardio da função de aptidão formada pelos modelos de Markov interpolados. (Uma inicialização que, no domínio relativo, se traduza em símbolos encontrados no modelo de ordem 0, por exemplo, pode antecipar a maximização dessa função.)
- Uma proposta de heurísticas que ataquem algumas das deficiências observadas nas composições geradas — o comprimento, a falta de estrutura frasal, o excesso de notas com duração de um *quantum* etc.

- O uso de operadores de variação musicalmente “inteligentes”, e.g., copiar um trecho do cromossomo para outro conjunto de genes no mesmo indivíduo, transpor os valores dos alelos etc. (Esse tipo de operador pode acelerar a convergência para soluções “boas” que apresente conteúdo temático.)
- Uma avaliação da correlação entre os esforços do modelo de Markov/da gramática gerativa e das heurísticas. (Provavelmente, a dimensão de $f_1(x)$ está sobrecarregada com algumas tarefas que o modelo em $f_2(x)$ já executa. Esse tipo de redundância pode diminuir o desempenho do algoritmo genético.)

Referências Bibliográficas

- [1] HILLER, JR., L. A., ISAACSON, L. M. *Experimental Music: Composition with an Electronic Computer*. New York, USA, McGraw-Hill, 1959.
- [2] MCCORMACK, J. “Grammar Based Music Composition”. In: Stocker, R., Jelinek, H., Durnota, B., et al. (Eds.), *Complex Systems 96: From Local Interactions to Global Phenomena*, IOS Press, pp. 320–336, Amsterdam, Nederland, 1996.
- [3] LEACH, J., FITCH, J. “Nature, Music, and Algorithmic Composition”, *Computer Music Journal*, v. 19, n. 2, pp. 23–33, fev. 1995.
- [4] OLSON, H. F. *Music, Physics and Engineering*. 2 ed. New York, USA, Dover Publications, 1967.
- [5] XENAKIS, I. *Formalized Music: Thought and Mathematics in Composition*. New York, USA, Pendragon Press, 1992.
- [6] JACOB, B. L. “Algorithmic Composition as a Model of Creativity”, *Organised Sound*, v. 1, n. 3, pp. 157–165, dez. 1996.
- [7] NIERHAUS, G. *Algorithmic Composition: Paradigms of Automated Music Generation*. Mörlenbach, Deutschland, Springer-Verlag, 2009.
- [8] MAIA, L. S. *Representação e Análise de Interpretações Expressivas — Um Estudo da Dinâmica e da Agógica*. Projeto de graduação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2013.
- [9] HEDGES, S. A. “Dice Music in the Eighteenth Century”, *Music & Letters*, v. 59, n. 2, pp. 180–187, abr. 1978.
- [10] TEMPERLEY, D. *The Cognition of Basic Musical Structures*. Cambridge, USA, The MIT Press, 2004.
- [11] FERNÁNDEZ, J. D., VICO, F. “AI Methods in Algorithmic Composition: A Comprehensive Survey”, *Journal of Artificial Intelligence Research*, v. 48, pp. 513–582, nov. 2013.

- [12] SHANNON, C. E. “A Mathematical Theory of Communication”, *The Bell System Technical Journal*, v. 27, n. 3, pp. 379–423, jul. 1948.
- [13] COVER, T. M., THOMAS, J. A. *Elements of Information Theory*. 2 ed. Hoboken, USA, John Wiley & Sons, 2006.
- [14] COHEN, J. E. “Information Theory and Music”, *Behavioral Science*, v. 7, n. 2, pp. 137–163, abr. 1962.
- [15] HUDSON, N. J. “Musical Beauty and Information Compression: Complex to the Ear but Simple to the Mind?” *BMC research notes*, v. 4, n. 9, jan. 2011.
- [16] BROOKS, F., HOPKINS, A., NEUMANN, P., et al. “An Experiment in Musical Composition”, *IRE Transactions on Electronic Computers*, v. EC-6, n. 3, pp. 175–182, set. 1957.
- [17] TURING, A. M. “Computing Machinery and Intelligence”, *Mind*, v. 59, n. 236, pp. 433–460, out. 1950.
- [18] PACHET, F. “The Continuator: Musical Interaction With Style”. In: *Proceedings of the 2002 International Computer Music Conference*, pp. 211–218, Göteborg, Sverige, set. 2002. International Computer Music Association.
- [19] COPE, D. “Computer Modeling of Musical Intelligence in EMI”, *Computer Music Journal*, v. 16, n. 2, pp. 69–83, fev. 1992.
- [20] HILLER, JR., L. A., ISAACSON, L. M. “Musical Composition with a High Speed Digital Computer”. In: *Audio Engineering Society Convention 9*, New York, USA, out. 1957. Audio Engineering Society.
- [21] BASHARIN, G. P., LANGVILLE, A. N., NAUMOV, V. A. “The Life and Work of A. A. Markov”, *Linear Algebra and its Applications*, v. 386, pp. 3–26, jul. 2004.
- [22] SUNG, B. *Ars Conjectandi: Translations from James Bernoulli*. In: Technical Report 2, Harvard University, Department of Statistics, Cambridge, USA, 1966.
- [23] SENETA, E. “Markov and the Birth of Chain Dependence Theory”, *International Statistical Review*, v. 64, n. 3, pp. 255–263, dez. 1996.
- [24] FELLER, W. *An Introduction to Probability Theory and Its Applications*, v. I. 3 ed. New York, USA, John Wiley & Sons, 1968.

