



Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

Aplicação de Algoritmos Evolucionários
à
Extracção de Padrões Musicais

Carlos Fernando Almeida Grilo

Coimbra

Novembro de 2002

Dissertação submetida à
Universidade de Coimbra
para obtenção do grau de
Mestre em Engenharia Informática
Área Científica de Inteligência Artificial

Aplicação de Algoritmos Evolucionários
à
Extracção de Padrões Musicais

Carlos Fernando Almeida Grilo

Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática
Novembro de 2002

Dissertação realizada sob a orientação do
Professor Doutor Fernando Amílcar Bandeira Cardoso
Professor Auxiliar
do Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia
da Universidade de Coimbra

Resumo

A extracção de padrões é um problema que se coloca em várias áreas como, por exemplo, a biologia molecular ou a área financeira, e que pode ser considerado, do ponto de vista da inteligência artificial, como uma forma de aprendizagem não supervisionada. No domínio musical, o problema pode ser definido, informalmente, da seguinte forma: dada uma peça musical (ou várias), identificar as partes dessa peça que se repitam, aproximadamente ou não, e que possuam um significado relevante no contexto dessa peça.

O objectivo deste trabalho consistiu em estudar a viabilidade da aplicação de algoritmos evolucionários ao problema da extracção de padrões musicais. Para levar a cabo o estudo proposto desenvolvemos duas abordagens utilizando dois tipos diferentes de algoritmos evolucionários: a programação genética e os algoritmos genéticos. Em cada uma das abordagens o objectivo é essencialmente o mesmo: encontrar uma segmentação de uma peça que permita identificar os padrões mais importantes nela existentes. Devido às características de cada um dos algoritmos, a representação utilizada para os indivíduos é diferente. Assim, enquanto que na abordagem baseada em programação genética cada indivíduo é um programa que produz como resultado uma determinada peça, constituindo ao mesmo tempo uma descrição da sua estrutura de segmentos, na abordagem baseada em algoritmos genéticos cada indivíduo consiste numa sequência de símbolos que representa uma hipótese de segmentação da peça a analisar. Embora as funções de avaliação utilizadas nas duas abordagens também sejam diferentes, ambas beneficiam os indivíduos que apresentem o conjunto dos padrões mais importantes existentes na peça. Para ambas as abordagens foi também desenvolvido um método que permite realizar uma segunda segmentação de uma peça a partir dos segmentos identificados na primeira segmentação.

Os resultados experimentais obtidos com a abordagem baseada em programação genética que desenvolvemos permitem-nos verificar que esta abordagem apresenta bastantes dificuldades na resolução deste tipo de problemas. Pelo contrário, a abordagem baseada em algoritmos genéticos permitiu obter resultados que nos levam a considerar que a aplicação desta abordagem a este tipo de problemas é viável.

Abstract

Pattern extraction is a problem that occurs in several areas like, for example, molecular biology and finance, and can be viewed, from the point of view of artificial intelligence, as a kind of unsupervised learning. In the musical domain, the problem can be informally defined in the following way: given a musical piece (or more), identify the meaningful recurrent parts of that piece.

The goal of this work is to study the viability of applying evolutionary algorithms to the problem of musical pattern extraction. In order to take this study, we develop two approaches based on two different types of evolutionary algorithms: genetic programming and genetic algorithms. The goal in both approaches is essentially the same: find a segmentation of a musical piece that allows the identification of the most meaningful patterns that exist in that piece. Due to the character of each type of algorithm, the representation used to represent individuals in each approach is different. Hence, while in the genetic programming based approach an individual is a program that produces as a result a musical piece, being at the same time a description of the structure of that piece, in the genetic algorithms based approach each individual is a sequence of symbols that represent a possible segmentation of the musical piece that is being analyzed. The two approaches also use different fitness functions, but both have in common the fact that they give a better fitness value to individuals that present the set of most meaningful patterns. For both approaches we also developed a method to make a second segmentation of a musical piece using the segments identified in the first segmentation.

The experimental results obtained with the genetic programming based approach allowed us to verify that this approach has great difficulties in the resolution of this type of problems. On the contrary, with the genetic algorithms based approach we obtained results that allow us to believe that this approach can be useful in the resolution of this type of problems.

Agradecimentos

São várias as pessoas e entidades que permitiram que este trabalho fosse, não apenas possível, mas também mais agradável de realizar e a quem gostaria de agradecer:

- Ao meu orientador, o Professor Amílcar Cardoso, pelo apoio, críticas e sugestões dados ao longo do tempo, essenciais para que este trabalho se realizasse.
- Ao Professor Ernesto Costa, pela disponibilidade para discutir alguns aspectos relacionados com este trabalho.
- Aos meus colegas do LIA pelas discussões, sugestões e amizade. Em especial, gostaria de agradecer ao Penousal e aos Xicos o contributo que deram em diversos momentos da realização deste trabalho.
- À Escola Superior de Tecnologia e Gestão (ESTG) do Instituto Politécnico de Leiria pelas facilidades concedidas para a realização deste trabalho, nomeadamente a concessão da dispensa de serviço docente durante um semestre. O meu agradecimento também aos meus colegas do Departamento de Engenharia Informática da ESTG pelo companheirismo e apoio dado.
- Aos meus pais pelos enormes sacrifícios que sempre fizeram para que os seus filhos tivessem uma boa educação. Espero sinceramente ser digno desse esforço.
- À Sandrinha, claro.

Índice

1	Introdução	1
1.1	O Problema	1
1.2	Objectivos e Enquadramento da Dissertação	3
1.3	Estrutura da Dissertação	5
2	Análise Musical	7
2.1	Análise Paradigmática	8
2.1.1	Análise Semiótica	8
2.1.2	O Procedimento de Análise de Ruwet	10
2.1.3	Problemas da Análise Paradigmática	13
2.2	Teoria Generativa da Música Tonal	14
2.3	Teoria Geral Computacional da Estrutura Musical	17
2.3.1	Estrutura Geral da TGCEM	18
2.3.2	Modelo de Detecção Local de Limites	21
3	Extracção de Padrões Musicais	25
3.1	Estrutura Geral de um Sistema de Extracção de Padrões	26
3.2	Representação Musical	29
3.2.1	Sequências Musicais Monofónicas	30
3.2.2	Informação Melódica	32
3.2.3	Informação Rítmica	34
3.3	Comparação de Sequências Musicais	35
3.3.1	Distância de Levenshtein	37
3.3.2	Programação Dinâmica	38
3.3.3	Programação Dinâmica e Comparação de Sequências Musicais	39
3.4	Sistemas de Extracção de Padrões Musicais	41
3.4.1	EMI	41
3.4.2	Charm	43
3.4.3	Sistemas Musicais Interactivos	45
3.4.4	FIEXPath	47
3.4.5	AIPS	51
4	Computação Evolucionária	55
4.1	Terminologia	55
4.2	O Algoritmo Evolucionário Típico	56
4.3	Métodos de Selecção	59
4.4	Principais Modelos de Algoritmos Evolucionários	62
4.4.1	Programação Evolucionária	62
4.4.2	Estratégias Evolucionárias	63
4.4.3	Algoritmos Genéticos	63
4.5	Programação Genética	65
4.5.1	Representação e População Inicial	66
4.5.2	Avaliação e Selecção	67
4.5.3	Operadores Genéticos	67
4.5.4	Funções Definidas Automaticamente	68

4.6	Programação Genética Aplicada ao Cálculo da Complexidade de Kolmogorov	69
4.7	Programação Genética Aplicada à Composição Musical	71
5	Programação Genética e Extração de Padrões Musicais	75
5.1	Programas como Descrições da Estrutura de Peças Musicais	75
5.1.1	Programas “independentes”	76
5.1.2	Programas “dependentes”	78
5.2	Descrição da Abordagem	80
5.2.1	Representação dos Indivíduos	80
5.2.2	Avaliação	85
5.2.3	Representação das Peças	87
5.2.4	Operadores de Melhoramento	89
5.3	Experimentação	89
5.3.1	Ambiente Experimental	90
5.3.2	Funções e Terminais Utilizados	90
5.3.3	Parâmetros do Processo Evolutivo	92
5.3.4	«Be m’anperdut...»	94
5.3.5	«Maria muoter reinû maît»	96
5.3.6	«Kalenda Maya»	100
5.3.7	4ª Variação das «Seis variaciones sobre um tema de Milán»	103
5.3.8	«Syrinx»	104
5.3.9	Segunda Segmentação	107
5.4	Conclusão	109
6	Algoritmos Genéticos e Extração de Padrões Musicais	113
6.1	Descrição da Abordagem	113
6.1.1	Representação dos Indivíduos	113
6.1.2	Avaliação	116
6.1.3	Representação das Peças	119
6.1.4	Operadores Genéticos e Correção dos Indivíduos	120
6.1.5	Operadores de Aprendizagem	122
6.1.6	Segunda Segmentação	126
6.2	Experimentação	128
6.2.1	Ambiente Experimental	128
6.2.2	Parâmetros do Processo Evolutivo	129
6.2.3	«Be m’anperdut...»	131
6.2.4	«Maria muoter reinû maît»	135
6.2.5	«Kalenda Maya»	138
6.2.6	Variação 4 das «Seis variaciones sobre um tema de Milán»	140
6.2.7	«Syrinx»	144
6.3	Conclusão	149
7	Conclusões e Trabalho Futuro	151
7.1	Conclusões	151
7.1.1	Programação Genética e Extração de Padrões Musicais	151
7.1.2	Algoritmos Genéticos e Extração de Padrões Musicais	153
7.2	Trabalho Futuro	155
	Bibliografia	157

Índice de Figuras

Figura 2.1 - Análise da peça «Maria muoter reinû maît» por Ruwet.	13
Figura 2.2 - Estrutura geral da TGCEM.	19
Figura 2.3 - VLs calculados para três valores de uma determinada dimensão (altura das notas, dinâmica, pausas ou articulação) separados por dois intervalos.	22
Figura 2.4 - VLs calculados para o valor de três pontos de ataque separados por dois intervalos de pontos de ataque e duas durações.	23
Figura 2.5 - Exemplos de VLs calculados para sequência melódicas com mais de três notas.	24
Figura 2.6 - Proto-segmetação do primeiro tema da sinfonia em Sol menor de Mozart.	24
Figura 3.1 - Estrutura geral de um sistema de extracção de padrões.	27
Figura 3.2 - Representações baseadas em notas de uma pequena sequência melódica.	31
Figura 3.3 - Representações baseadas na relação entre as notas de uma pequena sequência melódica.	31
Figura 3.4 - Exemplo de dois segmentos em tonalidades diferentes e cuja relação com a tónica é igual.	32
Figura 3.5 - Sequência com uma transposição modal e uma transposição cromática.	33
Figura 3.6 - As primeiras quatro ocorrências de um motivo na peça <i>Vingt Regards sur l'Enfant Jésus(III-L'échange)</i> de Messiaen.	34
Figura 3.7 - Uma sequência rítmica e uma sua variação com metade da duração.	34
Figura 3.8 - Correspondência entre os símbolos de duas sequências.	37
Figura 3.9 - Cálculo da distância das sequências “abcdefg” e “acdefhi”.	38
Figura 3.10 - Alinhamento possível entre duas sequências quando se utilizam as operações de fragmentação e consolidação.	40
Figura 3.11 - Exemplo de uma fragmentação.	40
Figura 3.12 - Relatório fornecido durante a execução do algoritmo de análise.	44
Figura 3.13 - Transformações identificadas pelo algoritmo.	45
Figura 3.14 - Exemplo de uma estrela no grafo de semelhanças.	50
Figura 3.15 - Os padrões mais importantes da canção <i>Frère Jacques</i> descobertos pelo AIPS aplicado apenas a intervalos escalares e pela função de selecção utilizando os pesos $(a, b, c) = (3, 3, 4)$.	53
Figura 3.16 - Soma pesada e normalizada dos limites calculados com o MDLL e com o AIPS/função de avaliação para a canção <i>Frère Jacques</i> .	54
Figura 4.1 - Um algoritmo evolucionário típico.	57
Figura 4.2 - Operadores de recombinação e mutação mais utilizados nos algoritmos genéticos.	65
Figura 4.3 - Exemplo de um indivíduo.	73
Figura 5.1 - Exemplo de um programa capaz de produzir uma sequência de notas de forma totalmente independente.	77
Figura 5.2 - Exemplo de uma sequência de notas e de um programa que descreve a sua estrutura.	80
Figura 5.3 - Repetições consecutivas e não consecutivas descritas através da função <i>conc</i> .	81
Figura 5.4 - Dois programas capazes de produzir a mesma sequência de notas.	83
Figura 5.5 - Dois indivíduos diferentes mas equivalentes na forma como descrevem uma peça.	85
Figura 5.6 - Dois indivíduos com funções principais diferentes e com as mesmas FDAs descrevendo a mesma peça. O indivíduo da direita é, segundo a função de avaliação, melhor que o indivíduo da esquerda.	86
Figura 5.7 - Indivíduo que produz uma sequência ligeiramente diferente da sequência a analisar.	87
Figura 5.8 - Estrutura da canção «Frère Jacques» e a sua representação utilizando intervalos cromáticos.	88
Figura 5.9 - Indivíduo que produz uma sequência ligeiramente diferente da sequência a analisar devido à utilização de um segmento cujos limites não estão correctos.	89
Figura 5.10 - Representação simplificada de dois indivíduos equivalentes.	92
Figura 5.11 - Três dos indivíduos obtidos nas experiências realizadas, antes da aplicação dos operadores de melhoramento.	95

Figura 5.12 - Dois indivíduos com uma estrutura AABB obtidos nas experiências com a peça «Maria muoter reinû maît».	98
Figura 5.13 - Exemplo de um indivíduo que os operadores não conseguem melhorar.	98
Figura 5.14 - Indivíduo com uma estrutura AAX obtido nas experiências com a peça «Maria muoter reinû maît».	99
Figura 5.15 - Indivíduo obtido nas experiências realizadas com a peça «Kalenda Maya».	101
Figura 5.16 - Dois indivíduos capazes de produzir sem erros a peça «Kalenda Maya».	103
Figura 5.17 - Indivíduo obtido nas experiências realizadas com a 4ª Variação das «Seis variaciones sobre um tema de Milán».	104
Figura 5.18 - Os dois melhores indivíduos obtidos nas experiências realizadas com a peça «Syrinx».	106
Figura 5.19 - Melhor indivíduo obtido em experiências realizadas com a peça «Syrinx», nas quais a duração das notas não foi tida em conta.	107
Figura 5.20 - Indivíduo que representa uma segunda segmentação da peça «Be m'anperdut...».	108
Figura 5.21 - Indivíduo que representa uma segunda segmentação da peça «Maria muoter reinû maît».	109
Figura 5.22 - Indivíduo que representa uma segunda segmentação da 4ª variação das «Seis variaciones sobre um tema de Milán».	109
Figura 6.1 - Exemplo de uma peça representada como uma sequência de notas e de um indivíduo que representa uma sua segmentação.	115
Figura 6.2 - Exemplo de uma peça representada como uma sequência de intervalos e de um indivíduo que representa uma sua segmentação.	115
Figura 6.3 - Dois indivíduos que representam duas segmentações diferentes da uma mesma peça representada como uma sequência de intervalos.	118
Figura 6.4 - Exemplo da aplicação do operador de mutação em dois pontos diferentes de um indivíduo.	120
Figura 6.5 - Exemplo da aplicação do operador de recombinação de dois pontos em dois pontos diferentes de um indivíduo.	120
Figura 6.6 - Ilustração do processo de correcção de segmentos.	122
Figura 6.7 - Descrição do processo de aprendizagem.	123
Figura 6.8 - Exemplo da aplicação do primeiro operador.	124
Figura 6.9 - Exemplo da aplicação do segundo operador.	124
Figura 6.10 - Exemplo da aplicação do terceiro operador.	125
Figura 6.11 - Algoritmo de classificação utilizado na escolha dos segmentos a utilizar na segunda segmentação.	127
Figura 6.12 - Estrutura dos melhores indivíduos obtidos na primeira segmentação da peça «Be m'anperdut...».	132
Figura 6.13 - Indivíduos com desempenho igual a 30 gerados na segunda segmentação da peça «Be m'anperdut...».	135
Figura 6.14 - Estrutura dos melhores indivíduos obtidos na primeira segmentação da peça «Maria muoter reinû maît».	136
Figura 6.15 - Estrutura dos melhores indivíduos obtidos na segunda segmentação da peça «Maria muoter reinû maît».	137
Figura 6.16 - Estrutura dos melhores indivíduos obtidos na primeira segmentação da peça «Kalenda Maya».	139
Figura 6.17 - Estrutura dos melhores indivíduos obtidos na segunda segmentação da peça «Kalenda Maya».	140
Figura 6.18 - Estrutura dos melhores indivíduos obtidos na primeira segmentação da Variação 4.	141
Figura 6.19 - Estrutura dos melhores indivíduos obtidos na segunda segmentação da Variação 4.	143
Figura 6.20 - Estrutura do melhor indivíduo obtido na primeira segmentação da peça «Syrinx».	145
Figura 6.21 - Estrutura dos melhores indivíduos obtidos na segunda segmentação da peça «Syrinx».	149

Índice de Tabelas

Tabela 3.1 - Principais características dos sistemas descritos neste capítulo.	54
Tabela 4.1 - Funções primitivas do TOY Lisp.	70
Tabela 4.2 - Programas obtidos para cada uma das sequências.	71
Tabela 4.3 - Funções primitivas.	72
Tabela 5.1 - Parâmetros do processo evolutivo.	93
Tabela 5.2 - Resultados dos melhores indivíduos obtidos para a peça «Be m'anperdut...», antes da aplicação dos operadores de melhoramento.	94
Tabela 5.3 - Resultados dos melhores indivíduos obtidos para a peça «Be m'anperdut...», depois da aplicação dos operadores de melhoramento.	94
Tabela 5.4 - Resultados dos melhores indivíduos obtidos para a peça «Maria muoter reinû maît», antes da aplicação dos operadores de melhoramento.	97
Tabela 5.5 - Resultados dos melhores indivíduos obtidos para a peça «Maria muoter reinû maît», depois da aplicação dos operadores de melhoramento.	97
Tabela 5.6 - Resultados dos melhores indivíduos obtidos para a peça «Kalenda Maya», antes da aplicação dos operadores de melhoramento.	100
Tabela 5.7 - Resultados dos melhores indivíduos obtidos para a peça «Kalenda Maya», depois da aplicação dos operadores de melhoramento.	101
Tabela 5.8 - Resultados dos melhores indivíduos obtidos para a 4ª variação das «Seis variaciones sobre um tema de Milán», antes da aplicação dos operadores de melhoramento.	104
Tabela 5.9 - Resultados dos melhores indivíduos obtidos para a 4ª variação das «Seis variaciones sobre um tema de Milán», depois da aplicação dos operadores de melhoramento.	104
Tabela 5.10 - Resultados dos melhores indivíduos obtidos para a peça «Syrinx», depois da aplicação dos operadores de melhoramento.	105
Tabela 6.1 - Parâmetros do processo evolutivo.	130
Tabela 6.2 - Média e desvio padrão do desempenho dos melhores indivíduos obtidos na primeira segmentação da peça «Be m'anperdut...» quando se utilizam e quando não se utilizam operadores de aprendizagem.	132
Tabela 6.3 - Resultados obtidos para os melhores indivíduos gerados na primeira segmentação da peça «Be m'anperdut...» quando são utilizados operadores de aprendizagem.	132
Tabela 6.4 - Média e desvio padrão do desempenho dos melhores indivíduos obtidos na segunda segmentação da peça «Be m'anperdut...» quando se utilizam e quando não se utilizam operadores de aprendizagem.	133
Tabela 6.5 - Resultados obtidos para os melhores indivíduos gerados na segunda segmentação da peça «Be m'anperdut...» quando são utilizados operadores de aprendizagem.	134
Tabela 6.6 - Média e desvio padrão do desempenho dos melhores indivíduos obtidos na primeira segmentação da peça «Maria muoter reinû maît» quando se utilizam e quando não se utilizam operadores de aprendizagem.	136
Tabela 6.7 - Resultados obtidos para os melhores indivíduos gerados na primeira segmentação da peça «Maria muoter reinû maît» quando são utilizados operadores de aprendizagem.	136
Tabela 6.8 - Média e desvio padrão do desempenho dos melhores indivíduos obtidos na segunda segmentação da peça «Maria muoter reinû maît» quando se utilizam e quando não se utilizam operadores de aprendizagem.	137
Tabela 6.9 - Resultados obtidos para os melhores indivíduos gerados na segunda segmentação da peça «Maria muoter reinû maît», quando são utilizados operadores de aprendizagem.	137
Tabela 6.10 - Média e desvio padrão do desempenho dos melhores indivíduos obtidos na primeira segmentação da peça «Kalenda Maya» quando se utilizam e quando não se utilizam operadores de aprendizagem.	138

Tabela 6.11 - Resultados obtidos para os melhores indivíduos gerados na primeira segmentação da peça «Kalenda Maya» quando são utilizados operadores de aprendizagem.	138
Tabela 6.12 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na segunda segmentação da peça «Kalenda Maya» quando se utilizam e quando não se utilizam operadores de aprendizagem.	139
Tabela 6.13 - Resultados obtidos para os melhores indivíduos gerados na segunda segmentação da peça «Kalenda Maya» quando são utilizados operadores de aprendizagem.	140
Tabela 6.14 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na primeira segmentação da Variação 4 quando se utilizam e quando não se utilizam operadores de aprendizagem.	141
Tabela 6.15 - Resultados obtidos para os melhores indivíduos gerados na primeira segmentação da Variação 4 quando são utilizados operadores de aprendizagem.	141
Tabela 6.16 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na segunda segmentação da Variação 4 quando se utilizam e quando não se utilizam operadores de aprendizagem.	142
Tabela 6.17 - Resultados obtidos para os melhores indivíduos gerados na segunda segmentação da Variação 4 quando são utilizados operadores de aprendizagem.	142
Tabela 6.18 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na primeira segmentação da peça «Syrinx» quando se utilizam e quando não se utilizam operadores de aprendizagem.	144
Tabela 6.19 - Resultados obtidos para os melhores indivíduos gerados na primeira segmentação da peça «Syrinx» quando são utilizados operadores de aprendizagem.	145
Tabela 6.20 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na segunda segmentação da peça «Syrinx» quando se utilizam e quando não se utilizam operadores de aprendizagem.	148
Tabela 6.21 - Resultados obtidos para os melhores indivíduos gerados na segunda segmentação da peça «Syrinx» quando são utilizados operadores de aprendizagem.	148

Índice de Gráficos

Gráfico 5.1 - Média do desempenho dos melhores indivíduos ao longo das 100 gerações.	96
Gráfico 5.2 - Média da distância entre a peça produzida pelos melhores indivíduos ao longo das 100 gerações e a peça a analisar.	96
Gráfico 5.3 - Média do número de notas utilizadas pelos melhores indivíduos ao longo das 100 gerações.	96
Gráfico 5.4 - Média do desempenho dos melhores indivíduos ao longo das 100 gerações.	100
Gráfico 5.5 - Média da distância entre a peça produzida pelos melhores indivíduos ao longo das 100 gerações e a peça a analisar.	100
Gráfico 5.6 - Média do número de notas utilizadas pelos melhores indivíduos ao longo das 100 gerações.	100

Lista de Siglas

AIPS - Algoritmo de Indução de Padrões em Sequências.

FDA - Função Definida Automaticamente.

MDLL - Modelo de Detecção Local de Limites.

RGIM - Regra Geral da Identidade-Mudança.

RIM - Regra da Identidade-Mudança.

RP - Regra da Proximidade.

TGCEM - Teoria Geral Computacional da Estrutura Musical.

TGMT - Teoria Generativa da Música Tonal.

VL - Valor de Limite.

Capítulo 1

Introdução

1.1 O Problema

Os seres humanos parecem ter uma forte propensão, seja ela inata ou aprendida, para descobrir padrões onde quer que eles existam. Esta capacidade tem sido ao longo do tempo essencial para a sobrevivência da espécie humana, uma vez que permite que se possam fazer previsões com base nas regularidades observadas (veja-se, por exemplo, a capacidade de prever as estações ou o tempo). No entanto, nem todos os padrões que ocorrem num determinado fenómeno têm obrigatoriamente que ter a mesma importância ou utilidade. Por exemplo, no mercado financeiro existe um padrão muito conhecido denominado “three advancing white soldiers” que quando ocorre permite com alguma segurança confiar na subida de um título. Este padrão ocorre quando, depois de atingir um mínimo histórico, o valor de um título sobe gradualmente durante três dias consecutivos. No entanto, segundo os especialistas, se o valor de um título nas mesmas condições subir apenas durante dois dias seguidos, não existem razões para acreditar que o ciclo negativo desse título terminou. Este exemplo demonstra que o simples facto de um padrão existir não significa que este tenha que ter alguma utilidade ou importância. Conforme o fenómeno em causa, a importância dos padrões pode depender de vários factores como sejam, por exemplo, a frequência de ocorrência ou o seu tamanho.

Em inteligência artificial, é comum designar o processo da procura dos padrões mais importantes que ocorrem num determinado fenómeno por extracção de padrões. Uma sub-área importante desta área de investigação dedica-se ao problema de encontrar os padrões mais importantes existentes em fenómenos que possam ser representados como sequências de símbolos. Este é um problema que tem um grande interesse em vários domínios, como por exemplo a biologia molecular, a área

financeira ou a música. No domínio da música, que é aquele que é estudado nesta dissertação, a automatização do processo de descoberta dos padrões mais importantes existentes numa peça musical tem um especial interesse por várias razões, entre as quais destacamos as seguintes:

- Do ponto de vista da análise musical, é muito importante saber quais os padrões mais importantes existentes numa peça ou conjunto de peças. Entre outras coisas, estes padrões podem servir para ajudar a caracterizar o estilo de um determinado compositor, ou simplesmente para ajudar a saber em que segmentos é que uma peça se divide e qual a relação entre esses segmentos. Se atendermos ao facto de que este processo é um processo demorado e que um estudo musicológico pode implicar a análise de um número considerável de peças musicais, facilmente se reconhece que a sua automatização pode ter grandes vantagens.
- Com o crescimento da Internet, surgiu também o interesse pela criação de sistemas de bases de dados musicais (e não só) que permitam pesquisas baseadas em conteúdos. Neste tipo de sistemas, por vezes denominados “What You Hum Is What You Get”, o utilizador tem a possibilidade de pesquisar e licitar uma determinada peça musical cantarolando um pequena parte dessa peça. De modo a tornar este processo mais rápido seria útil que estas bases de dados fossem indexadas pelos padrões mais importantes existentes nas peças, os quais coincidem frequentemente com as partes que as pessoas lembram mais facilmente (os refrões, por exemplo).
- Em sistemas computacionais de composição musical, o conhecimento dos padrões tipicamente utilizados por um determinado compositor e a caracterização do contexto em que cada um desses padrões é utilizado pode ser importante se o objectivo for a composição de peças ao estilo desse compositor. Por outro lado, em sistemas interactivos de improvisação/composição musical em grupo, é importante que o sistema seja capaz de identificar em tempo real os padrões mais importantes que o ou os seus companheiros produzem, de modo a que possa também ele utilizar esse material. Desta forma é possível conferir alguma estrutura à peça que vai

sendo produzida.

1.2 Objectivos e Enquadramento da Dissertação

Esta dissertação tem como objectivo estudar a viabilidade da aplicação de algoritmos evolucionários ao problema da extracção de padrões em peças musicais monofónicas¹ representadas de forma simbólica. A razão que nos levou a iniciar este estudo prende-se com a necessidade que sentimos, no âmbito do projecto SICOM [Pereira, 1997], desenvolvido no Laboratório de Inteligência Artificial do Centro de Informática e Sistemas da Universidade de Coimbra, de dispor de um conjunto de peças musicais analisadas. Neste projecto é utilizado raciocínio baseado em casos para realizar composição musical. Mais concretamente, o objectivo é a composição de novas peças ao estilo de um determinado compositor, cujas peças, acompanhadas da respectiva análise, estão armazenadas numa base de casos. Entre outras coisas, da análise de cada peça deve constar informação acerca dos segmentos em que ela está dividida e qual a relação entre esses segmentos. Numa primeira fase do projecto, utilizámos uma base de casos composta por três peças (o que é manifestamente pouco) e as respectivas análises realizadas por especialistas em análise musical. No entanto, depressa nos apercebemos que o enriquecimento da base de casos com um número razoável de peças seria um trabalho demasiado moroso, sendo de todo desejável automatizar, senão todo o processo, pelo menos parte dele.

Nas abordagens existentes ao problema da extracção de padrões musicais, sobretudo as que não se destinam a ser utilizadas em sistemas cuja resposta tem que ser dada em tempo real, é comum o espaço de segmentos possíveis de uma peça ser percorrido quase na totalidade de modo a que possam ser identificados todos os padrões nela existentes. Depois de identificados, estes padrões são ordenados utilizando um ou mais critérios de importância para que seja possível saber quais são os padrões mais importantes. Devido à grande quantidade de padrões que podem ser encontrados, não é fácil estabelecer critérios que permitam ordenar correctamente esses padrões. Por exemplo, pode ser difícil evitar que haja um grande grau de sobreposição entre os padrões mais importantes, situação que é normalmente indesejável.

¹ Peças com apenas uma linha melódica.

Estas dificuldades deixam espaço aberto ao estudo da viabilidade de aplicação de outros métodos. No leque de métodos existentes, os algoritmos evolucionários são possíveis candidatos a esse estudo. Trata-se de algoritmos de pesquisa paralela e estocástica inspirados na teoria da evolução de Darwin e na biologia molecular que permitem evoluir um conjunto de potenciais soluções para um problema. O processo de pesquisa inicia-se geralmente com um conjunto de potenciais soluções criado aleatoriamente e evolui depois de forma iterativa até que um determinado critério de paragem seja satisfeito (por exemplo, até que seja encontrada uma solução para o problema). Ao longo do processo evolutivo, o conjunto de potenciais soluções vai sofrendo alterações provocadas por operadores, geralmente denominados operadores genéticos, que permitem criar novos indivíduos a partir dos indivíduos já existentes. Uma vez que a aplicação continuada destes operadores faria aumentar progressivamente o tamanho do conjunto de potenciais soluções, em cada iteração é também aplicado um operador de selecção que escolhe probabilisticamente os melhores elementos da iteração anterior. Para que se possa aplicar este operador é necessário utilizar uma função de avaliação que, quando aplicada a um determinado elemento, indique o valor desse elemento enquanto solução para o problema. Este processo permite concentrar a pesquisa em zonas mais promissoras do espaço, aumentando progressivamente as possibilidades, não só de se encontrar uma solução para o problema, mas também de encontrar melhores soluções.

De seguida apresentamos as razões que, à partida, nos levaram a considerar que o estudo da viabilidade da aplicação de algoritmos evolucionários a este problema poderia valer a pena:

- A capacidade dos algoritmos evolucionários em encontrar soluções para problemas complexos e que envolvam um grande espaço de pesquisa, já demonstrada em outros domínios.
- A possibilidade de representar os indivíduos candidatos a solução para o problema como hipóteses de segmentação de uma peça em que não haja sobreposição de segmentos. Desta forma, no final do processo de pesquisa já não seria necessário realizar processamento extra de modo a decidir quais, de entre os padrões identificados, os que fazem realmente parte da segmentação.

- Verificar até que ponto é possível identificar os padrões mais importantes existentes numa peça sem que se tenha que percorrer todo ou quase todo o espaço de segmentos possíveis.
- Tanto quanto sabemos, esta seria a primeira tentativa de aplicar este tipo de algoritmos a este problema. Este estudo poderia, assim, contribuir para alargar o leque de métodos existentes para a resolução do problema.

Para levar a cabo o estudo proposto desenvolvemos duas abordagens utilizando dois tipos diferentes de algoritmos evolucionários: a programação genética e os algoritmos genéticos. Em cada uma das abordagens o objectivo é essencialmente o mesmo: encontrar uma segmentação de uma peça que permita identificar os segmentos mais importantes nela existentes. Devido às características de cada um dos algoritmos, a representação utilizada para os indivíduos é diferente. Assim, enquanto que na abordagem baseada em programação genética cada indivíduo é um programa que produz como resultado uma determinada peça, constituindo ao mesmo tempo uma descrição da sua estrutura de segmentos, na abordagem baseada em algoritmos genéticos cada indivíduo consiste numa sequência de símbolos que representa uma hipótese de segmentação da peça a analisar. Embora as funções de avaliação utilizadas nas duas abordagens também sejam diferentes, ambas beneficiam os indivíduos que apresentem o conjunto dos padrões mais importantes existentes na peça. Para ambas as abordagens foi também desenvolvido um método que permite realizar uma segunda segmentação de uma peça a partir dos segmentos identificados na primeira segmentação.

Para cada uma das abordagens foi realizado um conjunto de testes experimentais. Os resultados obtidos nestes testes permitiram-nos chegar a algumas conclusões acerca da viabilidade de utilização dos algoritmos evolucionários ao problema, pelo que consideramos que o estudo realizado representa um contributo válido para esta área de investigação.

1.3 Estrutura da Dissertação

Apresentamos de seguida uma perspectiva geral do conteúdo desta dissertação. No próximo capítulo descrevemos três teorias musicais que julgamos serem importantes

para a contextualização e compreensão do nosso trabalho. Qualquer uma destas teorias permite obter uma descrição estrutural de uma peça musical, e todas têm em comum o facto de terem na noção de semelhança um dos seus pilares. As noções musicais que são necessárias para compreender o conteúdo não só deste capítulo mas também dos seguintes são muito básicas. Julgamos, por isso, não ser necessário dedicar uma parte desta dissertação à introdução dessas noções, as quais podem ser facilmente obtidas em qualquer manual de introdução à educação musical ou numa das muitas páginas Web dedicadas ao assunto. De entre estas últimas aconselhamos a página [Music Theory Online].

O capítulo 3 é dedicado aos sistemas computacionais de extracção de padrões musicais. Neste capítulo, além de descrevermos alguns dos sistemas computacionais de extracção de padrões mais conhecidos, abordaremos aspectos com uma importância central neste tipo de sistemas, como sejam a representação e a comparação de sequências musicais.

No capítulo 4 fazemos uma breve introdução à área da computação evolucionária. Neste capítulo damos especial atenção aos algoritmos genéticos e à programação genética, uma vez que são estas as duas técnicas utilizadas nas abordagens que desenvolvemos. Ainda neste capítulo, descrevemos dois trabalhos que serviram de ponto de partida para a nossa abordagem baseada em programação genética.

Nos capítulos 5 e 6 apresentamos as duas abordagens à extracção de padrões musicais que desenvolvemos, baseadas, respectivamente, em programação genética e em algoritmos genéticos. Nestes capítulos apresentamos e analisamos também os resultados experimentais obtidos com cada uma destas abordagens.

Finalmente, no último capítulo tecemos algumas conclusões acerca do trabalho desenvolvido e sugerimos algumas possíveis direcções para trabalho futuro.

Capítulo 2

Análise Musical

A análise de uma peça musical compreende essencialmente duas partes: o inventário dos elementos que a constituem, e a formulação das regras de combinação e de funcionamento desses elementos. Tradicionalmente, a teoria musical utiliza, nesta tarefa, métodos impressionistas e não formais. Isto é, os musicólogos, ao descreverem estilos ou peças musicais, contentam-se geralmente em mencionar as características que julgam mais importantes, omitindo a descrição do método ou métodos utilizados para atingir a suas conclusões. Nos últimos anos, no entanto, a necessidade da criação de teorias musicais baseadas em abordagens científicas provenientes de áreas como a psicologia cognitiva, a inteligência artificial, a semiótica ou a psicoacústica, mas não só, tem sido defendida por diversos investigadores (ver, por exemplo, [Ruwet, 1972], [Laske, 1992] ou [Camilleri, 1992]).

Neste capítulo, abordamos três teorias musicais formais que surgiram como resposta a esta necessidade, e que julgamos serem relevantes para a compreensão e contextualização, não só do nosso trabalho, mas também das técnicas descritas no capítulo 3. Dada a complexidade e abrangência de qualquer uma das teorias, focamos, de cada uma, apenas os aspectos que julgamos estarem mais relacionados com o nosso trabalho.

A primeira teoria que descrevemos, a análise paradigmática [Nattiez, 1975], fornece uma metodologia geral que permite decompor uma peça musical num conjunto de segmentos musicais e agrupar esses segmentos num conjunto de classes. A segunda, a Teoria Generativa da Música Tonal [Lerdahl & Jackendoff, 1983], fornece um conjunto de regras que explicam as estruturas inferidas por um ouvinte experiente ao ouvir uma peça tonal. Finalmente, a Teoria Geral Computacional da Estrutura Musical [Cambouropoulos, 1998a], é um teoria formulada segundo modelos computacionais e

que, tal como as duas anteriores, permite obter uma descrição estrutural de uma peça musical.

2.1 Análise Paradigmática

A análise paradigmática é o primeiro passo de um tipo de análise mais abrangente denominado análise semiótica. Iremos, por isso, começar por fazer uma breve introdução à análise semiótica de modo a que se entenda melhor o contexto em que a análise paradigmática se insere.

2.1.1 Análise Semiótica

Segundo o dicionário da Porto Editora [Porto Editora], a semiologia é a “*ciência dos sinais*” ou o “*estudo das mudanças que a significação das palavras, como sinais das ideias, sofre no espaço ou no tempo*”. A semiologia musical ou simplesmente análise semiótica [Nattiez, 1975], como é usualmente referida, é considerada um ramo desta ciência. A questão que se coloca de imediato é a de saber em que consiste o estudo da música em termos de sinais e como se pode realizar esse estudo. Em [Cook, 1987] são referidas duas abordagens possíveis. A primeira consiste em estudar o significado da música e a forma como as estruturas musicais nos comunicam diferentes significados. É, no entanto, muito complicado seguir esta abordagem, pois os significados que nos são comunicados pela música são bastante difíceis de estabelecer. A segunda abordagem é inspirada na forma como os linguistas analisam um determinado texto: primeiro decide-se quais são as unidades pertencentes ao texto que tenham significado linguístico, e depois estuda-se a forma como essas unidades se distribuem de modo a estabelecer os princípios que governam a sua distribuição. Assim, segundo esta abordagem, a análise semiótica de uma peça musical divide-se em duas partes denominadas, respectivamente, análise paradigmática e análise sintagmática. A análise paradigmática consiste, em traços gerais, na divisão da peça num conjunto de unidades, seguida do estudo das relações existentes entre essas unidades. A análise sintagmática consiste na construção de um conjunto de regras que expliquem a distribuição das unidades ao longo da peça, bem como as relações existentes entre elas. A validade destas regras pode depois ser confirmada recorrendo, por exemplo, a abordagens generativas em que um conjunto de regras é utilizado para produzir novos

espécimes do fenómeno observado [Sundberg, 1976], [Bent, 1987]. Se for possível produzir espécimes do mesmo estilo que os originais, então as regras, a partir dos quais esses espécimes foram criados, estão correctas.

A análise semiótica de peças musicais, e por conseguinte a análise paradigmática, surgiu devido à necessidade que os musicólogos sentiam de disporem de métodos de análise com carácter científico, em que as obras musicais fossem analisadas apenas pelo seu conteúdo, independentemente das interpretações individuais de cada analista. Pretendia-se também que os analistas tornassem explícitos os critérios utilizados durante a análise de modo que os resultados das suas conclusões pudessem ser confirmados e reproduzidos por outros analistas. Por exemplo, em [Ruwet, 1972] é assinalado que até à época nunca tinha sido explicitamente proposto qualquer modelo formal de análise musical e que mesmo as melhores análises nunca eram acompanhadas da descrição dos critérios de descoberta em que se baseavam. Por outro lado, em [Forte, 1966] é referido que *”Quando um analista experimentado examina uma peça musical, associa determinados sinais com outros de modo a formar unidades e toma uma série de decisões básicas acerca da duração dessas unidades e da sua estrutura interna.[...] Apesar de as decisões de um analista se basearem em anos de prática, elas são frequentemente pouco sistemáticas e sujeitas a muitas influências que não são facilmente identificáveis. Apesar disso, parece evidente que em pelo menos parte do tempo ele toma decisões baseando-se em regras de algum tipo. Isto sugere a possibilidade de que se possam estabelecer regras.”*

É neste contexto que em [Ruwet, 1972] é proposto, numa tentativa de sistematização do processo de análise musical, um procedimento de análise que tem como principal objectivo a inventariação formal e objectiva, das repetições, transformações e contrastes existentes numa peça musical. O objectivo é que esta inventariação seja realizada sem que se tenham em conta, por exemplo, as intenções do compositor ou as percepções de quem a ouve. Pretende-se, portanto, que a análise seja o mais neutra possível, isto é, independente da intuição e do contexto cultural em que o analista está inserido. Em [Nattiez, 1975], documento onde é extensamente discutida a semiologia musical, bem como toda a teoria que a fundamenta, este procedimento foi adoptado como exemplo de como realizar análise paradigmática de peças musicais.

2.1.2 O Procedimento de Análise de Ruwet

O procedimento é relativamente simples e consiste, através de um conjunto de operações, na divisão de uma peça musical em segmentos e na associação desses segmentos em classes de acordo com o grau de semelhança existente entre eles. Como principal critério de divisão, é utilizada a noção de repetição. Esta escolha deve-se, segundo Ruwet, à constatação empírica do papel fundamental desempenhado na música, a todos os níveis, pela repetição. Mas, para que dois segmentos sejam considerados idênticos, ou como uma transformação um do outro, é necessário definir primeiro quais as dimensões (altura e/ou duração das notas, intensidade, timbre, etc.) que serão utilizadas para os comparar. Esta escolha depende do tipo de análise que o analista pretenda realizar. Se, por exemplo, se pretender realizar uma análise rítmica então apenas a duração das notas deverá ser considerada.

Segue-se uma descrição detalhada dos diversos passos em que consiste o procedimento:

- a) *Identificação de repetições elementares*: partindo do princípio de que existe um mecanismo capaz de reconhecer repetições elementares entre segmentos, a primeira operação consiste em identificar os maiores segmentos que se repitam ao longo da peça. Estes são considerados como unidades de nível I. Desta primeira operação resultam estruturas como $A + X + A$, $A + A + X$, $A + A + B + Y + B + X$, etc., em que as letras mais próximas do início do alfabeto correspondem a segmentos repetidos, e as últimas aos restantes segmentos, os quais se designam por restos.
- b) *Tratamento dos restos (primeira fase)*: os restos que tiverem uma duração semelhante à dos segmentos de nível I, já identificados, consideram-se também como elementos de nível I. Por exemplo, se da operação a) tiver resultado a estrutura $A + A + X$ e se X tiver uma duração semelhante à de A então pode-se, pela notação adoptada, substituir X por B resultando na estrutura $A + A + B$. Os resultados desta operação podem ser consolidados recorrendo a outros critérios como sejam, a análise da colocação das pausas ou, se for o caso, através da análise do texto agregado à música.
- c) *Tratamento dos restos (segunda fase)*: podem ocorrer duas situações distintas no

tratamento dos restos não abrangidos pela operação b).

- 1) O *resto* é muito mais pequeno que os segmentos de nível I já identificados. Neste caso, remete-se o *resto* para estados posteriores da análise.
 - 2) O *resto* é muito mais longo que os segmentos de nível I. Neste caso, ou o *resto* é segmentado em unidades de nível I através da aplicação das operações b) e d), ou é segmentado em unidades de nível II através de nova aplicação da operação a).
- d) *Identificação de transformações*: o objectivo desta operação é a identificação de transformações que permitam estabelecer relações entre os segmentos, sejam eles de nível I ou restos. Por exemplo, a estrutura $A + A + X$ poderá ser reescrita, depois de identificada uma relação entre A e X , como $A + A + A'$. É essencial que antes da aplicação desta operação sejam discriminadas todas as transformações a aplicar e os procedimentos que as permitam identificar. Esta operação pode dividir-se em quatro sub-operações que, por sua vez, correspondem à identificação de outros tantos tipos de transformações.
- d1) Identificação de transformações rítmicas e/ou melódicas.
 - d2) Identificação de transformações que envolvam permutações, inserção ou supressão de certos elementos.
 - d3) Identificação de sub-unidades comuns. Para tal, volta-se a aplicar a operação a) aos restos e aos segmentos de nível I já identificados de modo que se cheguem a conclusões do tipo, $A = a + b$ e $B = a + c$, em que a , b e c são unidades de nível II.
 - d4) Consideremos a estrutura $A + x + B + y$, com restos muito pequenos. Se $A + x$ for, do ponto de vista da duração, por exemplo, idêntico a $A + y$, pode-se considerar que $A + x$ é um só segmento e que $A + y$ é uma sua transformação. Neste caso pode-se reescrever a estrutura como $A + A'$.
- e) *Agrupamento de segmentos*. Suponhamos que, como resultado, a operação a) forneceu a estrutura $A + X + A + Y$. Podemos considerar que $A + X$ e $A + Y$ são segmentos de nível 0 se, por exemplo, X e Y terminarem com pausa ou com uma nota longa, ou se durante a operação d) se verificar que Y é

uma transformação de X.

Este procedimento, caso se pretenda, pode ser novamente aplicado aos segmentos de nível I de modo a identificar os segmentos de nível II, e assim sucessivamente, até que os segmentos identificados deixem de poder ser divididos. Ruwet assinala que não é imperativo que as operações sejam executadas pela ordem apresentada para que se cheguem aos mesmos resultados.

Como se referiu em cima, um dos objectivos deste procedimento é a classificação dos segmentos de acordo com o grau de semelhança existente entre eles. Essa classificação, implícita na descrição do procedimento, consiste em agrupar na mesma classe os segmentos que sejam idênticos do ponto de vista das dimensões escolhidas, ou que se relacionem através das transformações utilizadas. Utilizando a notação introduzida, isto equivale a dizer que os segmentos que são representados pela mesma letra formarão uma classe (por vezes, as classes denominam-se por *paradigmas* e ao segmento pertencente a um *paradigma* que primeiro ocorre na peça dá-se o nome de *cabeça do paradigma*). Devemos referir, no entanto, que nos textos sobre análise musical a que tivemos acesso, onde a análise paradigmática é descrita ([Cook, 1987], [Bent, 1987]), [Nattiez, 1975], excepto [Ruwet, 1972]) a classificação é realizada, segundo determinados critérios, numa fase posterior à aplicação do procedimento de Ruwet, servindo este apenas como uma referência na fase de segmentação.

A Figura 2.1 ilustra a análise resultante da aplicação do procedimento ao canto alemão do século XIV «Maria muoter reinû maît» (exemplo retirado de [Ruwet, 1972]). Tal como no exemplo original, os segmentos do segundo nível pertencentes à mesma classe são representados na mesma coluna de modo a facilitar a leitura. Pela figura podemos ver que no primeiro nível existem duas classes: uma formada por A e A' e a outra por B. No segundo nível existem quatro classes: uma constituída por *a*, outra por *b* e as suas duas variações (ambas representadas por *b'*), outra constituída por *c* e outra por *d*.



Figura 2.1 - Análise da peça «Maria muoter reinû maît» por Ruwet.

2.1.3 Problemas da Análise Paradigmática

A análise paradigmática tem sido alvo de algumas críticas que se prendem, sobretudo, com o facto de, na prática, tanto a segmentação como a classificação de segmentos estarem muito dependentes da intuição do analista. Mais especificamente, e no que diz respeito à classificação, é apontado por Nattiez que “*As pessoas decidem associar várias unidades num único paradigma devido a critérios semânticos ou psicológicos que não expressam conscientemente.*” (citado em [Cook, 1987], p.180).

Outro problema, também atribuído a este tipo de análise, diz respeito ao facto de a escolha dos critérios de classificação ser normalmente realizada, explicitamente ou não, apenas em função da peça a analisar [Cook, 1987]. Como consequência, torna-se difícil a comparação de análises de peças diferentes.

Os analistas consideram frequentemente que a *cabeça do paradigma* é o segmento mais representativo do *paradigma*, contra o qual todos os outros são comparados. Acontece que, por vezes, este segmento não é o mais representativo da classe, sendo apenas uma mera primeira ocorrência, o que pode dificultar a identificação de outros segmentos, ou mesmo levar a algumas inconsistências na segmentação. Este problema deve-se ao facto de a análise ser normalmente realizada do princípio para o fim da peça. Em [Anagnostopoulos, 1997] é descrita uma abordagem computacional à classificação de segmentos em que um dos objectivos é a resolução deste problema.

Nesta abordagem, depois de segmentada a peça (parte-se de uma peça já segmentada), cada segmento é representado por um conjunto de características, que correspondem aos critérios que irão ser utilizados na classificação. A classificação é depois realizada utilizando uma rede neuronal, que depois de treinada deve apresentar um conjunto de vectores que representam as classes formadas.

Finalmente, coloca-se a questão de saber até que ponto se pode apreender o significado de uma peça musical através da análise das relações abstractas existentes entre as diversas partes que a constituem [Cook, 1987], mas essa é uma questão que não abordaremos neste trabalho.

2.2 Teoria Generativa da Música Tonal²

A Teoria Generativa da Música Tonal (TGMT) [Lerdahl & Jackendoff, 1983] é uma teoria detalhada sobre hierarquias musicais que consiste numa gramática de música tonal, cujo objectivo é modelar a intuição musical. Esta gramática tem a forma de um conjunto de regras que permitem descrever a estrutura que um ouvinte experiente infere quando ouve uma determinada música. Esta teoria, tal como a próxima, difere da anterior no facto de tentar explicar uma capacidade cognitiva.

Os principais componentes da intuição musical sobre os quais a teoria se debruça são: a estrutura de agrupamento³, que permite expressar a segmentação hierárquica de uma peça em motivos, frases e secções; a estrutura métrica⁴, que expressa a intuição de que os eventos de uma peça estão relacionados a diversos níveis hierárquicos com uma alternância regular de batimentos fortes e fracos; a redução do âmbito temporal⁵, que atribui às notas uma importância hierárquica tendo em conta a sua posição na estrutura de agrupamento e na estrutura métrica; a redução prolongacional⁶, que atribui às notas uma hierarquia que expressa a intuição da tensão e o relaxamento harmónico e melódico, bem como a continuidade e a progressão harmónica e melódica ([Lerdahl & Jackendoff, 1983], págs. 8-9). Destes componentes iremos

² do inglês '*Generative Theory of Music*'

³ do inglês '*grouping structure*'

⁴ do inglês '*metrical structure*'

⁵ do inglês '*time-span reduction*'

⁶ do inglês '*prolongational reduction*'

apenas abordar o primeiro, a estrutura de agrupamento, por ser aquele que mais directamente se relaciona com o nosso trabalho.

Cada componente da TGMT, incluindo a estrutura de agrupamento, é formado a partir de dois tipos de regras: as regras de consistência⁷, que especificam quais as estruturas que é possível formar e as regras de preferência⁸, que especificam quais, de entre as estruturas possíveis, as que mais se aproximam das intuições de um ouvinte experiente. A estrutura de agrupamento é formada por cinco regras de consistência (RC) e sete regras de preferência (RP). Começemos por ver as cinco regras de consistência:

RC1: Qualquer sequência contígua de eventos sonoros pode constituir um grupo, e apenas sequências contíguas podem constituir um grupo.

RC2: Uma peça constitui um grupo.

RC3: Um grupo pode conter grupos mais pequenos.

RC4: Se um grupo G1 contém parte de um grupo G2 então G1 deve conter G2 na totalidade.

RC5: Se um grupo G1 contém um grupo mais pequeno G2, então G1 deve ser exaustivamente dividido em grupos mais pequenos.

Repare-se que a regra RC4 impede que dois grupos se sobreponham. Por seu lado, a regra RC5 impede que depois de realizada a análise existam notas ou grupos de notas não atribuídas a qualquer grupo.

Vejamos agora as regras de preferência (não apresentamos a regra RP7 uma vez que esta regra relaciona a componente de agrupamento com outras componentes da teoria que não iremos descrever):

RP1: Evitar fortemente grupos contendo um único evento (forma alternativa: evitar análises com grupos muito pequenos).

RP2 (Proximidade): Considere-se a sequência $n_1n_2n_3n_4$. Tudo o resto permanecendo

⁷ do inglês 'well-formedness rules'

⁸ do inglês 'preference rules'

igual, a transição n_2-n_3 pode ser ouvida como o limite entre dois grupos se

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

RP3 (*Mudança*): Considere-se a sequência $n_1n_2n_3n_4$. Tudo o resto permanecendo igual, a transição n_2-n_3 pode ser ouvida como o limite entre dois grupos se

- a. (*Registo*) a transição n_2-n_3 envolver uma maior distância intervalar que as transições n_1-n_2 e n_3-n_4 ou se
- b. (*Duração*) n_2 e n_3 têm durações diferentes e ambos os pares n_1, n_2 e n_3, n_4 não diferem na duração.

Em relação a esta regra, são apresentadas também alternativas semelhantes para a dinâmica e a articulação. Os autores referem ainda que a regra pode ser também aplicada a outras dimensões como por exemplo, o timbre.

RP4 (*Intensificação*): Onde os efeitos das regras RP2 e RP3 forem mais pronunciados pode formar-se o limite de um grupo maior (de um nível acima, que englobe grupos mais pequenos).

RP5 (*Simetria*): Preferir análises de agrupamento que favoreçam a subdivisão de grupos em partes de igual duração.

RP6 (*Paralelismo*): Se dois ou mais segmentos da peça puderem ser construídos como paralelos, estes devem preferencialmente formar partes paralelas de grupos.

Como se pode ver, na TGMT, ao contrário do que acontece com o procedimento de Ruwet, a formação de grupos (ou segmentação) é realizada tendo em conta, não só a semelhança entre segmentos (regras RP5 e RP6), mas também aspectos locais como a proximidade ou diferença entre eventos (regras RP2, RP3 e RP5). Não é, no entanto, fornecida qualquer estratégia para a aplicação das regras, nem qualquer indicação do peso ou prioridade de cada regra. Este último aspecto é também importante uma vez

que pode acontecer que, em determinadas situações, a aplicação de duas ou mais regras diferentes resultem em segmentações diferentes, sendo necessário decidir qual das regras aplicar.

Outro aspecto que não é tratado na TGMT, é a forma com se detecta a existência de paralelismo, isto é, a semelhança entre diferentes segmentos. Ou seja, é dada importância à semelhança entre segmentos na formação de grupos, sobretudo através da regra RP6, mas não é fornecido qualquer mecanismo capaz de detectar e medir essa semelhança, tal como é, aliás, reconhecido pelos autores: “[...] *sentimos que a nossa incapacidade para materializar a noção de paralelismo é uma falha na nossa tentativa de formular uma teoria do entendimento musical totalmente explícita.*” ([Lerdahl & Jackendoff, 1983], pág. 53).

Apesar dos aspectos referidos, a TGMT é uma teoria bastante bem formulada que deixa espaço a futuros desenvolvimentos, e que pode ser comparada e testada experimentalmente.

2.3 Teoria Geral Computacional da Estrutura Musical⁹

A Teoria Geral Computacional da Estrutura Musical (TGCEM) [Cambouropoulos, 1998a] é uma teoria que utiliza modelos computacionais para modelar a intuição musical de um ouvinte não experiente. A teoria procura formalizar cinco aspectos da estrutura musical:

1. *Superfície musical.* É o nível mais baixo de representação musical, que corresponde às notas ou intervalos entre as notas, utilizado como ponto de partida para a análise.
2. *Segmentação.* Tal como na teoria anterior, a formalização do modo como é realizada a segmentação tem em conta aspectos locais, como a percepção de discontinuidades (por exemplo, uma nota longa ou um intervalo melódico grande), e aspectos não locais, como a semelhança entre segmentos.
3. *Classificação.* A superfície musical pode ser descrita em termos de classes com significado musical. Cada classe consiste num conjunto de entidades musicais

⁹ do inglês ‘*General Computational Theory of Musical Structure*’

(por exemplo, segmentos) que são associadas em função de um determinado conjunto de critérios. Como se pode ver facilmente, a segmentação e a classificação correspondem à análise paradigmática descrita anteriormente.

4. *Organização temporal*: As categorias podem ser organizadas temporalmente sendo necessário definir as relações e funções existentes entre os diversos materiais musicais. Este aspecto equivale à análise sintagmática, o primeiro passo da análise semiótica, referido anteriormente.
5. *Redução*: Alguns eventos musicais são perceptualmente mais proeminentes do que outros. Estes podem fazer parte e dar origem a representações mais abstractas (reduções) da superfície musical. A segmentação, a categorização e a organização temporal podem ser aplicadas a diferentes reduções de uma peça, permitindo descrições mais sofisticadas que reflectam qualidades hierárquicas do material musical existente na peça.

A formalização destes aspectos assenta num pequeno conjunto de princípios provenientes da psicologia cognitiva, que o autor assume fazerem parte do modo como as pessoas percebem o mundo. O princípio mais fundamental é o princípio lógico de *Identidade-Diferença*: duas entidades são idênticas se partilham exactamente as mesmas propriedades de um determinado domínio, e diferentes em caso contrário. Se for possível ordenar os diferentes valores de uma propriedade e, na comparação entre duas entidades, for possível ter em conta o número de propriedades comuns às duas, então pode-se estabelecer um grau de diferença (ou distância) entre as duas entidades. Os outros princípios em que se baseia a teoria são os princípios cognitivos gerais de *economia*, *suficiência*¹⁰ e *naturalidade*, que permitem limitar os efeitos do princípio de *Identidade-Diferença* que, por ser um princípio lógico e não cognitivo, pode resultar em associações psicologicamente impertinentes.

2.3.1 Estrutura Geral da TGCEM

Nesta secção iremos descrever a estrutura geral da TGCEM, ilustrada na Figura 2.2.

Como informação de entrada, é recebida uma superfície musical (superfície musical (0)) que consiste num conjunto de símbolos que representam as notas da peça a

¹⁰ do inglês 'informativeness'

analisar. Como saída, a teoria produz um conjunto de análises alternativas, cada uma constituindo uma interpretação diferente da estrutura da peça analisada (assume-se, portanto, que não há uma única interpretação de uma peça).

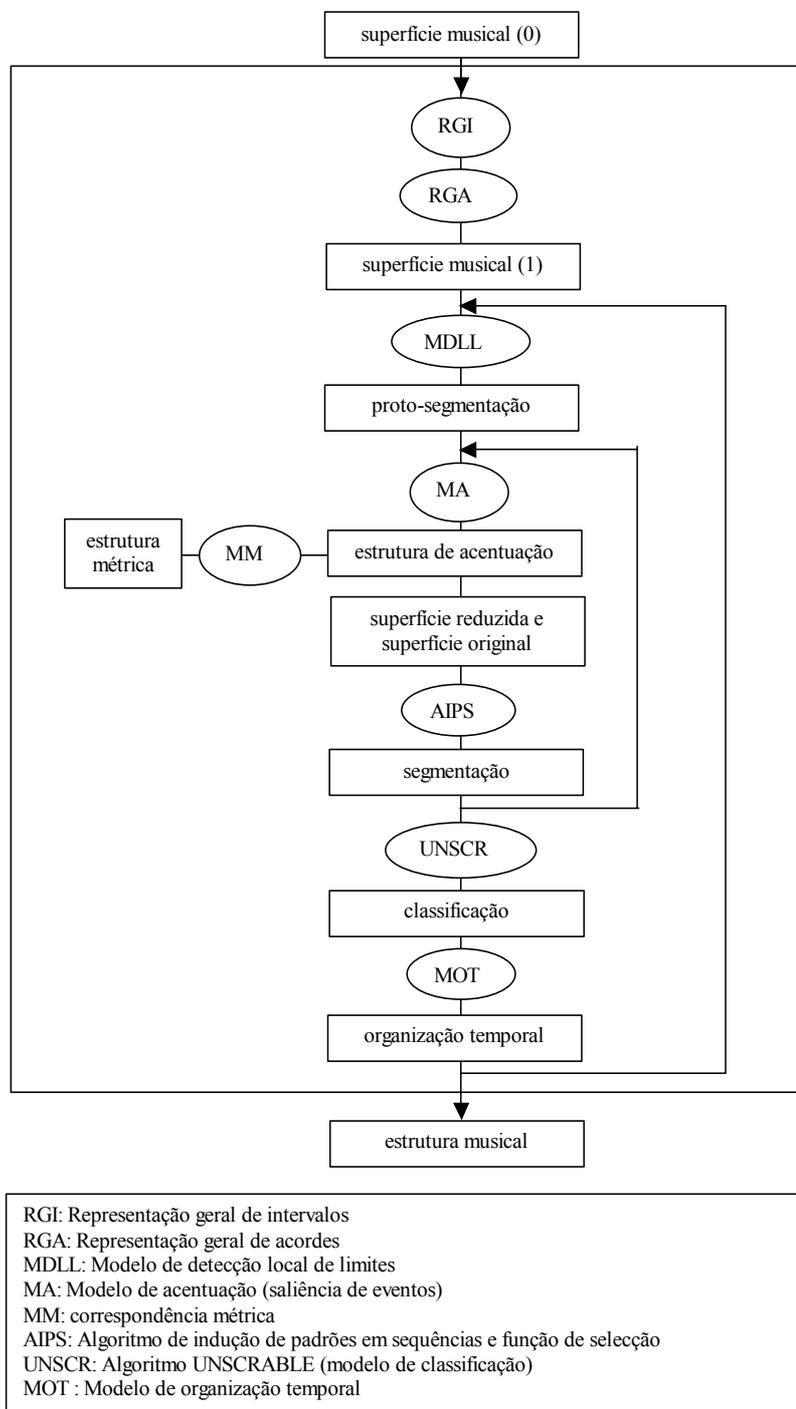


Figura 2.2 - Estrutura geral da TGCEM.

A teoria consiste num conjunto de módulos, alguns dos quais (os mais importantes para a contextualização do nosso trabalho), passamos a descrever:

a) A *Representação Geral de Intervalos (RGI)*. Neste módulo, a representação inicial (superfície musical (0)), baseada no valor absoluto das notas, é convertida para uma representação mais sofisticada (superfície musical (1)), baseada nos intervalos entre as notas (ou intervalos musicais). Esta representação compreende vários tipos de intervalos relativos às dimensões de altura e duração das notas. Relativamente à altura, são derivados, por exemplo, intervalos cromáticos (em meios tons) e intervalos escalares (diferença entre a posição de duas notas numa determinada escala).

b) O *Modelo de Detecção Local de Limites (MDLL)*. O MDLL permite detectar pontos de mudança/descontinuidade máxima na superfície musical. Estes pontos têm maiores possibilidades de ser entendidos pelo ouvinte como limites entre segmentos. O resultado da aplicação deste módulo é uma segmentação provisória denominada por proto-segmentação. Na secção 2.3.2 descreveremos mais detalhadamente este módulo.

c) O *Algoritmo de Indução de Padrões em Sequências (AIPS)*. O AIPS consiste num algoritmo que permite induzir padrões em sequências de símbolos (por exemplo, sequências de intervalos musicais), começando pelos padrões mais pequenos e acabando nos padrões maiores. Depois de aplicado o algoritmo, é utilizada uma função de selecção que permite seleccionar os padrões mais pertinentes do ponto de vista cognitivo. O AIPS complementa o módulo anterior na tarefa de segmentação permitindo revelar passagens musicais paralelas como, por exemplo, motivos, temas, etc., podendo também ser aplicado a reduções da superfície musical original. Uma vez que este módulo faz uso de técnicas que só serão descritas no capítulo 3, deixaremos para esse capítulo a sua descrição mais detalhada.

d) O algoritmo *UNSCRABLE*. Este módulo consiste num algoritmo de aprendizagem simbólica não supervisionado que permite organizar os segmentos resultantes dos dois módulos anteriores em classes que sejam pertinentes em termos cognitivos.

Como se pode ver na Figura 2.2, a TGCEM não é uma teoria linear segundo a qual uma análise é realizada em apenas uma direcção. Em determinadas alturas da análise, é possível voltar atrás e utilizar os resultados já obtidos para refinar o trabalho já

realizado ou para prosseguir para uma análise num nível hierárquico diferente.

2.3.2 Modelo de Detecção Local de Limites

O Modelo de Detecção Local de Limites (MDLL) permite definir limites locais numa determinada superfície musical. Ao contrário do que acontece com a Teoria Generativa da Música Tonal descrita anteriormente, em que são utilizadas regras heurísticas para sugerir algumas localizações prováveis para limites entre segmentos (regras RP2, RP3 e RP5), com o MDLL, para cada ponto da superfície musical é calculado um valor que reflecte as possibilidades de ocorrência de um limite nesse ponto. Este cálculo é baseado nas duas regras que descrevemos a seguir.

Regra Geral da Identidade-Mudança (RGIM): Apenas podem ser introduzidos limites entre duas entidades diferentes. Duas entidades iguais não sugerem a existência de limites entre si.

Se as entidades comparadas forem intervalos (por exemplo intervalos entre a altura das notas ou entre os pontos de ataque) esta regra pode ser formulada de forma mais específica:

Regra da Identidade-Mudança (RIM): Podem introduzir-se limites em qualquer dos intervalos formados por três objectos sucessivos se esses intervalos forem diferentes. Dois intervalos consecutivos idênticos não sugerem qualquer limite.

Quando, através da aplicação desta regra a dois intervalos consecutivos, se detecta uma diferença e se sugere um limite local, esse limite é ambíguo, uma vez que pode ser colocado em qualquer um dos lados do objecto do meio. A segunda regra permite resolver esta ambiguidade dando preferência ao maior dos dois intervalos:

Regra da Proximidade (RP): Se três objectos sucessivos formarem intervalos diferentes entre si, deve introduzir-se um limite no intervalo maior, isto é, os objectos que estão mais próximos (ou que são mais semelhantes) tendem a formar um grupo.

Iremos agora mostrar como é que estas regras são utilizadas no MDLL para calcular, para cada ponto da superfície musical, um valor que reflecta a possibilidade de

ocorrência de um limite (VL)¹¹ nesse ponto.

Os VLs são calculados tal como é ilustrado na Figura 2.3, se as dimensões consideradas forem a altura das notas, a dinâmica, as pausas ou a articulação. Consideram-se as pausas como sendo a distância entre o final da nota anterior e o início (ou ponto de ataque) da próxima. Os fenómenos, como ligamentos ou indicações de respiração, são considerados como sendo pausas expressivas que são inseridas entre as notas a que se referem como se fossem pausas normais, cujo valor é uma fracção da nota precedente.

<p>a)</p> <p style="text-align: center;">• • •</p> <p style="text-align: center;">Dx Dy</p> <p style="text-align: center;">0</p> <p>RIM: 0 0</p> <p><u>RP:</u> 0 0</p> <p>Total: 0 0</p>	<p>b)</p> <p style="text-align: center;">• • •</p> <p style="text-align: center;">Dx Dy</p> <p style="text-align: center;">+</p> <p>RIM: 1 1</p> <p><u>RP:</u> 1 0</p> <p>Total: 2 1</p>	<p>c)</p> <p style="text-align: center;">• • •</p> <p style="text-align: center;">Dx Dy</p> <p style="text-align: center;">-</p> <p>RIM: 1 1</p> <p><u>RP:</u> 0 1</p> <p>Total: 1 2</p>
--	--	--

Figura 2.3 - VLs calculados para três valores de uma determinada dimensão (altura das notas, dinâmica, pausas ou articulação) separados por dois intervalos.

Na figura, cada ponto representa o valor de uma determinada dimensão de uma nota e a distância entre os pontos representa a diferença entre os valores ou intervalo (representada por Dx e Dy , no início de cada intervalo). A relação entre dois intervalos pode ser de identidade ou diferença. Devido à assimetria existente (ver, já a seguir, o cálculo dos VLs quando se considera a dimensão temporal), a relação de diferença entre dois intervalos é representada com os símbolos ‘+’ ou ‘-’, consoante o valor do primeiro intervalo seja, respectivamente, maior ou menor que o do segundo intervalo. Cada regra atribui a cada intervalo um valor segundo o tipo de relação existente (segundo o autor, é necessário mais investigação para escolher os valores mais apropriados):

RIM: 0 para a relação de identidade (0 para cada intervalo)

2 para a relação de mudança (1 para cada intervalo)

RP: 0 para a relação de identidade (0 para cada intervalo)

1 para a relação de mudança (1 para o intervalo maior)

¹¹ Valor de limite, do inglês ‘boundary strength’

Se for considerada a dimensão temporal, que inclui o ponto de ataque e a duração das notas, o cálculo dos VLs é realizado tal como se ilustra na Figura 2.4. A dimensão temporal é tratada de forma diferente de modo a reflectir o facto de que as notas são objectos assimétricos. Esta diferença permite que a estrutura de agrupamento resultante da análise da versão original de uma melodia seja diferente da estrutura de agrupamento resultante da análise da sua versão invertida.

	a)	b)	c)
	0	+	-
RIM (p. ataque):	0 0	1 1	1 1
RP (p. ataque):	0 0	1 0	0 1
RGIM (duração):	0	2	2
Total:	0 0	4 1	3 2

Figura 2.4 - VLs calculados para o valor de três pontos de ataque separados por dois intervalos de pontos de ataque e duas durações.

O cálculo dos VLs para a dimensão temporal tem em conta o ponto de ataque da nota e a sua duração. A duração de uma nota musical é um atributo interno da nota enquanto que os intervalos entre pontos de ataque são distâncias entre dois eventos sucessivos. Assim, para os pontos de ataque, são aplicadas as regras RIM e RP tal como vimos antes. Para a sequência de durações é aplicada a regra RGIM. Neste caso, o valor a atribuir ao intervalo entre as duas durações é 2, se as durações forem diferentes, e 0 em caso contrário. Como se pode ver na Figura 2.4, as relações ‘+’ e ‘-’ não são simétricas. Segundo o autor, não é possível aplicar os princípios de organização perceptual ao domínio temporal da música sem introduzir assimetria local.

O cálculo dos VLs para uma determinada dimensão em melodias mais longas (Figura 2.5) é realizado para todos os tipos de relações (0, +, -) que existam entre todos os pares de intervalos consecutivos. Se ocorrerem três ou mais relações consecutivas do tipo ‘+’ ou ‘-’ (por exemplo, +++, ---), apenas a última contribui para o valor final. Este valor é calculado somando os VLs correspondentes a cada uma das relações em que cada intervalo é interveniente. Na sequência final de VLs, os máximos locais sugerem a ocorrência de limites locais segundo a dimensão considerada.

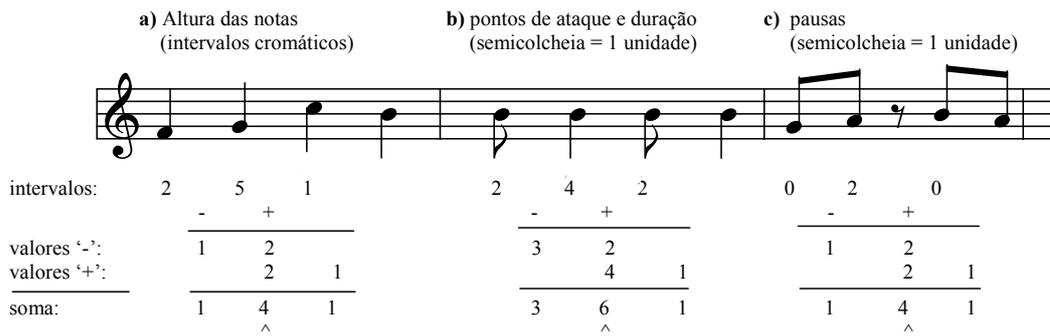


Figura 2.5 - Exemplos de VLs calculados para seqüência melódica com mais de três notas.

O cálculo descrito deve ser realizado para todas as dimensões que se julguem de interesse para a análise a realizar. Depois, as seqüências de valores obtidos para todas as dimensões devem ser somadas (esta soma pode ser pesada, conforme a importância atribuída a cada uma das dimensões consideradas). Tal como acontece para cada dimensão individualmente, os máximos locais indicam pontos preferenciais para a ocorrência de limites entre segmentos, definindo, assim, a proto-segmentação resultante da aplicação do MDLL. A Figura 2.6 ilustra o cálculo destes pontos para o tema inicial da sinfonia em Sol menor de Mozart, considerando os pontos de ataque/durações, a altura das notas (intervalos cromáticos) e as pausas.

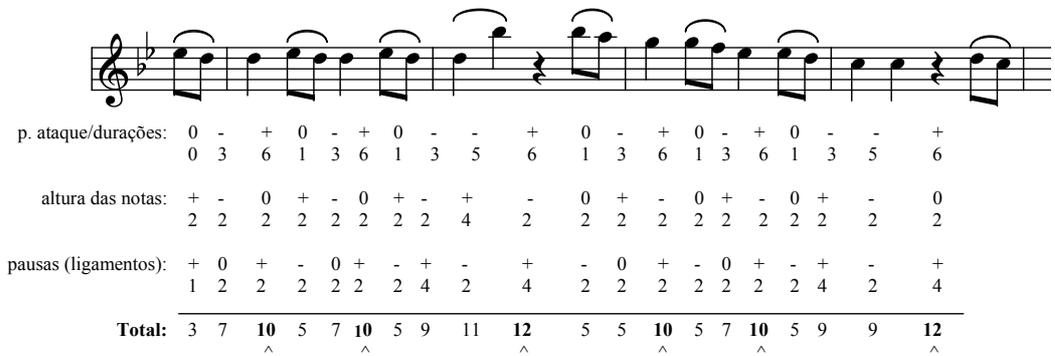


Figura 2.6 - Proto-segmentação do primeiro tema da sinfonia em Sol menor de Mozart.

Capítulo 3

Extracção de Padrões Musicais

Como foi referido no capítulo anterior, a primeira parte da análise de uma peça musical consiste na inventariação dos elementos que a constituem, sendo que parte desta tarefa consiste na sua decomposição num conjunto de segmentos musicais. Vimos, também, que esta decomposição é fortemente baseada na identificação de partes da peça que sejam idênticas, ou muito parecidas. A identificação de padrões, termo que utilizaremos ao longo deste capítulo para designar o material recorrente numa peça, pode ser realizada sobre qualquer dimensão musical a qualquer nível de abstracção: altura, duração, dinâmica ou timbre de notas, acordes, etc. Como veremos mais adiante, uma peça musical é, normalmente, representada como uma sequência de pares ordenados de atributos discretos que podem ser interpretados como símbolos de um alfabeto finito. Assim, pode ver-se uma peça musical como uma sequência de símbolos, sobre a qual é possível aplicar algoritmos de processamento de padrões¹², e a tarefa de identificação de material recorrente como um problema de extracção de padrões.

A extracção de padrões é um problema que se coloca em várias áreas como, por exemplo, a biologia molecular ou o processamento de texto, e que pode ser considerado, do ponto de vista da inteligência artificial, como uma forma de aprendizagem não supervisionada. No domínio musical, o problema pode ser definido, informalmente, da seguinte forma: dada uma peça musical (ou várias), identificar as partes dessa peça que se repitam, aproximadamente ou não, e que possuam um significado relevante no contexto dessa peça.

Como se pode ver por esta definição, nem todos os padrões existentes numa determinada peça interessam, sendo, por isso, necessário definir os critérios que

¹² do inglês *'pattern-matching'*.

permitam avaliar da importância dos padrões encontrados. Neste aspecto, a extracção de padrões diferencia-se de outros problemas de processamento de padrões como, por exemplo, a pesquisa de sequências melódicas em peças musicais. Nestes problemas, é decidido *a priori*, qual a melodia (o padrão) a pesquisar, ou seja, o utilizador, ou outra entidade, decide previamente que aquele é um segmento importante, o que simplifica de algum modo a resolução do problema.

Outra questão que esta definição coloca, é a de saber o que se entende por “ocorrência aproximada de uma parte da peça”. As sequências musicais podem ser transformadas de muitas maneiras de modo a preservar algo do seu carácter original, pelo que se torna necessário definir exactamente o significado desta expressão e quais os métodos que podem ser utilizados para detectar essas transformações. Esta questão, ao contrário da anterior, coloca-se igualmente noutros tipos de problemas de processamento de padrões musicais, como aquele que já referimos.

Neste capítulo, iremos debruçar-nos sobre sistemas computacionais de extracção de padrões musicais. Assim, começamos, na secção 1.1 por descrever a estrutura geral de um sistema de extracção de padrões. Nas secções 1.2 e 1.3 descrevemos respectivamente, algumas das técnicas de representação e de comparação de segmentos musicais mais utilizadas. Como veremos, o sucesso destes sistemas está fortemente dependente da forma como as peças são representadas, bem como das técnicas utilizadas para comparar segmentos. Finalizamos este capítulo descrevendo alguns dos sistemas computacionais mais conhecidos, onde a extracção de padrões desempenha um papel fulcral, quando não central.

Para uma perspectiva geral sobre sistemas de extracção de padrões musicais, ver [Rolland, 2000].

3.1 Estrutura Geral de um Sistema de Extracção de Padrões

A estrutura dos sistemas de extracção de padrões musicais, bem como a definição dos aspectos referidos anteriormente, varia, sobretudo, em função do tipo de objectivos que se pretenda atingir, sendo que as principais diferenças ocorrem entre sistemas que são executados em tempo real e sistemas em que o tempo de resposta não é crucial. No entanto, e apesar dessas diferenças, pode considerar-se que a estrutura geral destes

sistemas se aproxima daquela que é ilustrada na Figura 3.1, onde os dados são representados por rectângulos e os passos executados, por elipses.

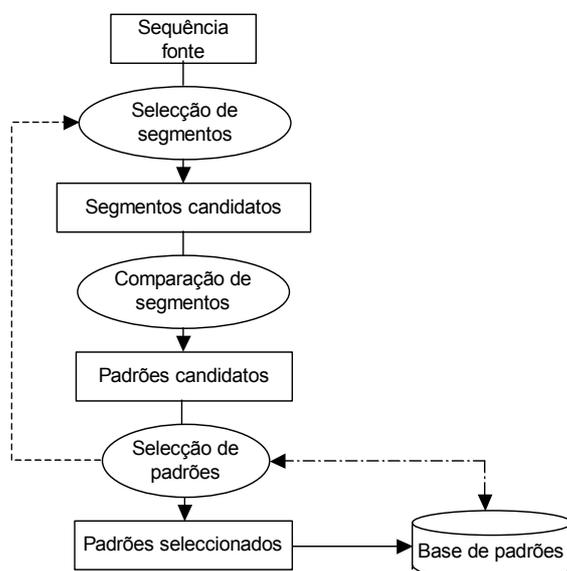


Figura 3.1 - Estrutura geral de um sistema de extração de padrões.

A informação de entrada de um sistema de extração de padrões musicais é um conjunto de peças (por vezes, apenas uma). O primeiro passo do processo consiste em seleccionar os segmentos que irão mais tarde ser comparados entre si de modo a determinar quais os que ocorrem mais do que uma vez. Frequentemente, são seleccionados todos os segmentos possíveis, o que não implica que seja necessário enumerá-los ou representá-los explicitamente. Nos casos em que apenas alguns segmentos são seleccionados, se for necessária uma enumeração explícita torna-se necessário proceder previamente à segmentação das peças. Esta segmentação pode ser baseada, por exemplo, em regras musicais ou em princípios cognitivos, como é o caso do MDLL que vimos no capítulo anterior.

A determinação da natureza recorrente de um segmento é realizada comparando-o com todos os outros segmentos seleccionados, de modo a verificar se existe pelo menos um que seja idêntico ou semelhante a si. Geralmente, da comparação de dois segmentos, resulta um valor numérico que reflecte o grau de semelhança (ou diferença) entre os dois e, se esse valor for superior a um determinado limite, diz-se que os segmentos são equivalentes ou paralelos, assumindo o estatuto de padrões. Este processo pode, depois, resultar na criação de um conjunto de classes, cada uma

constituída apenas por segmentos equivalentes entre si. Ao segmento mais representativo de cada classe dá-se, geralmente, a designação de *protótipo*. Na última secção deste capítulo descreveremos um algoritmo simples, que permite não só realizar a classificação de segmentos, mas também determinar o protótipo de uma classe.

Depois de identificados todos os padrões, segue-se a selecção dos segmentos mais importantes. Esta selecção pode ser baseada em vários critérios, como sejam o tamanho dos padrões (quanto maior, melhor), o número de padrões com os quais existe uma relação de equivalência (*idem*), o grau de sobreposição em relação a outros padrões (quanto menor, melhor), ou o *quorum*, utilizado apenas quando o processo é realizado sobre várias peças, e definido como o número mínimo de peças em que um padrão, ou um seu equivalente, deve ocorrer.

Como já referimos, as principais diferenças na estrutura dos sistemas de extracção de padrões ocorrem entre sistemas executados em tempo real e sistemas em que isso não acontece, sendo que a estrutura dos últimos corresponde quase sempre ao que foi acima descrito. Resta-nos, portanto, fazer referência às características próprias dos primeiros. Estes sistemas são normalmente utilizados como componentes de sistemas interactivos mais abrangentes capazes de modificar o seu comportamento em função da informação musical que vão recebendo em cada momento. Por exemplo, em tarefas de improvisação em grupo, é necessário que o sistema seja capaz de identificar os padrões produzidos pelo(s) companheiro(s) de modo a utilizar esse material, após uma eventual transformação. Assim, nestes sistemas, a peça musical sobre o qual o componente de extracção de padrões opera não é fornecida toda de uma vez (normalmente esta está a ser simultaneamente produzida por outra entidade), sendo necessário repetir várias vezes o processo ilustrado na Figura 3.1. Em cada iteração do ciclo, o sistema recebe uma nova versão da peça, consistindo o primeiro passo em seleccionar os n últimos segmentos, em que n é normalmente um valor pequeno, por vezes 1. Este passo, tal como nos sistemas não interactivos, pode implicar a utilização de técnicas de segmentação local. No passo seguinte os segmentos seleccionados são comparados, embora isso possa não acontecer se apenas o último segmento for seleccionado. A selecção de padrões realiza-se da seguinte forma: todos os segmentos seleccionados são comparados com os padrões encontrados nas iterações anteriores, já

armazenados numa base de padrões; se um segmento for equivalente a um desses padrões, não é guardado, podendo-se, eventualmente, aumentar o valor de importância associado ao padrão já existente; caso contrário, o segmento é considerado único e acrescentado à base de padrões. Os padrões que são armazenados ficam, assim, disponíveis para serem utilizados por outros componentes do sistema de modo a que este possa responder de forma sensível à informação que vai recebendo.

3.2 Representação Musical

Quando se aplicam algoritmos de processamento de padrões em sequências musicais, surgem algumas questões relacionadas com a representação dos elementos que fazem parte de uma sequência deste tipo [Camboroupoulos, 1999b]. Por exemplo, será que uma melodia deve ser representada ao mais baixo nível como uma única sequência de pares ordenados *<altura da nota, duração da nota>*, ou será que as diferentes dimensões devem ser tratadas em sequências separadas? A altura das notas deverá ser representada em termos absolutos (Exemplo: Re#4 ou), em relação a um centro tonal (Exemplo: terceira) ou utilizando intervalos (Exemplo: -4)? A sequência rítmica deve ser representada como uma sequência de durações ou de razões entre durações? Não sendo nosso objectivo discutir estas questões com profundidade, nesta secção iremos descrever algumas das representações mais utilizadas em problemas que envolvem o processamento de padrões em sequências musicais, bem como algumas vantagens e desvantagens da utilização de cada uma delas. Para uma discussão mais aprofundada sobre questões relacionadas com a representação musical consultar, por exemplo, [Anagnostopoulou, 1999], [Camboroupoulos, 1999b], [Selfridge-Field, 1998], [Huron, 1992] ou [Ó Maidín, 1992].

A escolha de uma representação musical adequada tem sido reconhecida por vários autores (os que acabámos de referir, por exemplo) como um factor crítico em qualquer abordagem computacional, e em especial em problemas de extracção de padrões. A representação a utilizar pode depender de vários factores como, por exemplo, o formato da informação (MIDI, imagem de uma pauta, som, etc.) que é fornecida como entrada para o sistema a desenvolver, da quantidade de processamento que se queira/possa realizar de modo a converter essa informação na

representação pretendida e, como é óbvio, do tipo de problema que se pretenda resolver. Existem várias abordagens possíveis para a representação de sequências musicais que podem ser utilizadas em sistemas computacionais, constituindo as representações mais comuns, tais como a notação musical tradicional ou o MIDI, apenas uma parte dessas abordagens. Uma vez que neste trabalho só são utilizadas sequências musicais monofónicas, iremos apenas tratar dos aspectos necessários para representar este tipo de sequências, deixando de lado aspectos específicos da representação de sequências polifónicas. Assim, de agora em diante, quando empregarmos a expressão “sequência musical”, leia-se “sequência musical monofónica”.

Outra questão que não abordaremos será a das representações estruturadas. O conteúdo de uma peça musical pode ser altamente estruturado, pelo que pode ser descrito a diferentes níveis de abstracção, começando na simples descrição de todas as notas que fazem parte da peça (superfície musical), até descrições sucessivamente mais estruturadas (por vezes denominadas por reduções da superfície musical) baseadas, por exemplo, apenas nas notas mais importantes, a nível métrico e/ou harmónico, do nível anterior. Alguns autores, como [Camboroupoulos, 1998a], defendem que, em problemas de processamento de padrões musicais, devem ser utilizadas, preferencialmente, representações estruturadas. No entanto, a tarefa de extrair a estrutura de uma sequência representada de uma forma não estruturada pode não ser trivial. Além disso, as representações estruturadas não abundam, pelo que se torna difícil desenvolver um sistema que tenha informação deste tipo como entrada. Por estas razões, iremos concentrar-nos, apenas, em representações não estruturadas.

3.2.1 Sequências Musicais Monofónicas

Uma sequência musical monofónica é representada, normalmente, como um conjunto de tuplos de atributos, ordenados pela ordem da sua ocorrência no tempo, que descrevem as características das notas ou a relação entre cada duas notas consecutivas pertencentes à sequência. Em algumas abordagens, associam-se também à sequência algumas características de carácter global, como sejam a tonalidade, o andamento, o tipo de compasso, etc. No entanto, esta informação nem sempre é disponibilizada directamente ao sistema, sendo necessário realizar algum processamento extra, ou

pedi-la ao utilizador, para a conseguir. O sistema FExPat, descrito em [Rolland, 1999], utiliza este tipo de informação, mas o autor não descreve a forma como ela é extraída.

Quanto às abordagens à representação individual dos elementos que constituem uma sequência musical, podemos dividi-las em dois grandes grupos: as que utilizam uma representação baseada em notas, e as que utilizam uma representação baseada na relação entre as notas ou intervalos. Em qualquer um destes dois grupos de abordagens é comum representar cada elemento da sequência como um par ordenado \langle informação melódica, informação rítmica \rangle . A estas duas dimensões juntam-se ainda, por vezes, outras como sejam a dinâmica (intensidade) ou o timbre. A Figura 3.2 e a Figura 3.3 ilustram algumas das formas de representação mais comuns da informação melódica e rítmica utilizadas, respectivamente, em abordagens baseadas em notas e em abordagens baseadas na relação entre as notas.



Figura 3.2 - Representações baseadas em notas de uma pequena sequência melódica.