- [25] PEEBLES JR., P. Z. *Probability, random variables, and random signal principles*. 2 ed. New York, USA, McGraw-Hill, 1987.
- [26] OLSON, H. F., BELAR, H. “Aid to Music Composition Employing a Random Probability System”, *The Journal of the Acoustical Society of America*, v. 33, n. 9, pp. 1163–1170, set. 1961.
- [27] AMES, C. “The Markov Process as a Compositional Model: A Survey and Tutorial”, *Leonardo*, v. 22, n. 2, pp. 175–187, 1989.
- [28] PONSFORD, D., WIGGINS, G., MELLISH, C. “Statistical learning of harmonic movement”, *Journal of New Music Research*, v. 28, n. 2, pp. 150–177, jun. 1999.
- [29] RANDEL, D. M. *The Harvard Dictionary of Music*. 4 ed. Cambridge, USA, Belknap Press of Harvard University Press, 2003.
- [30] AMES, C. “Tutorial on Automated Composition”. In: *Proceedings of the 1987 International Computer Music Conference*, pp. 1–8, Urbana, USA, ago. 1987. International Computer Music Association.
- [31] CONKLIN, D., WITTEN, I. H. “Multiple Viewpoint Systems for Music Prediction”, *Journal of New Music Research*, v. 24, n. 1, pp. 51–73, mar. 1995.
- [32] TRIVIÑO RODRIGUEZ, J., MORALES-BUENO, R. “Using Multiattribute Prediction Suffix Graphs to Predict and Generate Music”, *Computer Music Journal*, v. 25, n. 3, pp. 62–79, set. 2001.
- [33] LO, M. Y., LUCAS, S. M. “Evolving Musical Sequences with N-Gram Based Trainable Fitness Functions”. In: *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pp. 601–608, Vancouver, Canada, jul. 2006. IEEE.
- [34] FARBOOD, M., SCHONER, B. “Analysis and Synthesis of Palestrina-Style Counterpoint Using Markov Chains”. In: *Proceedings of the 2001 International Computer Music Conference*, pp. 471–474, Havana, Cuba, set. 2001. International Computer Music Association.
- [35] CHEN, S. F., GOODMAN, J. *An Empirical Study of Smoothing Techniques for Language Modeling*. Technical Report TR-10-98, Harvard University, Computer Science Group, Cambridge, USA, 1998.

- [36] ALLAN, M. *Harmonising Chorales in the Style of Johann Sebastian Bach*. Master's thesis, University of Edinburgh, School of Informatics, Edinburgh, Scotland, 2002.
- [37] SCHOLZ, R., VINCENT, E., BIMBOT, F. "Robust Modeling of Musical Chord Sequences Using Probabilistic N-grams". In: *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 53–56, Taipei, Taiwan, abr. 2009. IEEE.
- [38] CHOMSKY, N. *Syntactic Structures*. 2 ed. Berlin, Deutschland, Mouton de Gruyter, 2002.
- [39] DE SAUSSURE, F. *Cours de Linguistique Générale*. 5 ed. Paris, France, Payot et Rivages, 1995.
- [40] KOERNER, E. *Ferdinand de Saussure: Origin and Development of his Linguistic Theory in Western Studies of Language*. Doctor's thesis, Simon Fraser University, Vancouver, Canada, 1971.
- [41] FALK, J. S. "Saussure and American Linguists". In: Sanders, C. (Ed.), *The Cambridge Companion to Saussure*, Cambridge University Press, cap. 7, pp. 107–123, Cambridge, England, 2004.
- [42] BLOOMFIELD, L. *Language*. New York, USA, Henry Holt and Company, 1933.
- [43] BLOOMFIELD, L. "A Set of Postulates for the Science of Language", *Language*, v. 2, n. 3, pp. 153–164, set. 1926.
- [44] SEARLE, J. R. "Chomsky's Revolution in Linguistics", *The New York Review of Books*, v. 18, n. 12, pp. 12–29, jun. 1972.
- [45] CHOMSKY, N. *Aspects of the Theory of Syntax*. Cambridge, USA, The MIT Press, 1965.
- [46] CAMACHO, R. G. "O Papel do Contexto Social na Teoria Linguística", *Alfa*, v. 38, pp. 19–36, 1994.
- [47] PULLUM, G. K. "Creation Myths of Generative Grammar and the Mathematics of Syntactic Structures". In: Ebert, C., Jäger, G., Michaelis, J. (Eds.), *The Mathematics of Language*, v. 6149, *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 238–254, Los Angeles, USA, 2010.
- [48] HOPCROFT, J. E., ULLMAN, J. D. *Formal Languages and Their Relation to Automata*. Boston, USA, Addison-Wesley Publishing Company, 1969.