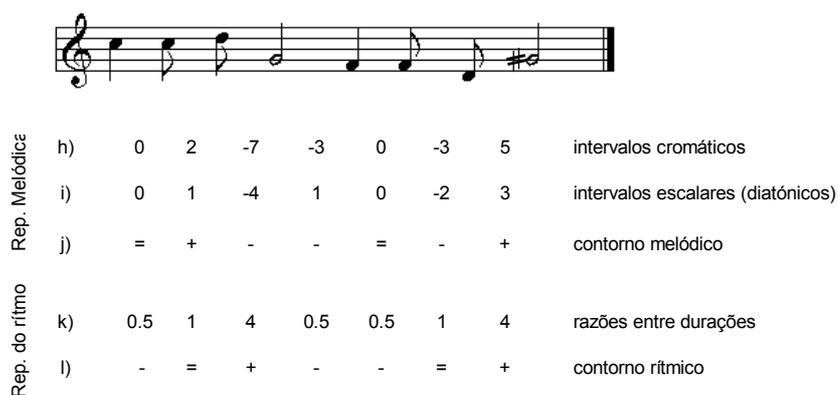


Figura 3.3 - Representações baseadas na relação entre as notas de uma pequena sequência melódica.

3.2.2 Informação Melódica

Existem muitos sistemas computacionais que, para representar a informação melódica, utilizam a notação ocidental tradicional e os valores absolutos das notas (representações *a*) e *b*) na Figura 3.2). Um dos principais problemas da utilização destas abordagens em sistemas de processamento de padrões prende-se com o facto de não ser fácil detectar a semelhança de um segmento musical com outro que seja uma sua transposição (modal ou cromática), obrigando a que o sistema possua funções que permitam realizar essa transformação. Igualmente, é difícil detectar a semelhança de um segmento com outro que esteja numa tonalidade diferente, mas que seja igual, tanto no que diz respeito à relação entre as notas que o compõem, como à relação destas com a tónica, isto é, que seja uma modelação do primeiro (Figura 3.4). À representação *b*) acresce ainda a desvantagem de não ser possível distinguir entre, por exemplo, um *dó sustenido* de um *ré bemol* (ambos tomam o mesmo valor), o que, entre outras coisas, reduz a expressividade da representação.

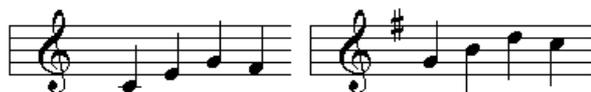


Figura 3.4 - Exemplo de dois segmentos em tonalidades diferentes e cuja relação com a tónica é igual.

As representações em que o valor das notas é dado pela sua relação com a tónica (representações, *c*) e *d*) na Figura 3.2) permitem resolver o problema da modelação de segmentos desde que a relação com a tónica dos dois segmentos seja a mesma. No entanto, no que diz respeito às transposições, apenas permitem identificar semelhanças entre segmentos transpostos à oitava, pois só nesse caso a relação das notas com a tónica permanece inalterada.

Estes problemas podem ser resolvidos recorrendo às representações baseadas em intervalos (representações *b*) e *i*) na Figura 3.3), embora cada uma delas, individualmente, não o permita resolver por completo uma vez que, numa mesma sequência musical, podem existir transposições modais e transposições cromáticas (Figura 3.5). Para ultrapassar este problema, e de modo a cobrir um maior número de situações, alguns sistemas utilizam simultaneamente vários tipos de representações para descrever a informação melódica (ver a Tabela 3.1 no fim deste capítulo).



Figura 3.5 - Sequência com uma transposição modal e uma transposição cromática.

Das representações de que já falámos, as representações *c)*, *d)* e *b)* necessitam de informação acerca da tonalidade em que se encontra a sequência ou das tonalidades locais. Isto implica que, se se quiser representar uma determinada sequência melódica apenas a partir de uma representação baseada em valores absolutos, seja necessário determinar previamente a(s) tonalidade(s) dessa sequência e posteriormente ter que resolver alguns possíveis conflitos como por exemplo: se a tonalidade da peça for *Do*, a nota 68 é um *sol susenido* ou um *lá bemol*? Em [Camboroupoulos, 1998] é descrito um sistema baseado num pequeno conjunto de regras que permite converter uma representação baseada em valores absolutos (representação *b)*) para uma representação em graus diatónicos (representação *b)*). No entanto, este sistema não garante uma conversão totalmente correcta.

Ainda no domínio da representação melódica, há a considerar a representação baseada no contorno melódico (representação *j)*) que não é mais do que uma abstracção da codificação por intervalos, que por sua vez também pode ser considerada uma abstracção das representações baseadas em notas. O contorno permite, em certos casos, revelar semelhanças entre segmentos, difíceis de identificar utilizando outros tipo de representação. Por exemplo, se os segmentos da Figura 3.6 forem codificados utilizando valores absolutos, não é possível identificar nenhuma semelhança significativa; se forem utilizados intervalos cromáticos, é possível detectar um padrão nos cinco primeiros intervalos; se for utilizado o contorno melódico, é possível detectar um padrão que engloba todos os intervalos. No entanto, devido à perda de informação inerente a este tipo de representação, a sua utilização em problemas de processamento de padrões pode levar a que se detectem “falsas” semelhanças entre segmentos, isto é, semelhanças que do ponto de vista musical não fazem sentido.

2 2 4 5 2 -5 1 -2	intervalos cromáticos	2 2 4 5 2-7 3 -4
+ + + + - + -	contorno melódico	+ + + + - + -

2 2 4 5 2 -9 5 -6	intervalos cromáticos	2 2 4 5 2 -11 7 -8
+ + + + - + -	contorno melódico	+ + + + - + -

Figura 3.6 - As primeiras quatro ocorrências de um motivo na peça *Vingt Regards sur l'Enfant Jésus(III-L'échange)* de Messiaen.

3.2.3 Informação Rítmica

Os problemas que se colocam na representação da informação rítmica são muito semelhantes aos que se colocam na representação melódica de uma sequência musical. Assim, as representações baseadas na duração (representações e) e f) na Figura 3.2), apesar de muito utilizadas, não permitem detectar a semelhança entre um segmento musical com outro que seja uma sua variação rítmica, a menos que o sistema permita produzir essa variação. Este problema, como se pode ver na Figura 3.7, pode ser resolvido se for utilizada uma representação baseada em razões de durações, facilmente obtida a partir de uma representação baseada em durações. No entanto, isto só é totalmente válido se a variação for uniforme, isto é, se todas as notas forem multiplicadas pelo mesmo valor. Se isso não acontecer, esta representação permite, ainda assim, identificar semelhanças entre todos intervalos que mantenham a mesma relação entre as durações.

	12	2	2	8	6	1	1	4
duração								
razões	1/6	1	4		1/6	1	4	

Figura 3.7 - Uma sequência rítmica e uma sua variação com metade da duração.

Apesar das desvantagens da utilização exclusiva das durações, isso não invalida que

estes tipos de representação não devam também ser utilizados. De facto, se além de se pretender identificar semelhanças entre segmentos, se pretender também identificar o tipo de variação rítmica (se existir), as durações podem ser de extrema utilidade: no exemplo acima, depois de ser identificada a semelhança entre os dois segmentos, utilizando uma representação baseada em razões, se dividirmos a duração da segunda sequência pela duração da primeira chegamos ao valor que permitiu realizar a variação. Isto não seria possível se apenas fosse utilizada uma representação baseada em razões.

Quanto às representações baseadas na duração, há ainda a referir que a representação *f)* tem, sobre a representação *e)*, a vantagem de ser imune às variações de andamento, pois enquanto que a primeira apenas depende da unidade de tempo do compasso, que permanece inalterada ao longo da peça¹³, a segunda reflecte todas as alterações de andamento, dificultando a identificação de eventuais semelhanças entre segmentos.

3.3 Comparação de Sequências Musicais

Existe uma grande variedade de algoritmos de processamento de padrões, normalmente aplicados em sequências de texto ou sequências biológicas (ADN ou sequências de proteínas, por exemplo), que são também utilizados na resolução de problemas no domínio musical que necessitem deste tipo de processamento. Destes problemas fazem parte, por exemplo, a pesquisa de peças musicais em bases de dados de grandes dimensões, onde se pode tentar encontrar uma peça musical cantando ou escrevendo apenas uma pequena parte, a improvisação em grupo ou, como é o nosso caso, a segmentação de uma peça musical. Em qualquer destes casos, a comparação de segmentos musicais ocupa um papel central, na medida em que a escolha de um método apropriado ao problema que se pretenda resolver afecta fortemente os resultados finais. Nos textos [Meredith, 2001], [Perttu, 2000], [Lemström, 2000], [Cambouropoulos, 1999a], [Rolland, 1999], [Crawford, 1998], [Ó Maidín, 1998], [Smith, 1998] ou [Mongueau & Sankoff, 1990] podem ser encontrados vários algoritmos de processamento de padrões em sequências musicais, e em especial algoritmos de comparação. Para uma descrição de vários algoritmos de processamento de padrões em sequências genéricas de símbolos, incluindo, também, a comparação de

¹³ Isto pode não ser válido se houver, no decorrer da peça, uma mudança de compasso.

sequências, consultar [Stephen, 1992] ou [Crochemore, 1998].

Segundo [Camborououlos, 1999b], existem essencialmente dois tipos de abordagem à comparação de sequências musicais: a comparação exacta e a comparação aproximada. A primeira abordagem é, normalmente, aplicada não só à superfície musical mas também a várias reduções desta. Para cada nível hierárquico a comparação é realizada várias vezes, considerando, de cada vez, apenas uma dimensão musical (ex: primeiro apenas o ritmo, depois apenas a dimensão melódica, etc.). Esta abordagem baseia-se no pressuposto de que dois segmentos musicais só devem ser considerados como equivalentes ou paralelos se forem, num dos níveis hierárquicos considerados (superfície musical ou uma das reduções), exactamente iguais segundo pelo menos uma dimensão musical. Assim, uma vez que a distância entre os dois segmentos tem que ser 0, não é propriamente necessário um método que calcule o quão diferentes eles são, mas que apenas permita saber se são diferentes ou não. Um dos métodos tradicionais de calcular a distância entre duas sequências de símbolos com o mesmo tamanho, suficiente para este problema, denomina-se por distância de Hamming. Neste método, cada símbolo de uma sequência é comparado com o símbolo que ocupa a mesma posição na outra sequência, e a diferença é dada pelo número de desigualdades encontradas (por vezes considera-se que dois símbolos são iguais se a sua diferença for menor que um determinado valor). No caso da comparação exacta, assim que se encontrar dois símbolos diferentes, pode-se interromper o processo de comparação. De entre as vantagens desta abordagem salientamos o facto de o tempo de comparação entre dois segmentos ser de ordem linear (proporcional ao tamanho n dos segmentos), e o facto de se tornarem explícitas as razões por que dois segmentos são considerados equivalentes (exemplo: dois segmentos são equivalentes porque têm o ritmo igual). Por outro lado, esta abordagem exige uma quantidade significativa de processamento sobre a superfície musical de modo a que os segmentos possam ser comparados a diferentes níveis hierárquicos.

A segunda abordagem é, normalmente, aplicada apenas a descrições não estruturadas das sequências musicais (superfícies musicais) e baseia-se no pressuposto de que dois segmentos podem ser considerados como equivalentes sem que haja uma correspondência total entre todos os seus elementos. Assim, a comparação de dois segmentos implica a escolha de uma função de distância que, conjuntamente com um

algoritmo que implemente essa função, permita calcular a distância entre os dois segmentos. Uma vez que esta abordagem é a mais generalizada, sendo também a que é seguida no nosso trabalho, iremos dedicar-lhe o resto desta secção.

3.3.1 Distância de Levenshtein

Na teoria moderna da comparação de sequências finitas de símbolos, a abordagem geral consiste em procurar, de entre todos os alinhamentos possíveis, um alinhamento óptimo entre os elementos das duas sequências a comparar (que não têm que ter necessariamente o mesmo tamanho), satisfazendo algumas condições como, por exemplo, a preservação da ordem dos elementos. Ao invés, pode procurar-se o alinhamento que implique o menor número de elementos não alinhados, caso em que a comparação se torna no cálculo da distância entre as duas sequências. Uma forma natural de quantificar a distância entre duas sequências de símbolos $A = a_1a_2 \dots a_m$ e $B = b_1b_2 \dots b_n$ consiste em calcular o número mínimo de operações de edição (escolhidas de entre um conjunto pré-definido de operações) necessárias para transformar uma sequência na outra. Quando estas operações são as operações de remoção, inserção, e substituição de um símbolo por outro, designa-se a distância calculada como distância de Levenshtein.

Tal como mostra a Figura 3.8, se utilizarmos a distância de Levenshtein, o número mínimo de operações para transformar a sequência $A = \text{“abcdefg”}$ na sequência $B = \text{“acdefhi”}$ são três: remoção de ‘b’, substituição de ‘g’ por ‘h’ e inserção de ‘i’. Como também se pode ver na figura, quando se calcula esta distância, um símbolo pode estar ligado a apenas um outro símbolo e as linhas nunca se podem cruzar.

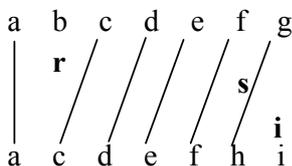


Figura 3.8 - Correspondência entre os símbolos de duas sequências.

Estes conceitos podem ser generalizados associando a cada operação um peso. Desta forma, a distância entre duas sequências, deixa de ser necessariamente o número de operações, mas a soma mínima de pesos de entre todas as séries de operações que

convertam uma seqüência na outra.

3.3.2 Programação Dinâmica

O método de programação dinâmica de Wagner e Ficher [Stephen, 1992] permite calcular a distância de Levenshtein entre duas seqüências de símbolos, baseando-se no cálculo da distância entre prefixos das seqüências cada vez mais longos, à custa dos resultados obtidos para prefixos mais curtos. Seja d_{ij} a distância entre $a_1a_2 \dots a_i$ e $b_1b_2 \dots b_j$ e $w(a, \phi)$ o peso associado à operação de remover a_i , $w(a, b_j)$ o peso associado à operação de substituir a_i por b_j (assume-se que $w(a, b_j) = 0$ se $a = b_j$), e $w(\phi, b_j)$ o peso associado à operação de inserir b_j , o cálculo da distância entre duas seqüências é realizado, em tempo proporcional a $m \times n$, através da equação

$$d_{ij} = \min \begin{cases} d_{i-1,j} + w(a_i, \phi) \\ d_{i-1,j-1} + w(a_i, b_j) \\ d_{i,j-1} + w(\phi, b_j) \end{cases}$$

para $1 \leq i \leq m$ e $1 \leq j \leq n$, e com as seguintes condições iniciais:

$$\begin{aligned} d_{00} &= 0; \\ d_{i0} &= d_{i-1,0} + w(a_i, \phi), i \geq 1; \\ d_{0j} &= d_{0,j-1} + w(\phi, b_j), j \geq 1 \end{aligned}$$

A Figura 3.9 ilustra o cálculo da distância entre as seqüências “abcdefg” e “acdefhi” com os pesos de todas as operações iguais a 1. O valor da distância entre as duas seqüências é dado pelo valor calculado na posição do canto inferior direito da matriz.

	j	0	1	2	3	4	5	6	7
i		a	b	c	d	e	f	g	
0		0	1	2	3	4	5	6	7
1	a	1	0	1	2	3	4	5	6
2	c	2	1	1	1	2	3	4	5
3	d	3	2	2	2	1	2	3	4
4	e	4	3	3	3	3	1	2	3
5	f	5	4	4	4	4	2	1	2
6	h	6	5	5	5	5	3	2	2
7	i	7	6	6	6	6	4	3	3

Figura 3.9 - Cálculo da distância das seqüências “abcdefg” e “acdefhi”.

No final do cálculo da distância entre as seqüências, é possível saber quais são as operações que permitem transformar uma seqüência na outra. Para isso, basta fazer o

caminho desde o canto inferior direito da matriz até ao canto superior esquerdo seguindo sempre pelos menores valores que se encontrarem. Sempre que se passar de um valor maior para um menor isso significa a realização de uma operação de edição: se o valor menor estiver na mesma coluna e na linha acima isso significa a inserção na sequência horizontal, de um carácter da sequência representada na vertical; se o valor menor estiver na mesma linha e na coluna imediatamente à esquerda isso significa a remoção de um carácter na sequência horizontal; finalmente, se o valor menor estiver na coluna imediatamente à esquerda e na linha acima isso significa a substituição de um carácter na sequência horizontal, por um carácter da sequência representada na vertical. Repare-se que podem existir vários caminhos possíveis, correspondendo cada caminho a um conjunto de operações alternativo. Por exemplo, no quadro anterior, dois dos caminhos possíveis são $\{[7, 7], [6, 7], [5, 6], [4, 5], [3, 4], [2, 3], [2, 2], [1, 1], [0, 0]\}$ ou $\{[7, 7], [6, 6], [5, 6], [4, 5], [3, 4], [2, 3], [1, 2], [1, 1], [0, 0]\}$, em que cada posição do caminho tem o significado $[linha, coluna]$.

3.3.3 Programação Dinâmica e Comparação de Sequências Musicais

A programação dinâmica tem sido ao longo dos últimos anos o método mais utilizado quando se pretendem comparar duas sequências musicais. De entre as adaptações deste método especialmente concebidas para a comparação de sequências musicais, a mais conhecida é a de [Mongueau & Sankoff, 1990]. A função de distância proposta por estes autores opera sobre sequências constituídas por pares $\langle \text{distância em semitons da nota à tónica, duração da nota em número de semicolcheias} \rangle$, sendo o custo de substituição de uma nota por outra dado por

$$w(a_i, b_j) = w_{\text{intervalo}}(a_i, b_j) + k \times w_{\text{duração}}(a_i, b_j),$$

onde $w_{\text{intervalo}}(a_i, b_j)$ corresponde a um valor que é tanto maior quanto maior for a dissonância entre as duas notas, $w_{\text{duração}}(a_i, b_j)$ corresponde à diferença de duração entre as duas notas, e onde k representa a contribuição relativa do valor da diferença de durações. Por seu lado, as operações de inserção e remoção são simplesmente definidas, respectivamente, como $w(\phi, b_j) = \text{duração}(b_j)$ e $w(a_i, \phi) = \text{duração}(a_i)$, onde $\text{duração}(n)$ corresponde ao valor da duração da nota n .

Além da adaptação do cálculo dos pesos das três operações típicas, Mongueau e Sankoff propõem também mais duas operações de edição denominadas por fragmentação e consolidação. A operação de fragmentação consiste em substituir uma nota por um conjunto de notas, enquanto que a operação de consolidação consiste na operação contrária. Tal como ilustra a figura seguinte, estas operações implicam que a condição referida anteriormente, de que a cada nota de uma sequência corresponda apenas uma outra na outra sequência, já não tenha que se verificar.

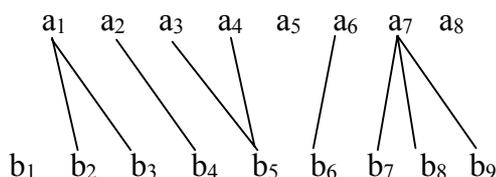


Figura 3.10 - Alinhamento possível entre duas sequências quando se utilizam as operações de fragmentação e consolidação.

À semelhança do que acontece com a operação de substituição, o peso destas operações é uma combinação linear de $w_{intervalo}$ e de $w_{duração}$. Para a fragmentação, $w_{intervalo}$ é definido como a soma dos $w_{intervalo}$ s entre cada uma das notas substitutas e a nota substituída, enquanto que $w_{duração}$ é definido como a diferença entre a duração total das notas substitutas e a duração da nota substituída. Por exemplo, na figura seguinte, uma vez que a soma da duração das notas substitutas é igual à duração da nota substituída, o peso associado à diferença total é zero. Por sua vez, $w_{intervalo}$ será igual à soma dos $w_{intervalo}$ s individuais correspondentes às linhas que ligam cada nota substituta à nota substituída. O cálculo do peso associado a uma operação de consolidação é realizado do forma similar mas, neste caso, a noção de nota substituta e nota substituída é invertida.

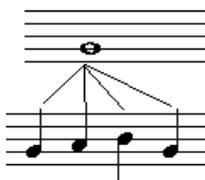


Figura 3.11 - Exemplo de uma fragmentação.

No sistema FLEXPAT [Rolland, 1999] (descrito mais abaixo), além das operações de

fragmentação e de consolidação, são também propostas as seguintes operações: multi-remoção, que consiste em remover várias notas simultaneamente; multi-inserção, o contrário da anterior; e multi-substituição, que consiste em substituir, de uma vez só, um conjunto de notas de uma sequência por outro conjunto de notas da outra sequência. Além disso, neste trabalho, é calculada uma função de semelhança em vez de uma função de distância, estratégia que é também adoptada, por exemplo, em [Rowe, 1993] ou [Dannenberg, 2000].

3.4 Sistemas de Extração de Padrões Musicais

Nesta secção iremos descrever alguns dos sistemas de extração de padrões musicais mais conhecidos. De modo a que o leitor mais facilmente se aperceba da evolução que tem havido nesta área, os sistemas serão descritos por ordem cronológica. No final do capítulo apresentaremos uma tabela com uma súmula das principais características de cada um.

3.4.1 EMI

O objectivo do sistema EMI ([Cope, 1991], [Cope, 1996]) é a replicação de estilos musicais, ou seja a composição de peças musicais ao estilo de um determinado compositor. Para tal o sistema começa por tentar extrair, de duas ou mais peças do compositor, um conjunto de padrões musicais considerados como “suficientemente recorrentes”, que o autor designa por assinaturas. Estes padrões consistem em sequências de notas representadas por pares *<informação melódica, informação rítmica>* ou por apenas umas destas dimensões. No processo de (re)composição, os padrões extraídos são fixados numa estrutura estilística predefinida (por exemplo, a estrutura da forma sonata), e uma rede aumentada de transições¹⁴ completa o resto da peça utilizando regras de composição, também elas aprendidas a partir do conjunto de peças fornecido.

O processo de extração de padrões consiste nos seguintes passos:

- Primeiro, procuram-se todos os padrões existentes em cada peça (sequência musical) cujo tamanho seja igual a um determinado valor *l* previamente definido

¹⁴ Do inglês ‘Augmented Transition Network’

pelo utilizador. Para isso, todos os pares de segmentos possíveis com tamanho igual a l são comparados tendo em conta apenas a dimensão rítmica, apenas a dimensão melódica ou as duas em conjunto. A informação melódica é representada sob a forma de intervalos cromáticos, de modo a tornar a comparação insensível às transposições. Uma vez que o tratamento da informação rítmica neste sistema é algo complexo, iremos descrever apenas a comparação de segmentos em que só a informação melódica é considerada. Assim, dois segmentos “ a_1, \dots, a_n ” e “ b_1, \dots, b_n ” são considerados equivalentes, passando a ser considerados como candidatos a assinatura, se para cada $k \in \{1, \dots, n\}$, $a_k \approx b_k$, onde $a_k \approx b_k$ significa que os dois intervalos a_k e b_k diferem no máximo em d semitons em que d é um parâmetro do sistema (tipicamente, $d = 1$).

- Segundo, os padrões encontrados em cada peça são comparados com os padrões encontrados nas outras peças. Se um padrão ocorrer em mais do que uma peça assume o estatuto de assinatura, podendo vir a ser utilizado posteriormente na composição de novas peças.

Como se pode ver por esta descrição, no EMI é utilizada uma espécie de comparação exacta de segmentos em que uma nota só é comparada com outra que ocupe a mesma posição, podendo haver uma pequena diferença entre elas. Embora, tal como foi referido antes, haja autores que defendam este tipo de comparação quando os segmentos são comparados a vários níveis hierárquicos, neste sistema os segmentos apenas são comparados ao nível da superfície musical, pelo que nos parece que o método utilizado é algo limitado para o problema.

Outro aspecto que consideramos uma forte limitação deste sistema é o facto de, em cada sessão de extracção de padrões, apenas serem considerados segmentos de um determinado tamanho previamente escolhido pelo utilizador. Repare-se que mesmo que as assinaturas de um determinado compositor fossem todas do mesmo tamanho (situação muito pouco verosímil) haveria, ainda assim, que escolher um valor correcto para o tamanho dos segmentos, de modo a que o processo fosse bem sucedido. De facto, segundo o próprio autor “*o controlador *tamanho-de-padrão* requer alguma experiência uma vez que a escolha de um valor demasiado pequeno implicará a identificação de apenas partes de assinaturas, enquanto que a escolha de um valor demasiado grande pode impedir a descoberta de*

qualquer assinatura?

3.4.2 Charm

Em [Smaill, 1993] é proposto o sistema geral de representação musical Charm, que segundo os autores, não é baseado em qualquer estilo musical, sistema tonal ou aplicação. De entre todos os aspectos previstos por este sistema de representação, interessa apenas aqui referir que cada nota é descrita pela estrutura

<<nome, acidente, oitava>, posição, lista de figuras rítmicas>

em que *posição* indica o sítio da peça em que ocorre a nota (distância temporal em relação ao início da peça) e *lista de figuras rítmicas* consiste num conjunto de figuras rítmicas básicas cuja soma dá a duração total da nota. De modo a mostrar a universalidade do sistema, os autores descrevem alguns exemplos da sua aplicação à composição e à análise. No que diz respeito a esta última, é proposto um algoritmo simples, baseado no procedimento de Ruwet já descrito no capítulo anterior, que consiste nos seguintes passos:

1. Primeiro, procura-se na peça segmentos que sejam repetições exactas de um segmento anterior. Quando todas as repetições forem identificadas, removem-se do espaço de pesquisa todas as ocorrências do segmento em questão.
2. Depois de todas as repetições exactas terem sido removidas, procuram-se segmentos que sejam uma transformação do segmento tratado no passo anterior, sendo as transformações possíveis previamente definidas. Também estes segmentos são removidos do espaço de pesquisa.
3. Repetir os dois passos anteriores até que não se consigam descobrir mais relações entre os restantes segmentos. Tal como no procedimento de Ruwet, no final deste processo poderão existir alguns segmentos não relacionados com nenhum dos segmentos identificados.

Repare-se que em cada iteração do algoritmo é formada implicitamente uma nova classe constituída pelos segmentos identificados nessa iteração. A descrição do algoritmo é omissa, no entanto, quanto ao modo como são seleccionados os segmentos a comparar e quais os critérios de selecção utilizados. Não é possível perceber, por exemplo, se a segmentação é realizada antes, e por que método, ou se

durante o algoritmo e por que ordem (começa-se por comparar segmentos mais pequenos ou comparam-se primeiro segmentos maiores?). Se for realizada antes, então a função do algoritmo consiste apenas em associar em diferentes classes um conjunto de segmentos pré-seleccionados, utilizando, para os comparar, as transformações definidas. Repare-se também que este algoritmo é capaz de identificar todas as transformações que forem definidas mas apenas essas.

Os autores exemplificam o funcionamento do algoritmo com a peça «Syrinx» de Claude Debussy cuja pauta apresentamos no Anexo I (na peça assinalámos os segmentos com 3 ou mais notas que devia ser possível identificar utilizando as transformações que foram utilizadas pelos autores). As transformações utilizadas para analisar esta peça foram as seguintes:

- *Identidade* - verdadeiro se um segmento for exactamente igual a outro.
- *Identidade Longa* - verdadeiro se um segmento for igual a outro, excepto no facto de a primeira nota ter uma duração maior.
- *Transposição* - verdadeiro se um segmento for a transposição em um número constante de meios tons (transposição cromática) de um outro segmento.
- *Transposição livre* - verdadeiro se um segmento for a transposição em um número constante de meios tons de um outro segmento, independentemente da duração das notas.

De seguida mostramos um excerto do relatório fornecido pelo sistema durante a execução do algoritmo (Figura 3.12), seguido das transformações identificadas (Figura 3.13).

```
Performing analysis in the style of Ruwet of syrx
Found repeated phrase mtf2 at e000 and e014
Searching for other occurrences of mtf2
Found all occurrences of mtf2
Searching for phrases related to mtf2
Found phrase similar to mtf2 under transpose(-12) at e000 and e039
Found phrase similar to mtf2 under transpose(-12) at e000 and e049
Found phrase similar to mtf2 under longer_identity at e000 and e140
Found phrase similar to mtf2 under longer_identity at e000 and e155
Found phrase similar to mtf2 under loose_transpose(-12) at e000 and e820
Found phrase similar to mtf2 under loose_transpose(-12) at e000 and e830
Found all phrases similar to mtf2 under known transformations
Found repeated phrase mtf11 at e024 and e028
Searching for other occurrences of mtf11
Found all occurrences of mtf11
Searching for phrases related to mtf11
...
```

Figura 3.12 - Relatório fornecido durante a execução do algoritmo de análise.

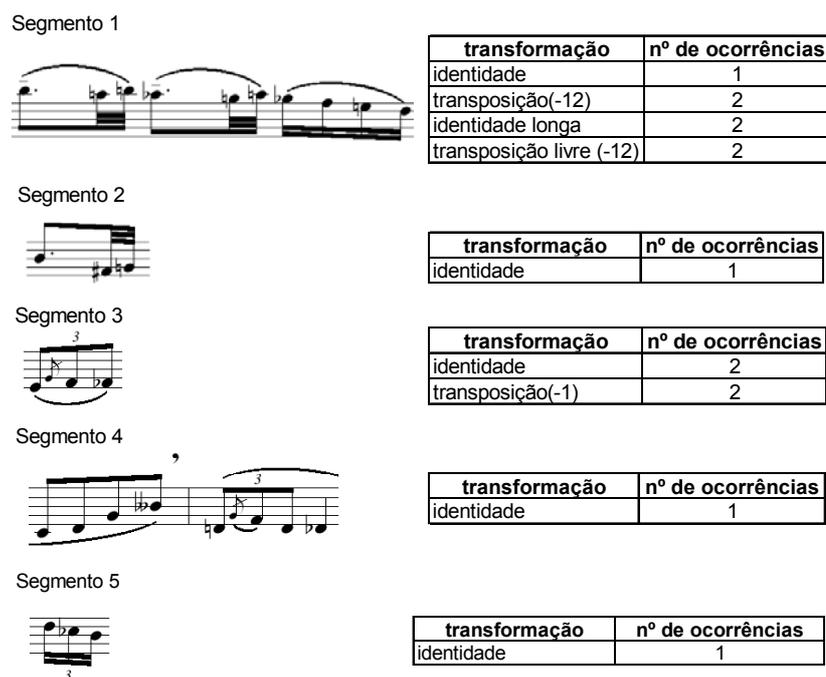


Figura 3.13 - Transformações identificadas pelo algoritmo.

Como se pode constatar, o algoritmo não identifica todas as transformações que devia identificar¹⁵ (comparem-se os segmentos que assinalámos na pauta a título de exemplo e os segmentos da figura anterior). Isso leva-nos a crer que nem todos os segmentos possíveis são seleccionados, ou seja, algumas transformações não são identificadas simplesmente porque os segmentos envolvidos não são comparados.

3.4.3 Sistemas Musicais Interactivos

O sistema descrito em [Pennycook, 1993] que, por comodidade, iremos designar por PSR, e o sistema Cypher [Rowe, 1993] são sistemas musicais interactivos em tempo real que possuem uma estrutura muito parecida. Ambos são baseados na interacção entre vários componentes, dos quais destacamos o *ouvinte*¹⁶ e o *artista*¹⁷, cuja função é, respectivamente, extrair padrões do material musical que vai chegando vindo dos outros companheiros de improvisação, e reutilizar esses padrões para responder a esse

¹⁵ Além disso, as duas transposições identificadas para o segmento 3 não parecem ser válidas se tivermos em conta a descrição que os autores fazem da transformação *Transposição*. Os segmentos, que são identificados como transposições do segmento 3 estão assinalados na pauta como E⁷.

¹⁶ do inglês 'listener'.

¹⁷ do inglês 'player'.

material.

No âmbito do *owinte*, as sequências de notas recebidas, representadas com valores absolutos, começam por ser transformadas de modo a que a comparação de segmentos seja mais sensível. Assim, enquanto que no Cypher apenas é utilizada a dimensão melódica, sendo as sequências recebidas transformadas em sequências de intervalos cromáticos, no PSR são utilizados pares <intervalos cromáticos, razões de durações>. Estas novas sequências são depois segmentadas por sub-componentes especializados. No Cypher, por exemplo, a segmentação é realizada, um pouco à semelhança do que acontece no MDLL descrito no capítulo anterior, tendo em conta descontinuidades locais que ocorram simultaneamente ao nível de várias dimensões, como sejam a informação melódica (intervalos grandes) ou a dinâmica (mudanças bruscas de intensidade sonora).

A fase de comparação de segmentos no Cypher consiste em comparar os dois últimos segmentos identificados utilizando o método da programação dinâmica. Estes segmentos são considerados equivalentes, se houver uma correspondência entre pelo menos quatro notas, independentemente do tamanho dos segmentos. A partir da descrição dos autores não é possível, no entanto, determinar se ambos os segmentos passam à fase de selecção de padrões, se apenas um deles, ou se é construído um padrão a partir dos dois segmentos, por um qualquer processo de generalização. De qualquer modo, é claro que, se os dois segmentos não forem considerados equivalentes, não é criado nenhum padrão candidato à fase seguinte, o que implica que os dois segmentos não serão posteriormente aproveitados pelo *artista*. Note-se que, deste modo, um segmento só pode assumir o estatuto de padrão caso se repita uma vez consecutivamente, o que é uma forte limitação no processo de identificação de padrões. Parece-nos, também, que o estabelecimento de um valor fixo para o limite mínimo de notas correspondentes a partir do qual se consideram equivalentes dois segmentos, é uma opção arriscada e desnecessária uma vez que, se os segmentos tiverem uma dimensão que ultrapasse consideravelmente este valor, facilmente se podem identificar falsas equivalências. Esta opção poder-se-ia justificar dada a necessidade de uma resposta rápida do sistema e caso a escolha deste limite para cada dois segmentos particulares fosse uma operação morosa, o que não é o caso, pois este valor pode ser calculado facilmente em função do tamanho dos segmentos. No

entanto, é possível que os autores do sistema imponham também um limite máximo para o tamanho dos segmentos de modo a que este valor seja sempre aceitável.

A fase de comparação de segmentos no PSR não existe uma vez que apenas o último segmento é considerado para a fase de selecção de padrões. Neste caso, não foi possível, através da descrição dos autores, determinar se esta escolha implica que alguns segmentos sejam ignorados, o que seria também uma forte limitação, pois algum do material musical recorrente poderia não ser aproveitado, ou se simplesmente o *ouvinte* trata um segmento de cada vez, caso em que todos os segmentos seriam candidatos à fase de selecção de padrões.

À semelhança do que foi referido na segunda secção deste capítulo, a fase de selecção de padrões nestes dois sistemas consiste em comparar cada um dos padrões identificados na fase anterior com os padrões seleccionados em iterações anteriores, entretanto armazenados numa base de padrões. Aqui, mais uma vez, a comparação é realizada, em ambos os sistemas, utilizando o método da programação dinâmica. Se um padrão for considerado equivalente a um dos padrões já armazenados, não é acrescentado à base de padrões aumentando-se, isso sim, o valor de importância associado ao padrão equivalente já armazenado; caso contrário o padrão é acrescentado à base de padrões. Como já referimos, os padrões armazenados ficam depois disponíveis para serem utilizados pelo *artista*, sendo a probabilidade de utilização de cada padrão tanto maior quanto maior for o seu valor de importância.

3.4.4 FIEXPAT

O sistema FIEXPAT [Rolland, 1999], ao contrário dos anteriores, foi concebido com o fim último de identificar os padrões existentes numa ou mais peças musicais. Neste sistema cada nota pertencente à sequência é representada pelas seguintes características:

- valor absoluto da nota;
- intervalo cromático, intervalo tonal, e direcção do intervalo entre a própria nota e a nota anterior; intervalo cromático, intervalo tonal, e direcção do intervalo entre a própria nota e a nota seguinte;
- grau da nota na tonalidade em que está a peça; grau da nota na tonalidade

local;

- duração da nota em segundos; razão entre a duração da própria nota e a duração da nota anterior, bem como a direcção da variação (igual, maior, menor);
- amplitude, isto é, a intensidade com que a nota soa;
- outros, não especificados pelo autor.

Além da descrição individual de cada nota, são também consideradas características de carácter global, para descrever as peças. Estas características incluem, por exemplo, a tonalidade da peça, o tipo de compasso, o esquema harmónico (utilizado quando o sistema é aplicado a sequências de Jazz), ou o *tempo*, isto é, a duração da unidade de compasso.

Como se pode constatar, neste sistema é utilizada uma representação bastante mais completa do que as que são utilizadas nos sistemas que vimos até aqui. Como também referimos na secção sobre representação musical, algumas destas descrições não são normalmente fornecidas à partida como entrada para o sistema sendo esse, também, o caso do FExPat. Isto implica que seja necessário realizar algum processamento extra de modo a extrair essa informação antes do início do processo de descoberta de padrões. No entanto, no caso do FExPat, não é possível determinar, através da descrição do autor, a forma como essa informação é extraída sendo, inclusivamente, difícil de perceber se todo o processo é realizado automaticamente ou apenas parte dele.

A selecção de segmentos no FExPat não é propriamente uma fase do processo de extracção de padrões embora nem todos os segmentos sejam considerados, pois é definido um limite mínimo (m_{min}) e um limite máximo (m_{max}) para o tamanho dos segmentos a considerar. Além disso, é utilizada uma função que controla a diferença máxima entre o tamanho de dois segmentos a comparar, pelo que nem todos os pares de segmentos são comparados.

A comparação de segmentos no FExPat é realizada utilizando o método da programação dinâmica mas, tal como referimos na secção sobre comparação de sequências musicais, em vez de uma função de diferença é calculada uma função de

semelhança. Ou seja, para cada operação, em vez do seu custo ($w(\phi, b_j)$, $w(a_i, \phi)$, $w(a_i, b_j)$, etc.), é calculado o valor ($contrib(\phi, b_j)$, $contrib(a_i, \phi)$, $contrib(a_i, b_j)$, etc.) com que essa operação contribui para a semelhança entre os segmentos. Por exemplo, a operação de substituir duas notas iguais terá o valor máximo, enquanto que substituir duas notas diferentes com uma duração grande contribuirá menos para a semelhança final do que substituir duas notas diferentes com uma duração pequena. Como também já referimos anteriormente, além das três operações de edição mais simples, no FExPat são também utilizadas outras operações de edição especialmente concebidas para o domínio musical, tais como: fragmentação, consolidação, multi-remoção, multi-inserção e multi-substituição. O valor resultante da comparação de dois segmentos é um valor normalizado entre 0 e 1. Se esse valor ultrapassar um determinado limite (0.75 ou 0.8) os dois segmentos são considerados equivalentes.

A ordem por que são comparados os segmentos é, neste sistema, realizada de modo a que a comparação entre cada dois segmentos possa ser feita à custa dos valores obtidos na comparação anterior de segmentos mais pequenos. Este facto permite reduzir bastante a complexidade temporal do processo embora, como se verá, isso seja feito à custa do aumento da complexidade espacial. Assim, cada segmento (i, m) , em que i designa a posição da primeira nota do segmento em toda a sequência S e m o número de notas do segmento, é comparado sucessivamente com os segmentos $(1, m_{min})$, $(1, m_{min}+1)$, ..., $(1, m_{max})$, $(2, m_{min})$, $(2, m_{min}+1)$, ..., $(2, m_{max})$, $(i, 1)$, $(i, m-1)$. O valor obtido em cada uma destas comparações é guardado na matriz Sim , de quatro dimensões, de modo a que possa ser utilizado na comparação de segmentos de dimensões maiores tal como mostra a equação seguinte

$$Sim_{i'm'}^{im} = \max \begin{cases} contrib(S_{i+m-1}, S_{i'+m'-1}) + Sim_{i'm'-1}^{im-1} \\ contrib(S_{i+m-1}, \phi) + Sim_{i'm'}^{im-1} \\ contrib(\phi, S_{i'+m'-1}) + Sim_{i'm'-1}^{im} \end{cases}$$

$$Sim_{i'0}^{i0} = 0;$$

$$Sim_{i'0}^{im} \rightarrow Sim_0^{im} = contrib(S_{i+m-1}, \phi) + Sim_0^{im-1};$$

$$Sim_{i'm'}^{i0} \rightarrow Sim_{i'm'}^0 = contrib(\phi, S_{i'+m'-1}) + Sim_{i'm'-1}^0;$$

onde $Sim_{i'm'}^{im}$ é o valor que indica a semelhança entre os segmentos (i, m) e (i', m') .

A selecção de padrões neste sistema é realizada através de um algoritmo de

classificação denominado *Centro da Estrela*¹⁸. Para isso, depois de comparados todos os segmentos válidos, é criado um grafo onde cada nó corresponde a um padrão (ou seja, apenas aqueles segmentos para os quais foi identificada uma relação de equivalência com outro segmento) e cada arco a uma relação de equivalência entre dois padrões ao qual está associado o valor de semelhança entre os dois. Cada nó do grafo é considerado o protótipo de uma classe constituída por todos os nós que lhe são adjacentes (Figura 3.14).

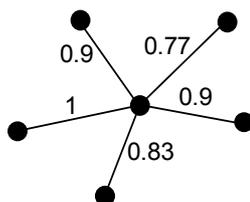


Figura 3.14 - Exemplo de uma estrela no grafo de semelhanças.

O algoritmo consiste em dois passos:

1. Para cada nó do grafo, calcular

$$valor(no) = \sum_{no' \in adj(no)} ValorAssociadoAoArco(no, no')$$

2. Ordenar os nós (os centros das estrelas) por ordem decrescente do valor calculado no passo anterior.

A importância de cada padrão é dada pelo valor calculado no segundo passo do algoritmo. Na verdade, quando um padrão é seleccionado, todos os que lhe estão associados por uma relação de equivalência são também seleccionados, pelo que se pode ver esta parte do processo como uma selecção de classes em vez de uma selecção de padrões¹⁹. Quanto ao limite a partir do qual uma classe é seleccionada, nada é referido pelo autor, sendo apenas dito que esta selecção pode ter em conta o *quorum* (quando a sequência a analisar é constituída por mais do que uma peça).

Como se pode calcular, deste processo resulta um grande número de classes e um número ainda maior de padrões. Daqui resultam duas questões que gostaríamos de discutir. Neste sistema dois segmentos podem ser considerados equivalentes mesmo

¹⁸ Do inglês 'Star Center'.

¹⁹ Este autor adopta uma definição de padrão diferente da que é adoptada neste texto, identificando as classes com os padrões, e denominado aquilo que neste texto são padrões como simples ocorrências do padrão.

que se sobreponham um ao outro, à semelhança do que acontece aliás com o sistema EMI e com o AIPS que iremos descrever a seguir. Se isso pode ser razoável quando o nível de sobreposição é pequeno e quando a primeira ou a última nota dos segmentos não coincide, já o mesmo não acontece quando, por exemplo, os segmentos [5 20] e [5 19] são considerados equivalentes. Não é que eles não sejam praticamente iguais, mas não nos parece de grande utilidade dizer que um segmento é equivalente a um seu sub-segmento. A outra questão explica-se melhor com um exemplo: consideremos a sequência “abcdefbnnnnnnnnabcdefnxxnnnnn”. Então, certamente que seria formada a classe A constituída pelas duas ocorrências do padrão “abcdef” com o valor de importância 1. Mas também seria formada uma classe B com as duas ocorrências do padrão “abcde” com o valor de importância igualmente 1. Por outro lado, a classe C constituída pelos padrões “nnnnnnnn” e “nxxnnnnn” teria um valor de importância menor que 1, uma vez que os dois padrões não são exactamente iguais. Ou seja, uma classe (B) constituída por sub-padrões de padrões existentes teria mais importância do que uma classe (C) com importância óbvia no contexto da sequência.

3.4.5 AIPS

O AIPS ([Cambouropoulos, 1998a], [Cambouropoulos, 1998b]), é um dos módulos da TGCEM que descrevemos no capítulo anterior e consiste num algoritmo que permite induzir padrões em sequências musicais. Como também referimos antes, na TGCEM a sequência inicial (superfície musical(0)), baseada no valor absoluto das notas, é transformada numa sequência musical baseada em vários tipos de intervalos (superfície musical(1)), sendo sobre esta e sobre reduções desta que o algoritmo opera.

Tal como no FExPat, não existe concretamente uma fase de selecção de segmentos, embora nem todos os segmentos possíveis sejam considerados pois são estabelecidos limites mínimos e máximos para o tamanho dos segmentos. Além disso, os resultados da aplicação do MDLL (descrito no capítulo anterior) podem ser utilizados de modo que nenhum padrão possa começar antes e acabar depois dos limites locais já calculados. Esta parece-nos, no entanto, ser uma limitação demasiado forte uma vez que, por vezes, pode acontecer que os padrões que melhor descrevem a estrutura de uma peça não respeitem estes limites. Mais à frente veremos um exemplo que confirma isto mesmo.

O algoritmo começa por comparar os segmentos mais pequenos, acabando nos segmentos maiores. Cada segmento é comparado com todos os segmentos possíveis que tenham o mesmo tamanho, sendo o processo realizado de modo que dois segmentos nunca sejam comparados mais do que uma vez. Os segmentos que se repetem são etiquetados para tratamento posterior (os segmentos são comparados na sua forma original, retrógrada²⁰, invertida²¹ e retrógrada e invertida). Os segmentos para os quais não for identificada qualquer repetição são postos de parte e marcados de modo a que não voltem a fazer parte do processo, o que permite a redução do espaço de pesquisa à medida que o processo vai avançando. O algoritmo é aplicado utilizando as dimensões que se desejar, mas apenas uma dimensão de cada vez. Além disso, se cada dimensão for descrita de várias formas (exemplo para a dimensão melódica: intervalos cromáticos, intervalos escalares, contorno, etc.) o algoritmo só é aplicado a uma destas descrições de cada vez.

Também como no FLEXPAT, no AIPS o processo de comparação de segmentos resulta num grande número de padrões, alguns dos quais sem grande importância no contexto da peça. Assim, a seguir a esta fase os segmentos mais importantes são seleccionados tendo em conta os seguintes princípios:

- Preferir padrões maiores;
- Preferir padrões que ocorram com mais frequência;
- Evitar a sobreposição de padrões.

Esta selecção é realizada recorrendo a uma função, denominada *função de selecção*, que atribui a cada segmento um valor de importância que reflecte os princípios mencionados:

$$f(T, F, GS) = \frac{F^a \times T^b}{10^{c \times GS}}$$

onde a , b e c são constantes que servem para ajustar a importância relativa de cada um dos princípios, T é o tamanho do padrão, F o número de vezes que ocorre e GS o

²⁰ A retrogradação de um segmento musical consiste em trocar a ordem pela qual as notas aparecem (a primeira troca com a última, a segunda troca com a penúltima, etc.).

²¹ A inversão de um segmento musical consiste em trocar o sinal dos intervalos entre as notas (se tivermos um segmento cujas notas formam os intervalos 2 2 1, a sua inversão terá os intervalos -2 -2 -1).

grau de sobreposição²².

A aplicação do MDLL e do AIPS pode dar resultados contraditórios, sendo por isso necessário realizar mais algum processamento de modo a decidir qual a segmentação mais correcta. Por exemplo, para a canção *Frère Jacques*, representada na Figura 3.15, o MDLL daria o intervalo entre a terceira e a quarta nota como um ponto preferencial para a ocorrência de um limite entre dois segmentos, enquanto que o AIPS daria mais importância ao padrão *a* do que ao padrão *b* se os pesos utilizados na função de selecção fossem $(a, b, c) = (3, 3, 4)$ (valores sugeridos pelo autor).

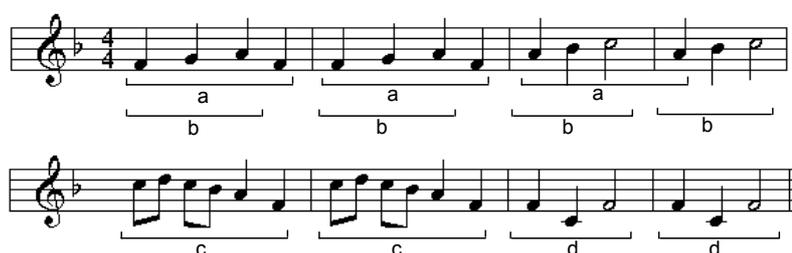


Figura 3.15 - Os padrões mais importantes da canção *Frère Jacques* descobertos pelo AIPS aplicado apenas a intervalos escalares e pela função de selecção utilizando os pesos $(a, b, c) = (3, 3, 4)$.

Na TGCEM, este problema é resolvido combinando o resultado do MDLL e do destes AIPS/função de selecção da seguinte forma: para cada ponto da sequência (intervalo entre duas notas), soma-se o valor de todos os padrões que comecem ou acabem nesse ponto (nenhum padrão é ignorado); cada valor obtido desta forma é combinado, através de uma soma pesada, com o valor na posição correspondente obtido com o MDLL. Os pontos de segmentação corresponderão aos máximos locais, sendo depois necessário verificar quais os padrões que melhor se “encaixam” nessa segmentação. Por exemplo, na canção *Frère Jacques*, depois de realizado este cálculo (Figura 3.16), facilmente se chega à conclusão de que a sequência de padrões que melhor descreve a peça é *aabbccdd*.

²² GS é definido como o número de elementos partilhados por alguns padrões dividido pelo número total de elementos nesses padrões, isto é $GS = (Total-U)/U$, onde: $Total$ é o número total de elementos de todas as ocorrências do padrão ($Total = F \times T$); U é o número de elementos do conjunto união das várias ocorrências do padrão (esta definição permite que, em alguns casos, o valor de GS exceda os 100%).

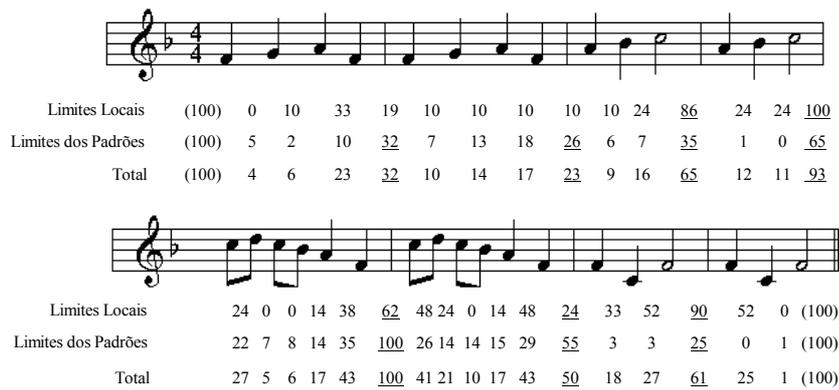


Figura 3.16 - Soma pesada e normalizada dos limites calculados com o MDLL e com o AIPS/função de avaliação para a canção *Frère Jacques*.

Para finalizar este capítulo mostramos na tabela seguinte algumas das principais características dos sistemas de extracção de padrões descritos neste capítulo.

	Informação melódica	Informação rítmica	Necessidade de pré-segmentação	Tipo de comparação de segmentos
EMI	intervalos cromáticos	razões de durações	não	exacta
SHARM	representação tradicional: <nome, acidente, oitava>	lista de figuras rítmicas cuja soma dá a duração total da nota	?	exacta com transformações
Cypher	intervalos cromáticos	não utiliza	sim	aproximada (programação dinâmica)
PSR	intervalos cromáticos	razões de durações	sim	aproximada (programação dinâmica)
FIExPat	intervalo escalares e cromático entre a nota e as notas adjacentes; grau da nota na tonalidade global e na tonalidade local.	duração em segundos; razão entre a duração da nota e a duração das notas adjacentes, bem como a direcção da variação.	não	aproximada (programação dinâmica)
AIPS	intervalos cromáticos, intervalos escalares e contorno.	razões de durações	não (embora os resultados da pré-segmentação possam ser utilizados)	exacta aplicada também a reduções da peça)

Tabela 3.1 - Principais características dos sistemas descritos neste capítulo.

Capítulo 4

Computação Evolucionária

A computação evolucionária é uma área das ciências da computação que se inspira na teoria da evolução de Darwin e na biologia molecular para a formulação de modelos computacionais que, por sua vez, são utilizados no desenvolvimento de sistemas de resolução de problemas. Existem vários modelos computacionais evolucionários que foram propostos e estudados, e aos quais se dá a denominação de algoritmos evolucionários. Estes algoritmos têm em comum o facto de simularem a evolução de um conjunto de estruturas individuais, através de processos de selecção e reprodução que dependem do desempenho dessas estruturas num determinado ambiente. A avaliação do desempenho é realizada tendo em vista um determinado objectivo, sob a forma de um critério de qualidade que é utilizado para medir e comparar um conjunto de candidatos a soluções num processo de refinamento progressivo. Se o processo for bem sucedido, é devolvida uma solução óptima ou quase óptima. Neste aspecto, estes algoritmos são mais parecidos com o “aperfeiçoamento” de raças realizado, por exemplo, pelos criadores de animais, um vez que também aí existe um critério de qualidade bem definido.

Neste capítulo, além de fazermos uma breve introdução à computação evolucionária, descrevemos, nas duas últimas secções, dois trabalhos que aplicam programação genética a dois domínios que, não correspondendo exactamente àquele que é tratado nesta dissertação, estão fortemente relacionados com ele.

4.1 Terminologia

A teoria da evolução das espécies de Darwin e a biologia molecular não só serviram de inspiração à área da computação evolucionária como lhe emprestaram algum do seu vocabulário. Iremos, pois, agora introduzir alguns termos da biologia molecular de

modo a facilitar a compreensão da descrição do funcionamento de um algoritmo evolucionário bem como dos vários modelos existentes:

- Os cromossomas consistem em cadeias de ADN (Ácido Desoxirribonucleico) constituídas por genes que codificam determinadas características dos indivíduos.
- Os diferentes valores que cada gene pode tomar designam-se por alelos.
- O conjunto de todo o material genético denomina-se genoma. Designa-se por genótipo o conjunto de genes contido no genoma.
- Os cromossomas são entidades orgânicas que codificam o fenótipo de um indivíduo, que corresponde às suas características observáveis.

No que aos algoritmos evolucionários diz respeito, o termo cromossoma é utilizado, normalmente, para designar um indivíduo candidato a solução, identificando-se com este. Dependendo do modelo de algoritmo evolucionário utilizado e/ou do problema a resolver, os cromossomas podem tomar várias formas como sejam, sequências de símbolos (binários, reais, alfanuméricos, etc.), grafos etc. Cada cromossoma será, por sua vez, constituído por um conjunto de genes, cada um representando uma determinada característica do indivíduo. Cada gene pode tomar diferentes valores, sendo que diferentes genes podem ter domínios diferentes. Os valores que cada gene pode tomar designam-se por alelos.

Para uma explicação mais contextualizada dos termos introduzidos nesta secção consultar [Mitchell, 1996].

4.2 O Algoritmo Evolucionário Típico

Como já foi mencionado, existem vários modelos de algoritmos evolucionários. Pode-se, no entanto considerar que a estrutura geral de qualquer destes algoritmos é, em grande parte, a mesma. A Figura 4.1 mostra um algoritmo típico que descreve os principais passos de qualquer algoritmo evolucionário.

```

t = 0.
gerar_população_inicial (P(t)).
avaliar (P(t)).
Enquanto critério de paragem não satisfeito faz
    P'(t) = seleccionar(P(t)).
    P''(t) = aplicar_operadores_genéticos(P'(t)).
    P(t + 1) = criar_população_seguinte(P(t), P''(t)).
    avaliar(P(t + 1)).
    t = t + 1.
FimEnquanto
Devolver melhor indivíduo.

```

Figura 4.1 - Um algoritmo evolucionário típico.

O primeiro passo do algoritmo consiste na criação da população inicial, isto é, o conjunto inicial de indivíduos candidatos a soluções. Geralmente, a criação desta população é realizada aleatoriamente, embora possa também ser utilizado conhecimento sobre o domínio de modo a criar indivíduos que estejam perto de possíveis soluções.

O segundo passo do algoritmo consiste em avaliar a qualidade dos indivíduos da população inicial utilizando um critério de qualidade predefinido. Na maior parte das situações, este critério toma a forma de uma função, designada função de avaliação, que atribui a cada indivíduo um valor numérico que reflecte o seu desempenho no ambiente em causa.

O algoritmo entra depois num ciclo que só termina quando um determinado critério de paragem for satisfeito, como por exemplo, existir um indivíduo na população actual que seja uma solução para o problema, ter sido atingido um número predefinido de gerações (iterações), ou não ocorrerem modificações significativas na população durante um determinado número de gerações.

Em cada iteração do ciclo, o primeiro passo consiste em seleccionar probabilisticamente os melhores indivíduos da população, criando-se, desta forma, uma população intermédia $P'(t)$. Existem vários métodos de selecção que se podem utilizar para este efeito, obedecendo todos eles aos seguintes princípios: os melhores indivíduos têm mais possibilidades de serem seleccionados; a selecção é realizada com reposição, ou seja, é possível escolher o mesmo indivíduo mais do que uma vez, de modo que os melhores possam ser escolhidos mais vezes.

A operação que se segue, por vezes denominada por fase de reprodução, consiste na

criação de novos indivíduos através da transformação dos indivíduos seleccionados no passo anterior, permitindo assim que se explorem outras regiões do espaço de pesquisa. As transformações são realizadas pelos chamados operadores genéticos sobre os indivíduos da população $P'(t)$. Os operadores mais comuns são: o operador de recombinação, que permite trocar partes de dois indivíduos criando dois outros; e o operador de mutação, que faz modificações num único indivíduo. Tipicamente, o operador de mutação é aplicado depois do operador de recombinação. A cada um deles é associada uma determinada probabilidade de ser aplicado, podendo acontecer que alguns indivíduos passem à fase seguinte sem qualquer transformação.

Da aplicação dos operadores genéticos, resulta a população $P''(t)$, que conjuntamente com a população $P(t)$, é utilizada para criar a população que irá substituir a população actual. Existem, basicamente, duas estratégias para criar a nova população. A primeira, denominada de estratégia geracional, consiste em substituir a população actual por $P''(t)$. Na outra estratégia, denominada por estado estável, a nova população é criada substituindo apenas um pequeno conjunto de indivíduos de $P(t)$, normalmente os piores. Em qualquer dos casos, a nova população deverá ter o mesmo tamanho N que a anterior. Para uma comparação entre estas duas estratégias consultar [Kinnear Jr., 1993] e [De Jong, 1993].

Antes de terminar a iteração actual do ciclo, a qualidade dos indivíduos da nova população é avaliada. Ao fim de uma série de iterações deste ciclo, a quase totalidade dos indivíduos da população deverá ser constituída pelo mesmo material genético. Quando isto acontece, diz-se que o algoritmo convergiu. Espera-se que a última população gerada contenha um indivíduo que seja uma solução óptima, ou quase óptima, para o problema que se pretenda resolver.

Uma vez descrito o algoritmo típico, é fácil identificar os aspectos que devem ser considerados quando se pretende utilizar um algoritmo evolucionário para resolver um determinado problema:

- Qual a representação e codificação das soluções.
- Qual a função de avaliação.
- Qual o método de selecção a utilizar.

- Quais os operadores genéticos a utilizar.
- Quais os parâmetros de controlo do algoritmo como, por exemplo, o tamanho da população, o número máximo de gerações, ou a probabilidade de aplicação de cada um dos operadores genéticos.
- Qual o critério de paragem.

O algoritmo descrito varia consoante os vários modelos de computação evolucionária existentes. As principais diferenças situam-se ao nível da representação dos indivíduos, e, conseqüentemente, no tipo de operadores genéticos utilizados. Existem ainda outras diferenças, nomeadamente, a ordem por que são executadas algumas das operações ou os métodos de selecção utilizados. Dada a importância do papel desempenhado pela selecção num algoritmo evolucionário e, uma vez que alguns métodos são utilizados por mais do que um modelo, descrevemos de seguida os métodos de selecção mais utilizados. Na secção 4.4 descreveremos quatro dos modelos de algoritmos evolucionários mais conhecidos.

4.3 Métodos de Selecção

Como vimos, depois de calculada a qualidade dos indivíduos da população, é necessário decidir quais irão produzir descendentes para a próxima geração, e em que proporção. Em geral, o método de selecção deve permitir que os melhores indivíduos sejam escolhidos mais vezes, na esperança de que os seus descendentes sejam ainda melhores, possibilitando assim que a população evolua até se encontrar uma solução. Existem vários textos, como por exemplo [Bäck, 1994] e [Blickle, 1995], que analisam e comparam métodos de selecção. Não existem, no entanto, técnicas que ajudem a decidir, *à priori*, qual o melhor método a utilizar num determinado problema. A seguir, iremos descrever brevemente alguns dos métodos de selecção mais utilizados.

Selecção proporcional ao desempenho

Descreveremos dois métodos de selecção proporcional ao desempenho. No método da roleta, a probabilidade de um indivíduo i ser seleccionado é

$$p_i = \frac{f_i}{\sum_j^N f_j}$$

em que N é o tamanho da população. Pode-se ver este mecanismo de selecção como se a cada indivíduo da população correspondesse uma fatia de uma roleta proporcional ao seu desempenho. A roleta é posta a girar N vezes e em cada uma é seleccionado um indivíduo. Obviamente, é esperado que o número de vezes que um indivíduo é seleccionado seja proporcional ao seu desempenho. No entanto, isto nem sempre acontece, sobretudo em populações pequenas [Mitchell, 1996], podendo suceder que indivíduos fracos são seleccionados bastante mais vezes do que seria esperado.

O método da amostragem universal estocástica procura resolver este problema colocando de uma vez só sobre a roleta N ponteiros igualmente distribuídos. Assim, um indivíduo será seleccionado tantas vezes quantos os ponteiros que estiverem sobre a sua fatia, ou seja, nunca menos que o valor esperado truncado, e nunca mais que o valor esperado truncado e incrementado.

Os métodos de selecção proporcional ao desempenho têm algumas desvantagens relativamente a outros métodos, nomeadamente, porque as probabilidades de selecção dependem fortemente do escalonamento da função de desempenho [Blickle, 1995]. Pode-se tentar resolver este problema utilizando métodos de escalonamento que permitem adaptar os valores da função de desempenho à média da população. Existe ainda outro problema que deriva do facto de o desvio padrão do desempenho dos indivíduos no início da evolução ser, geralmente, alto. Nestas condições, os indivíduos com maior desempenho têm tendência para se multiplicar consideravelmente dominando rapidamente a população (nestes casos diz-se que a pressão selectiva é elevada) e impedindo a exploração de outras regiões do espaço de pesquisa. Por outro lado, quando mais tarde todos indivíduos da população forem muito parecidos, o número de descendentes é quase igual tornando a convergência muito lenta.

Seleção por posição

Na selecção por posição os indivíduos da população são ordenados pelo valor do seu desempenho cabendo ao melhor indivíduo a posição N e ao pior a posição 1. A

probabilidade de selecção é atribuída linearmente ou exponencialmente de acordo com a posição de cada indivíduo. Se, por exemplo, for utilizada uma distribuição linear, a probabilidade de um indivíduo ser seleccionado é dada por

$$p_i = \frac{1}{N} \left[p^- + (p^+ - p^-) \frac{i-1}{N-1} \right]$$

em que p^-/N é a probabilidade do pior indivíduo ser seleccionado e p^+/N é a probabilidade de o melhor indivíduo ser seleccionado.

Este método procura reduzir a rápida convergência para um máximo local que ocorre nos métodos de selecção proporcional ao desempenho já descritos. Como a selecção é baseada na posição e não no valor relativo do desempenho dos indivíduos, é possível evitar que um pequeno conjunto de indivíduos com um desempenho muito superior ao do resto da população domine as populações subsequentes. Por outro lado, se o desvio padrão do desempenho dos indivíduos for baixo, a pressão selectiva pode ser mantida.

Seleção por truncamento

Neste método, os indivíduos da população são ordenados pelo seu desempenho e depois, apenas uma fracção T dos melhores indivíduos pode ser seleccionada, tendo estes a mesma probabilidade de serem seleccionados. A pressão selectiva será tanto maior quanto mais pequeno for o valor de T .

Seleção Sôfrega²³

Este método é algo parecido com o anterior mas permite que os indivíduos mais fracos possam também ser escolhidos. Para o aplicar é necessário começar por ordenar os indivíduos da população por ordem decrescente do seu desempenho. Depois, divide-se a população em dois grupos: o grupo I, formado pelos melhores indivíduos responsáveis por $\ell\%$ da soma do valor do desempenho de todos os indivíduos da população, e o grupo II, formado pelos restantes. A selecção dos indivíduos é depois realizada da seguinte forma: $p\%$ das vezes é escolhido um indivíduo do grupo I e $(100 - p)\%$ das vezes é escolhido um indivíduo do grupo II. Dentro de cada grupo os indivíduos são escolhidos com o método de selecção

²³ Do inglês 'greedy overselection'.

proporcional ao desempenho. A pressão selectiva deste método será tanto maior quanto maior for o valor de c e p .

Seleccção por torneio

O método de selecção por torneio funciona repetindo N vezes o seguinte procedimento: escolhe-se aleatoriamente um número t de indivíduos da população e selecciona-se o melhor deste grupo. É frequente o torneio ser disputado entre dois indivíduos, embora se possa utilizar um valor maior para t , o tamanho do torneio. Quanto maior for este valor, maior será a pressão selectiva.

Este método tem como vantagem o facto de ser computacionalmente mais eficiente que os métodos já descritos pois não necessita de uma comparação centralizada entre todos os indivíduos. Esta característica permite acelerar consideravelmente o processo de evolução, além de que permite paralelizar facilmente o algoritmo [Banzhaf, 1998].

4.4 Principais Modelos de Algoritmos Evolucionários

Nesta secção, descreveremos quatro dos principais modelos de computação evolucionária existentes. Como foi referido antes, estes algoritmos apresentam algumas diferenças. Tentaremos nesta descrição salientar as principais características de cada um, com especial relevo para os algoritmos genéticos e para a programação genética, que são os modelos utilizados neste trabalho.

Para uma introdução aos algoritmos evolucionários consultar os seguintes textos: [Spears, 1993], [Bäck, 1996], [Michalewicz, 1997], [Mitchell, 1999], [Silva, 2000].

4.4.1 Programação Evolucionária

A programação evolucionária foi desenvolvida por Lawrence Fogel em 1966 com o objectivo de gerar máquinas de estados finitos para tarefas de predição de sequências de símbolos. Actualmente, é utilizada sobretudo em problemas de optimização numérica. A representação dos indivíduos é normalmente dependente do tipo de problema que se queira resolver: Na versão original, por exemplo, cada indivíduo era uma máquina de estados finitos representada por um grafo. Em problemas de optimização numérica, os indivíduos são representados por vectores de valores reais e para o problema do caixeiro viajante, por exemplo, são utilizadas listas ordenadas. Os

operadores de mutação também são definidos consoante o problema e estão sujeitos a adaptação durante o processo evolutivo. A recombinação não é normalmente utilizada, dada a flexibilidade existente na definição dos operadores de mutação.

A primeira operação do processo evolutivo, depois de criada a população inicial, consiste na aplicação de um ou mais operadores de mutação a todos os indivíduos da população. Do resultado desta operação, é formada uma população intermédia, de tamanho $2 \times N$, constituída pelos progenitores e pelos seus descendentes. Seguidamente, todos os indivíduos desta população são avaliados, sendo seleccionados apenas os N melhores para passar à geração seguinte. Repare-se que, neste processo, a aplicação de operadores genéticos é realizada a todos os elementos da população actual antes da operação de selecção.

Para uma introdução à programação evolucionária, ver [Fogel, 1995].

4.4.2 Estratégias Evolucionárias

O modelo das estratégias evolucionárias foi desenvolvido por Rechenberg e Schwefel durante os anos 60 para problemas de optimização de um conjunto de parâmetros reais. Na versão original, apenas o operador de mutação era utilizado tal como no modelo anterior, e cada população era constituída apenas por um indivíduo. Em 1981, Schwefel introduziu o operador de recombinação, bem como populações com mais de um indivíduo. Também neste modelo o operador de mutação é adaptado à medida que o algoritmo vai sendo executado.

Depois da inicialização, são seleccionados aleatoriamente N indivíduos aos quais são aplicados os operadores de recombinação e mutação de modo a serem produzidos M descendentes, tal que $M > N$. A operação de selecção é depois tipicamente realizada utilizando um dos dois seguintes métodos. O primeiro selecciona simplesmente os N melhores descendentes para passar à próxima geração. O segundo selecciona os N melhores indivíduos de entre os N progenitores e os M descendentes.

Para uma breve introdução às estratégias evolucionárias ver [Bäck, 1996a].

4.4.3 Algoritmos Genéticos

Os algoritmos genéticos foram desenvolvidos por John Holland em 1975. O

algoritmo original tem como principais características o facto de cada indivíduo ser representado por uma sequência binária de 1s e 0s de tamanho fixo, e de utilizar sobretudo o método de selecção proporcional ao desempenho. Este método de selecção é, aliás, um dos elementos chave de um teorema formulado por Holland cuja ideia fundamental é a de que um algoritmo genético é capaz de combinar, através da aplicação repetida dos operadores de recombinação e de mutação, os elementos (os chamados blocos construtores) que constituem uma solução óptima. Para uma análise detalhada deste teorema consultar [Goldberg, 1989].

Como em qualquer outro modelo evolucionário, a escolha da representação e da codificação dos indivíduos é uma tarefa chave para o sucesso da aplicação do algoritmo. Quando se utiliza um algoritmo genético, cada sequência deve ser codificada de modo que a cada símbolo, ou grupo de símbolos, seja atribuído um determinado significado, estabelecendo-se assim um mapeamento de cada indivíduo no espaço de pesquisa. Embora a representação baseada em alfabetos binários seja ainda hoje a mais comum, outras têm sido também utilizadas, como sejam as representações baseadas em vectores de números reais [Goldberg, 1989]. Mais recentemente, têm surgido também abordagens em que são utilizadas outras estruturas que não sequências de símbolos como, por exemplo, listas ligadas [Whitley, 1989] ou redes neuronais [Harp, 1991], para representar os indivíduos. Quanto ao tamanho dos cromossomas, existem várias abordagens que não utilizam um tamanho fixo ([Neves, 1997], [Xiao, 1997]). Nestas abordagens é permitido que os indivíduos tenham tamanhos diferentes e que este tamanho varie ao longo do processo evolutivo, uma vez que não se sabe, à partida, qual o tamanho ideal.

Como já foi referido, durante alguns anos os algoritmos genéticos utilizaram preferencialmente os métodos de selecção proporcional ao desempenho. No entanto, devido às suas desvantagens, estes métodos têm vindo a perder terreno para outros como sejam o método da selecção por torneio ou outros descritos na secção 4.3.

Nos algoritmos genéticos, os operadores genéticos mais utilizados são o operador de recombinação e o operador de mutação, com especial relevo para o primeiro, cuja probabilidade de aplicação é, geralmente, alta. O operador de mutação, pelo contrário, é, normalmente, aplicado com uma probabilidade muito baixa. Existem vários tipos de

recombinação, sendo os mais conhecidos a recombinação de um corte, a recombinação de dois cortes e a recombinação uniforme. A recombinação de um corte consiste em escolher aleatoriamente um ponto de corte em dois indivíduos diferentes previamente seleccionados, trocando de seguida entre eles os segmentos situados num dos lados do corte. Na recombinação de dois cortes, é trocado o segmento situado entre dois pontos de corte definidos também aleatoriamente. Na recombinação uniforme existe uma probabilidade p de trocar cada uma das posições. O operador de mutação consiste em escolher aleatoriamente uma ou mais posições e substituir o valor dessas posições por outros. A Figura 4.2 ilustra o funcionamento destes quatro operadores. Em [Spears, 1998], é discutido com profundidade o papel desempenhado pelos operadores de recombinação e de mutação nos algoritmos genéticos.

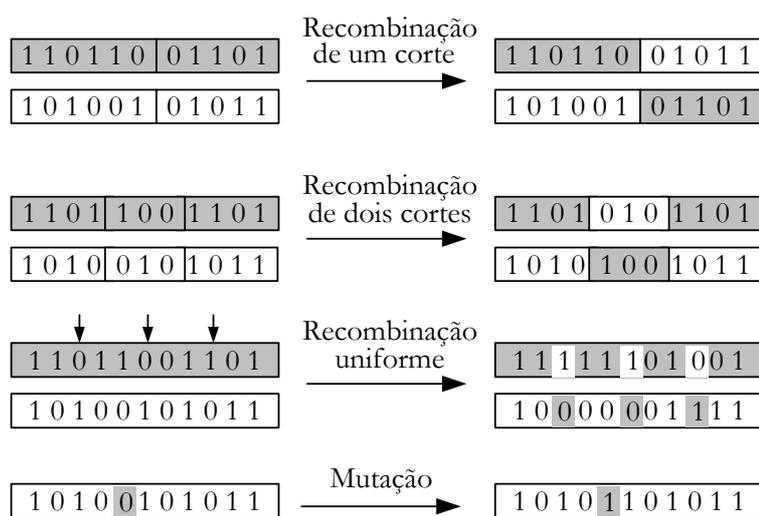


Figura 4.2 - Operadores de recombinação e mutação mais utilizados nos algoritmos genéticos.

Para uma introdução aos algoritmos genéticos ver [Goldberg, 1989], [Michalewicz, 1996], [Mitchell, 1996].

4.5 Programação Genética

No início dos anos 90, alguns investigadores, entre os quais John Koza [Koza, 1992], introduziram a programação genética, cujo objectivo é a evolução de programas que resolvam um determinado problema. Com este modelo, em vez de se utilizar um sistema evolucionário que descubra uma solução para o problema, utiliza-se um

sistema evolucionário que procure um programa que permita resolver o problema. Desta forma, a pesquisa deixa de ser feita no espaço de candidatos a soluções para o problema, e passa a ser realizada no espaço de possíveis programas.

4.5.1 Representação e População Inicial

A estrutura em árvore tem sido, até hoje, a forma mais comum de representar os programas, embora sejam também utilizadas outras representações tais como grafos ([Poli, 1997], [Pereira, 1999]) ou estruturas lineares ([Huelsenbergen, 1996], [Keller, 1996]). As funções e os terminais são as primitivas com as quais são construídos os programas: as funções são utilizadas para representar nós internos e os terminais são utilizados como nós folha. Quando se pretende utilizar programação genética para resolver um determinado problema deve-se, portanto, definir um conjunto de funções e um conjunto de terminais adequado a esse problema. O espaço de pesquisa será o hiperespaço de programas válidos, utilizando as funções e os terminais escolhidos, representados sob a forma de uma árvore.

Tal como nos outros modelos já descritos, o primeiro passo do algoritmo consiste na criação da população inicial. Uma vez que as árvores têm tamanho variável, é importante limitar de alguma forma o seu tamanho. Isso pode ser conseguido definindo um valor máximo para o número de nós e/ou para a profundidade das árvores. São três os métodos mais conhecidos para gerar os indivíduos da população inicial:

1. Criação completa: para cada árvore é definido um valor para a sua profundidade. A árvore é depois criada de modo que a distância do nó raiz a qualquer nó terminal seja igual a esse valor.
2. Criação por crescimento: O nó raiz começa por ser escolhido aleatoriamente da reunião do conjunto de funções com o conjunto de terminais. Este procedimento é depois executado recursivamente de modo a gerar sub-árvores para todos os nós da árvore que necessitem delas. Quando a profundidade máxima é atingida, o nó deve ser escolhido apenas a partir do conjunto de terminais.
3. Criação mista: consiste simplesmente em criar 50% dos indivíduos com o

método de criação completa e os outros 50% com o método de criação por crescimento. Este método é tipicamente o mais utilizado por ser aquele que produz uma maior variedade de formas e tamanhos para as árvores.

Para qualquer um destes métodos é possível especificar uma “rampa” de valores para a profundidade máxima da árvore em vez de apenas um. Por exemplo, se a “rampa” for 2-5, então 25% dos indivíduos são criados com uma profundidade máxima de 2, 25% com uma profundidade máxima de 3, e assim por diante.

4.5.2 Avaliação e Selecção

Uma vez que os indivíduos são programas, é necessário executá-los de modo a que se possa avaliar o seu desempenho. Em determinados problemas, os indivíduos são executados num conjunto de casos de teste, sendo-lhes atribuído um valor pelo resultado conseguido com cada um desses casos. Os valores conseguidos são depois combinados, através da função de avaliação, de modo a produzir um valor final que reflecta o desempenho do indivíduo. Por exemplo, num problema de regressão simbólica, cada indivíduo é executado para um conjunto de valores para os quais se conhece o resultado correcto. O desempenho do indivíduo poderá ser medido, por exemplo, pela soma dos módulos das distâncias entre o resultado correcto e o resultado obtido para cada um dos casos. Eventualmente, poderá ser útil ter em conta outras características dos programas que não o resultado da sua execução, como sejam o tamanho ou o tempo de execução. Na secção 4.6, descreveremos um trabalho onde o número de nós influencia o valor da função de avaliação.

Tal como nos algoritmos genéticos, em programação genética os métodos de selecção mais utilizados começaram por ser os métodos de selecção proporcional ao desempenho. No entanto, nos últimos anos começou-se, tal como nos algoritmos genéticos, a utilizar com grande frequência o método de selecção por torneio por este ser computacionalmente menos dispendioso.

4.5.3 Operadores Genéticos

Já se disse em cima que os operadores genéticos dependem fortemente do modo como se representam os indivíduos. Uma vez que em programação genética existem vários tipos de representação, iremos descrever apenas os operadores mais utilizados

com a representação em árvore que, como já foi referido, é o tipo de representação mais comum e aquele que é utilizado neste trabalho.

O operador genético ao qual se atribui maior importância é o operador de recombinação, sendo também utilizados, embora com menor probabilidade de ocorrência, os operadores de reprodução e de mutação²⁴. Ao contrário do que acontece, por exemplo, nos algoritmos genéticos, os operadores não são aplicados consecutivamente, ou seja, cada indivíduo é sujeito a apenas um operador. A probabilidade de cada operador ser aplicado é atribuída de modo a que a soma das probabilidades de todos os operadores seja igual a 1.

O operador de recombinação consiste em seleccionar aleatoriamente uma sub-árvore em cada um dos indivíduos progenitores trocando, de seguida, essas sub-árvores entre os dois indivíduos. O operador de mutação consiste em seleccionar uma sub-árvore do indivíduo e substituí-la por outra criada aleatoriamente. O operador de reprodução, o mais simples de todos, consiste, simplesmente, em copiar o indivíduo para a próxima população. Com a aplicação deste operador procura-se garantir que alguns dos melhores indivíduos da população actual passem intactos para a próxima população.

4.5.4 Funções Definidas Automaticamente

Tal como em qualquer um dos modelos já descritos, existem em programação genética várias variantes do algoritmo base, como por exemplo as que envolvem a utilização de nós fortemente tipados [Montana, 1994] ou a utilização de estruturas de dados [Langdon, 1995]. A variante que julgamos mais relevante para o nosso trabalho é claramente inspirada na forma como as pessoas programam e consiste na utilização de subrotinas denominadas por Funções Definidas Automaticamente (FDA) [Koza, 1993]. Quando se utilizam FDAs, cada programa é constituído por uma função principal e por um conjunto de subrotinas, também elas sujeitas ao processo de evolução. Cada FDA pode ter o seu próprio conjunto de funções e terminais. Deve-se por isso, restringir a aplicação do operador de recombinação de modo a que este ocorra apenas entre subrotinas que usem o mesmo conjunto de funções e terminais.

²⁴ Na versão original de Koza, o operador de mutação não era sequer utilizado.

Podem também existir referências hierárquicas entre subrotinas, isto é, uma subrotina pode pertencer ao conjunto de funções de outra subrotina. É necessário, no entanto, ter cuidado de modo a evitar que ocorra recursividade.

Koza mostrou que a utilização de FDAs permite melhorar a exploração das regularidades que eventualmente possam existir no espaço de pesquisa, o que pode levar a uma diminuição do número de indivíduos que é necessário processar até se encontrar uma solução, bem como à melhoria da qualidade dessa solução. As FDAs são largamente analisadas em [Koza, 1994].

4.6 Programação Genética Aplicada ao Cálculo da Complexidade de Kolmogorov

A complexidade de Kolmogorov define-se como o tamanho do programa mais pequeno capaz de executar uma determinada tarefa. Este tipo de complexidade é frequentemente aplicado a sequências de símbolos, caso em que se define como o tamanho do programa mais pequeno capaz de produzir uma determinada sequência. Pode, também, ser interpretada como representando a quantidade de informação contida pela sequência. Por exemplo, uma sequência constituída por n símbolos iguais é extremamente simples, uma vez que o programa mais pequeno capaz de a reproduzir apenas tem que imprimir o mesmo símbolo n vezes.

A abordagem de Conte [Conte, 1997] ao estudo deste problema consiste em utilizar programação genética para procurar o programa mais pequeno capaz de produzir, como resultado, uma determinada sequência de símbolos pertencentes ao alfabeto binário $\mathcal{A} = \{0, 1\}$. Os programas são codificados na linguagem de programação TOY Lisp. Os símbolos terminais consistem nos átomos 1, 0, () e $w, t, y, u, s, j, k, z, x, v$ para os átomos definidos automaticamente. O conjunto de símbolos correspondentes às funções utilizadas é composto pelos símbolos das funções primitivas do TOY Lisp (Tabela 4.1) e pelos símbolos $a(g), d(lm), f(pqr), b(h), e(no), c(i)$ para as funções definidas automaticamente, onde os símbolos que estão entre parêntesis representam variáveis. Portanto, os genótipos, ou cromossomas, são programas em TOY-Lisp codificados como estruturas em árvore sem um limite fixo para o tamanho.

Função	Símbolo	Nº de Arg.	Valor
<i>Quote</i>	'	1	'(xy)z → (xy)z
<i>Atom</i>	.	1	.x → 1. .'(x) → 0
<i>Equal</i>	=	2	=xx → 1 =xy → 0
<i>Car</i>	+	1	+'(xy)z → x +x → x
<i>Cdr</i>	-	1	-'(xy)z → (yz) -x → x
<i>Cons</i>	*	2	*x'(yz) → (xy)z *xy → x
<i>If-then-else</i>	/	3	/1xy → x /0xy → y
<i>Eval</i>	!	1	!x → avalia x
<i>Append</i>	^	2	^(xy)'(z)k → (xy)z'k
<i>Define</i>	&	2	&xy → x toma o valor de y &(yx)z → y toma o valor de &(x)z
<i>Try</i>	?	3	?yxz → avalia x vezes y com o argumento z
<i>Let</i>	:	3	:xy'z → ('&(x)z)y :(yx)z'k → ('&(y)k'&(x)z)

Tabela 4.1 - Funções primitivas do TOY Lisp.

Na função de avaliação são considerados os seguintes aspectos:

- O tamanho da sequência gerada: quanto mais próximo estiver do tamanho da sequência objectivo melhor;
- A distância de Hamming (ver capítulo 3) entre a sequência gerada e a sequência objectivo: quanto menor melhor;
- Tamanho do programa: quanto menor melhor.

A pesquisa do programa mais pequeno é, assim, reduzida a um problema de minimização. A função de avaliação consiste numa soma pesada de termos, em que cada um expressa um dos aspectos descritos, resultando na função

$$F(p) = a \sum_{i=1}^k d(s_i, o_i) + b |t(s) - t(o)| + c(t(p) - t(s))$$

onde a , b e c são constantes, p é o programa a avaliar cujo resultado é a sequência $s = s_1 \dots s_n$, $o = o_1 \dots o_m$ é a sequência objectivo, $t(x)$ é o tamanho da sequência x ,

$k = \min(t(s), t(o))$, e

$$d(s_i, o_i) = \begin{cases} 0 & \text{se } s_i = o_i \\ 1 & \text{se } s_i \neq o_i, s_i \in \{0,1\} \\ 2 & \text{se } s_i \neq o_i, s_i \notin \{0,1\} \end{cases}$$

representa a distância de Hamming entre as duas sequências.

A tabela seguinte mostra, na coluna da direita, os programas que foram obtidos em experiências realizadas com as sequências da primeira coluna.

Sequência	Programa
$o = 1^n, n = 64$	$p = : (ag) \wedge gg \text{ aaaaaa}'(1)$
$o = [10]^n, n = 32$	$p = : (ag) \wedge gg \text{ aaaaa}'(10)$
$o = 1^n 0^n, n = 24$	$p = : (ag) \wedge gg \wedge \text{aaa}'(111)\text{aaa}'(000)$
$o = 1^n 01^n, n = 32$	$p = : (ag) \wedge gg - a * 0\text{aaaaa}'(10)$
$o = 1^n 0^n 1^n, n = 24$	$p = : (ag) \wedge gg \wedge \wedge \text{aaa}'(111)\text{aaa}'(000)\text{aaa}'(111)$

Tabela 4.2 - Programas obtidos para cada uma das sequências.

Repare-se que, embora nesta abordagem não sejam utilizadas FDAs, existe um mecanismo, fornecido pela função primitiva ‘:’ que permite a criação dinâmica de funções. Pode-se ver na tabela que em todos os programas foi definida dinamicamente uma função a capaz de explorar as regularidades de cada sequência de modo a produzir a sequência final. Esta função, que é igual para todos os programas (porque será?), recebe uma sequência g e devolve outra que é a concatenação de g consigo própria ($\wedge gg$). A função é depois utilizada para produzir a sequência em questão.

Os autores referem que foram também realizadas experiências com sequências do tipo $1010^210^4 \dots 10^{2n}, 1010^310^5 \dots 10^{2n+1}$ mas que, embora tenham sido gerados programas capazes de as produzir, nenhum o fazia tirando partido das regularidades existentes nas sequências.

4.7 Programação Genética Aplicada à Composição Musical

Em [Johanson, 1997], é apresentado um sistema interativo que permite ao utilizador evoluir pequenas sequências musicais utilizando programação genética. Neste sistema, cada indivíduo é um programa cujo resultado é um trecho musical. É o utilizador que

desempenha o papel de função de avaliação pois é ele que, geração após geração, avalia cada um dos indivíduos, conforme o seu gosto musical, atribuindo-lhes um determinado valor.

Os conjuntos de terminais e funções utilizados são, respectivamente, as notas utilizadas nas melodias, e uma pequena coleção de funções que permitem modificar seqüências de notas. O conjunto de terminais é constituído por notas codificadas no formato XM que permite representar peças musicais como um conjunto de seqüências de notas. Neste formato, cada nota é descrita pelo seu nome, acidente e oitava. Por exemplo, C#4 corresponde a um dó sustenido pertencente à quarta oitava. São utilizados também, como terminais, um conjunto de sete pseudo-acordes. Cada um destes acordes é constituído por 3 notas tocadas sequencialmente, daí a designação de pseudo-acorde. Finalmente, é também utilizado o símbolo RST (abreviatura do inglês *rest*) para designar as pausas.