- [49] CHOMSKY, N. “Three Models for the Description of Language”, *IRE Transactions on Information Theory*, v. 2, n. 3, pp. 113–124, set. 1956.
- [50] ADORNO, T. W., PADDISON, M. “On the Problem of Musical Analysis”, *Music Analysis*, v. 1, n. 2, pp. 169–187, jul. 1982.
- [51] SCHENKER, H. *Der Freie Satz*, v. 3, *Neue musikalische Theorien und Phantasien*. 2 ed. Wien, Austria, Universal-Edition, 1935.
- [52] KIRLIN, P. B., UTGOFF, P. E. “A Framework for Automated Schenkerian Analysis”. In: *Proceedings of the 9th International Conference on Music Information Retrieval*, pp. 363–368, Philadelphia, USA, set. 2008. International Society for Music Information Retrieval.
- [53] DE BARROS, G. S., GERLING, C. C. “Análise Schenkeriana: Interpretação e Crítica”. In: Budasz, R. (Ed.), *Pesquisa em Música no Brasil: Métodos, Domínios e Perspectivas*, v. 1, *Pesquisa em Música no Brasil*, ANPPOM, pp. 87–121, Goiânia, Brasil, 2009.
- [54] LERDAHL, F., JACKENDOFF, R. S. *A Generative Theory of Tonal Music*. 1 ed. Cambridge, USA, The MIT Press, 1983.
- [55] MELLO, M. *Reflexões sobre Linguística e Cognição Musical*. Dissertação de mestrado, Universidade Estadual de Campinas, Campinas, Brasil, 2003.
- [56] DELIÈGE, I. “Grouping Conditions in Listening to Music: An Approach to Lerdahl & Jackendoff’s Grouping Preference Rules”, *Music Perception*, v. 4, n. 4, pp. 325–360, jul. 1987.
- [57] LERDAHL, F., JACKENDOFF, R. S. “An Overview of Hierarchical Structure in Music”, *Music Perception*, v. 1, n. 2, pp. 229–252, dez. 1983.
- [58] TEMPERLEY, D., SLEATOR, D. “Modeling Meter and Harmony: A Preference-Rule Approach”, *Computer Music Journal*, v. 23, n. 1, pp. 10–27, mar. 1999.
- [59] HAMANAKA, M., HIRATA, K., TOJO, S. “Implementing ‘A Generative Theory of Tonal Music’”, *Journal of New Music Research*, v. 35, n. 4, pp. 249–277, dez. 2006.
- [60] TEMPERLEY, D. *Music and Probability*. Cambridge, USA, The MIT Press, 2010.
- [61] BOD, R. “A Unified Model of Structural Organization in Language and Music”, *Journal of Artificial Intelligence Research*, v. 17, pp. 289–308, fev. 2002.