As funções, descritas na Tabela 4.3, operam sobre uma ou mais seqüências de notas. A função *play-two* é essencial, pois é ela que permite a formação de seqüências com mais do que uma nota. As outras funções destinam-se a conferir alguma estrutura à seqüência final.

Função	Descrição
<i>Play_two</i>	Recebe duas seqüências e devolve a sua concatenação
<i>add-space</i>	Adiciona uma pausa a seguir a cada uma das notas pertencentes à seqüência recebida
<i>play-twice</i>	Recebe uma seqüência e concatena-a consigo própria
<i>shift-up</i>	Acrescenta um grau diatónico a todas as notas da seqüência (transposição modal)
<i>shift-down</i>	Subtrai um grau diatónico a todas as notas da seqüência (transposição modal)
<i>mirror</i>	Inverte a ordem das notas pertencentes à seqüência
<i>Play_and mirror</i>	A seqüência é invertida e concatenada consigo própria

Tabela 4.3 - Funções primitivas.

Note-se que, ao contrário do que acontece na maioria dos sistemas baseados em

programação genética, neste sistema, tal como no anterior, o resultado de um programa não é um valor simples, mas sim uma sequência de notas. Além disso, os programas não recebem nenhum argumento, como geralmente acontece. A Figura 4.3 ilustra como é construída uma sequência. Cada nó da árvore propaga para o nó imediatamente acima uma sequência que é depois modificada por este, e assim sucessivamente, até ser atingida a raiz da árvore. No final é devolvida a sequência final.

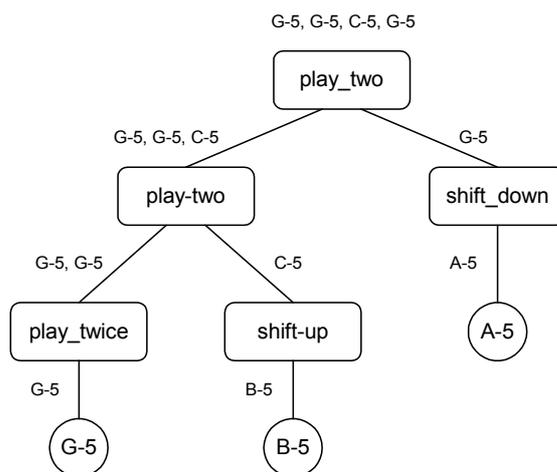


Figura 4.3 - Exemplo de um indivíduo.

O autor refere que foram também realizadas experiências com FDAs de modo a permitir a reutilização de material musical interessante ao longo da sequência ou, no caso de FDAs com argumentos, de modo a criar novas capacidades estilísticas que pudessem, igualmente, ser reutilizadas. Curiosamente, o resultado destas experiências, segundo o autor, foram desastrosos, tendo a utilização de FDAs implicado apenas um aumento considerável no tamanho das sequências geradas.

Capítulo 5

Programação Genética e Extração de Padrões Musicais

Neste capítulo descrevemos como é que abordámos o problema da extração de padrões utilizando programação genética. Começamos, no início do capítulo, por descrever os princípios que serviram de base a esta parte do nosso trabalho. Segue-se a descrição detalhada da abordagem e a apresentação dos resultados experimentais obtidos. Finalizamos o capítulo com algumas conclusões acerca da abordagem e dos resultados experimentais.

5.1 Programas como Descrições da Estrutura de Peças Musicais

A ideia que deu origem à nossa abordagem ao problema da extração de padrões utilizando programação genética pode ser formulada da seguinte forma:

“Um programa cujo resultado seja uma determinada peça musical, constitui, de algum modo, uma descrição da estrutura dessa peça.”

Embora possamos aceitar esta ideia como razoável, é óbvio que nem todos os programas capazes de produzir uma determinada peça musical constituirão uma descrição muito útil da sua estrutura. Por exemplo, programas do tipo

“ tocar dó, tocar ré, tocar mi, tocar dó, tocar ré, tocar mi, etc.”

não serão certamente de grande utilidade, pois nada nos dizem acerca da estrutura da peça, limitando-se a enunciar sucessivamente todas as suas notas. Interessam sim programas que, à semelhança do que os analistas humanos fazem, tirem o máximo partido das regularidades existentes na peça a analisar. Ou seja, interessam programas que descrevam, tanto quanto possível, umas partes da peça à custa de outras, estabelecendo, deste modo, relações entre elas. Para tal, é necessário que estes

programas sejam constituídos por primitivas que sejam pertinentes não só do ponto de vista musical de um modo geral, mas também do ponto de vista da análise musical. Queremos com isto dizer que ao construir um programa que constitua uma descrição da estrutura de uma peça se deve garantir que as relações que o programa é capaz de descrever são relações que façam sentido do ponto de vista da análise musical. Por exemplo, no limite seria possível descrever todas as notas de uma peça em função da primeira nota, mas isso não nos esclareceria muito quanto à sua estrutura. Eventualmente, pode também ser possível gerar uma parte de uma peça a partir de outra utilizando uma qualquer função matemática complicada, mas isso teria igualmente pouca utilidade para o músico que apenas está interessado em conhecer a estrutura da peça e não em saber como se pode representar toda a peça a partir, por exemplo, das suas dez primeiras notas.

Admitindo então que um programa pode constituir uma descrição da estrutura de uma peça musical, resta-nos, de modo a podermos orientar a nossa pesquisa, estabelecer os critérios que permitam decidir qual, de entre um conjunto de programas, o que melhor descreve a estrutura da peça. Um critério óbvio consiste em considerar que quanto mais parecida com a peça a analisar for a peça produzida por um programa, melhor é esse programa. Apesar de necessário, este critério por si só não garante, no entanto, que um programa seja uma boa descrição da estrutura da peça que produz como resultado (veja-se, por exemplo, o programa que descrevemos em cima). É, por isso, necessário complementá-lo com outro, ou outros, que ajudem a escolher os melhores programas. Um princípio a partir do qual se podem estabelecer estes critérios é o famoso princípio denominado “a navalha de Occam”²⁵ que pode ser enunciado da seguinte forma: “Preferir a hipótese mais simples que explique um determinado fenómeno” [Mitchell, 1997]. Na prática, no entanto, este princípio pode dar origem a critérios algo diferentes, dependendo do tipo de programa que se pretende...

5.1.1 Programas “independentes”

Por exemplo, podemos querer que o programa seja capaz de produzir a peça de modo

²⁵ Do inglês ‘Occam’s razor’.

completamente independente, tal como o fazem os programas de compressão de informação. Neste caso, e admitindo que as funções utilizadas pelo programa para produzir uma parte da peça a partir de outra são sempre mais pequenas do que a parte da peça a ser produzida, um programa será tanto melhor quanto menor for o seu tamanho. Este critério resulta do facto de o tamanho dos programas variar na ordem inversa do número de relações que forem descobertas entre as várias partes da peça (quanto maior for este número, menor será o número de notas a que é necessário fazer referência e vice-versa). Repare-se que, embora o nosso objectivo não seja o cálculo da complexidade da peça a analisar, podemos ver esta abordagem como se do cálculo da complexidade de Kolmogorov se tratasse.

Baseando-nos no trabalho de Conte, e também no de Johanson, descritos no capítulo anterior, a primeira tentativa que fizemos nesta parte do nosso trabalho consistiu em tentar gerar programas que, de um modo completamente independente, fossem capazes de produzir uma determinada peça. Na figura seguinte apresentamos um pequeno programa que ilustra como, nesta primeira abordagem, eram representados os programas.

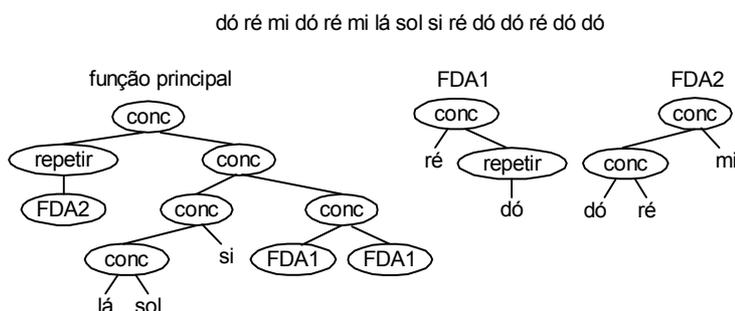


Figura 5.1 - Exemplo de um programa capaz de produzir uma sequência de notas de forma totalmente independente.

Tal como no trabalho de Johanson, os nós terminais eram notas (na verdade, sequências de uma nota) e o conjunto de funções constituído por funções que permitiam concatenar e modificar sequências de notas. A reutilização de material musical era realizada recorrendo a funções como, por exemplo, a função *repetir*, que recebia uma sequência e devolvia a concatenação dessa sequência consigo própria, e a FDAs com e sem argumentos.

A função de avaliação foi definida como uma soma pesada de dois termos: o primeiro

termo consistia no número de nós do programa, e o segundo na distância entre a peça produzida pelo programa e a peça a analisar, calculada com o algoritmo de Wagner-Fischer, descrito no capítulo 3. O objectivo era, tal como no trabalho de Conte, encontrar o programa que minimizasse o valor desta função.

Apesar das expectativas, os resultados experimentais obtidos com esta abordagem foram muito fracos. Nas experiências que realizámos, apenas para peças musicais muito pequenas e com uma estrutura muito simples foi o sistema capaz de produzir a peça e de tirar partido das regularidades existentes. Com peças com uma estrutura um pouco mais complexa, como as do Anexo I, os melhores programas gerados, além de não tirarem partido das regularidades existentes, não garantiam que a estrutura apresentada fosse a da peça a analisar, pois a distância entre esta e a peça produzida era geralmente grande. Esta abordagem tem ainda a desvantagem de os programas gerados não serem de fácil interpretação, pois, mesmo para sequências pequenas, os estes facilmente atingem um tamanho considerável (veja-se, por exemplo, o programa da Figura 5.1). Estes resultados e a difícil interpretação dos programas levaram-nos a tentar a abordagem que descrevemos a seguir.

5.1.2 Programas “dependentes”

Uma alternativa à criação de programas “independentes” consiste em permitir que os programas possam fazer referência às notas da peça sem as nomear explicitamente, assinalando apenas onde é que estas se encontram na peça. Utilizando esta abordagem, a sequência de notas “dó ré mi dó ré mi dó ré mi dó ré mi” pode ser descrita, por exemplo, pelo programa “*repetir(repetir ([0 2]))*”, onde [0 2] representa o segmento que vai da primeira à terceira nota e *repetir* é uma função que devolve a concatenação da sequência recebida consigo própria²⁶. A principal vantagem desta abordagem relativamente à anterior reside no facto de o sistema não perder tempo à procura das notas que fazem parte da peça nem a ordená-las pela ordem certa. Apenas é necessário identificar os segmentos mais importantes que constituem a peça e a forma como estes se relacionam, que é, afinal, o problema que nos propomos tentar resolver.

²⁶ Na verdade, seria mais correcto representar o programa da seguinte forma: *repetir(repetir(ler([0, 3])))*, em que *ler* é uma função de leitura que recebe os limites de um determinado segmento e devolve a sequência de notas da peça que compõem esse segmento.

Naturalmente, um programa deste tipo não é capaz de produzir a peça sem que esta “esteja por perto”, mas isso não deverá constituir um problema se o único objectivo for determinar a estrutura da peça.

Seguindo esta abordagem, que outros critérios, além do critério que favorece os programas que produzam peças mais parecidas com a peça a analisar, poderão ser utilizados para avaliar os programas? O critério do programa mais pequeno não pode agora ser utilizado uma vez que todos os segmentos de notas utilizados no programa ocupam o mesmo tamanho independentemente do número de notas do segmento (no limite, o programa “[0 11]” seria, segundo este critério, o melhor programa para descrever a sequência anterior!). Ou seja, o tamanho do programa é agora directamente proporcional ao número de relações que forem descobertas entre as várias partes da peça. De seguida descrevemos dois critérios que podem ser utilizados quando se segue esta abordagem.

O primeiro critério baseia-se no facto de que, quanto maior for a reutilização de material musical, menor será o número de notas a que o programa terá que fazer referência. Uma vez que nesta abordagem os programas só implicitamente fazem referência às notas da peça a analisar podemos enunciar este critério da seguinte forma: um programa será tanto melhor quanto menor for o número de notas da peça a ser analisada a que fizer implicitamente referência. Por exemplo, segundo este critério, dados os programas

“concatenar([0 3], concatenar([4 7], [0 3]))”

e

“concatenar([0 3], concatenar([4 7], [8 11]))”,

o primeiro descreve melhor a estrutura da sequência “dó dó dó dó ré mi fá sol dó dó dó dó” porque faz referência a apenas 8 notas da sequência a analisar, enquanto que o segundo faz referência a 12.

O segundo critério consiste em considerar que um programa será tanto melhor quanto maior for o número de notas, da peça que produzir, correspondentes aos segmentos utilizados mais do que uma vez. Segundo este critério, dados os dois programas anteriores e a mesma sequência de notas, o primeiro programa seria novamente

considerado melhor porque 8 das 12 notas da peça que produz como saída resultam de utilizar duas vezes o mesmo segmento, enquanto que no segundo, como não há reutilização de material musical, esse número é 0.

Durante a realização do nosso trabalho, ambos os critérios descritos foram testados. No entanto, os resultados obtidos com o segundo critério foram muito inferiores aos obtidos com o primeiro. Assim, na próxima secção descrevemos apenas o trabalho desenvolvido utilizando o primeiro critério. Como se verá na secção dedicada à apresentação dos resultados, apesar de melhor que o segundo, também este critério apresenta algumas dificuldades em descobrir os padrões existentes em algumas das peças que utilizámos para testar a abordagem.

5.2 Descrição da Abordagem

Nesta secção iremos descrever detalhadamente como utilizámos programação genética para tentar encontrar um programa que produza, como resultado, uma determinada peça musical e que, ao mesmo tempo, constitua uma descrição da estrutura dessa peça. Na nossa abordagem, apesar de a peça produzida por cada programa dever ser o mais parecida possível com a peça a analisar, não estamos interessados nesse resultado como um fim último a atingir. Este resultado só é importante na medida em que assim se garante que o programa é uma descrição da estrutura da peça a analisar, independentemente do valor dessa descrição. O que aqui realmente nos interessa é que o programa mostre quais são os segmentos que constituem a peça e quais as relações existentes entre eles.

5.2.1 Representação dos Indivíduos

A figura seguinte ilustra, através de um exemplo, a forma como são representados os indivíduos.

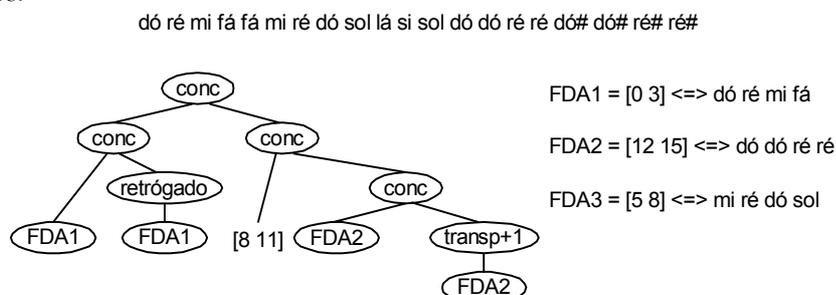


Figura 5.2 - Exemplo de uma sequência de notas e de um programa que descreve a sua estrutura.

Cada indivíduo consiste num programa que é constituído por uma função principal, representada sob a forma de árvore, que descreve as relações entre os segmentos, e por um conjunto de FDAs que representam segmentos concretos. A função principal é constituída a partir de um conjunto de funções que, além de concatenar os diferentes segmentos de modo a que o programa produza a sequência final de notas, servem para produzir transformações rítmicas, melódicas ou estruturais. Uma das funções mais importantes que é utilizada é a função que denominámos por *conc*. Esta função recebe, como argumentos, dois segmentos de notas e devolve, como resultado, a sua concatenação. Além de permitir a concatenação de todo o material musical utilizado, esta função permite também descrever implicitamente a ocorrência de repetições de segmentos, consecutivas ou não, de uma forma bastante simples e clara (Figura 5.3).



Figura 5.3 - Repetições consecutivas e não consecutivas descritas através da função *conc*.

Além da função *conc*, também podem ser utilizadas outras funções que permitam modificar segmentos rítmica, melódica ou estruturalmente. Por exemplo, a função *retrógrado* da Figura 5.2 inverte a ordem das notas do segmento recebido como argumento. Ainda nesse exemplo, a função *transp+1* permite transpor meio tom para cima as notas de um segmento. Não apresentamos aqui um conjunto de funções bem definido uma vez que o número de transformações que podem ser utilizadas para transformar segmentos musicais é virtualmente infinito. Existem apenas duas condições para que uma função possa fazer parte do conjunto de funções a utilizar. A primeira já foi referida no início deste capítulo e consiste no facto de as funções terem que fazer sentido do ponto de vista da análise musical. A segunda consiste em as funções terem que devolver valores que possam ser recebidos pelas outras funções do conjunto de funções, de modo a que este conjunto tenha a propriedade de fecho. No nosso caso, todas as funções devem receber sequências de notas como argumentos e devolver, igualmente, sequências de notas.

As folhas da árvore que representa a função principal são constituídas por dois tipos de terminais: segmentos e FDAs que, por sua vez, também representam segmentos. Cada segmento é representado como um par $[e d]$, onde e e d representam, respectivamente, o limite esquerdo e o limite direito do segmento. Na verdade, cada par funciona como uma função de leitura que recebe os limites de um segmento e que devolve a sequência de notas da peça correspondente a esses limites. Por seu lado, as FDAs são funções sem argumentos que consistem em apenas um terminal, correspondendo este a um segmento representado igualmente por um par $[e d]$. Cada FDA devolve, assim, a sequência de notas correspondente ao intervalo especificado pelo seu par $[e d]$.

5.2.1.1 *FDAs*

Existem três razões para utilizarmos FDAs na nossa abordagem. Primeiro, e tal como referimos no capítulo anterior, as FDAs permitem uma melhor exploração das regularidades do espaço de pesquisa. No nosso caso, estamos interessados em que os padrões que permitem otimizar a reutilização de material musical sejam identificados e mantidos de forma que o mesmo programa possa reutilizá-los facilmente. Segundo, as FDAs, quando utilizadas mais do que uma vez na função principal, permitem mostrar de forma elegante que duas ou mais partes da peça estão relacionadas. Por último, se as FDAs não fossem utilizadas, seria necessário realizar processamento extra para verificar se um determinado segmento é utilizado mais do que uma vez na função principal do programa (seria necessário compará-lo com todos os outros segmentos também utilizados pelo programa). Este último aspecto é importante no cálculo do desempenho dos indivíduos uma vez que, como se verá mais adiante, a função de avaliação tem em conta a quantidade de material musical da peça utilizado para a produzir/descrever.

Apesar de, na representação que descrevemos, as FDAs serem constituídas apenas por um nó terminal, nada impede que estas possam ter um conjunto de funções próprio. Isso representaria, aliás, uma melhoria na expressividade da representação. Vejamos, por exemplo, os dois indivíduos da Figura 5.4. O indivíduo 1 é constituído por uma função principal e por FDAs que consistem apenas num terminal. Por seu lado, o indivíduo 2 é constituído por uma função principal, por FDAs que consistem apenas num terminal e por uma FDA (FDA3) cujos conjuntos de terminais e funções

coincidem com os da função principal. O conjunto de funções de ambos os indivíduos é constituído apenas pela função *conc*.

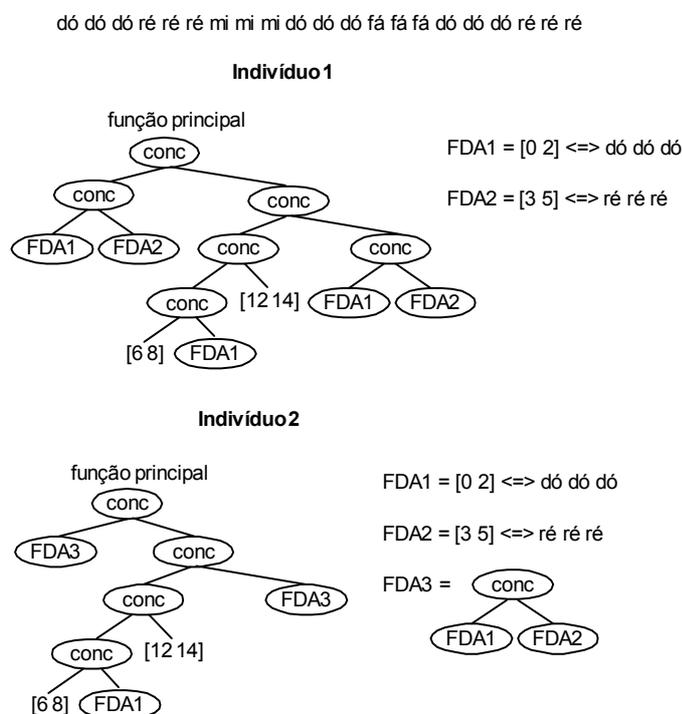


Figura 5.4 - Dois programas capazes de produzir a mesma sequência de notas.

Como se pode ver pela figura, o indivíduo 2 expressa melhor, não só o facto de que o segmento “dó dó dó ré ré ré” inicia e termina a peça, mas também que este segmento se divide em dois mais pequenos, um dos quais aparece noutra parte da peça inserido num contexto diferente. Além de uma maior expressividade, esta representação permite que sequências de segmentos como a sequência “FDA1 FDA2” necessitem de ser geradas apenas uma vez e não tantas vezes quantas as que aparecem na peça. No entanto, e apesar destas vantagens, as experiências que realizámos com esta representação deram resultados bastante inferiores aos que foram obtidos com a representação em que as FDAs consistem apenas num terminal. Assim, na secção 5.3 apresentaremos apenas os resultados obtidos com a primeira representação que descrevemos.

5.2.1.2 Restos

A utilização directa de segmentos na função principal justifica-se com a necessidade de representar restos, ou seja, segmentos que não se repetem ao longo da peça (veja-se,

por exemplo, os programas da figura anterior). Na verdade, numa primeira fase começámos por não utilizar segmentos directamente na função principal, utilizando como terminais apenas FDAs (esta abordagem encontra-se descrita em [Grilo, 2001]). No entanto, esta representação limita o número de segmentos diferentes a que o programa pode fazer referência ao número de FDAs que cada programa tiver, sendo que este número é fixo e igual para todos os programas. Por exemplo, utilizando esta representação seria impossível descrever a estrutura de uma peça musical com a estrutura ABACADAEAF... se o número de restos (segmentos B, C, D, E, F...) fosse maior que o número de FDAs menos um, uma vez que também estes segmentos teriam que ser representados por FDAs. A utilização directa de segmentos permite resolver este problema pois a função principal pode utilizar directamente tantos segmentos quantos os que forem necessários para representar os restos (que podem, ainda assim, continuar a ser representados por FDAs que sejam chamadas apenas uma vez na função principal). Repare-se, no entanto, que o número de segmentos utilizados para representar material musical reutilizável continua a estar limitado ao número de FDAs disponíveis para cada programa.

5.2.1.3 *Interpretação dos programas*

Ainda no que diz respeito à representação dos programas devemos referir que com esta representação não se pretende que os programas mostrem hierarquias de segmentos. Por exemplo, na Figura 5.5 o facto de no programa da esquerda os dois primeiros segmentos estarem ligados por um *con* não significa que os dois juntos formem um segmento de um nível hierárquico superior com significado na estrutura global da peça. O único significado que se deve atribuir a essa parte do programa é o de que o segmento “sol sol sol sol” vem a seguir ao segmento “dó ré mi fá”. Ou seja, o programa deve ser interpretado da seguinte forma: primeiro vem o segmento “dó ré mi fá”, seguido do segmento “sol sol sol sol”, seguido novamente pelo segmento “dó ré mi fá” transposto dois graus tonais para cima e retrogradado (repare-se que o termo “novamente” surge aqui devido ao uso repetido da FDA1 na função principal). Segundo esta interpretação o programa da esquerda e o programa da direita descrevem exactamente a mesma estrutura. Aliás, o valor da função de avaliação - que utilizamos nesta abordagem e que descrevemos na próxima secção - resultante da avaliação de cada um destes programas seria exactamente igual.

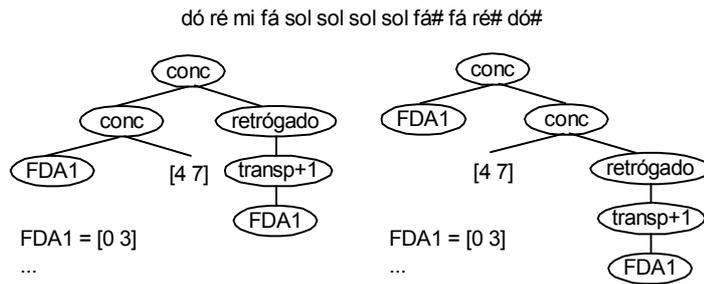


Figura 5.5 - Dois indivíduos diferentes mas equivalentes na forma como descrevem uma peça.

5.2.2 Avaliação

O desempenho de cada programa é medido tendo em conta (i) a distância entre a peça produzida pelo programa e a peça a ser analisada, e (ii) o número de notas da peça a ser analisada que é efectivamente utilizado pelo programa. O desempenho está inversamente relacionado com estes valores, pelo que o objectivo é minimizar o valor da função de avaliação, que pode ser descrita pela seguinte soma pesada de dois termos

$$f(x) = a \times d(s, p) + b \times \left(\sum_{i=1}^m t(r_i) + \sum_{j=1}^n t(FDA_j) \right).$$

Nesta expressão, x representa o programa a ser avaliado, a e b são constantes, $d(s, p)$ representa a distância entre a sequência de notas s produzida pelo programa e a peça p a ser analisada, calculada utilizando o algoritmo de Wagner-Fischer descrito no capítulo 3, m representa o número de segmentos que são utilizados directamente pela função principal, $t(r_i)$ representa o tamanho do i -ésimo segmento utilizado directamente na função principal, n representa o número de FDAs de cada indivíduo (como já referimos, este valor é igual para todos os indivíduos) e

$$t(FDA_j) = \begin{cases} d_j - e_j + 1 & \text{se } FDA_j \text{ for utilizada na função principal} \\ 0 & \text{se } FDA_j \text{ não for utilizada na função principal} \end{cases}$$

representa o tamanho de cada segmento que é utilizado na função principal através de uma FDA. Note-se que, mesmo que uma FDA seja utilizada mais do que uma vez na função principal (situação que é desejável), o tamanho do segmento que ela representa só é contabilizado uma vez. Este aspecto permite que os indivíduos capazes de

produzir a peça a ser analisada (ou uma muito próxima) utilizando menos notas sejam considerados os melhores. Por exemplo, na figura seguinte o indivíduo da direita é, de acordo com a função de avaliação, melhor que o indivíduo da esquerda porque utiliza menos notas da peça original ou, posto de outra forma, porque faz uma melhor reutilização dos segmentos que tem disponíveis.

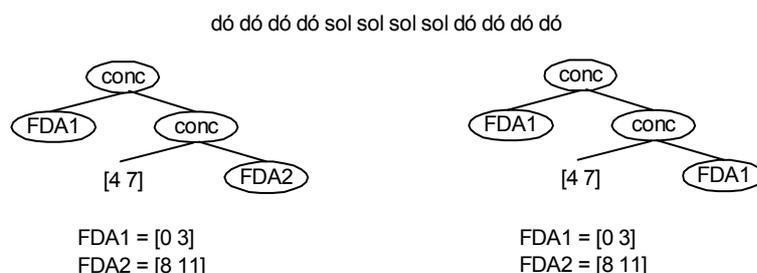


Figura 5.6 - Dois indivíduos com funções principais diferentes e com as mesmas FDAs descrevendo a mesma peça. O indivíduo da direita é, segundo a função de avaliação, melhor que o indivíduo da esquerda.

Um aspecto para o qual julgamos ser necessário chamar a atenção é o de que não é necessário que um indivíduo produza uma peça exactamente igual à peça a ser analisada para que possa ser considerado uma boa análise dessa peça. Por exemplo, podemos considerar que o indivíduo da Figura 5.7 descreve correctamente a estrutura da sequência de notas apresentada na mesma figura apesar da sequência que produz não ser exactamente igual à sequência analisada. Uma alternativa seria incluir no conjunto de funções, por exemplo, a função “*diminuir em meio tom a penúltima nota do segmento*”, pois dessa forma seria possível gerar um indivíduo que produzisse uma sequência exactamente igual à sequência a analisar. No entanto, julgamos que, em vez de utilizar um conjunto de funções com várias funções específicas como esta, é preferível utilizar um conjunto de funções mais reduzido mesmo que isso implique a existência de algumas diferenças entre a peça produzida e a peça a analisar. A utilização desse tipo de funções, além de não constituir uma mais valia em termos de clareza das análises, implicaria um aumento considerável do espaço de pesquisa, conduzindo a uma provável redução no desempenho do sistema.

dó dó dó dó dó dó ré ré ré ré ré ré dó dó dó dó si dó

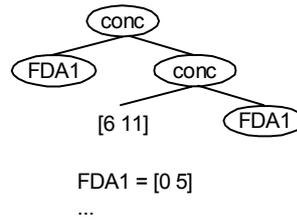


Figura 5.7 - Indivíduo que produz uma sequência ligeiramente diferente da sequência a analisar.

5.2.3 Representação das Peças

Nesta primeira abordagem ao problema da extracção de padrões decidimos utilizar uma representação baseada em notas para representar as peças musicais a analisar e os segmentos utilizados pelos programas e, por conseguinte, as peças por estes produzidas. Assim, cada peça/segmento é representada/o como uma sequência de notas em que cada nota é representada pela estrutura seguinte, baseada no sistema de representação musical Charm descrito no capítulo 3:

<<nome da nota, acidente, oitava>, duração da nota em número de semínimas>.

Além da sequência de estruturas que descrevem as notas que constituem as peças, não é utilizado nenhum outro tipo de informação (por exemplo, características de carácter global da peça). A comparação de sequências não tem necessariamente que ter em conta as dimensões melódica e rítmica em simultâneo. Por exemplo, se se pretender realizar apenas uma análise rítmica, então apenas se deve considerar a duração das notas. Igualmente, pode-se ignorar, por exemplo, a oitava a que pertence uma nota.

São duas as razões porque, nesta abordagem, optámos por uma representação baseada em notas e não por uma representação baseada em intervalos, opção que é normalmente preferível, tal como vimos no capítulo 3. Ambas estão relacionadas com o facto de, ao contrário de outras abordagens que já vimos, nesta abordagem ser necessário gerar uma peça que seja igual ou muito parecida à peça que se pretende analisar. Consideremos, na figura seguinte, a canção «Frère Jaques» e a sua representação em intervalos cromáticos:

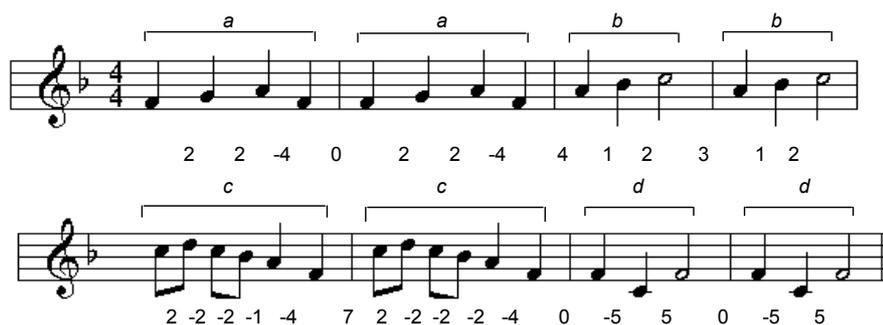


Figura 5.8 - Estrutura da canção «Frère Jacques» e a sua representação utilizando intervalos cromáticos.

Como se pode ver na figura, os segmentos que se repetem são os segmentos $a = "2\ 2\ -4"$, $b = "1\ 2"$, $c = "2\ -2\ -2\ -1\ -4"$ e $d = "-5\ 5"$. Mas, para que um programa possa produzir, por exemplo, a sequência de segmentos " ad " é necessário que produza também o intervalo que separa os dois segmentos, que neste caso é o intervalo 0. Para tal, o programa teria que utilizar o segmento " $2\ 2\ -4\ 0$ " ou o segmento " $0\ 2\ 2\ -4$ ". No entanto nenhum destes reflectiria correctamente a estrutura da peça. Além disso, a repetição de um destes segmentos impediria que a peça produzida pelo programa fosse exactamente igual ao original.

A outra razão que nos levou a optar por uma representação baseada em notas consiste no facto de, nesta abordagem, não ser possível utilizar simultaneamente representações baseadas em intervalos cromáticos e em intervalos escalares, pois cada programa só pode produzir uma sequência de intervalos de um tipo. Este facto implicaria que fosse impossível detectar transposições escalares se fosse utilizada uma representação baseada em intervalos cromáticos e vice versa.

Além destas duas razões podemos acrescentar ainda uma terceira que consiste no facto de os problemas associados às representações baseadas em notas poderem, nesta abordagem, ser resolvidos através da inclusão, no conjunto de funções, de funções que permitam transpor segmentos. Isso não significa, no entanto, que baste incluir esse tipo de funções neste conjunto para que o sistema de programação genética seja capaz de gerar programas que descrevam as transposições existentes numa peça. Significa, isso sim, que com esta representação é possível fornecer ao sistema ferramentas que lhe permitam, caso a pesquisa seja bem sucedida, gerar programas capazes de descrever as transposições existentes.

5.2.4 Operadores de Melhoramento

Durante a realização das experiências constatámos que, em alguns casos, o melhor indivíduo gerado durante o processo evolutivo, apesar de constituir uma boa descrição da peça a analisar, só não tinha um desempenho melhor porque os limites dos segmentos que utilizava não eram os mais adequados. A Figura 5.9 ilustra esta situação através de um exemplo simples. Repare-se que o indivíduo representado na figura constituiria uma descrição perfeita da estrutura da sequência de notas apresentada se, em vez do segmento [6 12], utilizasse o segmento [6 11].

dó dó dó dó dó dó ré ré ré ré ré ré dó dó dó dó dó dó

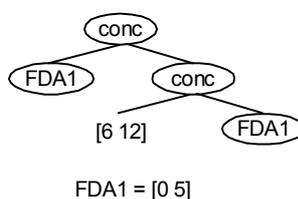


Figura 5.9 - Indivíduo que produz uma sequência ligeiramente diferente da sequência a analisar devido à utilização de um segmento cujos limites não estão correctos.

De modo a corrigir este problema, resolvemos aplicar, ao melhor indivíduo gerado em cada experiência, operadores de melhoramento que permitem modificar em uma unidade e em ambos os sentidos cada um dos limites dos segmentos utilizados na função principal do indivíduo. Estes operadores são sucessivamente aplicados a um segmento de cada vez, sendo o segmento efectivamente modificado apenas se essa modificação implicar uma melhoria no desempenho do indivíduo. Sempre que isso acontecer o operador é novamente aplicado ao segmento. Como se verá na próxima secção, estes operadores permitiram melhorar em alguns casos o desempenho do melhor indivíduo gerado durante o processo evolutivo.

5.3 Experimentação

Nesta secção apresentamos os resultados experimentais obtidos com esta abordagem para as peças «Be m'enperdout...» de Bernard de Ventadour, «Maria muoter reinû maît» atribuída a Paul Runge, «Kalenda Maya» de Raïmbaut de Vaqueiras, «Syrinx» de Claude Debussy e a 4ª variação das «Seis variaciones sobre um tema de Milán» de J. Nin-Culmell, cujas pautas são apresentadas no Anexo I. As três primeiras são canções de trovadores utilizadas por Ruwet em [Ruwet, 1972] para exemplificar o seu

procedimento de análise, já descrito no capítulo 2. Nestas peças estão assinalados os segmentos identificados por Ruwet, sendo a forma de apresentação também idêntica à que foi utilizada por este autor. A peça «Syrinx» foi escolhida por permitir comparar os nossos resultados com os da abordagem apresentada em [Smail, 1993], descrita no capítulo 3. Nesta peça assinalámos um conjunto de segmentos que devia ser possível identificar utilizando as transformações descritas pelos autores (os segmentos realmente identificados foram já apresentados no capítulo 3). Por último, a 4ª variação das «Seis variaciones sobre um tema de Milán» foi escolhida, não porque tenhamos algum termo de comparação, mas porque, à semelhança, por exemplo, da peça «Maria muoter reinû maît», apresenta uma estrutura hierárquica de segmentos que nos pareceu bastante interessante. Nenhuma das peças tem uma dimensão muito grande, sendo a mais pequena a peça «Kalenda Maya» com 96 notas, e a maior «Syrinx» com 244 (sem contabilizar as notas de embelezamento como tríolos e apogiaturas). No entanto, a estrutura das peças é, como se verá, suficiente para colocar problemas à pesquisa de programas capazes de as descrever.

5.3.1 Ambiente Experimental

A nossa abordagem à extracção de padrões utilizando programação genética foi desenvolvida com o sistema de programação genética *lil-gp* [Zongker, 1996]. A principal vantagem deste sistema consiste na facilidade com que permite definir novos problemas de programação genética. Para tal, basta definir, em linguagem C, as funções que cada indivíduo pode utilizar, os terminais, a função de avaliação e a estrutura dos indivíduos (quantas FDAs tem cada indivíduo e qual o conjunto de funções a utilizar pela função principal e pelas FDAs). Além disso, é disponibilizado um grande número de métodos de selecção e de parâmetros que podem ser utilizados para controlar o processo evolutivo. Os operadores genéticos disponibilizados são os operadores de reprodução, recombinação e mutação descritos no capítulo anterior. As experiências foram realizadas num processador AMD Athlon XP a 1.54 GHz com 256 MB de memória RAM.

5.3.2 Funções e Terminais Utilizados

As experiências com as quais obtivemos os resultados aqui apresentados foram

realizadas utilizando apenas a função *conc*. Como já referimos antes, esta função permite descrever implicitamente relações de repetição ou quase repetição, o que é suficiente para a quase totalidade das relações existentes nas peças que utilizámos. Para além deste tipo de relações existem, nestas peças, algumas transposições (segmentos a' e c' na 4ª variação e segmento a' em «Syrinx») e transformações rítmicas (segmentos A^{iii} e A^{iv} em «Syrinx»). As primeiras podem ser identificadas como repetições através da utilização da função *conc* se, na comparação das peças, não se tiver em conta a oitava a que as notas pertencem. Assim, nas experiências realizadas, esta dimensão das notas não foi considerada. Para a 4ª variação e para a peça «Syrinx», realizámos também experiências com funções que permitiam realizar transposições (*transp+1*, *transp-1*, *transp+3*, *transp-3*, *transp+8* e *transp-8*) mas os resultados obtidos foram muito inferiores. Com a peça «Syrinx», realizámos, igualmente, experiências com a função *ritmo1* que permitia transformar o segmento A no segmento A^{iii27} de modo a que fosse possível identificar e descrever explicitamente a transformação rítmica existente mas, mais uma vez, os resultados foram inferiores. As transformações rítmicas podem, no entanto, ser também identificadas como simples repetições através da utilização da função *conc*, sendo necessário, para isso, que não se considere a componente rítmica das notas na comparação das peças. Além das funções já referidas realizámos ainda, para todas as peças, experiências em que era utilizada a função *rep* que recebia um segmento e o concatenava consigo próprio, permitindo, assim, a descrição explícita de repetições consecutivas. Mais uma vez, os resultados foram inferiores aos que foram obtidos utilizando apenas a função *conc*.

Na abordagem que estamos a descrever não existe uma fase de selecção de segmentos que, tal como já referimos, são representados como terminais dos programas. Isso não significa, no entanto, que qualquer segmento possa ser utilizado. Nas experiências descritas, por exemplo, não permitimos que fossem gerados terminais que implicassem segmentos com tamanho inferior a três e superior a metade do tamanho da peça a ser analisada. Por um lado, esta gama de valores impede que a peça possa ser descrita à custa de relações entre segmentos de uma e duas notas. Por outro, não nos parece necessário permitir a geração de segmentos com um tamanho superior a

²⁷ A segmento A^{iv} é igual ao segmento A^{iii} excepto no facto de ter a última nota mais longa.

metade do número de notas da peça a analisar. Por exemplo, mesmo que a peça tenha uma estrutura AA', um dos dois segmentos tem necessariamente que ter um tamanho menor ou igual a metade do tamanho da peça, sendo sempre possível descrever os dois segmentos recorrendo apenas ao segmento mais pequeno. Outro exemplo: se a peça tiver uma estrutura AAX, tendo X um tamanho superior a metade do tamanho da peça, isso não tem que constituir um problema uma vez que é possível um programa utilizar mais do que um segmento para descrever o *resto* X (AAYZ). Depois de concluído o processo evolutivo pode-se resolver o problema submetendo o programa a um processo de simplificação trivial que permita juntar restos seguidos.

Nota: A utilização de um conjunto de funções constituído apenas pela função *conc* permite que os programas possam ser representados de uma forma mais simples do que aquela que temos vindo a fazer até aqui. Como se pode ver na Figura 5.10, a função principal passará a ser representada simplesmente como uma sequência de segmentos na qual a operação de concatenação está apenas implícita. Esta forma de representação tem ainda a vantagem de permitir que, por exemplo, os dois indivíduos da figura sejam representados da mesma forma. Outra simplificação que faremos consistirá, tal como também mostra a figura, em mostrar apenas as FDAs utilizadas na função principal.

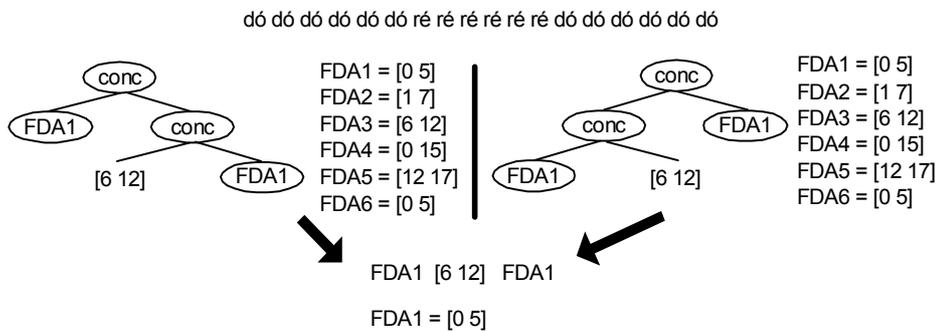


Figura 5.10 - Representação simplificada de dois indivíduos equivalentes.

5.3.3 Parâmetros do Processo Evolutivo

Nesta secção descrevemos os parâmetros do processo evolutivo que utilizámos nas experiências cujos resultados mostramos nas próximas secções. A configuração aqui apresentada é apenas uma das muitas que experimentámos, sendo aquela que melhores resultados permitiu obter. A tabela seguinte descreve os valores dos

parâmetros que foram utilizados em cada experiência. Para cada peça foram realizadas 30 experiências independentes utilizando estes parâmetros.

Descrição do Parâmetro	Valor
Tamanho da população	3000
Método de geração da população inicial	Criação mista
“rampa” de profundidades	4-7
Máximo de nós	500
Número de FDAs por indivíduo	6
Condição de paragem	100 gerações
Probabilidade de aplicação do operador de recombinação	0.6
Método de selecção dos indivíduos para o operador de recombinação	Torneio ($t = 2$)
Probabilidade de aplicação do operador de reprodução	0.1
Método de selecção dos indivíduos para o operador de reprodução	Seleccção sôfrega ($c = 10.7\%$; $p = 80\%$)
Probabilidade de aplicação do operador de mutação	0.3
Método de selecção dos indivíduos para o operador de mutação	Torneio ($t = 2$)
Valor do parâmetro a da função de avaliação	5
Valor do parâmetro b da função de avaliação	1

Tabela 5.1 - Parâmetros do processo evolutivo.