- [62] MARCUS, M. P., MARCINKIEWICZ, M. A., SANTORINI, B. “Building a Large Annotated Corpus of English: The Penn Treebank”, *Computational Linguistics*, v. 19, n. 2, pp. 313–330, jun. 1993.
- [63] SCHAFFRATH, H. “The Essen Folksong Collection in the Humdrum Kern Format”. 1995.
- [64] STEEDMAN, M. J. “A Generative Grammar for Jazz Chord Sequences”, *Music Perception*, v. 2, n. 1, pp. 52–77, out. 1984.
- [65] JOHNSON-LAIRD, P. N. “Jazz Improvisation: A Theory at the Computational Level”. In: Howell, P., West, R., Cross, I. (Eds.), *Representing Musical Structure*, v. 5, *Cognitive Science Series*, 1 ed., Academic Press, cap. 9, pp. 291–325, London, UK, 1991.
- [66] ROADS, C. “Composing Grammars”. In: *Proceedings of the 1977 International Computer Music Conference*. International Computer Music Association, 1977.
- [67] SUNDBERG, J., LINDBLOM, B. “Generative Theories in Language and Music Descriptions”, *Cognition*, v. 4, n. 1, pp. 99–122, mar. 1976.
- [68] KOHONEN, T. “A Self-Learning Musical Grammar, or ‘Associative Memory of the Second Kind’”. In: *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–5. IJCNN, jun. 1989.
- [69] CRUZ-ALCÁZAR, P. P., VIDAL-RUIZ, E. “Learning Regular Grammars to Model Musical Style: Comparing Different Coding Schemes”. In: *Proceedings of the International Colloquium on Grammatical Inference*, pp. 211–222, Ames, USA, jul. 1998. ICGI.
- [70] DARWIN, C. *The Origin of Species*. 6 ed. London, UK, John Murray, 1872.
- [71] LAMARCK, J.-B. P. A. *Philosophie Zoologique, ou Exposition des Considérations Relatives à l’Histoire Naturelle des Animaux*. Paris, France, Muséum d’Histoire Naturelle, 1809.
- [72] BOWLER, P. J. *The Eclipse of Darwinism: Anti-Darwinian Evolution Theories in the Decades Around 1900*. Baltimore, USA, The John Hopkins University Press, 1983.
- [73] MALTHUS, T. R. *An Essay on the Principle of Population*. 6 ed. London, UK, John Murray, 1826.

- [74] EIBEN, A. E., SMITH, J. E. *Introduction to Evolutionary Computing*. Natural Computing Series. Berlin, Deutschland, Springer, 2003.
- [75] HINE, R., MARTIN, E. (Eds.) *A Dictionary of Biology*. 7 ed. Oxford, UK, Oxford University Press, 2015.
- [76] MIRANDA, E. R., BILES, J. A. (Eds.) *Evolutionary Computer Music*. London, UK, Springer-Verlag, 2007.
- [77] WOLPERT, D. H., MACREADY, W. G. “No Free Lunch Theorems for Optimization”, *IEEE Transactions on Evolutionary Computation*, v. 1, n. 1, pp. 67–82, abr. 1997.
- [78] BOWLER, P. J. *Evolution: the History of an Idea*. 2 ed. Berkeley e Los Angeles, USA, University of California Press, 1989.
- [79] REEVES, C. R. “Fitness Landscapes”. In: Burke, E. K., Kendall, G. (Eds.), *Search Methodologies*, 2 ed., Springer, cap. 22, pp. 681–705, New York, USA, 2014.
- [80] MAYR, E. *What Makes Biology Unique? Considerations on the Autonomy of a Scientific Discipline*. New York, USA, Cambridge University Press, 2004.
- [81] JONG, K. A. D. “Genetic Algorithms are NOT Function Optimizers”. In: Whitley, L. D. (Ed.), *Foundations of Genetic Algorithms*, v. 2, Morgan Kaufmann, cap. 1, pp. 5–18, San Mateo, USA, 1993.
- [82] HOLLAND, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Ann Arbor, USA, University of Michigan Press, 1975.
- [83] HORNER, A., GOLDBERG, D. E. “Genetic Algorithms and Computer-Assisted Music Composition”. In: *Proceedings of the 1991 International Computer Music Conference*, pp. 479–482, Montreal, Canada, out. 1991. International Computer Music Association.
- [84] BILES, J. A. “GenJam: A Genetic Algorithm for Generating Jazz Solos”. In: *Proceedings of the 1994 International Computer Music Conference*, pp. 131–137, Aarhus, Denmark, set. 1994. International Computer Music Association.
- [85] TOWSEY, M., BROWN, A., WRIGHT, S., et al. “Towards Melodic Extension Using Genetic Algorithms”, *Journal of Educational Technology and Society*, v. 4, n. 2, pp. 54–65, abr. 2001.

- [86] ALFONSECA, M., CÉBRIAN, M., ORTEGA, A. “Evolving Computer-Generated Music by Means of the Normalized Compression Distance”. In: *Proceedings of the 5th WSEAS International Conference on Simulation, Modelling and Optimization*. World Scientific and Engineering Academy and Society, ago. 2005.
- [87] PAPADOPOULOS, G., WIGGINS, G. “A Genetic Algorithm for the Generation of Jazz Melodies”. In: *Proceedings of the Finnish Conference on Artificial Intelligence (STeP)*. Finnish Artificial Intelligence Society, set. 1998.
- [88] JOHNSON, M. D., TAURITZ, D. R., WILKERSON, R. W. “Evolutionary Computation Applied to Melody Generation”. In: *Proceedings of the Artificial Neural Networks in Engineering Conference*. ASME Press, nov. 2004.
- [89] PAZ, E. A. *500 Canções Brasileiras*. 2 ed. Brasília, Brasil, MusiMed, 2010.
- [90] SHEIKHOLHARAM, P., TESHNEHLAB, M. “Music Composition Using Combination of Genetic Algorithms and Kohonen Grammar”. In: *Proceedings of the 1st International Symposium on Computational Intelligence and Design*. IEEE, out. 2008.
- [91] CUTHBERT, M. S., ARIZA, C. “music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data”. In: *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pp. 637–642. International Society for Music Information Retrieval, ago. 2010.
- [92] FORTIN, F.-A., DE RAINVILLE, F.-M., GARDNER, M.-A., et al. “DEAP: Evolutionary Algorithms Made Easy”, *Journal of Machine Learning Research*, v. 13, pp. 2171–2175, jul. 2012.
- [93] DEB, K., PRATAP, A., AGARWAL, S., et al. “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”, *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, pp. 182–197, abr. 2002.
- [94] MOORER, J. A. “Music and Computer Composition”, *Communications of the ACM*, v. 15, n. 2, pp. 104–113, fev. 1972.
- [95] BAHL, L. R., JELINEK, F., MERCER, R. L. “A Maximum Likelihood Approach to Continuous Speech Recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. PAMI-5, n. 2, pp. 179–190, mar. 1983.

- [96] FRANZ, A. *Automatic Ambiguity Resolution in Natural Language Processing: An Empirical Approach*, v. 1171, *Lecture Notes in Artificial Intelligence*. Berlin, Deutschland, Springer, 1996.

Apêndice A

Conjunto de Medidas para Função de Aptidão

Neste apêndice é encontrado o conjunto de medidas proposto por Towsey *et al.* em [85], separadas de acordo com o tipo de características que analisam. O símbolo # deve ser lido como “o número de”, i.e., representa a quantidade de eventos de um dado tipo [85]. Os intervalos são medidos em semitons e um *quantum* temporal é a menor duração observada nas melodias, definida como a duração de uma semicolcheia [85]. Exceto quando indicado, as pausas são sempre consideradas nas medidas.

A Seção A.6 foi desenvolvida pelo próprio autor deste trabalho e apresenta uma modificação nas heurísticas 18 a 21.

A.1 Medidas de *Pitch*

Variedade de *pitch*

A razão entre *pitches* distintos e notas. Essa é uma medida da diversidade de classes de *pitch* utilizadas na produção da melodia.

$$F_1(x) = \frac{\#pitches \text{ distintos}}{\#notas}. \quad (\text{A.1})$$

Extensão de *pitch*

A extensão de *pitch* dividida pelo maior extensão aceitável. Essa medida indica a amplitude do contorno melódico.

$$F_2(x) = \frac{\text{maior } pitch - \text{menor } pitch}{24}. \quad (\text{A.2})$$

Tabela A.1: Graus de dissonância associados a cada tipo de intervalo (medido em semitons). Adaptado de [85].

Intervalo	Grau de dissonância
0–5, 7–9, 12	0,0
10	0,5
6, 11, ≥ 13	1,0

A maior extensão aceitável foi definida como duas oitavas. Caso o numerador exceda esse valor, então $F_2(x) = 1,0$.

A.2 Medidas de Tonalidade

Centralidade tonal

A proporção de *quanta* em que o *pitch* é primário, i.e., é tônica ou dominante¹. Essa medida fornece uma indicação de quão forte é o “sentido de tonalidade” da peça.

$$F_3(x) = \frac{\#quanta \text{ com } pitch \text{ primário}}{\#quanta}. \quad (\text{A.3})$$

Notas fora de escala

A proporção de *quanta* em que o *pitch* não faz parte da escala (maior ou menor, conforme a tonalidade da peça). Essa medida fornece uma indicação de quão “fortemente tonal” é a peça.

$$F_4(x) = \frac{\#quanta \text{ com } pitch \text{ fora da escala}}{\#quanta}. \quad (\text{A.4})$$

Intervalos dissonantes

A fração de intervalos dissonantes. Um grau de dissonância é associado a cada intervalo, conforme indicado na Tabela A.1. Pausas são ignoradas nessa medida.

$$F_5(x) = \frac{\sum_{\text{intervalo}} \text{grau de dissonância}}{\#\text{intervalos}}. \quad (\text{A.5})$$

Observação: Towsey *et al.* definem $\#\text{intervalos} = \#\text{notas} - 1$ [85].