Dos parâmetros apresentados, a condição de paragem e os parâmetros da função de avaliação merecem-nos alguns comentários.

Começamos pela condição de paragem. Esta foi definida como sendo um número máximo de gerações, uma vez que é impossível determinar, a partir do valor da função de avaliação e respectivos termos, se um determinado indivíduo é ou não uma solução para o problema. Por exemplo, não podemos saber se um indivíduo poderia utilizar menos notas, mesmo que produza uma peça exactamente igual àquela que se pretende analisar. Este facto impede-nos, igualmente, de apresentar os resultados obtidos baseando-nos apenas no desempenho dos indivíduos. Por isso, os resultados serão apresentados comparando sobretudo os segmentos utilizados pelos indivíduos obtidos nas experiências com os que são assinalados nas peças no Anexo I. Isso não nos impede, no entanto, de utilizarmos o valor da função de avaliação ou de um dos seus componentes para, logo à partida, pormos de parte indivíduos que julgemos não deverem ser considerados solução para o problema. Assim, decidimos estabelecer um limite para a distância entre a peça a analisar e a peça produzida pelos indivíduos, a partir do qual estes não são considerados soluções mesmo que apresentem uma

estrutura correcta. Este limite impede que consideremos como solução indivíduos que produzam uma peça muito diferente da que se pretende analisar. O limite foi fixado em 5% do tamanho da peça a analisar.

Quanto aos parâmetros da função de avaliação, julgamos que é necessário referir que o parâmetro a deve ter um valor superior ao parâmetro b . Se assim não acontecer, o sistema tende a favorecer indivíduos que usem menos notas, ainda que isso implique que a peça produzida seja muito diferente da peça a analisar.

5.3.4 «Be m'anperdut...»

Nas tabelas 5.2 e 5.3 mostramos os resultados obtidos para os melhores indivíduos nas 30 experiências realizadas com a peça «Be m'anperdut...», respectivamente, antes e depois da aplicação dos operadores de melhoramento. Cada experiência demorou, em média, 103.6 segundos, com um desvio padrão de 12.4 segundos.

Desempenho	Distância entre a peça produzida e a peça a analisar	Número de notas utilizadas	Número de indivíduos
100	0	100	19
104	1	99	7
105	1	100	1
108	2	98	1
110	2	100	1
112	3	97	1

Tabela 5.2 - Resultados dos melhores indivíduos obtidos para a peça «Be m'anperdut...», antes da aplicação dos operadores de melhoramento.

Desempenho	Distância entre a peça produzida e a peça a analisar	Número de notas utilizadas	Número de indivíduos
100	0	100	30

Tabela 5.3 - Resultados dos melhores indivíduos obtidos para a peça «Be m'anperdut...», depois da aplicação dos operadores de melhoramento.

Em todas as experiências realizadas foram obtidos indivíduos que apresentam uma estrutura de segmentos igual à estrutura AAX, de segmentos de nível I, identificada por Ruwet. Antes da aplicação dos operadores de melhoramento, 19 dos indivíduos, com desempenho igual a 100, apresentam segmentos que correspondem exactamente aos segmentos identificados por Ruwet. Os outros 11 apresentam segmentos ligeiramente diferentes, daí o seu desempenho inferior. Como se pode ver pela

Tabela 3, a aplicação dos operadores de melhoramento permitiu que todos estes 11 indivíduos melhorassem o seu desempenho para o valor 100. Isto equivale a dizer que no final de todo o processo todos os 30 indivíduos apresentam uma sequência de segmentos exactamente igual à sequência de segmentos de nível I assinalada no Anexo I. A figura seguinte mostra 3 indivíduos obtidos depois do processo evolutivo, antes da aplicação dos operadores de melhoramento.

Indivíduo 1		
FDA4	FDA4	FDA3
FDA3 = [84 141] = X	(desempenho = 100; diferença = 0; notas utilizadas = 100)	
FDA4 = [42 83] = A		
Indivíduo 2		
FDA5	FDA5	FDA1
FDA1 = [83 141] ≈ X	(desempenho = 105; diferença = 1; notas utilizadas = 100)	
FDA5 = [0 40] ≈ A		
Indivíduo 3		
FDA5	FDA5	FDA1
FDA1 = [84 140] ≈ X	(desempenho = 104; diferença = 1; notas utilizadas = 99)	
FDA5 = [0 41] ≈ A		

Figura 5.11 - Três dos indivíduos obtidos nas experiências realizadas, antes da aplicação dos operadores de melhoramento.

Os gráficos 5.1, 5.2 e 5.3 permitem visualizar a média, respectivamente, do desempenho, da distância entre a peça produzida e a peça a analisar, e do número de notas utilizadas pelos melhores indivíduos ao longo das 100 gerações das 30 experiências realizadas. Nestes gráficos pode ver-se que, ao contrário do que acontece com a distância entre a peça produzida e a peça a analisar, o número de notas utilizadas cresce ao longo das 100 gerações, ocorrendo apenas uma descida abrupta entre a geração 3 e a geração 7, imediatamente recuperada na geração 8. Este fenómeno, que nos parece algo surpreendente, acontece em todas as peças à excepção da peça «Maria muoter reinû maît», pelo que só para essa peça voltaremos a apresentar gráficos deste tipo. A razão de, normalmente, acontecer uma subida, e não uma descida como seria de esperar, parece-nos estar relacionada com o facto de esta componente da função de avaliação ter um peso menor do que a distância entre a peça produzida e a peça a analisar. Devido a esta desigualdade nos pesos é frequente ser possível melhorar o desempenho dos indivíduos diminuindo a distância, ainda que isso implique o aumento do número de notas utilizadas. Com os pesos que foram

utilizados nestas experiências, por exemplo, pode melhorar-se o desempenho de um indivíduo diminuindo a distância em uma unidade e aumentando o número de notas utilizadas em 4 unidades.

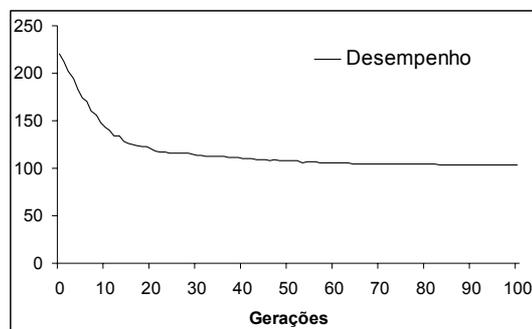


Gráfico 5.1 - Média do desempenho dos melhores indivíduos ao longo das 100 gerações.

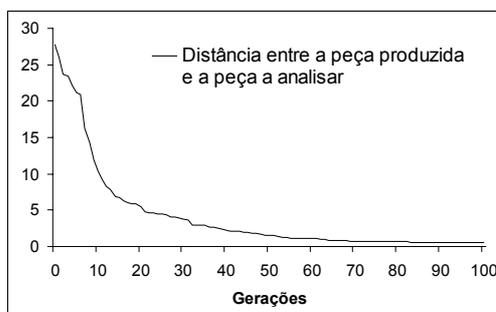


Gráfico 5.2 - Média da distância entre a peça produzida pelos melhores indivíduos ao longo das 100 gerações e a peça a analisar.

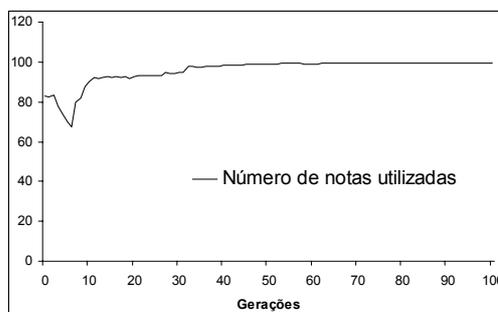


Gráfico 5.3 - Média do número de notas utilizadas pelos melhores indivíduos ao longo das 100 gerações.

5.3.5 «Maria muoter reinû maît»

Nas tabelas 5.4 e 5.5 mostramos os resultados obtidos para os melhores indivíduos nas 30 experiências realizadas com a peça «Maria muoter reinû maît», respectivamente, antes e depois da aplicação dos operadores de melhoramento. Cada experiência demorou, em média, 66 segundos, com um desvio padrão de 8 segundos.

Desempenho	Distância entre a peça produzida e a peça a analisar	Número de notas utilizadas	Número de indivíduos
69	3	54	12
70	3	55	6
78	5	53	4
79	5	54	3
88	3	73	1
89	3	74	2
96	5	71	1
103	6	73	1

Tabela 5.4 - Resultados dos melhores indivíduos obtidos para a peça «Maria muoter reinû maît», antes da aplicação dos operadores de melhoria.

Desempenho	Distância entre a peça produzida e a peça a analisar	Número de notas utilizadas	Número de indivíduos
69	3	54	12
70	3	55	8
78	5	53	2
79	5	54	1
88	3	73	3
89	3	74	2

Tabela 5.5 - Resultados dos melhores indivíduos obtidos para a peça «Maria muoter reinû maît», depois da aplicação dos operadores de melhoria.

Em ambas as tabelas podemos dividir os indivíduos em dois grupos: os que têm um desempenho menor ou igual que 79 e os que têm um desempenho superior a este valor. Os indivíduos do primeiro grupo apresentam uma estrutura de segmentos idêntica à estrutura AABB, de segmentos de nível I, identificada por Ruwet. A diferença entre os indivíduos com um desempenho 69 e os que têm desempenho 70 reside no facto de os primeiros utilizarem, como segmento A, o segmento [36 70], enquanto que os segundos utilizam o segmento [0 35]. Ou seja, os primeiros utilizam o segmento correspondente ao segmento que na pauta é identificado com A', enquanto que os segundos utilizam o segmento identificado na pauta como A. A diferença no desempenho destes indivíduos deve-se ao facto de o segmento A' ter menos uma nota que o segmento A. A figura seguinte mostra dois indivíduos obtidos nas experiências realizadas, um com desempenho 69 e o outro com desempenho 70. Repare-se que, enquanto o primeiro utiliza a segunda ocorrência do segmento B ([90 108]), o segundo utiliza a primeira ocorrência ([71 89]). Esta diferença é, no entanto, totalmente irrelevante, tanto no que diz respeito à estrutura descrita como ao

desempenho.

Indivíduo 1			
FDA6	FDA6	FDA1	FDA1
FDA1 = [90 108] = B		(desempenho = 69; diferença = 3; notas utilizadas = 54)	
FDA6 = [36 70] = A'			

Indivíduo 2			
FDA5	FDA5	FDA2	FDA2
FDA2 = [0 35] = A		(desempenho = 70; diferença = 3; notas utilizadas = 55)	
FDA5 = [71 89] = B			

Figura 5.12 - Dois indivíduos com uma estrutura AABB obtidos nas experiências com a peça «Maria muoter reinû mãit».

O pior desempenho dos outros indivíduos do primeiro grupo deve-se ao facto de os limites dos segmentos que utilizam serem ligeiramente diferentes dos limites correctos. Através da aplicação dos operadores de melhoria, foi possível melhorar o desempenho de alguns destes indivíduos. Assim, dois dos indivíduos com desempenho 78 foram transformados em indivíduos com desempenho 69, enquanto que dois dos indivíduos com desempenho 79 foram transformados em indivíduos com desempenho 70. Os operadores não são capazes de melhorar o desempenho dos outros três indivíduos (dois com desempenho 78 e um com desempenho 79 na Tabela 5.5) porque, para isso, seria necessário modificar mais do que um limite ao mesmo tempo. A Figura 5.13 mostra um destes indivíduos. Para melhorar o seu desempenho seria necessário aplicar operadores que diminuíssem simultaneamente em uma unidade os limites esquerdo e direito do segmento representado pela FDA5 e em uma unidade o limite esquerdo do segmento representado pela FDA3. Este problema ocorre igualmente nas outras peças, pelo que não voltaremos a fazer referência a ele nas próximas secções.

FDA5	FDA5	FDA3	FDA3
FDA3 = [37 71] ≈ A'		(desempenho = 78; diferença = 5; notas utilizadas = 53)	
FDA5 = [72 89] ≈ B			

Figura 5.13 - Exemplo de um indivíduo que os operadores não conseguem melhorar.

Quanto aos indivíduos do segundo grupo, estes apresentam uma estrutura de segmentos que podemos descrever como AAX. Isto significa que a função principal destes indivíduos consiste na repetição consecutiva de uma FDA, seguida de um

segmento utilizado apenas uma vez. Os indivíduos com desempenho 88 e 89 utilizam, respectivamente, os segmentos [36 70] e [0 35] na FDA correspondente ao segmento A, ou seja, utilizam os segmentos que Ruwet identifica como A' e A. O terceiro segmento utilizado por estes indivíduos na função principal é o segmento [71 108] que corresponde às duas ocorrências do segmento que Ruwet identifica como B. Isto significa que os indivíduos deste grupo constituem uma descrição mais pobre da estrutura desta peça do que os indivíduos do primeiro grupo, uma vez que não mostram a repetição do segmento B. A figura seguinte mostra um destes indivíduos, com desempenho 88.

FDA3	FDA3	FDA6	
FDA3 = [36 70] = A'			(desempenho = 88; diferença = 3; notas utilizadas = 73)
FDA6 = [71 108] = BB			

Figura 5.14 - Indivíduo com uma estrutura AAX obtido nas experiências com a peça «Maria muoter reinû maît».

Tal como acontece no primeiro grupo, o pior desempenho dos outros indivíduos do segundo grupo (com desempenho 96 e 103) deve-se à utilização de segmentos com limites ligeiramente diferentes dos limites correctos. No entanto, como se pode confirmar na tabela 5, neste caso foi possível, através da aplicação dos operadores de melhoramento, transformar estes indivíduos em indivíduos com desempenho 88.

Os gráficos 5.4, 5.5 e 5.6 permitem visualizar a média, respectivamente, do desempenho, da distância entre a peça produzida e a peça a analisar, e do número de notas utilizadas pelos melhores indivíduos ao longo das 100 gerações das 30 experiências realizadas. Tal como referimos antes, esta é a única peça, das que utilizámos, para a qual a média do número de notas utilizadas pelo melhor indivíduo no final do processo evolutivo é menor do que a média do número de notas utilizadas no início do processo. Nestes gráficos podemos comprovar que para melhorar o desempenho dos indivíduos descendo o número de notas utilizadas ao mesmo tempo que se aumenta a distância entre a peça produzida e a peça a analisar, é necessário que essa descida seja muito grande comparada com a subida da distância. Por exemplo, na geração 7 a média da distância entre a peça produzida e a peça a analisar sobe ligeiramente, sendo necessária uma descida abrupta da média do número de notas utilizadas para que o desempenho melhore. Tal como também referimos

anteriormente, este fenómeno deve-se à diferença dos pesos atribuídos a cada um dos termos da função de avaliação.

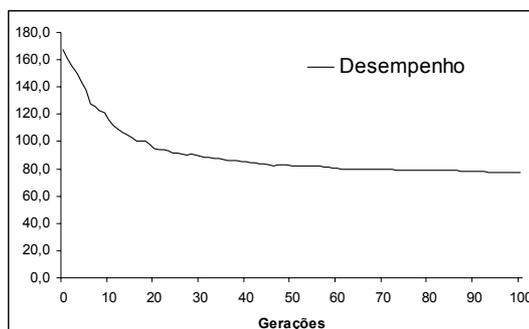


Gráfico 5.4 - Média do desempenho dos melhores indivíduos ao longo das 100 gerações.

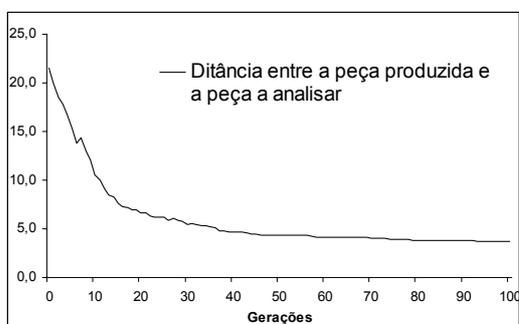


Gráfico 5.5 - Média da distância entre a peça produzida pelos melhores indivíduos ao longo das 100 gerações e a peça a analisar.

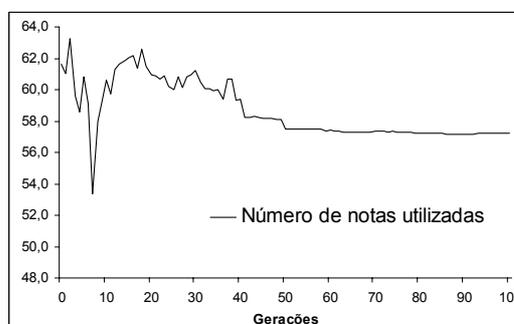


Gráfico 5.6 - Média do número de notas utilizadas pelos melhores indivíduos ao longo das 100 gerações.

5.3.6 «Kalenda Maya»

Nas tabelas 5.6 e 5.7 mostramos os resultados dos melhores indivíduos obtidos nas 30 experiências realizadas com a peça «Kalenda Maya», respectivamente, antes e depois da aplicação dos operadores de melhoramento. Cada experiência demorou, em média, 58.2 segundos, com um desvio padrão de 11.3 segundos.

Desempenho	Distância entre a peça produzida e a peça a analisar	Número de notas utilizadas	Número de indivíduos
74	0	74	6
78	1	73	23
82	2	72	1

Tabela 5.6 - Resultados dos melhores indivíduos obtidos para a peça «Kalenda Maya», antes da aplicação dos operadores de melhoramento.

Desempenho	Distância entre a peça produzida e a peça a analisar	Número de notas utilizadas	Número de indivíduos
74	0	74	30

Tabela 5.7 - Resultados dos melhores indivíduos obtidos para a peça «Kalenda Maya», depois da aplicação dos operadores de melhoramento.

Em todas as experiências realizadas com esta peça foram obtidos indivíduos que apresentam uma estrutura AAX. Como se pode comprovar pelas tabelas, os operadores de melhoramento permitiram melhorar o desempenho de 24 indivíduos, que passaram a ter um desempenho de 74. Assim, no final da aplicação dos operadores, todos os indivíduos apresentam uma estrutura equivalente à estruturado indivíduo apresentado na Figura 5.15. Isto significa que todos os indivíduos obtidos nas 30 experiências constituem uma fraca descrição da estrutura da peça, se os compararmos com a descrição de Ruwet.

FDA5 FDA5 [49 95]	
FDA5 = [0 21] = A	(desempenho = 74; diferença = 0; notas utilizadas = 74)

Figura 5.15 - Indivíduo obtido nas experiências realizadas com a peça «Kalenda Maya».

A razão de o sistema não ser capaz de produzir indivíduos que mostrem outros segmentos que não apenas as duas ocorrências do segmento A pode ser explicada, pelo menos em parte, pela diferença de pesos dos dois termos da função de avaliação. Por exemplo, com os pesos que utilizámos nas experiências cujos resultados aqui apresentamos não há vantagem nenhuma em descrever o segmento D' em função do segmento D e vice-versa. Isto, porque o erro resultante dessa reutilização de material musical implicaria um acréscimo de 30 unidades no valor dado pela função de avaliação (correspondente à diferença de 6 notas entre os segmentos D e D') enquanto que o máximo que se conseguiria “poupar” em notas utilizadas seria apenas de 11 unidades (correspondente ao número de notas do segmento D'), caso se utilizasse o segmento D.

Este argumento perde alguma força no caso dos segmentos B e B' uma vez que ambos são constituídos por 10 notas e que a distância entre eles é de 2 notas. Ou seja, em termos do valor do desempenho, é indiferente descrever ou não um segmento à

custa do outro porque o número de notas que se “poupa” ao fazê-lo é exactamente igual ao número de notas de diferença entre os dois multiplicado por 5. Isto significa que talvez fosse de esperar que o sistema gerasse pelo menos alguns indivíduos em que o segmento B' fosse substituído pelo segmento B (ou o contrário) na peça produzida. No entanto, tal como mostram os resultados, isso não acontece, parecendo, pelo contrário, que o sistema, “em caso de dúvida”, “prefere” indivíduos mais simples no que ao número de segmentos utilizados diz respeito²⁸. Devemos referir que, além da configuração de parâmetros aqui apresentada, outras foram testadas, as quais tornavam vantajosa a utilização de um destes segmentos para descrever o outro. Nomeadamente, foram utilizadas as seguintes configurações para os pesos da função de avaliação: $(a = 4, b = 1)$, $(a = 3, b = 1)$ e $(a = 2, b = 1)$. No entanto, em nenhuma das experiências realizadas com estes pesos foi gerado um indivíduo que permitisse descrever o segmento B em função do segmento B', ou o contrário. Este facto torna ainda mais difícil atribuir a culpa à incapacidade do sistema em gerar indivíduos melhores apenas à diferença entre os pesos da função de avaliação.

Uma possível justificação para esta incapacidade pode ser o número de operações que, em determinadas situações, é necessário fazer sobre um indivíduo para que este se transforme num outro com melhor desempenho. Consideremos, por exemplo, os dois indivíduos da Figura 5.16, capazes de produzir sem qualquer erro a peça «Kalenda Maya». O indivíduo 1 é equivalente aos indivíduos obtidos nas experiências cujos resultados aqui apresentamos. Este indivíduo, porque consegue gerar a peça sem quaisquer erros e porque o consegue fazer reutilizando algum material musical, pode ser considerado um mínimo local. Por seu lado, o indivíduo 2 apresenta uma estrutura AABBX onde A (FDA1) e B (FDA3) coincidem, respectivamente, com os segmentos A e B identificados por Ruwet, e X (FDA4) engloba a sequência de segmentos que Ruwet descreve por ccDccD'. Ou seja, o indivíduo 2, apesar de também ser um mínimo local, constitui uma melhor descrição da estrutura da peça do que o indivíduo 1. Mas, para conseguir transformar o indivíduo 1 no indivíduo 2 é necessário fazer, sobre o primeiro, pelo menos as seguintes alterações:

²⁸ Será que é o princípio da “navalha de Occam” a funcionar?

1. Modificar a FDA3 de modo a que passe a representar o segmento [44 53].
2. Modificar a FDA4 de modo a que passe a representar o segmento [64 95].
3. Substituir o terminal FDA2 na função principal por uma sub-árvore equivalente à sub-árvore $\text{conc}(\text{conc}(\text{FDA3}, \text{FDA3}), \text{FDA4})$ ²⁹.

Cada uma destas modificações necessita de ser realizada através da aplicação de, pelo menos um operador de mutação ou de recombinação. Uma vez que cada indivíduo é sujeito no máximo a um operador genético por geração, seriam necessárias pelo menos três gerações para transformar o indivíduo 1 no indivíduo 2 sem que houvesse, entretanto, qualquer melhoria no seu desempenho. Este facto, parece-nos, pode ser responsável pela dificuldade do sistema em gerar indivíduos melhores. No entanto, julgamos que é necessário mais estudo para o confirmar.

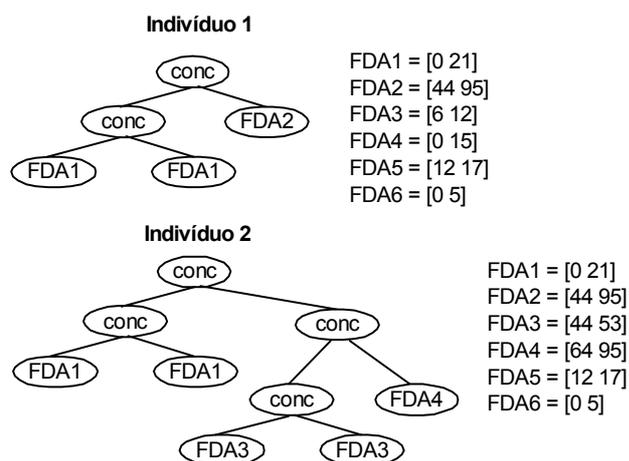


Figura 5.16 - Dois indivíduos capazes de produzir sem erros a peça «Kalenda Maya».

5.3.7 4ª Variação das «Seis variaciones sobre um tema de Milán»

Nas tabelas 5.8 e 5.9 mostramos os resultados dos melhores indivíduos obtidos nas 30 experiências realizadas com a 4ª variação das «Seis variaciones sobre um tema de Milán», respectivamente, antes e depois da aplicação dos operadores de melhoramento. Cada experiência demorou, em média, 203.1 segundos, com um desvio padrão de 13.9 segundos.

²⁹ Também podia ser a sub-árvore $\text{conc}(\text{FDA3}, \text{conc}(\text{FDA3}, \text{FDA4}))$.

Desempenho	Distância entre a peça produzida e a peça a analisar	Número de notas utilizadas	Número de indivíduos
151	0	151	8
155	1	150	8
156	1	151	3
157	1	152	1
159	2	149	4
161	2	151	1
162	2	152	1
163	3	148	2
165	3	150	1
166	3	151	1

Tabela 5.8 - Resultados dos melhores indivíduos obtidos para a 4ª variação das «Seis variaciones sobre um tema de Milán», antes da aplicação dos operadores de melhoramento.

Desempenho	Distância entre a peça produzida e a peça a analisar	Número de notas utilizadas	Número de indivíduos
151	0	151	29
160	2	150	1

Tabela 5.9 - Resultados dos melhores indivíduos obtidos para a 4ª variação das «Seis variaciones sobre um tema de Milán», depois da aplicação dos operadores de melhoramento.

Em todas as experiências realizadas foram obtidos indivíduos que apresentam uma estrutura de segmentos igual à estrutura AXA, de segmentos de nível I, por nós identificada utilizando o método de Ruwet. Dos indivíduos obtidos, apenas os que têm um desempenho de 151 utilizam segmentos exactamente iguais aos que assinalámos na peça. Os outros, embora tenham uma estrutura correcta, utilizam segmentos ligeiramente diferentes. Isto significa que, depois da aplicação dos operadores de melhoramento, só um indivíduo não utiliza segmentos exactamente iguais àqueles por nós identificados. A figura seguinte mostra um dos indivíduos com desempenho 151 obtidos nas experiências realizadas com esta peça.

FDA6	FDA2	FDA6	
FDA2 = [70 150] = X			(desempenho = 151; diferença = 0; notas utilizadas = 151)
FDA6 = [0 69] = A			

Figura 5.17 - Indivíduo obtido nas experiências realizadas com a 4ª Variação das «Seis variaciones sobre um tema de Milán».

5.3.8 «Syrinx»

A peça «Syrinx» foi aquela com que se obtiveram os piores resultados. Na tabela

seguinte mostramos os resultados dos melhores indivíduos obtidos nas 30 experiências realizadas com esta peça, depois da aplicação dos operadores. Cada experiência demorou, em média, 220.9 segundos, com um desvio padrão de 15.7 segundos.

Desempenho	Distância entre a peça produzida e a peça a analisar	Número de notas utilizadas	Número de indivíduos
233	0	233	1
237	1	232	1
244	0	244	1
245	3	230	4
246	1	241	1
247	1	242	1
252	2	242	2
253	3	238	3
255	3	240	3
256	3	241	6
260	4	240	2
262	4	242	1
264	5	239	1
267	6	237	1
276	8	236	1
289	14	219	1

Tabela 5.10 - Resultados dos melhores indivíduos obtidos para a peça «Syrinx», depois da aplicação dos operadores de melhoramento.

Como se pode ver nesta tabela, para esta peça o sistema é praticamente incapaz de gerar indivíduos que façam reutilização de material musical de modo a produzirem umas partes da peça à custa de outras. Por exemplo, o indivíduo com desempenho 244 (a peça tem 244 notas) produz uma peça exactamente igual à original sem repetir nenhuma das FDAs na função principal. As únicas excepções dignas de registo são os indivíduos com desempenho 233 e 237 representados na Figura 5.18. O indivíduo 1 utiliza a primeira ocorrência do segmento A (mais uma nota) para produzir a segunda, enquanto que o indivíduo 2 utiliza a primeira ocorrência do segmento F para produzir a sua segunda ocorrência.

Indivíduo 1 (desempenho = 233; diferença = 0; notas utilizadas = 233)

FDA1 FDA6 FDA1 FDA4 [123 243]

FDA1 = [0 10] \approx A

FDA4 = [25 122]

FDA6 = [11 13]

Indivíduo 2 (desempenho = 237; diferença = 1; notas utilizadas = 232)

FDA2 FDA4 FDA4 [142 243]

FDA2 = [0 118]

FDA4 = [119 129] = F

Figura 5.18 - Os dois melhores indivíduos obtidos nas experiências realizadas com a peça «Syrinx».

É possível confirmar na tabela apresentada que o sistema não tem quaisquer problemas em gerar indivíduos capazes de produzir a peça original. Na realidade, apenas um dos indivíduos gerados produz uma peça cuja distância em relação à peça original ultrapassa 5% do tamanho desta. O problema está, assim, na dificuldade em gerar indivíduos que reutilizem material musical. Tendo em conta este facto, e tal como fizemos para as outras peças, realizámos experiências em que a diferença dos pesos dos termos da função de avaliação não era tão grande. O objectivo desta redução da diferença entre os pesos consistia em colocar o ênfase da pesquisa na procura de programas que fizessem reutilização de material musical e não tanto de programas que produzissem uma peça muito próxima da original. No entanto, os resultados destas experiências foram igualmente desencorajadores.

Outra experiência que realizámos consistiu em ignorar a componente rítmica das notas na comparação das peças. Tal como referimos antes, esta opção permite que um segmento possa ser descrito à custa de outro se os dois segmentos forem semelhantes no que à componente melódica diz respeito, independentemente da duração das respectivas notas. No caso da peça «Syrinx», esta opção permite, se a pesquisa for bem sucedida, que os segmentos A^{III} e A^{IV} possam ser descritos à custa, por exemplo, do segmento A e vice versa, sem que isso implique o aumento da distância entre a peça produzida e a peça original. De facto, nas experiências realizadas, embora o valor da média do desempenho dos melhores indivíduos tenha piorado (a média dos indivíduos da Tabela 5.10 é de 254.3, enquanto que nesta experiência é de 259.4), foi obtido um indivíduo, apresentado na figura seguinte, que permite produzir/descrever os segmentos A^{II} e A^{III} à custa do segmento A^{IV} . Apesar disso, não podemos

considerar como satisfatórios os resultados obtidos nesta experiência, uma vez que nesta peça existem muito mais relações que podem ser descobertas.

[0 75] FDA5 FDA4 FDA4 FDA4 [231 243]
FDA4 = [221 230] = A ^N (desempenho = 224; diferença = 0; notas utilizadas = 224)
FDA5 = [76 200]

Figura 5.19 - Melhor indivíduo obtido em experiências realizadas com a peça «Syrinx», nas quais a duração das notas não foi tida em conta.

5.3.9 Segunda Segmentação

Algumas das peças que utilizámos nas experiências realizadas têm uma estrutura hierárquica de segmentos, ou seja, são constituídas por segmentos de maiores dimensões que podem ser decompostos em segmentos mais pequenos. Tal como foi possível ver para algumas delas, os indivíduos capazes de descrever correctamente a sua estrutura - quando o sistema os consegue gerar - fazem-no, praticamente sempre, utilizando apenas os maiores segmentos existentes na peça. Esta tendência do sistema em gerar apenas este tipo de indivíduos, não sendo um facto negativo em si mesmo, deixa em aberto a questão de como realizar, utilizando programação genética, uma segunda segmentação de uma peça de modo a identificar os seus segmentos mais pequenos.

Uma forma óbvia de realizar uma segunda segmentação consiste em aplicar exactamente o mesmo processo de segmentação a cada um dos segmentos já identificados como se cada um deles fosse uma peça a analisar. No entanto, esta abordagem pode, em certas situações, ser inconsequente, uma vez que os segmentos anteriormente identificados podem não ser compostos por sub-segmentos relacionados. Consideremos, por exemplo, que uma peça tem uma estrutura ABA e que $A = ab$ e que $B = cb$. Se tentarmos analisar separadamente os segmentos A e B, então seremos incapazes de identificar os segmentos a , b e c , pois não existe qualquer relação entre a e b e entre b e c .

Outra abordagem possível consiste em voltar a aplicar todo o processo à peça a analisar mas, desta vez, impondo a restrição de que só possam ser gerados, como terminais, segmentos que pertençam aos segmentos identificados na primeira segmentação (Esta abordagem encontra-se igualmente descrita em [Grilo, 2001]). Por

exemplo, se quiséssemos fazer uma segunda segmentação da peça «Maria muoter reinû maît» a partir do indivíduo 1 da Figura 5.12 então só poderiam ser gerados segmentos pertencentes aos segmentos [36 70] e [90 108]. Esta abordagem tem, no entanto, a desvantagem de ter que se gerar a sub-estrutura dos segmentos anteriormente identificados tantas vezes quantas as que estes aparecerem na peça. Por exemplo, para a peça «Maria muoter reinû maît», a sequência de segmentos *ab'cb'* teria que ser gerada duas vezes na função principal porque o segmento A' aparece duas vezes (embora não exactamente igual).

Este problema pode ser resolvido se, em vez de tentarmos gerar novamente a peça a analisar, tentarmos gerar uma outra constituída pelos segmentos anteriormente identificados mas sem que estes se repitam. Por exemplo, se quiséssemos fazer uma segunda segmentação da peça «Maria muoter reinû maît» a partir do indivíduo 1 da Figura 5.12, a peça a analisar seria a peça “[36 70][90 108]”. Tal como na abordagem anterior, os segmentos gerados como terminais teriam que pertencer aos segmentos [36 70] e [90 108].

Apesar de os resultados obtidos com esta última abordagem terem sido melhores do que os que foram obtidos com a abordagem anterior, apenas para a peça «Be m'anperdut...» foram obtidos resultados estatisticamente interessantes. A segunda segmentação desta peça foi realizada a partir do indivíduo 1 da Figura 5.11, pelo que a peça a analisar consistia na sequência de notas “[42 83] [84 141]”. A figura seguinte mostra um dos melhores indivíduos gerados nas 30 experiências realizadas.

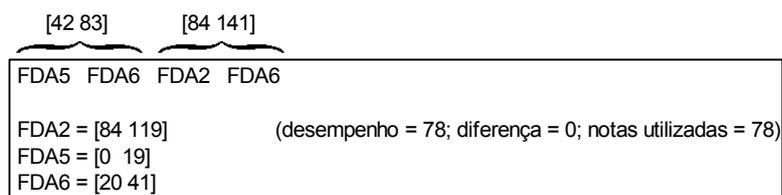


Figura 5.20 - Indivíduo que representa uma segunda segmentação da peça «Be m'anperdut...».

Este indivíduo permite verificar que os segmentos A ([0 41] ou [42 83]) e X ([84 141]) assinalados na peça têm um sub-segmento em comum. Praticamente todos os outros indivíduos obtidos nas experiências têm uma estrutura igual à deste indivíduo, excepto no facto de o sub-segmento comum aos segmentos A e X ser de maiores dimensões.

Os resultados obtidos nas experiências realizadas com as outras peças foram muito inferiores, embora tenhamos que reconhecer que a estrutura dessas peças seja algo mais complexa do que a da peça «Be m'anperdut...». Assim, para a peça «Maria muoter reinû maît», apenas um dos indivíduos gerados permite identificar as ocorrências do segmento *d1*, bem como duas das ocorrências do segmento *b'*. A figura seguinte mostra este indivíduo. Como se pode ver, a segunda segmentação desta peça foi realizada utilizando os segmentos [36 70] e [90 108], identificados na primeira segmentação.

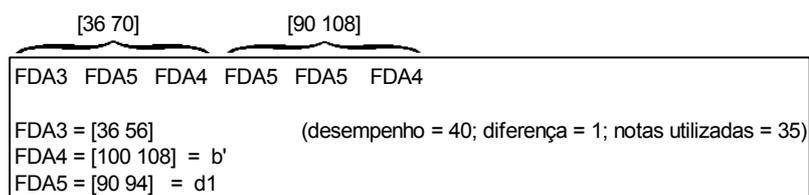


Figura 5.21 - Indivíduo que representa uma segunda segmentação da peça «Maria muoter reinû maît».

Quanto à 4ª variação, praticamente não foram gerados indivíduos capazes de reutilizar material musical, sendo uma das únicas exceções o indivíduo da figura seguinte, o qual permite identificar as duas ocorrências do segmento *e*.

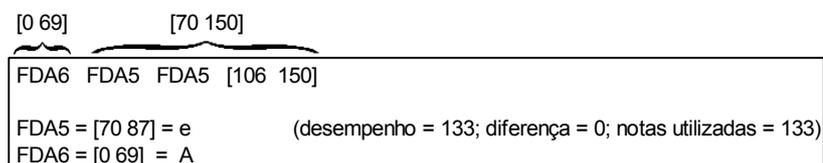


Figura 5.22 - Indivíduo que representa uma segunda segmentação da 4ª variação das «Seis variaciones sobre um tema de Milán».

Nas experiências realizadas com a peça «Kalenda Maya» não foram obtidos indivíduos dignos de registo. Finalmente, não foram realizadas segundas segmentações sobre a peça «Syrinx» dado os resultados obtidos na primeira segmentação não serem satisfatórios.

5.4 Conclusão

Neste capítulo descrevemos a primeira das duas abordagens ao problema da extracção de padrões baseadas em algoritmos evolucionários que apresentamos nesta dissertação. Inicialmente, a programação genética pareceu-nos ser uma técnica com

potencial para ser aplicada a este problema, sobretudo porque oferecia uma forma simples de descrever outras relações existentes entre as várias partes de uma peça musical que não apenas a relação de semelhança. Ou seja, se a pesquisa fosse bem sucedida, o melhor programa deveria ser capaz de “mostrar” não só relações de semelhança entre os segmentos, mas também as transformações melódicas e rítmicas existentes na peça. Além desta razão devemos acrescentar ainda a reconhecida capacidade das técnicas evolucionárias, entre as quais se encontra a programação genética, em realizar, com sucesso, pesquisas em espaços de pesquisa complexos e de grandes dimensões.

Um dos aspectos positivos da abordagem que descrevemos parece-nos ser o facto de a representação por nós proposta para os programas permitir descrever de uma forma bastante simples de entender por um ser humano a estrutura de uma peça musical. Mas, não basta que os programas sejam fáceis de entender. É necessário que, utilizando essa representação, o sistema seja capaz de gerar programas que descrevam correctamente a estrutura da peça a analisar. E é neste aspecto fundamental que o sistema que descrevemos apresenta os maiores problemas:

- Apesar de se terem obtido resultados bastante interessantes na primeira segmentação de três das cinco peças utilizadas nas experiências, isso só foi possível quando se utilizaram populações muito grandes (3000). Como consequência, as experiências demoram bastante tempo a ser realizadas. Lembremos que, para a peça mais pequena, cada experiência demorou, em média, quase um minuto.
- A capacidade de pesquisa do sistema é fortemente afectada sempre que se aumenta o tamanho do conjunto de funções. Este problema impediu que se obtivessem resultados dignos de registo quando utilizámos funções que permitiam realizar transformações melódicas, rítmicas ou estruturais.
- Mesmo com populações grandes e com um conjunto de funções unitário, o sistema demonstrou dificuldades óbvias em descobrir programas que descrevessem a estrutura de duas das peças utilizadas. Estas dificuldades aumentaram quando se tentou realizar uma segunda segmentação às peças, tendo apenas sido obtidos resultados estatisticamente interessantes para a peça

«Be m'anperdut...».

São sobretudo duas as razões que julgamos poderem ser responsáveis pela incapacidade do sistema baseado em programação genética descrito neste capítulo em encontrar indivíduos que descrevam correctamente a estrutura de algumas peças musicais. A primeira razão já foi apresentada anteriormente e consiste no facto de, em certas situações, serem necessárias várias operações para transformar um indivíduo num outro com melhor desempenho, sem que as configurações intermédias impliquem uma melhoria no seu desempenho. Este problema ocorre sobretudo quando essa transformação implica uma modificação estrutural do indivíduo como seja, por exemplo, a divisão de um segmento em vários outros. Em termos do espaço de pesquisa podemos ver este problema como se o indivíduo a transformar fosse um mínimo local e que, para saltar para um ponto mais baixo do espaço, fosse necessário subir primeiro uma montanha. Uma forma de resolver este problema poderá passar, eventualmente, pela utilização de operadores genéticos que realizem, de uma só vez, as modificações necessárias que permitam melhorar o desempenho de alguns indivíduos.

A outra razão que julgamos poder ser responsável pelas dificuldades do sistema consiste no facto de a posição que os terminais ocupam na função principal ser determinante no desempenho do indivíduo. Ou seja, uma vez que cada indivíduo tem que gerar a peça a analisar, a ordem por que um segmento aparece na sequência de segmentos da função principal é muito importante. Isto significa que se a peça a analisar tiver, por exemplo, uma estrutura AXA e um indivíduo apresentar uma estrutura AAX ou XAA, o seu desempenho vai ser muito mau, mesmo que o conteúdo dos seus segmentos A e X esteja correcto. Este facto dificulta o melhoramento do desempenho dos indivíduos através da aplicação dos operadores genéticos, uma vez que, além de “inserir” o segmento certo no indivíduo certo, é também necessário inseri-lo na posição correcta. Tal como referimos anteriormente acerca do problema anterior, apesar de nos parecer que este problema possa ser um dos principais responsáveis pela dificuldade do sistema em gerar indivíduos melhores, julgamos que é necessário mais estudo para o confirmar.

Capítulo 6

Algoritmos Genéticos e Extração de Padrões Musicais

O desenvolvimento da abordagem ao problema da extração de padrões musicais que descrevemos neste capítulo aconteceu cronologicamente depois da abordagem que descrevemos no capítulo anterior. Podemos, por isso, dizer que algumas das ideias que deram origem à abordagem baseada em algoritmos genéticos foram surgindo à medida que nos fomos apercebendo dos problemas da abordagem baseada em programação genética. Nomeadamente, nesta parte do nosso trabalho procurámos evitar os dois problemas que referimos no final do capítulo anterior, a saber: a dificuldade em gerar indivíduos com melhor desempenho através da aplicação dos operadores genéticos, e a importância que a posição relativa dos segmentos tem no desempenho do indivíduo. Como se verá, um aspecto que distingue as duas abordagens é o facto de, na abordagem descrita neste capítulo, os indivíduos não serem programas.

Neste capítulo, começamos por descrever como é que abordámos o problema da extração de padrões musicais utilizando algoritmos genéticos, seguindo-se a apresentação dos resultados experimentais obtidos. Finalizamos o capítulo com algumas conclusões acerca dos resultados a que chegámos.

6.1 Descrição da Abordagem

6.1.1 Representação dos Indivíduos

Na abordagem que descrevemos neste capítulo, cada indivíduo representa uma segmentação da peça musical a analisar, ou seja, a sequência de segmentos em que a peça se divide. A representação que utilizámos para os indivíduos é igual, quer as peças sejam representadas como sequências de notas, quer sejam representadas como sequências de intervalos. A diferença, como veremos, reside no modo como os

indivíduos devem ser interpretados quando se utiliza um ou outro método de representação das peças. Assim, de modo a facilitar a compreensão da forma como devem ser interpretados os indivíduos, começaremos por ver quais as diferenças entre os segmentos resultantes da segmentação de uma peça representada como uma sequência de notas e os segmentos resultantes de uma segmentação equivalente da mesma peça representada como uma sequência de intervalos.

Consideremos a sequência de notas “ $nota_0 \quad nota_1 \quad \dots \quad nota_n$ ”. A sequência de intervalos correspondente será a sequência “ $intervalo_0 \quad intervalo_1, \dots, intervalo_{n-1}$ ”, em que $intervalo_i$ representa o intervalo formado pelas notas $nota_i$ e $nota_{i+1}$. Se dividirmos a mesma sequência de notas nos dois segmentos “ $nota_0 \quad \dots \quad nota_i$ ” e “ $nota_{i+1} \quad \dots \quad nota_n$ ” então os segmentos de intervalos equivalentes serão os segmentos “ $intervalo_0 \quad \dots \quad intervalo_{i-1}$ ” e “ $intervalo_{i+1} \quad \dots \quad intervalo_{n-1}$ ”. Utilizando uma notação semelhante à que foi utilizada para representar os indivíduos no capítulo anterior, os dois segmentos resultantes da divisão da sequência de notas são os segmentos $[0 \quad i]$ e $[i+1 \quad n]$, e os segmentos resultantes da divisão da sequência de intervalos são os segmentos $[0 \quad i-1]$ e $[i+1 \quad n-1]$. Ou seja, os limites dos segmentos resultantes da segmentação de uma peça representada como uma sequência de notas e os limites dos segmentos resultantes de uma segmentação equivalente da mesma peça representada como uma sequência de intervalos, são diferentes. Além disso, quando se utiliza uma representação baseada em notas, todas as notas da sequência são incluídas num dos segmentos, enquanto que, quando se utilizam intervalos, entre cada dois segmentos consecutivos há sempre um intervalo que não pertence a nenhum dos dois (por exemplo, o intervalo $intervalo_i$ não pertence nem ao segmento $[0 \quad i-1]$ nem ao segmento $[i+1 \quad n-1]$). Como se poderá comprovar já a seguir, estes dois aspectos são importantes para se perceber como é que o método de representação das peças determina a forma como devem ser interpretados os indivíduos.

As figuras 6.1 e 6.2 ilustram a forma como são representados e interpretados os indivíduos quando a peça a analisar é representada, respectivamente, como uma sequência de notas e como uma sequência de intervalos.

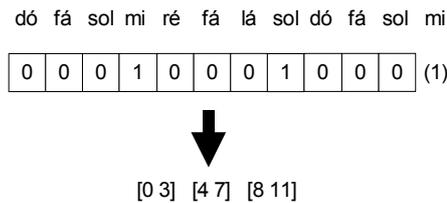


Figura 6.1 - Exemplo de uma peça representada como uma sequência de notas e de um indivíduo que representa uma sua segmentação.

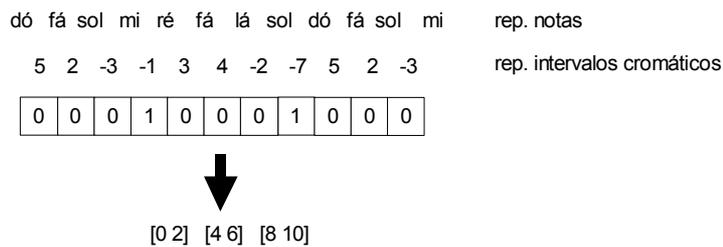


Figura 6.2 - Exemplo de uma peça representada como uma sequência de intervalos e de um indivíduo que representa uma sua segmentação.

Tal como ilustram as figuras, independentemente do modo como são representadas as peças, o genótipo de cada indivíduo é constituído por apenas um cromossoma que consiste numa sequência de uns e zeros. Todos os cromossomas têm o mesmo tamanho, sendo este igual ao número de notas da peça menos um. Por sua vez, o fenótipo não é mais do que uma representação do tipo “[$e_0 d_0$] ... [$e_n d_n$]” em que [$e_i d_i$] representa o i -ésimo segmento da peça, e e_i e d_i representam, respectivamente, o limite esquerdo e o limite direito desse segmento. Este tipo de representação, além de ser de leitura mais fácil, tem, sobre a representação binária, a vantagem de permitir aceder com mais facilidade aos limites de cada segmento. Assim, embora os operadores genéticos (ver secção 6.1.4) operem sobre a representação binária dos indivíduos, depois da aplicação destes operadores, todos os indivíduos são convertidos numa estrutura “[$e_0 d_0$] ... [$e_n d_n$]” de modo a facilitar, entre outras coisas, o processo de avaliação.

Quando a peça a analisar é representada como uma sequência de notas, o dígito que ocupa a posição n do cromossoma corresponde à nota da peça que ocupa a posição n . Neste caso, todos os segmentos, excepto o último, correspondem a sequências 0...01, em que o primeiro 0 representa a primeira nota do segmento e o 1 representa a última. Ao contrário de todos os outros, o último segmento não corresponde a uma sequência

terminada por um 1, uma vez que a última nota da peça não é explicitamente representada no cromossoma. Segundo esta interpretação, o indivíduo da Figura 6.1 corresponde à sequência de segmentos (de notas) “[0 3][4 7][8 11]”.

Se a peça a analisar for representada como uma sequência de intervalos, então o dígito que ocupa a posição n na sequência corresponderá ao intervalo que ocupa a posição n da peça. Neste caso, os segmentos correspondem simplesmente às sequências de 0s, servindo os 1s para representar os intervalos que, não pertencendo a nenhum segmento, servem para os separar. Segundo esta interpretação, o indivíduo da Figura 6.2 corresponde à sequência de segmentos (de intervalos) “[0 2][4 6][8 10]”.

De modo a facilitar a leitura, neste capítulo representaremos, sempre que possível, os indivíduos através do seu fenótipo e não através do genótipo.

6.1.2 Avaliação

Os indivíduos, quer sejam, ou não, representados por uma sequência binária, não contêm qualquer informação acerca dos padrões existentes na peça, permitindo apenas identificar em que segmentos é que esta está dividida. Posto de outro modo, os indivíduos não permitem identificar quais os segmentos que se repetem, exactamente ou não, ao longo da peça e quais os que não se repetem de todo. Como veremos já a seguir, esta informação é associada a cada indivíduo durante o processo de avaliação sob a forma de relações entre segmentos.

A avaliação do desempenho de um indivíduo é realizada comparando, entre si, todos os seus segmentos. Se a distância entre dois segmentos for menor que um determinado valor, diz-se que os segmentos são semelhantes e é criada uma relação entre os dois, caso contrário, não é criada nenhuma relação e os segmentos são considerados dissimilantes. Sempre que é criada uma relação, é-lhe associado o valor da distância entre os dois segmentos. O objectivo do sistema consiste em encontrar um indivíduo para o qual o número de notas dos segmentos que participam em relações seja o máximo possível, e para o qual a soma da diferença associada às relações seja o mínimo possível. Ou seja, o objectivo do sistema consiste em encontrar um indivíduo que maximize o valor da seguinte função de avaliação:

$$f(x) = a \times \sum_{i=1}^m d(rel_i) + b \times \sum_{i=1}^n area(s_i)$$

onde x representa o indivíduo a ser avaliado, a é uma constante negativa, b é uma constante positiva, m representa o número de relações criadas como consequência do processo de comparação de segmentos, $d(rel_i)$ representa a diferença associada à relação rel_i calculada com o algoritmo de Wagner-Fischer, n representa o número de segmentos do indivíduo, e

$$area(s_i) = \begin{cases} d_i - e_i + 1, & \text{se o segmento } s_i \text{ fizer parte de alguma relação,} \\ 0, & \text{em caso contrário} \end{cases}$$

representa o tamanho de cada segmento que participe numa relação. Note-se que o tamanho de um segmento só é contabilizado uma vez, mesmo que este participe em mais do que uma relação.

Quanto ao valor máximo da distância entre dois segmentos, abaixo do qual eles são considerados semelhantes, este foi definido empiricamente do seguinte modo: se o menor dos dois segmentos tiver um tamanho superior a 20, então a distância máxima é igual a 5; caso contrário, a distância máxima é estabelecida como sendo um quarto do tamanho do menor dos dois segmentos. Julgamos, no entanto, poder ser necessário mais estudo para escolher os valores mais apropriados para este valor.

Além do valor da distância entre dois segmentos, outro aspecto que pode impedir a criação de uma relação é o tamanho dos segmentos. Nesta abordagem é definido um valor mínimo para o tamanho dos segmentos - que denominaremos por *tam_min_seg* - de modo a que um segmento com um tamanho inferior a esse valor não se possa relacionar com nenhum outro segmento. Note-se, no entanto, que isso não significa que não possam existir segmentos com tamanho inferior a *tam_min_seg* (os valores que definem realmente as dimensões máximas e mínimas dos segmentos e o modo como é garantido que todos eles obedecem a estas dimensões são tratados na secção 6.1.4). A existência do parâmetro *tam_min_seg* permite que se defina, por exemplo, que na primeira segmentação de uma peça só se possam identificar os maiores padrões nela existentes, embora possam coexistir com esses padrões restos de pequenas dimensões. Suponhamos, por exemplo, que se pretende realizar uma primeira segmentação a uma

peça com uma estrutura AXA, em que X tem uma dimensão muito inferior à do segmento A. Neste caso, se não se permitir a existência de segmentos com a dimensão do segmento X, dificilmente se consegue obter um indivíduo que permita identificar a semelhança entre a primeira e a segunda ocorrência do segmento A. Por outro lado, se permitirmos que segmentos como o segmento X possam fazer parte de uma relação, pode acontecer que em vez de se identificar as duas ocorrências dos segmentos A, se divida este segmento em segmentos mais pequenos. Embora este último aspecto não constitua por si só uma grande desvantagem, o objectivo do utilizador pode não ser o de identificar desde logo os segmentos mais pequenos existentes na peça, cabendo ao parâmetro *tam_min_seg* permitir essa maleabilidade.

Como se pode ver pela definição da função de avaliação, os restos, que são os segmentos que não participam em nenhuma relação, não contribuem, de forma directa, nem positiva nem negativamente para o valor do desempenho dos indivíduos. Apesar disso, estes segmentos influenciam o desempenho dos indivíduos pois, quanto maior for a área por si ocupada, menor será a área ocupada pelos segmentos que fazem parte de relações e, conseqüentemente, pior será o desempenho. Por exemplo, na figura seguinte, pode-se dizer que o indivíduo 1 tem um melhor desempenho do que o indivíduo 2 porque tem menos área ocupada por restos³⁰.

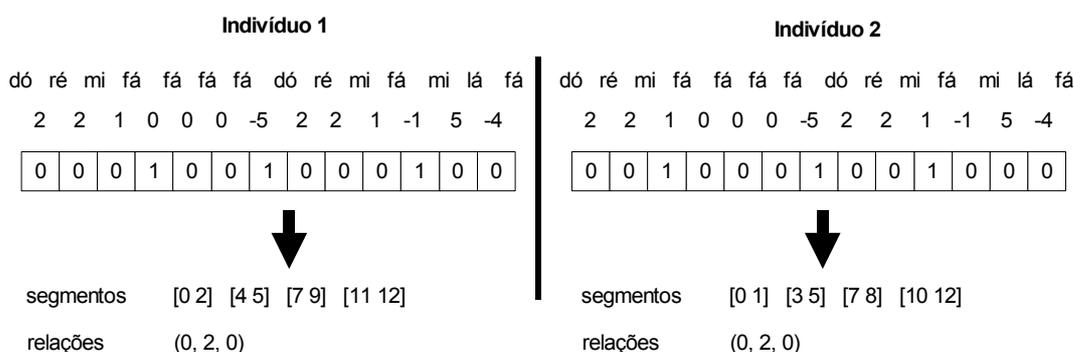


Figura 6.3 - Dois indivíduos que representam duas segmentações diferentes da uma mesma peça representada como uma sequência de intervalos.

Nesta figura, juntámos a cada indivíduo as relações criadas durante o processo de avaliação. Cada relação é representada por uma estrutura (*indice_segmento*,

³⁰ Que é o mesmo que dizer que o indivíduo 1 é melhor porque a área ocupada pelos segmentos que participam em relações é maior do que no indivíduo 2.

$índice_segmento_2$, $distância$), onde $índice_segmento_1$ e $índice_segmento_2$ representam os índices dos segmentos que fazem parte de relação (considera-se que o índice do primeiro segmento da peça é igual a 0), e $distância$ representa a distância entre os dois segmentos. Embora, do ponto de vista do algoritmo genético, as relações não façam parte dos indivíduos, é óbvio que a informação nelas contidas é essencial para se perceber o valor da segmentação que cada um representa. Assim, no resto do capítulo, faremos acompanhar os indivíduos das relações sempre que isso for possível.

Um aspecto que julgamos importante referir, ainda no âmbito da avaliação dos indivíduos, é o facto de, no algoritmo que desenvolvemos, os segmentos serem comparados apenas na sua forma original. Isso não significa, no entanto, que eles não possam ser também comparados, por exemplo, na sua forma retrogradada, invertida ou ambas, tal como acontece no AIPS descrito no capítulo 3.

6.1.3 Representação das Peças

Nesta parte do nosso trabalho utilizámos sobretudo peças representadas como sequências de intervalos, embora tenhamos realizado também experiências com representações baseadas em notas. Esta orientação deveu-se simplesmente ao facto de as representações baseadas em intervalos terem algumas vantagens sobre as representações baseadas em notas. Assim, cada peça é representada como uma sequência do tipo

“ $\langle intCr(nota_1, nota_2), rDur(nota_1, nota_2) \rangle, \dots, \langle intCr(nota_{n-1}, nota_n), rDur(nota_{n-1}, nota_n) \rangle$ ”,

onde $intCr(nota_i, nota_{i+1})$ e $rDur(nota_i, nota_{i+1})$ representam, respectivamente, o intervalo cromático formado por duas notas consecutivas e a razão da duração entre essas duas notas. As pausas não são representadas explicitamente. Em vez disso, o valor da duração de cada pausa é acrescentado ao da nota que a precede. Tal como na abordagem anterior, não é utilizada nenhuma informação de carácter global acerca da peça. Também como antes, a comparação de segmentos não tem necessariamente que ter em conta as dimensões melódica e rítmica em simultâneo, podendo incidir apenas sobre uma delas. Finalmente, note-se que, embora não tenhamos utilizado intervalos escalares, estes podem ser facilmente integrados nesta representação e utilizados conjuntamente com os intervalos cromáticos e com as razões das durações na

comparação de segmentos.

No resto deste capítulo os indivíduos deverão ser sempre interpretados em função da forma de representação das peças que descrevemos nesta secção.

6.1.4 Operadores Genéticos e Correção dos Indivíduos

No algoritmo genético que desenvolvemos são utilizados dois dos operadores genéticos clássicos já descritos no capítulo 4: a mutação e a recombinação de dois cortes. As duas figuras seguintes ilustram a aplicação destes dois operadores e como é que as modificações operadas ao nível do genótipo se repercutem ao nível do fenótipo.

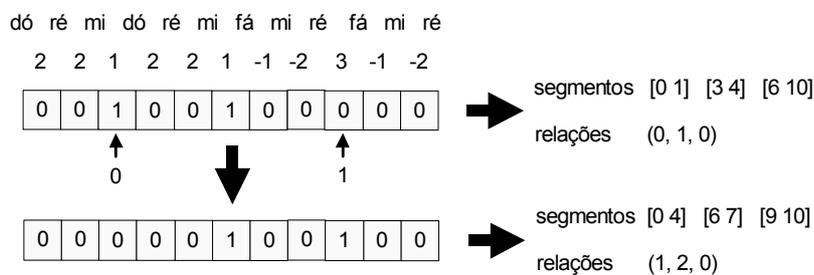


Figura 6.4 - Exemplo da aplicação do operador de mutação em dois pontos diferentes de um indivíduo.

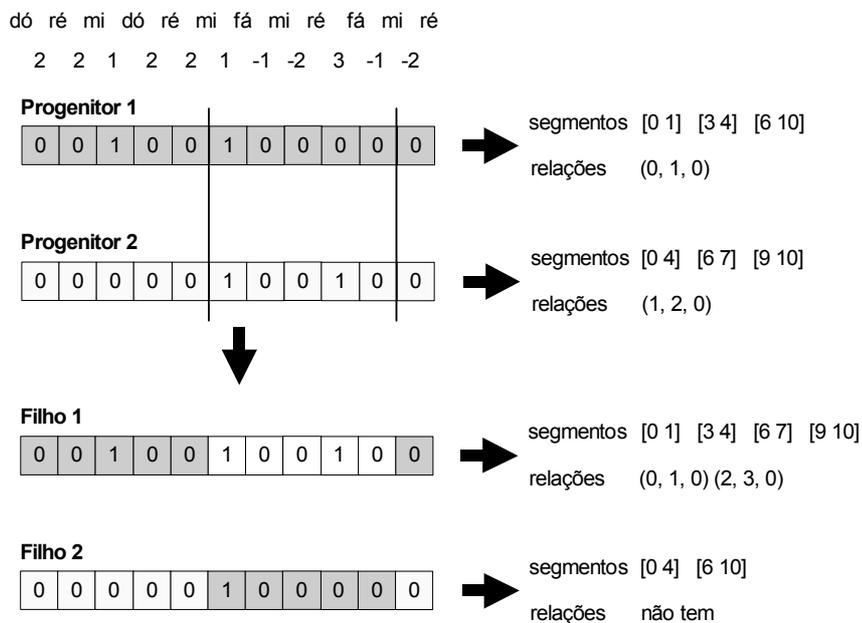


Figura 6.5 - Exemplo da aplicação do operador de recombinação de dois pontos em dois pontos diferentes de um indivíduo.

Além destes dois, foram ainda testados dois outros operadores genéticos concebidos por nós: um de mutação e outro de recombinação. O operador de mutação que testámos funciona do seguinte modo: cada gene com valor 1 tem uma determinada probabilidade de trocar o seu valor com um dos genes adjacentes, sendo igual a probabilidade de a troca ocorrer com o gene da esquerda ou com o gene da direita. Este operador foi concebido com o objectivo de permitir ao algoritmo genético ajustar os limites dos segmentos de modo a melhorar o desempenho dos indivíduos. Por sua vez, o operador de recombinação que testámos consiste num operador de recombinação de dois cortes com a seguinte característica: os pontos de corte têm necessariamente que coincidir com os limites dos segmentos, podendo a parte que se copia incluir mais do que um segmento. Este operador foi concebido com o objectivo de impedir que o material genético trocado entre os dois indivíduos possa ser constituído apenas por zeros. Apesar destes dois operadores terem sido desenvolvidos especificamente para este problema, os resultados obtidos nas experiências em que foram utilizados foram sempre inferiores aos resultados obtidos com os operadores clássicos. Assim, na secção 6.2 apenas os resultados obtidos com os operadores clássicos serão apresentados.

Da aplicação dos operadores genéticos podem resultar indivíduos com segmentos demasiado pequenos ou demasiado grandes. Por exemplo, no limite seria possível que da aplicação dos operadores resultassem indivíduos em que todos os genes tivessem valor 0, ou em que todos tivessem valor 1. De modo a corrigir situações deste tipo, todos os indivíduos são sujeitos a um processo de correcção depois da aplicação dos operadores genéticos. Este processo consiste simplesmente em fundir segmentos quando ocorre um segmento demasiado pequeno e em dividir ao meio segmentos que sejam demasiado grandes. Tal como na abordagem anterior, a dimensão mínima foi definida como sendo dois intervalos (equivalente a três notas) e a dimensão máxima foi definida como metade do tamanho da peça. A figura seguinte ilustra como funciona o processo de correcção.

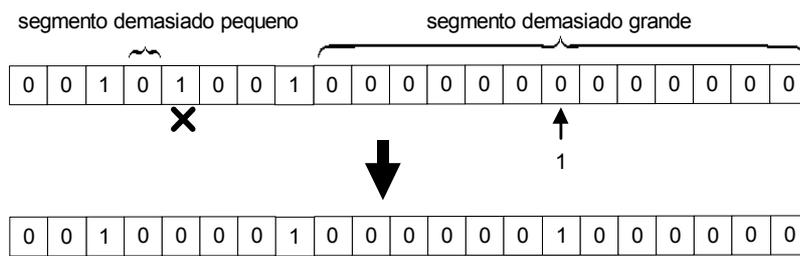


Figura 6.6 - Ilustração do processo de correcção de segmentos.

6.1.5 Operadores de Aprendizagem

Por vezes, quando se utilizam algoritmos evolucionários, além dos operadores genéticos utiliza-se também outro tipo de operadores que é comum denominar por operadores de aprendizagem. A utilização destes operadores inspira-se na facto de o desempenho dos organismos naturais não ser só determinado pelo seu material genético, podendo também ser melhorado ao longo da vida através de processos de aprendizagem. Uma diferença fundamental entre os operadores genéticos e os operadores de aprendizagem consiste em que os primeiros são sempre aplicados ao genótipo dos indivíduos, enquanto que os segundos são sempre aplicados ao seu fenótipo. Em sistemas artificiais, os operadores de aprendizagem são, geralmente, aplicados depois do processo de avaliação, apenas a um pequeno sub-conjunto de indivíduos da população escolhido aleatoriamente. Para cada indivíduo seleccionado, os operadores são aplicados de forma sequencial mas, para cada operador, o indivíduo só é realmente modificado se essa modificação implicar um melhoramento no seu desempenho ou se este não piorar. Os vários operadores podem ser aplicados durante um número fixo de iterações ou até que a sua aplicação deixe de implicar um melhoramento do desempenho do indivíduo. Para mais informação acerca da interacção entre evolução e aprendizagem em ambientes de computação evolucionária consultar [Hart, 1994], [Mayley, 1996] ou [Pereira, 2002].

Na abordagem que descrevemos neste capítulo são utilizados, durante o processo evolutivo, quatro operadores de aprendizagem que são aplicados a um pequeno conjunto de indivíduos da população escolhido aleatoriamente. Além disso, depois do processo evolutivo, o melhor indivíduo é também sujeito a estes operadores. Tal como mostra a Figura 6.7, os operadores são aplicados a cada indivíduo de forma

iterativa até que já não seja possível melhorar o seu desempenho. No final de cada ciclo, o indivíduo é corrigido de forma semelhante à que foi descrita na secção anterior só que, agora, ao nível do fenótipo. A correcção do indivíduo torna-se necessária uma vez que, da aplicação dos operadores, podem resultar segmentos com um tamanho não desejado. Este procedimento pode, no entanto, implicar uma redução no desempenho do indivíduo para níveis inferiores aos que este apresentava antes do processo de aprendizagem. Assim, as modificações operadas pelos operadores e pelo procedimento de correcção só adquirem carácter definitivo se no final do processo houver uma melhoria efectiva no desempenho do indivíduo. No final do todo o processo, se o desempenho do indivíduo for melhorado, o seu genótipo é actualizado com as modificações operadas sobre o seu fenótipo (em ambientes de computação evolucionária, quando as alterações realizadas sobre o fenótipo dos indivíduos durante o processo de aprendizagem se repercutem directamente no seu genótipo, diz-se que a estratégia de aprendizagem é Lamarckiana [Pereira, 2002]).

```

Aprendizagem(ind: indivíduo a submeter ao processo de
aprendizagem) .

fAntesAprendizagem = f(ind) .
indAntesAprendizagem = ind .
Fazer
    fAntesOperadores = f(ind) .
    ind = operador1(ind) .
    ind = operador2(ind) .
    ind = operador3(ind) .
    ind = operador4(ind) .
    ind = correção(ind) .
Enquanto f(ind) > fAntesOperadores

Se f(ind) > fAntesAprendizagem então actualizar genótipo de ind .
Senão ind = indAntesAprendizagem

Devolver ind .

```

Figura 6.7 - Descrição do processo de aprendizagem.

Iremos agora descrever os quatro operadores que são utilizados durante o processo de aprendizagem. O primeiro operador a ser aplicado, tal como o operador de mutação que descrevemos na secção anterior, foi concebido com o objectivo de permitir o ajustamento dos limites dos segmentos durante o processo evolutivo. Este operador actua do mesmo modo para todos os limites entre os segmentos do indivíduo: primeiro, tenta deslocar, para a esquerda, cada um dos limites entre os segmentos o

maior número de unidades que for possível, sendo que cada modificação só adquire carácter definitivo se não implicar uma diminuição do desempenho do indivíduo; caso não consiga deslocar um determinado limite para a esquerda, então tenta deslocá-lo para a direita o maior número de unidades que conseguir. A figura seguinte ilustra a deslocação, em uma unidade para a esquerda, do limite entre o primeiro e o segundo segmento de um indivíduo.

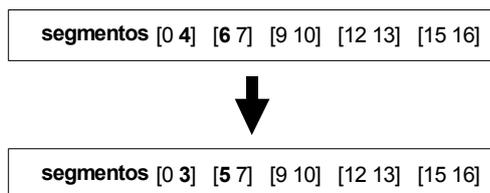
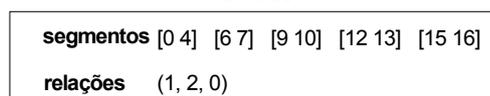


Figura 6.8 - Exemplo da aplicação do primeiro operador.

O segundo operador consiste simplesmente em juntar todos os restos consecutivos de cada um dos indivíduos seleccionados. Este operador, tal como o que é descrito a seguir, foi inicialmente concebido com o objectivo único de simplificar os indivíduos. No entanto, a sua aplicação pode também implicar uma melhoria imediata no desempenho dos indivíduos pois, é possível que sejam criadas novas relações nas quais participem os segmentos criados³¹. A figura seguinte ilustra esta situação.

dó ré mi fá mi dó lá si lá lá si lá dó ré mi fá mi dó
 2 2 1 -1 -4 9 2 -2 0 2 -2 -9 2 2 1 -1 -4

Indivíduo



(Os segmentos [12 13] e [15 16] são dois *restos* consecutivos)

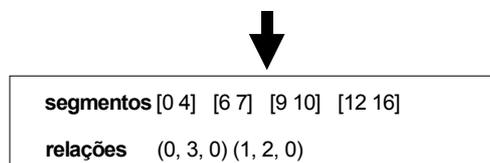


Figura 6.9 - Exemplo da aplicação do segundo operador.

O terceiro operador explica-se melhor através de um exemplo. Consideremos a peça da Figura 6.10, com uma estrutura de segmentos AXA. O indivíduo, também

³¹ Note-se que estas relações só são criadas durante o processo de avaliação que se segue à aplicação do operador.

representado na figura, apesar de permitir identificar todo o material recorrente existente na peça, apresenta uma estrutura que é claramente mais complexa do que o necessário (ABCXABC). O objectivo deste operador consiste, assim, em detectar situações deste tipo e em simplificar os indivíduos onde elas ocorram, caso isso não implique uma deterioração do desempenho do indivíduo.

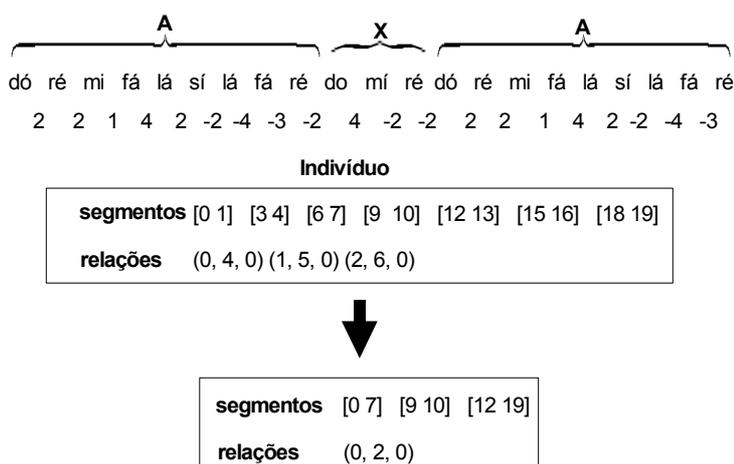


Figura 6.10 - Exemplo da aplicação do terceiro operador.

O quarto operador é parecido com o primeiro, só que actua simultaneamente sobre os limites de segmentos que façam parte de uma mesma relação. Consideremos, por exemplo, que os segmentos $[e_x d_x]$ e $[e_y d_y]$ fazem parte da mesma relação. O operador começa por tentar deslocar para a esquerda o limite esquerdo de ambos os segmentos, resultando nos novos segmentos $[e_x-1 d_x]$ e $[e_y-1 d_y]$ ³². Se esta modificação implicar uma melhoria no desempenho do indivíduo, então procede-se novamente do mesmo modo, e assim por diante, enquanto for possível melhorar o desempenho. Se este procedimento não for capaz de melhorar o desempenho do indivíduo, então tenta-se deslocar o limite esquerdo de ambos os segmentos para a direita enquanto for possível. Depois de tratados os limites esquerdos dos segmentos, repete-se o mesmo procedimento para os limites direitos.

Como se verá na secção 6.2, a aplicação destes operadores permite melhorar de forma substancial os resultados obtidos.

³² Estas modificações implicam que também sejam modificados os limites dos segmentos adjacentes aos segmentos que fazem parte da relação e sobre os quais o operador actua.

6.1.6 Segunda Segmentação

Tal como na abordagem baseada em programação genética, com esta abordagem também é possível realizar uma segunda segmentação de uma peça. Mais uma vez, para o fazer, podemos escolher, por exemplo, uma das três estratégias que descrevemos no capítulo anterior, a saber:

- Voltar a segmentar cada um dos segmentos identificados na primeira segmentação;
- Segmentar novamente a peça inteira, mas agora restringindo os segmentos de modo que os limites dos segmentos identificados na primeira segmentação não sejam violados pelos novos segmentos;
- Escolher apenas alguns segmentos identificados na primeira segmentação e aplicar o mesmo processo a uma “nova peça” constituída por estes segmentos.

As duas primeiras estratégias têm, como já vimos, algumas desvantagens pelo que, também nesta abordagem, foi escolhida a terceira. No entanto, a aplicação desta estratégia à abordagem que estamos a descrever reveste-se de algumas dificuldades ao nível da escolha dos segmentos. Assim, se na abordagem baseada em programação genética o processo de escolha dos segmentos é trivial, bastando escolher os segmentos que são utilizados na função principal, nesta abordagem a escolha pode não ser assim tão fácil. Por exemplo, entre três segmentos s_1 , s_2 e s_3 , se houver uma relação entre s_1 e s_2 , uma entre s_1 e s_3 e outra entre s_2 e s_3 , qual dos três segmentos se deve escolher? E se existir uma relação entre s_3 e um quarto segmento s_4 com o qual s_1 e s_2 não estão relacionados? Além disso, poderão existir na peça ainda outros segmentos que não se relacionem com nenhum destes quatro, podendo, no entanto relacionar-se entre si.

Embora a classificação de segmentos não faça parte dos objectivos desta dissertação, o problema de decidir quais os segmentos que devem ser escolhidos para compor a segunda “peça” a analisar pode ser visto como o de escolher os protótipos das classes de segmentos existentes na peça. Na Figura 6.11 é descrito o algoritmo de classificação que utilizámos na resolução deste problema. Este algoritmo é algo rudimentar, constituindo um dos aspectos a melhorar de futuro nesta abordagem. Uma das suas

características mais salientes consiste no facto de se considerar que a relação de semelhança é transitiva, o que não é verdade. Esta característica, no entanto, não se revelou problemática nas experiências que realizámos, pois em nenhuma delas foi criada qualquer classe a que pertencessem dois segmentos dissimelhantes.

```
Classificação (ind: indivíduo a partir do qual é realizada a
segunda segmentação).

Seja segs o conjunto de segmentos de ind.
Seja rels o conjunto de relações de ind.
Seja classes o conjunto de classes de segmentos de ind.
classes = {}.

Para todos os segmentos de segs que não participem em nenhuma
relação (restos) fazer

    Seja s o segmento que se está a tratar.

    Acrescentar a classe {s} a classes.
FimPara.

Para todas as relações de rels fazer

    Seja (s1, s2, dif) a relação que se está a tratar.

    Para todas as classes de classes fazer

        Seja classe a classe que se está a tratar.

        Se  $s_1 \in classe$  ou  $s_2 \in classe$  então
            Se  $s_1 \notin classe$  então acrescentar  $s_1$  a classe.
            Se  $s_2 \notin classe$  então acrescentar  $s_2$  a classe.
        FimSe.

    FimPara.

    Se nenhum dos segmentos da relação que se está a tratar
    foi acrescentado a alguma das classes já existentes então
    criar a classe {s1, s2} e acrescentá-la a classes.
FimPara.

Devolver classes
```

Figura 6.11 - Algoritmo de classificação utilizado na escolha dos segmentos a utilizar na segunda segmentação.

Depois da classificação dos segmentos é necessário decidir qual é o protótipo de cada uma das classes. Essa decisão é tomada da seguinte modo:

1. Se a classe só tem um segmento, então esse segmento é o protótipo da classe;
2. Se a classe tem apenas dois segmentos, o protótipo da classe é o segmento

mais pequeno;

3. Se a classe tem três ou mais segmentos, o protótipo é o segmento cuja soma das distâncias entre si e todos os outros segmentos da classe é menor.

A nova peça a analisar é criada justapondo os segmentos escolhidos por ordem de ocorrência na peça. Entre cada dois segmentos é inserido um intervalo especial de modo a que na segunda segmentação só sejam criados sub-segmentos dos segmentos identificados na primeira segmentação. Isto é conseguido atribuindo o valor 1 às posições do genótipo correspondentes aos intervalos especiais e impedindo que os operadores genéticos possam modificar o valor dessas posições. Além, disso a acção dos operadores de aprendizagem é restringida de modo a que os intervalos especiais nunca possam fazer parte de qualquer segmento.

6.2 Experimentação

Nesta secção apresentamos os resultados experimentais obtidos com esta abordagem para as peças apresentadas no Anexo I, também utilizadas para testar a abordagem que descrevemos no capítulo anterior.

6.2.1 Ambiente Experimental

Ao contrário do que aconteceu com a abordagem baseada em programação genética, o algoritmo genético utilizado nesta parte do nosso trabalho foi totalmente desenvolvido por nós. O algoritmo é praticamente igual ao que descrevemos no início do capítulo 4, sendo a principal diferença a existência de mais um passo, correspondente à aplicação dos operadores de aprendizagem. Além desta modificação, procurámos otimizar alguns aspectos do algoritmo com o objectivo de acelerar o processo evolutivo. Assim, uma vez que o processo de avaliação de um indivíduo é algo moroso, se houver na população vários indivíduos iguais, apenas um deles é avaliado, sendo o valor do desempenho dos outros simplesmente copiado daquele que é avaliado. Para isso, antes do processo de avaliação é necessário que todos os indivíduos da população sejam comparados. Esta comparação é realizada ao nível do fenótipo, de modo a tornar o processo mais rápido. Além desta optimização, outras foram postas em prática, a saber:

- O operador de recombinação não é aplicado sobre indivíduos que sejam iguais (para evitar que se comparem novamente os indivíduos, o resultado da comparação entre indivíduos realizada antes do processo de avaliação da iteração anterior é utilizado para este efeito);
- Se a diferença entre dois segmentos s_1 e s_2 for igual a zero, apenas um deles, seja s_1 , é comparado com os outros segmentos do indivíduo pelo que apenas para esse são eventualmente criadas relações (além da relação criada entre os dois). As relações do segmento s_2 com os outros segmentos não são criadas explicitamente embora na realidade essas relações existam. Note-se que, caso se pretenda, é trivial saber os segmentos com os quais s_2 se relaciona, bastando para tal verificar quais os segmentos com que s_1 se relaciona.
- O algoritmo de Wagner-Fischer foi alterado de modo a ser interrompido assim que a diferença máxima entre os dois segmentos a comparar seja ultrapassada.

À semelhança do que aconteceu com o sistema que utilizámos para realizar as experiências descritas no capítulo anterior, não foi desenvolvido nenhum interface gráfico que possibilite uma interacção mais agradável com o algoritmo. Este é um dos aspectos que deverá merecer futuramente a nossa atenção. Quanto ao processador utilizado para realizar as experiências, também para esta abordagem foi utilizado um processador AMD Athlon XP a 1.54 GHz com 256 MB de memória RAM.

6.2.2 Parâmetros do Processo Evolutivo

Na tabela seguinte são apresentados os valores dos parâmetros que foram utilizados nas experiências cujos resultados mostramos nas próximas secções. Para cada peça foram realizados dois grupos de 30 experiências independentes utilizando estes parâmetros. No primeiro grupo não foram utilizados operadores de aprendizagem, enquanto que no segundo grupo estes operadores foram utilizados. Assim, para o primeiro grupo deve ignorar-se o parâmetro que define o número de indivíduos a ser sujeito ao processo de aprendizagem.

Descrição do Parâmetro	Valor
Tamanho da população	100
Condição de paragem	100 gerações
Probabilidade de ocorrência de genes com o valor 1 nos indivíduos da população inicial	5%
Método de selecção	Torneio ($t = 5$)
Probabilidade de aplicação do operador de recombinação	70%
Probabilidade de aplicação do operador de mutação	1%
Número de indivíduos a ser sujeito ao processo de aprendizagem	1
Valor do parâmetro a da função de avaliação	-2
Valor do parâmetro b da função de avaliação	1
Valor do parâmetro tam_min_seg na primeira segmentação	10% do tamanho da peça
Valor do parâmetro tam_min_seg na segunda segmentação	2

Tabela 6.1 - Parâmetros do processo evolutivo.

Como se pode ver na tabela, a condição de paragem foi definida como um número máximo de gerações, tal como aconteceu na abordagem baseada em programação genética, uma vez que também nesta abordagem não era possível estabelecê-la com base no desempenho dos indivíduos. Por exemplo, mesmo que não exista qualquer diferença entre os segmentos para os quais se conseguiu estabelecer uma relação, não podemos determinar se seria possível aumentar as dimensões destes segmentos aumentando, assim, a área por eles ocupada. Tal como anteriormente, este facto impede-nos de apresentar os resultados obtidos baseando-nos apenas no desempenho dos indivíduos. Os resultados serão, assim, novamente apresentados comparando os padrões identificados pelos indivíduos obtidos nas experiências com os que são assinalados nas peças no Anexo I.

Nesta abordagem é necessário ter um cuidado especial com a probabilidade de ocorrência de cada um dos alelos nos cromossomas dos indivíduos da população inicial. O facto de a probabilidade de ocorrência do alelo 1 ser consideravelmente mais baixa do que a probabilidade de ocorrência do alelo 0 deve-se simplesmente ao facto de o número de limites entre segmentos ser, geralmente, bastante pequeno quando comparado com o tamanho das peças.

O valor do número de indivíduos sujeitos ao processo de aprendizagem foi definido depois de termos realizado várias experiências em que este valor foi variado de 1 a 5, tendo os melhores resultados sido obtidos quando apenas um indivíduo é sujeito a este processo. De modo a que seja possível ajuizar acerca da vantagem de utilizar, ou

não, operadores de aprendizagem durante o processo evolutivo, nas próximas secções compararemos as médias do desempenho dos indivíduos gerados nas experiências realizadas com e sem a utilização destes operadores.

Durante a fase de experimentação foi possível observar que se permitirmos a criação de relações entre segmentos muito pequenos logo na primeira segmentação, os melhores indivíduos gerados pelo algoritmo para algumas peças apresentam uma grande variedade de segmentos no que ao tamanho diz respeito. De modo a que na primeira segmentação sejam descobertos preferencialmente apenas os maiores padrões existentes nas peças, o valor do parâmetro *tam_min_seg* foi estabelecido em 10% do tamanho da peça a analisar, sendo reduzido para 2 só na segunda segmentação. A única exceção a esta regra foi a peça «Syrinx», pois nesta peça não existem padrões com mais de 10% do seu tamanho. Assim, para esta peça, tanto na primeira como na segunda segmentação, o valor de *tam_min_seg* foi definido como 2. O valor ideal para este parâmetro varia muito de peça para peça, pelo que não é fácil defini-lo de modo que, para todos os casos, sejam identificados apenas os maiores padrões. O valor definido para este parâmetro foi simplesmente consequência de um esforço da nossa parte em tentar uniformizar as experiências realizadas com as várias peças. Por exemplo, como se verá já a seguir, este valor não impede que para a peça «Be m'anperdut...» os melhores indivíduos apresentem divididos em dois os segmentos A e X identificados por Ruwet.

6.2.3 «Be m'anperdut...»

Na Tabela 6.2 apresentamos a média e o desvio padrão do desempenho dos melhores indivíduos gerados, com e sem a utilização de operadores de aprendizagem, na primeira segmentação da peça «Be m'anperdut...». Estes valores permitem-nos concluir que nestas experiências o desempenho do sistema foi significativamente superior (nível de significância: 0.05) quando se utilizaram operadores de aprendizagem. As experiências em que não foram utilizados operadores de aprendizagem demoraram, em média, 0.6 segundos, com um desvio padrão de 0.09 segundos. Por seu lado, as experiências em que estes operadores foram utilizados demoraram, em média, 0.86 segundos, com um desvio padrão de 0.12 segundos.

	Média	Desvio padrão
Sem operadores de aprendizagem	94.87	8.07
Com operadores de aprendizagem	100.83	0.91

Tabela 6.2 - Média e desvio padrão do desempenho dos melhores indivíduos obtidos na primeira segmentação da peça «Be m'anperdut...» quando se utilizam e quando não se utilizam operadores de aprendizagem.

Na Tabela 6.3 apresentamos os resultados obtidos para os melhores indivíduos gerados na primeira segmentação desta peça quando foram utilizados operadores de aprendizagem. De modo a não estender demasiado o texto, e sem prejuízo para o entendimento das capacidades do algoritmo, na exposição que se segue, para esta e para as outras peças, apenas analisamos os resultados obtidos quando os operadores de aprendizagem são utilizados. Esta opção deve-se ao facto de os resultados obtidos nas experiências em que se utilizaram estes operadores terem sido quase sempre superiores aos que foram obtidos nas experiências em que eles não foram utilizados, embora esta superioridade, quando existe, nem sempre tenha significado estatístico.

Desempenho	Distância acumulada entre segmentos semelhantes	Área ocupada	Número de indivíduos
101	0	101	29
96	0	96	1

Tabela 6.3 - Resultados obtidos para os melhores indivíduos gerados na primeira segmentação da peça «Be m'anperdut...» quando são utilizados operadores de aprendizagem.

Como se pode ver nesta tabela, só numa das experiências em que foram utilizados operadores de aprendizagem não foi gerado um indivíduo com um desempenho de 101. Este indivíduo tem a mesma estrutura dos outros indivíduos, sendo apenas ligeiramente diferente no que aos limites dos segmentos diz respeito. Todos os indivíduos com desempenho 101 são iguais ao que mostramos na figura seguinte. Nesta e nas próximas figuras do capítulo fazemos acompanhar cada segmento com o seu índice de modo a facilitar a interpretação dos indivíduos.

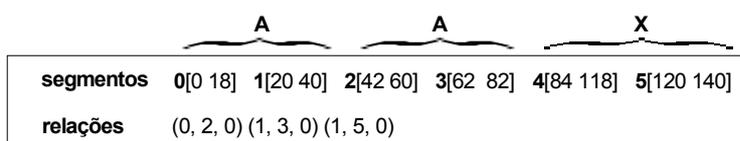


Figura 6.12 - Estrutura dos melhores indivíduos obtidos na primeira segmentação da peça «Be m'anperdut...».

Nesta figura é possível verificar que os indivíduos com desempenho 101 dividem em

dois os segmentos A e X identificados por Ruwet. Esta divisão permite que seja identificada a semelhança entre a parte final do segmento A e a parte final do segmento X. Tal como referimos antes, este é um dos casos em que o sistema, devido à sua capacidade de pesquisa, impede que sejam identificados apenas os maiores segmentos existentes na peça. Embora julgemos que este é um problema menor, para não dizer desejável, cabe-nos, ainda assim, propor formas de o resolver. Uma dessas formas é óbvia e consiste simplesmente em aumentar o valor do parâmetro *tam_min_seg*. Para o confirmar realizámos 30 experiências com *tam_min_seg* = 25. Todas estas experiências geraram indivíduos com uma estrutura AAX igual à que foi identificada por Ruwet. Outra forma de resolver este “problema” consiste em aplicar um processo de extracção de padrões, mas agora ao nível dos segmentos que foram identificados. Para isso, os segmentos da Figura 6.12, por exemplo, podem ser etiquetados da seguinte forma: [0 18] e [42 60] com a letra a; [20 40], [62 82] e [120 140] com a letra b; e o segmento [84 118] com a letra x. Ou seja, o processo de extracção de padrões irá actuar agora sobre a sequência de símbolos “ababxb”. Julgamos que não será exagerado dizer que, se não for permitida a semelhança entre segmentos de tamanho 1 nem a existência de erros entre segmentos relacionados, é trivial identificar nesta pequena sequência uma estrutura AAX em que A = “ab” e X = “xb”.