¹Towsey *et al.* se referem ao primeiro ou quinto graus da escala, $\hat{1}$ ou $\hat{5}$.

A.3 Medidas de Contorno

Direção do contorno

A soma de todos os intervalos ascendentes dividida pelo soma de todos os intervalos (em valor absoluto). Essa medida avalia a tendência geral da melodia de subir ou descer. Uma melodia que se inicie e termine na mesma nota é avaliada com $F_6(x) = 0,5$. Um valor mais alto indica uma tendência ascendente, enquanto um valor inferior significa um contorno descendente. Pausas são ignoradas nessa medida.

$$F_6(x) = \frac{\sum_{\text{intervalo} > 0} \text{intervalo}}{\sum |\text{intervalo}|}. \quad (\text{A.6})$$

Estabilidade do contorno

A proporção de intervalos para os quais o intervalo seguinte se dá na mesma direção. Essa é uma medida da estabilidade na direção da melodia. Dois intervalos consecutivos contêm três notas; se todas as notas possuem o mesmo *pitch*, isso é interpretado como sendo um movimento na mesma direção. Pausas são novamente ignoradas nessa medida.

$$F_7(x) = \frac{\#\text{intervalos consecutivos na mesma direção}}{\#\text{intervalos} - 1}. \quad (\text{A.7})$$

Movimento gradual

A proporção de intervalos que são passos diatônicos. Um valor alto indica uma curva melódica suave, com poucos saltos. Um passo diatônico compreende um ou dois semitons. Pausas são ignoradas.

$$F_8(x) = \frac{\#\text{passos diatônicos}}{\#\text{intervalos}}. \quad (\text{A.8})$$

Retorno de salto

A proporção de intervalos grandes (saltos) que não são sucedidos por um intervalo de retorno. Um salto é maior ou igual a oito semitons (intervalo de sexta menor). Um intervalo de retorno deve ser ao menos de um semitom, porém menor que o salto que o precedeu².

$$F_9(x) = \frac{\#\text{saltos sem retorno}}{\#\text{saltos}}. \quad (\text{A.9})$$

²Se a melodia não possui salto algum, define-se que o valor dessa medida é 0,0.

Intensidade do clímax

Medida como o inverso do número de vezes que a nota de clímax é repetida na melodia. O maior valor dessa medida (1,0) ocorre quando essa nota é utilizada uma única vez. O uso frequente da nota de clímax reduz seu impacto.

$$F_{10}(x) = \frac{1}{\#\text{usos da nota de clímax}}. \quad (\text{A.10})$$

A.4 Medidas Rítmicas

Densidade de notas

A razão entre notas e *quanta*. Essa medida indica a esparsidade ou aglomeração de notas em uma melodia.

$$F_{11}(x) = \frac{\#\text{notas}}{\#\text{quanta}}. \quad (\text{A.11})$$

Densidade de pausas

A proporção de *quanta* silenciosos. Essa medida indica o grau de silêncio ou esparsidade em uma melodia.

$$F_{12}(x) = \frac{\#\text{quanta silenciosos}}{\#\text{quanta}}. \quad (\text{A.12})$$

Variedade rítmica

Essa é uma medida do grau de utilização dos 16 diferentes valores de duração entre a semicolcheia e a semibreve³. Dá uma indicação da coerência rítmica de uma melodia.

$$F_{13}(x) = \frac{\#\text{durações distintas}}{16}. \quad (\text{A.13})$$

Extensão de ritmos

A extensão de valores rítmicos na melodia dividida pelo maior valor possível. O maior valor de extensão é definido como 16 (i.e., semibreve:semicolcheia)⁴. Essa medida é diferente da anterior, porque é possível que uma melodia tenha uma grande

³No caso do *subcorpus* 2 definido no Capítulo 5, há apenas 8 diferentes valores de duração possíveis. Em valores relativos à duração da semínima, são eles: 0,25 (semicolcheia); 0,5; 0,75; 1,0; 1,5; 2,0; 2,5; e 3,0 (mínima pontuada).

⁴Novamente, no caso do *subcorpus*, a razão máxima entre as durações de uma composição é dada por $\frac{3,0}{0,25} = 12$.

extensão de durações, mas não utilize valores intermediários.

$$F_{14}(x) = \frac{\text{maior duração/menor duração}}{16}. \quad (\text{A.14})$$

Síncopes

A proporção de notas que são sincopadas, i.e., a proporção de notas cuja duração é maior que um pulso e começam fora do pulso. Pausas e notas são diferenciadas nessa medida.

$$F_{15}(x) = \frac{\#\text{notas sincopadas}}{\#\text{notas}}. \quad (\text{A.15})$$

A.5 Padrões

Repetição de *pitch*

A proporção de intervalos nos quais as duas notas apresentam o mesmo *pitch*. É uma outra medida do movimento melódico (ou falta de movimento). Para essa medida, as pausas são ignoradas.

$$F_{16}(x) = \frac{\#\text{intervalos com } \textit{pitch} \text{ repetido}}{\#\text{intervalos}}. \quad (\text{A.16})$$

Repetição de valor rítmico

A proporção de intervalos nos quais as duas notas apresentam a mesma duração. Pausas são ignoradas.

$$F_{17}(x) = \frac{\#\text{intervalos com duração repetida}}{\#\text{intervalos}}. \quad (\text{A.17})$$

Repetição de padrões de *pitch* em 3 notas⁵

A proporção de sequências de três notas cuja sequência de *pitches* é subsequentemente repetida. Pausas são consideradas, porque fazem parte dos padrões rítmicos.

$$F_{18}(x) = \frac{\#\text{sequência repetida de 3 } \textit{pitches}}{\#\text{notas} - 4}. \quad (\text{A.18})$$

Repetição de padrões de *pitch* em 4 notas⁶

A proporção de sequências de quatro notas cuja sequência de *pitches* é subsequentemente repetida. Como a medida anterior, essa medida compara relações intervalares.

⁵cf. Seção A.6

⁶cf. Seção A.6

Pausas são consideradas, porque fazem parte dos padrões rítmicos.

$$F_{19}(x) = \frac{\#sequência\ repetida\ de\ 4\ pitches}{\#notas - 5}. \quad (\text{A.19})$$

Repetição de padrões rítmicos em 3 notas⁷

A proporção de sequências de três notas cuja sequência de durações é subsequentemente repetida. Essa é uma medida da coerência rítmica da melodia, desenvolvida através da repetição. A razão entre as durações é avaliada, em vez das durações absolutas, de forma que um padrão semínima–mínima–semínima equivale a um padrão colcheia–semínima–colcheia. Pausas são consideradas, porque fazem parte dos padrões rítmicos.

$$F_{20}(x) = \frac{\#sequência\ repetida\ de\ 3\ durações}{\#notas - 4}. \quad (\text{A.20})$$

Repetição de padrões rítmicos em 4 notas⁸

A proporção de sequências de quatro notas cuja sequência de durações é subsequentemente repetida. Essa é uma medida da coerência rítmica da melodia, desenvolvida através da repetição. Como na medida anterior, avalia-se a razão entre as durações, em vez das durações absolutas. Pausas são consideradas, porque fazem parte dos padrões rítmicos.

$$F_{21}(x) = \frac{\#sequência\ repetida\ de\ 4\ durações}{\#notas - 5}. \quad (\text{A.21})$$

A.6 Sobre Repetições de Padrões

A interpretação das heurísticas 18 a 21 da maneira que Towsey *et al.* propõem em [85] é que se avaliam os padrões (de *pitch* ou rítmicos) que são subsequente e imediatamente repetidos, sem sobreposição. Assim, para o comprimento do padrão $M = 3$, a sequência “A B C A B C A B C” contém o padrão “A B C” que é repetido uma vez e, em seguida, avalia-se o padrão “A B C” central que também é repetido e o algoritmo termina; no total, há 2 repetições. Para sequência “A B C D B C D”, a subsequência “A B C” não é subsequentemente repetida, mas o padrão “B C D” sim; há apenas uma repetição. A sequência “A B C D A B C”, por sua vez, não tem nenhuma repetição (a subsequência “A B C” não é imediatamente repetida). Se L é o comprimento de uma sequência, o valor máximo de padrões que podem ser

⁷cf. Seção A.6

⁸cf. Seção A.6

repetidos, de acordo com a definição acima é dado por:

$$\mathcal{D}(L, M) = \left\lfloor \frac{L}{M} \right\rfloor - 1, \quad (\text{A.22})$$

onde $\lfloor \cdot \rfloor$ indica a função piso (em inglês, “*floor*”), i.e., $\lfloor x \rfloor$ é o maior número inteiro menor ou igual a x . Essa expressão é sugerida, neste trabalho, para substituir o denominador das heurísticas 18 a 21.

Apêndice B

Cálculo dos Parâmetros de Interpolação de Modelos de Markov

Este apêndice apresenta brevemente como os parâmetros da interpolação de modelos de Markov podem ser obtidos através do algoritmo de maximização de expectativa (EM — *Expectation Maximization*).

B.1 Interpolação

Uma dificuldade comumente enfrentada na área de processamento de linguagem natural é a insuficiência de dados para treinamento, que dá origem ao problema de esparsidade dos modelos linguísticos. Conforme visto na Seção 2.4.1, uma forma de lidar com esse problema é através das técnicas de suavização, dentre as quais destaca-se a interpolação de modelos de diferentes ordens. Por exemplo, em uma base de dados com 1,5 milhão de palavras, trigramas que não ocorram recebem probabilidade nula de ocorrência em um estimador de máxima verossimilhança [95]. A probabilidade suavizada de ocorrência desse mesmo trigrama pode ser estimada com auxílio dos bigrama e unigrama que o compõem.

Se, tipicamente, a probabilidade de uma sequência s de comprimento L em um modelo de Markov de ordem $N - 1$ é dada por

$$p_{\text{ML}}(s) = \prod_{k=1}^L p(w_k | s_{k-N+1}^{k-1}), \quad (\text{B.1})$$

um método de interpolação combina as probabilidades de ocorrência da sequência em diferentes ordens, de modo que cada N -grama (ou equivalentemente cada transição) não é calculado somente por meio da Equação 2.21. Um exemplo expresso pela

Equação 2.23, repetida aqui por facilidade, é a técnica de suavização de Jelinek-Mercer:

$$p_{\text{interpolação}}(w_k | s_{k-N+1}^{k-1}) = \lambda_{s_{k-N+1}^{k-1}} p_{\text{ML}}(w_k | s_{k-N+1}^{k-1}) + (1 - \lambda_{s_{k-N+1}^{k-1}}) p_{\text{interp}}(w_k | s_{k-N+2}^{k-1}). \quad (\text{B.2})$$

De maneira genérica, a probabilidade interpolada de uma transição pode ser definida como:

$$p_{\text{interpolação}}(w_k | s_{k-N+1}^{k-1}) = \sum_{n=1}^N \lambda_n p_n(w_k | s_{k-n+1}^{k-1}), \quad (\text{B.3})$$

onde $p_n(\cdot)$ representa o modelo de ordem $n - 1$ (n -grama). Assim, a combinação de dois modelos, de primeira e de segunda ordens, é indicada por [96]:

$$p_{\text{interpolação}}(w_k | w_{k-2} w_{k-1}) = \lambda_1 p_1(w_k | w_{k-1}) + \lambda_2 p_2(w_k | w_{k-2} w_{k-1}). \quad (\text{B.4})$$

B.2 Algoritmo de Maximização de Expectativa

Os valores ótimos para os pesos λ_n podem ser determinados pelo algoritmo de maximização de expectativa. Uma das formas de efetuar esse cálculo compreende [96]: dividir o *corpus* amostral em 2 ou mais grupos; estimar os parâmetros de interpolação para o primeiro grupo e calcular as probabilidades dos n -gramas, $(n - 1)$ -gramas etc. com as sequências dos demais grupos. Repete-se o procedimento até que os parâmetros tenham sido estimados sobre todos os grupos [95]. Usualmente, os pesos λ_n também dependem dos contextos dos símbolos [96]. Uma regra prática consiste em agrupar os diferentes n -gramas de acordo com o número de ocorrências no *corpus* e utilizar o mesmo peso para todos as sequências de um mesmo grupo [96]. No caso geral, os passos do algoritmo EM são, então, para a combinação de modelos de bigramas e trigramas [96]:

1. Inicializar os pesos com valores aleatórios, com a restrição que $\lambda_1 + \lambda_2 = 1$;
2. Calcular as somas:

$$S_1 = \sum_{w_{k-2}^k} \frac{\lambda_1 p_1(w_k | w_{k-1})}{\lambda_1 p_1(w_k | w_{k-1}) + \lambda_2 p_2(w_k | w_{k-2} w_{k-1})}, \quad (\text{B.5})$$

$$S_2 = \sum_{w_{k-2}^k} \frac{\lambda_2 p_2(w_k | w_{k-2} w_{k-1})}{\lambda_1 p_1(w_k | w_{k-1}) + \lambda_2 p_2(w_k | w_{k-2} w_{k-1})}; \quad (\text{B.6})$$

3. Atualizar os parâmetros:

$$\lambda_1 = \frac{S_1}{S_1 + S_2} \quad \text{e} \quad \lambda_2 = \frac{S_2}{S_1 + S_2}; \quad (\text{B.7})$$

4. Retornar ao passo 2 enquanto o módulo da diferença na atualização for superior a uma constante de erro ϵ .

O caso em que mais de dois modelos são combinados é similar ao descrito acima.