Na Tabela 6.4 apresentamos a média e o desvio padrão do desempenho dos melhores indivíduos gerados, com e sem a utilização de operadores de aprendizagem, na segunda segmentação da peça «Be m’anperdut...». Estes valores permitem-nos concluir que na segunda segmentação desta peça o desempenho do sistema foi significativamente superior (nível de significância: 0.05) quando se utilizaram operadores de aprendizagem. As experiências em que não foram utilizados operadores de aprendizagem demoraram, em média, 0.28 segundos, com um desvio padrão de 0.01 segundos. Por seu lado, as experiências em que estes operadores foram utilizados demoraram, em média, 0.35 segundos, com um desvio padrão de 0.02 segundos.

	Média	Desvio padrão
Sem operadores de aprendizagem	24.70	3.59
Com operadores de aprendizagem	27.73	2.61

Tabela 6.4 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na segunda segmentação da peça «Be m’anperdut...» quando se utilizam e quando não se utilizam operadores de aprendizagem.

Na Tabela 6.5 apresentamos os resultados obtidos para os melhores indivíduos gerados na segunda segmentação desta peça quando foram utilizados operadores de aprendizagem. Estas experiências foram realizadas utilizando uma sequência de intervalos baseada nos melhores indivíduos obtidos na primeira segmentação. Ou seja, a sequência a analisar é composta pelos segmentos [0 18], [20 40] e [84 118] por esta ordem.

Desempenho	Distância acumulada entre segmentos semelhantes	Área ocupada	Número de indivíduos
30	2/3/4	34/36/38	5/9/1
28	3/4	34/36	1/2
26	2/3/4	30/32/34	1/1/4
24	1/3/4	26/30/32	1/2/2
22	2	26	1

Tabela 6.5 - Resultados obtidos para os melhores indivíduos gerados na segunda segmentação da peça «Be m'anperdut...» quando são utilizados operadores de aprendizagem.

Nesta tabela, optámos por associar na mesma linha todos os indivíduos com desempenho igual, mesmo que tenham características diferentes. A primeira linha, por exemplo, deve ser entendida do seguinte modo: dos indivíduos com desempenho igual a 30, 5 são indivíduos com distância 2 e 34 de área ocupada, 9 são indivíduos com distância 3 e 36 de área ocupada e 1 indivíduo tem distância 4 e 38 de área ocupada. Este procedimento é adoptado para todas as peças sempre que houver indivíduos diferentes com o mesmo desempenho.

Na figura seguinte mostramos dois tipos de indivíduos com desempenho igual a 30. Os outros indivíduos com desempenho 30 (3 de distância e 36 de área ocupada) têm exactamente a mesma estrutura que os indivíduos com 2 de distância e 34 de área ocupada, sendo apenas ligeiramente diferentes no que aos limites dos segmentos diz respeito ([0 3], [5 13], [96 103] e [105 113] em vez de [0 4], [6 13], [96 104] e [106 113]).

Indivíduos com 2 de distância e 34 de área coberta

segmentos	0[0 4]	1[6 13]	2[15 18]	3[20 21]	4[23 25]	5[27 30]	6[32 34]	7[36 37]
	8[39 40]	9[84 87]	10[89 90]	11[92 94]	12[96 104]	13[106 113]	14[115 118]	
relações	(1, 13, 1) (3, 7, 0) (5, 9, 1) (6, 11, 0)							

Indivíduos com 4 de distância e 38 de área coberta

segmentos	0[0 3]	1[5 13]	2[15 18]	3[20 25]	4[27 30]	5[32 37]	6[39 40]
	7[84 87]	8[89 90]	9[92 97]	10[99 103]	11[105 113]	12[115 118]	
relações	(1, 11, 2) (4, 7, 1) (5, 9, 1)						

Figura 6.13 - Indivíduos com desempenho igual a 30 gerados na segunda segmentação da peça «Be m'anperdut...».

Os segmentos apresentados por estes indivíduos não correspondem exactamente aos segmentos identificados por Ruwet. Isto deve-se ao facto de a medida de distância que utilizámos no nosso trabalho (algoritmo de Wagner-Fischer conjuntamente com o valor a partir do qual se consideram semelhantes dois segmentos), não permitir identificar a semelhança entre os segmentos assinalados por Ruwet³³ (por exemplo entre a parte final do segmento A e o segmento A1.3). Isso não significa, no entanto, que o desempenho do algoritmo não seja satisfatório, dada a medida de distância com que trabalha. De facto, os segmentos apresentados pelos 15 indivíduos com desempenho igual a 30 são os que é possível identificar com a medida de distância utilizada. O facto de existirem diferentes indivíduos capaz de o fazer significa simplesmente que não existe uma única interpretação possível da estrutura da peça.

Quanto aos indivíduos com desempenho inferior a 30, embora haja alguns com uma estrutura diferente, estes consistem, regra geral, em versões mais pobres dos melhores indivíduos gerados.

6.2.4 «Maria muoter reinû maît»

Na Tabela 6.6 apresentamos a média e o desvio padrão do desempenho dos melhores indivíduos gerados, com e sem a utilização de operadores de aprendizagem, na primeira segmentação da peça «Maria muoter reinû maît». Estes valores permitem-nos

³³ Apesar de esta ser uma justificação válida não nos é possível afirmar, no entanto, que o algoritmo seria capaz de identificar os segmentos identificados por Ruwet caso utilizasse uma medida de distância que lhe permitisse fazê-lo.

concluir que nestas experiências o desempenho do sistema foi significativamente superior (nível de significância: 0.05) quando se utilizaram operadores de aprendizagem. As experiências em que não foram utilizados operadores de aprendizagem demoraram, em média, 0.37 segundos, com um desvio padrão de 0.04 segundos. Por seu lado, as experiências em que estes operadores foram utilizados demoraram, em média, 0.42 segundos, com um desvio padrão de 0.02 segundos.

	Média	Desvio padrão
Sem operadores de aprendizagem	87.1	8.80
Com operadores de aprendizagem	95	0

Tabela 6.6 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na primeira segmentação da peça «Maria muoter reinû maît» quando se utilizam e quando não se utilizam operadores de aprendizagem.

Na Tabela 6.7 apresentamos os resultados obtidos para os melhores indivíduos gerados na primeira segmentação desta peça quando foram utilizados operadores de aprendizagem. Os indivíduos gerados nestas experiências, como se pode comprovar na Figura 6.14, correspondem todos à primeira segmentação realizada por Ruwet.

Desempenho	Distância acumulada entre segmentos semelhantes	Área ocupada	Número de indivíduos
95	5	105	30

Tabela 6.7 - Resultados obtidos para os melhores indivíduos gerados na primeira segmentação da peça «Maria muoter reinû maît» quando são utilizados operadores de aprendizagem.

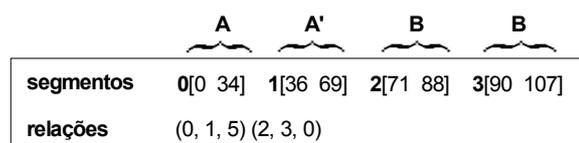


Figura 6.14 - Estrutura dos melhores indivíduos obtidos na primeira segmentação da peça «Maria muoter reinû maît».

Na Tabela 6.8 apresentamos a média e o desvio padrão do desempenho dos melhores indivíduos gerados, com e sem a utilização de operadores de aprendizagem, na segunda segmentação da peça «Maria muoter reinû maît». Neste caso, os valores obtidos não nos permitem concluir que na segunda segmentação desta peça o desempenho do sistema foi significativamente superior (nível de significância: 0.05) quando se utilizaram operadores de aprendizagem. As experiências em que não foram utilizados operadores de aprendizagem demoraram, em média, 0.13 segundos, com

um desvio padrão de 0.01 segundos. Por seu lado, as experiências em que estes operadores foram utilizados demoraram, em média, 0.17 segundos, com um desvio padrão de 0.01 segundos.

	Média	Desvio padrão
Sem operadores de aprendizagem	30.8	1.61
Com operadores de aprendizagem	31.3	1.21

Tabela 6.8 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na segunda segmentação da peça «Maria muoter reinû maît» quando se utilizam e quando não se utilizam operadores de aprendizagem.

Na Tabela 6.9 apresentamos os resultados obtidos para os melhores indivíduos gerados na segunda segmentação desta peça quando foram utilizados operadores de aprendizagem. Estas experiências foram realizadas utilizando uma sequência de intervalos baseada nos segmentos apresentados pelos indivíduos gerados na primeira segmentação. A sequência a analisar é, assim, composta pelos segmentos [36 69] e [71 88] por esta ordem.

Desempenho	Distância acumulada entre segmentos semelhantes	Área ocupada	Número de indivíduos
32	2	36	21
30	1/3/4	32/36/38	1/4/3
27	1	29	1

Tabela 6.9 - Resultados obtidos para os melhores indivíduos gerados na segunda segmentação da peça «Maria muoter reinû maît», quando são utilizados operadores de aprendizagem.

Os 21 indivíduos com desempenho igual a 32, cuja estrutura é apresentada na figura seguinte, apresentam praticamente os mesmos segmentos que são identificados por Ruwet. As únicas diferenças consistem no facto de os segmentos c e d aparecerem divididos em dois. Note-se que só deste modo é possível identificar as várias ocorrências do segmento d1.

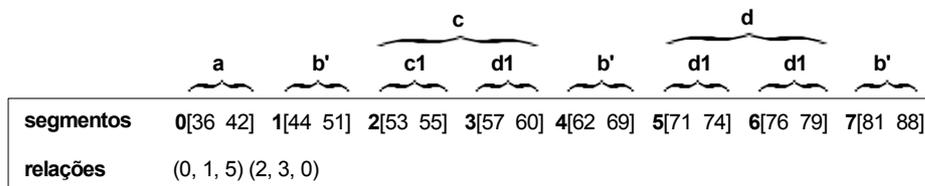


Figura 6.15 - Estrutura dos melhores indivíduos obtidos na segunda segmentação da peça «Maria muoter reinû maît».

Os indivíduos com desempenho inferior a 32 apresentam um conjunto de segmentos

consideravelmente diferente do que é identificado por Ruwet. Embora estes indivíduos constituam uma segmentação de pior qualidade, tal como o reflecte o seu desempenho, os segmentos que neles fazem parte de relações ocupam, mesmo assim, uma percentagem considerável da área da sequência a analisar. Esta situação só é possível porque existe realmente muito material recorrente na peça.

6.2.5 «Kalenda Maya»

Na Tabela 6.10 apresentamos a média e o desvio padrão do desempenho dos melhores indivíduos gerados, com e sem a utilização de operadores de aprendizagem, na primeira segmentação da peça «Kalenda Maya». Estes valores permitem-nos concluir que nestas experiências o desempenho do sistema foi significativamente superior (nível de significância: 0.05) quando se utilizaram operadores de aprendizagem. As experiências em que não foram utilizados operadores de aprendizagem demoraram, em média, 0.3 segundos, com um desvio padrão de 0.04 segundos. Por seu lado, as experiências em que estes operadores foram utilizados demoraram, em média, 0.37 segundos, com um desvio padrão de 0.01 segundos.

	Média	Desvio padrão
Sem operadores de aprendizagem	66.83	6.12
Com operadores de aprendizagem	70.67	4.79

Tabela 6.10 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na primeira segmentação da peça «Kalenda Maya» quando se utilizam e quando não se utilizam operadores de aprendizagem.

Na Tabela 6.11 apresentamos os resultados obtidos para os melhores indivíduos gerados na primeira segmentação desta peça quando foram utilizados operadores de aprendizagem.

Desempenho	Distância acumulada entre segmentos semelhantes	Área ocupada	Número de indivíduos
74	2/3/4	78/80/82	6/4/10
64	3	70	10

Tabela 6.11 - Resultados obtidos para os melhores indivíduos gerados na primeira segmentação da peça «Kalenda Maya» quando são utilizados operadores de aprendizagem.

Na figura seguinte apresentamos a estrutura dos indivíduos com desempenho igual a 74 e com 2 de diferença e 78 de área ocupada. Os outros indivíduos cujo desempenho é igual a 74 têm a mesma estrutura mas os limites dos últimos 4 segmentos são

ligeiramente diferentes.

	A	A	B	B'				
segmentos	0[0 20]	1[22 42]	2[44 52]	3[54 62]	4[64 72]	5[74 77]	6[79 87]	7[89 94]
relações	(0, 1, 0)	((2, 3, 2)	(4, 6, 0)					

Figura 6.16 - Estrutura dos melhores indivíduos obtidos na primeira segmentação da peça «Kalenda Maya».

Como é possível confirmar nesta figura, os quatro primeiros segmentos destes indivíduos são exactamente iguais aos quatro primeiros segmentos identificados por Ruwet. Quanto aos outros segmentos, uma vez que a distância entre os segmentos D e D' é demasiado grande (6), o algoritmo divide-os em dois de modo a poder apresentar dois outros segmentos (4 e 6 na figura) exactamente iguais. Estes segmentos são constituídos por duas ocorrências do segmento c mais o início comum dos segmentos D e D' e são, depois das duas ocorrências do segmento A, os dois maiores segmentos idênticos existentes na peça. Note-se que este comportamento do algoritmo é totalmente coerente, com o próprio procedimento de Ruwet descrito no capítulo dois, ou seja, primeiro deve identificar-se os maiores segmentos existentes na peça que sejam exactamente iguais.

Na Tabela 6.12 apresentamos a média e o desvio padrão do desempenho dos melhores indivíduos gerados, com e sem a utilização de operadores de aprendizagem, na segunda segmentação da peça «Kalenda Maya». Tal como para a segunda segmentação da peça anterior, os valores obtidos não nos permitem concluir que na segunda segmentação desta peça o desempenho do sistema foi significativamente superior (nível de significância: 0.05) quando se utilizaram operadores de aprendizagem. As experiências em que não foram utilizados operadores de aprendizagem demoraram, em média, 0.16 segundos, com um desvio padrão de 0.01 segundos. Por seu lado, as experiências em que estes operadores foram utilizados demoraram, em média, 0.18 segundos, com um desvio padrão de 0.03 segundos.

	Média	Desvio padrão
Sem operadores de aprendizagem	14.77	1.22
Com operadores de aprendizagem	14.80	1.63

Tabela 6.12 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na segunda segmentação da peça «Kalenda Maya» quando se utilizam e quando não se utilizam operadores de aprendizagem.

Na Tabela 6.13 apresentamos os resultados obtidos para os melhores indivíduos gerados na segunda segmentação desta peça quando foram utilizados operadores de aprendizagem. A segunda segmentação desta peça foi realizada tendo como base os indivíduos cuja estrutura é apresentada na Figura 6.16. A sequência a analisar é, assim, composta pelos segmentos [0 20], [44 52], [64 73], [75 77] e [90 94] por esta ordem.

Desempenho	Distância acumulada entre segmentos semelhantes	Área ocupada	Número de indivíduos
16	0	16	18
14	1	16	6
12	0/1	12/14	2/4

Tabela 6.13 - Resultados obtidos para os melhores indivíduos gerados na segunda segmentação da peça «Kalenda Maya» quando são utilizados operadores de aprendizagem.

A figura seguinte ilustra a estrutura dos indivíduos com desempenho igual a 16. Estes indivíduos, além das duas ocorrências do segmento c, permitem identificar a semelhança entre o início dos segmentos A e B/B' (segmentos 0 e 5 na figura), e a semelhança entre dois sub-segmentos do segmento A (segmentos 2 e 3 na figura).

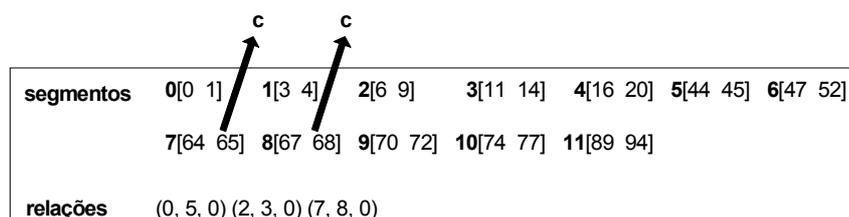


Figura 6.17 - Estrutura dos melhores indivíduos obtidos na segunda segmentação da peça «Kalenda Maya».

Quanto aos indivíduos com um desempenho inferior a 16, estes têm uma estrutura parecida à estrutura dos melhores indivíduos, mas os segmentos que apresentam são ligeiramente diferentes.

6.2.6 Variação 4 das «Seis variaciones sobre um tema de Milán»

Na Tabela 6.14 apresentamos a média e o desvio padrão do desempenho dos melhores indivíduos gerados, com e sem a utilização de operadores de aprendizagem, na primeira segmentação da Variação 4. Estes valores permitem-nos concluir que nestas experiências o desempenho do sistema foi significativamente superior (nível de significância: 0.05) quando se utilizaram operadores de aprendizagem. As experiências em que não foram utilizados operadores de aprendizagem demoraram, em média, 0.68

segundos, com um desvio padrão de 0.07 segundos. Por seu lado, as experiências em que estes operadores foram utilizados demoraram, em média, 0.98 segundos, com um desvio padrão de 0.04 segundos.

	Média	Desvio padrão
Sem operadores de aprendizagem	125.1	18
Com operadores de aprendizagem	138	0

Tabela 6.14 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na primeira segmentação da Variação 4 quando se utilizam e quando não se utilizam operadores de aprendizagem.

Na Tabela 6.15 apresentamos os resultados obtidos para os melhores indivíduos gerados na primeira segmentação desta peça quando foram utilizados operadores de aprendizagem. Todos os indivíduos gerados nestas experiências, como se pode comprovar na Figura 6.18, têm uma estrutura AXA correspondente à estrutura de segmentos de nível I que identificámos utilizando o procedimento de Ruwet..

Desempenho	Distância acumulada entre segmentos semelhantes	Área ocupada	Número de indivíduos
138	0	138	30

Tabela 6.15 - Resultados obtidos para os melhores indivíduos gerados na primeira segmentação da Variação 4 quando são utilizados operadores de aprendizagem.

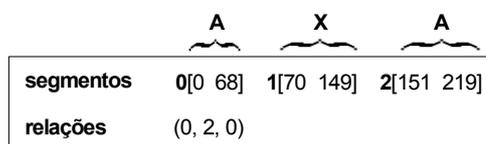


Figura 6.18 - Estrutura dos melhores indivíduos obtidos na primeira segmentação da Variação 4.

Na Tabela 6.16 apresentamos a média e o desvio padrão do desempenho dos melhores indivíduos gerados, com e sem a utilização de operadores de aprendizagem, na segunda segmentação da Variação 4. Estes valores permitem-nos concluir que na segunda segmentação desta peça o desempenho do sistema foi significativamente superior (nível de significância: 0.05) quando se utilizaram operadores de aprendizagem. As experiências em que não foram utilizados operadores de aprendizagem demoraram, em média, 0.53 segundos, com um desvio padrão de 0.02 segundos. Por seu lado, as experiências em que estes operadores foram utilizados demoraram, em média, 0.99 segundos, com um desvio padrão de 0.07 segundos.

	Média	Desvio padrão
Sem operadores de aprendizagem	103.90	5.72
Com operadores de aprendizagem	108.97	5.56

Tabela 6.16 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na segunda segmentação da Variação 4 quando se utilizam e quando não se utilizam operadores de aprendizagem.

Na Tabela 6.17 apresentamos os resultados obtidos para os melhores indivíduos gerados na segunda segmentação desta peça quando foram utilizados operadores de aprendizagem. Estas experiências foram realizadas utilizando uma sequência de intervalos baseada nos segmentos apresentados pelos indivíduos gerados na primeira segmentação. A sequência a analisar é, assim, composta pelos segmentos [0 68] e [70 149] por esta ordem.

Desempenho	Distância acumulada entre segmentos semelhantes	Área ocupada	Número de indivíduos
115	1	117	7
114	1	116	1
113	1	115	1
112	2	116	1
111	1	113	5
110	2	114	2
109	1	111	2
108	4	116	1
107	4	115	1
106	4	114	1
105	5	115	2
104	3	110	1
102	3	108	1
101	4/5	109/111	1/1
99	6	111	1
94	8	110	1

Tabela 6.17 - Resultados obtidos para os melhores indivíduos gerados na segunda segmentação da Variação 4 quando são utilizados operadores de aprendizagem.

As experiências realizadas no âmbito da segunda segmentação desta peça deram origem a uma grande variedade de indivíduos. Na Figura 6.19 mostramos a estrutura dos melhores indivíduos obtidos nestas experiências, ou seja, a estrutura dos indivíduos com desempenho igual a 115. Estes indivíduos apresentam um conjunto de segmentos de nível II praticamente igual ao que foi por nós identificado, existindo

apenas duas diferenças. A primeira diferença reside no facto de, nestes indivíduos, o segmento c' ter mais um intervalo (o intervalo entre a penúltima e a última nota do segmento A). Isto acontece simplesmente porque não é permitida a existência de segmentos com apenas um intervalo (se o segmento 9 da figura fosse exactamente igual ao segmento c', teria que existir o segmento [69 69] uma vez que o intervalo 70 não pode fazer parte de nenhum segmento devido às restrições impostas da primeira para a segunda segmentação). A outra diferença consiste em as duas ocorrências do segmento b serem divididas em dois segmentos cada uma. Esta é uma situação semelhante à que acontece na primeira segmentação da peça «Be m'anperdut...». Ou seja, esta divisão permite que seja identificada a semelhança entre a primeira parte do segmento b com o segmento b' (a semelhança entre os dois últimos intervalos das duas ocorrências do segmento b é identificada através da relação entre os segmentos 3 e 8).

	a	a	b		c	d	
segmentos	0[0 7]	1[9 16]	2[18 25]	3[27 28]	4[30 34]	5[36 40]	6[42 49]
	b		c + 1	e	e	b'	
	7[51 58]	8[60 61]	9[63 68]	10[70 86]	11[88 104]	12[106 107]	13[109 116]
	f		a	f	d		
	14[118 119]	15[121 122]	16[124 131]	17[133 134]	18[136 137]	19[139 146]	20[148 149]
relações	(0, 1, 0) (0, 16, 0) (2, 7, 0) (2, 13, 0) (3, 8, 0) (5, 9, 1) (6, 19, 0) (10, 11, 0) (15, 17, 0)						

Figura 6.19 - Estrutura dos melhores indivíduos obtidos na segunda segmentação da Variação 4.

Quanto aos outros indivíduos gerados nestas experiências, os indivíduos com desempenho igual a 114, 113 e 112 têm uma estrutura igual à estrutura dos melhores indivíduos, devendo-se a diferença de desempenho a ligeiras diferenças nos limites dos segmentos. A diferença entre os indivíduos com desempenho igual a 111 e os melhores indivíduos reside no facto de as duas ocorrências do segmento b não serem divididas em dois segmentos cada uma. Além disso, o segmento b' é dividido em três segmentos de dois intervalos cada um de modo a que seja identificada a semelhança entre o segmento f e o segmento formado pelos dois intervalos centrais do segmento b'. Os restantes indivíduos, como se pode avaliar pelos valores da coluna “Área ocupada” da Tabela 6.17, apesar de não terem uma estrutura que permita identificar

todo o material recorrente existente na peça, permitem, ainda assim, que grande parte deste material seja identificado.

6.2.7 «Syrinx»

Na Tabela 6.18 apresentamos a média e o desvio padrão do desempenho dos melhores indivíduos gerados, com e sem a utilização de operadores de aprendizagem, na primeira segmentação da peça «Syrinx». Mais uma vez, os valores obtidos não nos permitem concluir que na primeira segmentação desta peça o desempenho do sistema foi significativamente superior (nível de significância: 0.05) quando se utilizaram operadores de aprendizagem. As experiências em que não foram utilizados operadores de aprendizagem demoraram, em média, 1.39 segundos, com um desvio padrão de 0.07 segundos. Por seu lado, as experiências em que estes operadores foram utilizados demoraram, em média, 3.20 segundos, com um desvio padrão de 0.29 segundos.

	Média	Desvio padrão
Sem operadores de aprendizagem	108.57	5.88
Com operadores de aprendizagem	108.90	6.16

Tabela 6.18 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na primeira segmentação da peça «Syrinx» quando se utilizam e quando não se utilizam operadores de aprendizagem.

Na Tabela 6.19 apresentamos os resultados obtidos para os melhores indivíduos gerados na primeira segmentação desta peça quando foram utilizados operadores de aprendizagem. Tal como aconteceu com a segunda segmentação da Variação 4, as experiências realizadas no âmbito da primeira segmentação desta peça deram origem a uma grande variedade de indivíduos. Na Figura 6.20 apresentamos a estrutura do melhor indivíduo gerado nestas experiências, o qual permite identificar praticamente todo o material recorrente que assinalámos na pauta do Anexo I.

primeiro, e à semelhança de situações idênticas já encontradas noutras peças, consiste no facto de o segmento A estar dividido em dois segmentos. Neste caso, isso permite que seja identificada a semelhança entre o segmento composto pelos dois primeiros intervalos do segmento A (segmento 0 na figura) e os segmentos compostos respectivamente pelos intervalos 60 a 61, e 63 a 64 (segmentos 10 e 11 na figura). Segundo, devido à divisão em dois do segmento A, não é identificada a relação entre este segmento e as duas ocorrências do segmento A^{II}. Embora este facto não afecte o desempenho deste indivíduo, uma vez que é identificada a semelhança entre as duas ocorrências do segmento A^{II} (ou seja, a área de ambos os segmentos é contabilizada na função de avaliação), isso não significa, porém, que a existência de uma relação que permitisse identificar a semelhança entre o segmento A e o segmento A^{II} não enriquecesse a descrição da estrutura da peça. A criação desta relação pode ser conseguida se for realizado, sobre esta segmentação, um novo processo de extracção de padrões como o que descrevemos aquando da apresentação dos resultados obtidos com a peça «Be m'anperdut...». Por último, resta-nos referir que o facto de não ser identificada a semelhança entre as ocorrências do segmento A^{II} (ou A, se este segmento não aparecesse neste indivíduo dividido em dois) e os segmentos A^{III} e A^{IV} é perfeitamente normal, uma vez que estes dois segmentos têm uma estrutura rítmica completamente diferente da estrutura rítmica do segmento A^{II} (e A). A identificação destas relações só seria possível se o processo de extracção de padrões fosse realizado sem ter em conta a componente temporal das notas. Uma alternativa a este procedimento consistiria simplesmente em voltar a comparar os segmentos apresentados pelo indivíduo sem ter em conta esta componente, criando novas relações sempre que se confirmasse a semelhança entre dois segmentos.

Sobre a primeira segmentação desta peça, resta-nos ainda referir que, apesar de apenas ter sido gerado um indivíduo como o que descrevemos, isso não significa que os resultados obtidos nestas experiências não sejam satisfatórios. De facto, depois de analisados os outros indivíduos verifica-se que estes apresentam de, um modo geral, um conjunto de segmentos substancialmente parecido com o do indivíduo com desempenho igual a 128, sobretudo no que aos segmentos de maiores dimensões diz respeito. Por exemplo, as várias ocorrências do segmento A (o que inclui os segmentos A^I, A^{II}, A^{III} e A^{IV}) aparecem numa percentagem considerável dos

indivíduos obtidos. Por seu lado, as ocorrências dos segmentos D e H aparecem em praticamente todos os indivíduos. Uma outra forma de aferirmos o valor dos resultados obtidos nestas experiências consiste em comparar a área dos segmentos recorrentes existentes nos indivíduos gerados com a área dos segmentos identificados em [Smaill, 1993] (ver capítulo 3). Neste trabalho, se não se considerarem os segmentos que na pauta identificamos como F^I (a semelhança entre estes segmentos e o segmento F não devia ser identificada utilizando as transformações descritas pelos autores), a área ocupada pelos segmentos identificados é de 100 intervalos³⁴ (não considerando as notas de embelezamento ou apogiaturas) e a distância acumulada entre os segmentos é igual a 0. No resultados por nós obtidos a média da área ocupada é de 120.03 intervalos com um desvio padrão de 5.97, e a média da distância acumulada entre segmentos é de 5.57 com um desvio padrão de 1.98. Note-se que na nossa abordagem a existência de alguma distância acumulada é inevitável se se pretender considerar como semelhantes segmentos que não sejam exactamente iguais (tome-se como exemplo os segmentos A^{III} e A^{IV}).

Na Tabela 6.20 apresentamos a média e o desvio padrão do desempenho dos melhores indivíduos gerados, com e sem a utilização de operadores de aprendizagem, na segunda segmentação da peça «Syrinx». A segunda segmentação desta peça constitui o único caso em que a média do desempenho dos melhores indivíduos quando se utilizam operadores de aprendizagem é inferior à média obtida quando não se utilizam os operadores. No entanto, os valores obtidos não nos permitem concluir que o desempenho do sistema para este problema em particular é significativamente superior (nível de significância: 0.05) quando não são utilizados operadores de aprendizagem. As experiências em que não foram utilizados operadores de aprendizagem demoraram, em média, 0.58 segundos, com um desvio padrão de 0.03 segundos. Por seu lado, as experiências em que estes operadores foram utilizados demoraram, em média, 0.71 segundos, com um desvio padrão de 0.03 segundos.

³⁴ Se forem considerados os segmentos que na pauta assinalamos como F^2 , a área ocupada é de 104 intervalos.

	Média	Desvio padrão
Sem operadores de aprendizagem	25.37	2.31
Com operadores de aprendizagem	25.07	2.69

Tabela 6.20 – Média e desvio padrão do desempenho dos melhores indivíduos obtidos na segunda segmentação da peça «Syrinx» quando se utilizam e quando não se utilizam operadores de aprendizagem.

Na Tabela 6.21 apresentamos os resultados obtidos para os melhores indivíduos gerados na segunda segmentação desta peça quando foram utilizados operadores de aprendizagem. Estas experiências foram realizadas utilizando uma sequência de intervalos baseada nos segmentos apresentados pelo melhor indivíduo gerado na primeira segmentação. A sequência a analisar é, assim, composta pelos segmentos [0 1], [3 8], [10 12], [24 39], [41 42], [44 48], [66 73], [75 76], [90 92], [102 103], [105 106], [114 115], [117 127], [141 142], [144 145], [150 156], [158 159], [164 172], [177 183], [199 207], [209 217] e [229 240] por esta ordem.

Desempenho	Distância acumulada entre segmentos semelhantes	Área ocupada	Número de indivíduos
27	0	27	18
24	0	24	5
23	0	23	2
20	0	20	5

Tabela 6.21 - Resultados obtidos para os melhores indivíduos gerados na segunda segmentação da peça «Syrinx» quando são utilizados operadores de aprendizagem.

Na Figura 6.21 mostramos a estrutura dos indivíduos com desempenho igual a 27. Como se pode ver nesta figura, estes indivíduos permitem identificar a semelhança entre dois sub-segmentos do segmento A (segmentos 0 e 1 na figura), entre dois sub-segmentos do segmento D e entre dois sub-segmentos do segmento A^{III}. Além disso, é possível identificar também a semelhança entre partes do segmento A e partes dos segmentos A^{II} e A^{III}. Finalmente, é possível identificar a semelhança entre os segmentos que na pauta assinalámos com uma elipse (segmentos 5 e 18 na figura).

segmentos	0[0 1]	1[3 4]	2[6 8]	3[10 12]	4[24 26]	5[28 29]	6[31 39]
	7[41 42]	8[44 48]	9[66 67]	10[69 70]	11[72 73]	12[75 76]	13[90 92]
	14[102 103]	15[105 106]	16[114 115]	17[117 118]	18[120 121]	19[123 127]	20[141 142]
	21[144 145]	22[150 156]	23[158 159]	24[164 172]	25[177 183]	26[199 200]	27[202 203]
	28[205 207]	29[209 210]	30[212 213]	31[215 217]	32[229 240]		
relações	(0, 1, 0) (0, 27, 0) (2, 28, 0) (2, 31, 0) (5, 18, 0) (9, 10, 0) (29, 30, 0)						

Figura 6.21 - Estrutura dos melhores indivíduos obtidos na segunda segmentação da peça «Syrinx».

Quanto aos outros indivíduos, os indivíduos com desempenho 24 juntam os segmentos 1 e 2 da figura pelo que não permitem identificar a semelhança entre os sub-segmentos do segmento A. Por seu lado, os indivíduos com desempenho 23 não permitem identificar os segmentos que assinalámos com elipses. Finalmente, os indivíduos com desempenho 20 combinam as características dos indivíduos com desempenho 23 e 24.

6.3 Conclusão

Neste capítulo descrevemos a segunda das duas abordagens ao problema da extracção de padrões baseadas em algoritmos evolucionários que apresentamos nesta dissertação. De modo a melhor podermos avaliar de uma forma global o desempenho do algoritmo nas experiências realizadas, apresentamos agora uma breve súmula dos resultados obtidos. Antes disso, um aspecto que julgamos valer a pena voltar a referir é o facto de a utilização de operadores de aprendizagem permitir, de um modo geral, a obtenção de melhores resultados, embora nem sempre se possa atribuir significado estatístico à diferença entre os resultados obtidos quando se utilizam e quando não se utilizam estes operadores. Quanto aos resultados propriamente ditos, nas experiências realizadas com as três canções de trovadores utilizadas por Ruwet para demonstrar o seu procedimento de análise, o sistema foi capaz de gerar, com uma frequência considerável, indivíduos que representam segmentações muito semelhantes, quando não exactamente iguais, às que aquele autor realizou. Quanto à Variação 4 das «Seis variaciones sobre um tema de Milán», consideramos que os resultados são igualmente positivos: na primeira segmentação todos os indivíduos gerados apresentam uma estrutura AXA, e na segunda podemos considerar que pelo menos 15 dos 30

indivíduos gerados (indivíduos com desempenho 111 a 115) apresentam um conjunto de segmentos bastante semelhante ao que foi por nós identificado utilizando o procedimento de Ruwet. Finalmente, apesar da grande variedade de indivíduos gerados na primeira segmentação da peça «Syrinx», julgamos que o desempenho do sistema nas experiências realizadas com esta peça é igualmente positivo se tivermos em conta a considerável quantidade de material recorrente que é, em média, identificada pelos indivíduos gerados e o reduzido valor da média da distância acumulada entre segmentos. Estes resultados, julgamos, permitem que possamos considerar esta abordagem como uma possibilidade a considerar na resolução deste tipo de problemas.

Capítulo 7

Conclusões e Trabalho Futuro

7.1 Conclusões

O objectivo desta dissertação consistia em estudar a viabilidade da aplicação de técnicas evolucionárias ao problema da extracção de padrões musicais. Para realizar esse estudo desenvolvemos duas abordagens: uma baseada em programação genética e outra em algoritmos genéticos. Para cada uma das abordagens foi realizado um conjunto de testes experimentais de modo a podermos tirar algumas conclusões acerca da sua viabilidade. Uma vez que os resultados a que chegámos foram diferentes, iremos dividir as nossas conclusões em duas partes.

7.1.1 Programação Genética e Extracção de Padrões Musicais

A ideia principal da abordagem baseada em programação genética consiste em tentar gerar programas que, ao mesmo tempo que são capazes de produzir uma peça igual ou muito próxima da peça a analisar, constituam uma descrição da estrutura dessa peça, permitindo identificar os principais padrões nela existentes. Nesta parte do nosso trabalho fizemos essencialmente duas tentativas diferentes de aplicar programação genética à extracção de padrões musicais. Nomeadamente, foram testadas duas formas alternativas de representar os programas, bem como vários critérios de avaliação dos mesmos.

Na primeira tentativa que fizemos, cada programa é constituído por uma função principal e por um conjunto de Funções Definidas Automaticamente (FDAs). Os nós terminais da árvore utilizada para representar a função principal representam notas, e os nós internos consistem em funções que permitem concatenar e modificar sequências de notas. As FDAs têm a mesma estrutura que a função principal, e quando utilizadas por esta, devolvem uma sequência de notas. Os padrões

identificados pelo programa correspondem às sequências de notas devolvidas pelas FDAs que são utilizadas mais do que uma vez na função principal. A qualidade de cada programa é medida com base na distância entre a peça produzida e a peça a analisar e no tamanho dos programas, sendo que um programa é tanto melhor quanto menor for o seu tamanho.

Os resultados obtidos nesta primeira tentativa foram muito fracos, pois o algoritmo desenvolvido nem sequer é capaz de gerar indivíduos que produzam peças semelhantes às que utilizámos como conjunto de teste. É de referir ainda que, mesmo quando o algoritmo consegue gerar indivíduos capazes de produzir a peça a analisar e de reutilizar material musical nela existente (o que só acontece quando se utilizam sequências musicais muito pequenas e com uma estrutura muito simples), os programas gerados facilmente atingem grandes dimensões, o que dificulta a sua interpretação por um ser humano.

Na segunda tentativa, aquela que permitiu obter melhores resultados, os nós terminais da árvore utilizada para representar a função principal representam segmentos de notas (cada segmento é representado pela posição das notas da peça em que começa e em que termina) e os nós internos consistem em funções que representam operações que se podem realizar sobre segmentos. Por seu lado, cada FDA consiste apenas num nó terminal que representa um segmento. Tal como na primeira tentativa, os padrões identificados correspondem às FDAs que são utilizadas mais do que uma vez na função principal. A qualidade de cada programa é medida com base na distância entre a peça produzida e a peça a analisar e na capacidade do programa em reutilizar material musical. Este último aspecto é avaliado da seguinte forma: um programa é tanto melhor quanto menor for o tamanho dos segmentos distintos por ele utilizados (o tamanho dos segmentos utilizados mais do que uma vez só é contabilizado uma vez). No final do processo evolutivo são aplicados operadores de melhoramento ao melhor indivíduo gerado os quais, ao modificarem os limites dos segmentos desse indivíduo permitem, em algumas situações, melhorar o seu desempenho. Foi também desenvolvido um método que permite a realização de uma segunda segmentação de uma peça a partir dos segmentos identificados na primeira segmentação. No âmbito desta abordagem experimentámos ainda outras variantes no que diz respeito à representação dos indivíduos e aos critérios utilizados para avaliar os programas. No

entanto, os resultados obtidos com estas variantes foram sempre inferiores.

Apesar de bastante melhores do que os que foram obtidos na primeira tentativa, os resultados obtidos com esta abordagem não podem ser classificados como positivos. Como foi possível observar no capítulo 5, o algoritmo apresenta sérias dificuldades em descobrir os padrões mais importantes existentes em algumas das peças que utilizámos para testar as abordagens desenvolvidas. Como então assinalámos, são duas as principais razões que julgamos serem responsáveis por esta incapacidade. São elas a dificuldade em gerar indivíduos com melhor desempenho através da aplicação dos operadores genéticos, e a importância que a posição relativa dos segmentos apresentados por um indivíduo tem no desempenho deste. À incapacidade do algoritmo em produzir bons resultados, há ainda a acrescentar o facto de ser necessário utilizar populações com um tamanho considerável, o que leva a que o seu tempo de execução seja, quanto a nós, bastante elevado.

Os resultados obtidos levam-nos, assim, a concluir que esta abordagem não deve constituir uma hipótese a considerar se se pretender desenvolver um sistema no qual a extracção de padrões musicais desempenhe um papel importante. Com esta afirmação não pretendemos, no entanto, dizer que se deva pôr de parte qualquer tentativa de aplicar programação genética a este problema. A abordagem que desenvolvemos poderá ser apenas uma de muitas possíveis e, quiçá, talvez entre elas haja uma adequada ao problema. Embora, como referimos, tenhamos tentado várias alternativas no âmbito desta parte do nosso trabalho, é, inclusivamente, possível que algumas alterações no algoritmo desenvolvido possam modificar consideravelmente a sua capacidade de pesquisa. Estas alterações poderiam, eventualmente, passar pela utilização de outros operadores genéticos ou pela utilização de operadores de aprendizagem.

7.1.2 Algoritmos Genéticos e Extracção de Padrões Musicais

A ideia principal da abordagem baseada em algoritmos genéticos consiste em gerar uma sequência binária de 1s e 0s que represente uma segmentação da peça a analisar na qual seja possível identificar os padrões mais importantes nela existentes. Em traços gerais, as sequências de 0s correspondem aos segmentos e os 1s correspondem aos limites entre segmentos. Durante o processo de avaliação de um indivíduo é gerado

um conjunto de relações entre segmentos que permitem identificar o material recorrente existente na peça. Nesta abordagem, o desempenho dos indivíduos é medido tendo em conta a distância entre os segmentos semelhantes e a área ocupada por esses segmentos. Além de operadores genéticos, são também utilizados operadores de aprendizagem que permitem, em cada geração, realizar modificações sobre um pequeno conjunto de indivíduos da população, com o objectivo de melhorar o seu desempenho. Também para esta abordagem foi desenvolvido um método que permite realizar uma segunda segmentação de uma peça a partir dos segmentos identificados na primeira segmentação.

Ao contrário do que acontece com a abordagem baseada em programação genética, julgamos que os resultados obtidos com esta abordagem nos permitem concluir que a sua utilização num sistema de extracção de padrões é viável. Tal como foi possível observar no capítulo 6, para todo o conjunto de teste que utilizámos, o algoritmo não só consegue encontrar soluções de qualidade, como o consegue fazer com considerável frequência. Apesar disso, é um facto que esta abordagem, devido à sua natureza estocástica, não garante que sempre que o algoritmo é executado se consiga encontrar o conjunto de padrões mais importantes existentes na peça a analisar.

Embora não faça parte dos objectivos desta dissertação comparar as abordagens por nós desenvolvidas com as abordagens existentes, devemos lembrar a este propósito que mesmo as abordagens que recorrem a uma pesquisa exaustiva do espaço podem não ser capazes de atribuir a devida importância aos padrões identificados. Na nossa abordagem, uma vez que cada indivíduo constitui uma hipótese de segmentação da peça a analisar, o melhor indivíduo gerado apresenta já aquele que se julga ser o conjunto de padrões mais importantes da peça, não se colocando portanto o problema da ordenação dos segmentos pela sua importância. Também devido à natureza estocástica dos algoritmos evolucionários, com a nossa abordagem é possível obter diferentes soluções para uma mesma peça de cada vez que se executa o algoritmo. Deste modo, caso se considere que a solução encontrada, depois de uma execução do algoritmo, não é satisfatória, este pode ser executado novamente na tentativa de se encontrar uma solução melhor ou simplesmente diferente. Este é, aliás, um procedimento normal quando se utilizam algoritmos evolucionários.

Em suma, à pergunta “é viável a aplicação de algoritmos evolucionários à extracção de padrões musicais?” julgamos poder responder que existe pelo menos uma abordagem cuja aplicação é viável. O facto de termos desenvolvido também uma abordagem cuja aplicação imediata não julgamos viável leva-nos simplesmente a confirmar o que é ensinado em qualquer manual sobre algoritmos evolucionários: a forma como se representam os indivíduos e os operadores utilizados são determinantes para a capacidade de um algoritmo evolucionário em encontrar soluções para um problema.

7.2 Trabalho Futuro

Existem alguns aspectos do trabalho descrito nesta dissertação que julgamos merecerem um estudo mais aprofundado. Assim, para começar, apesar de o conjunto de teste utilizado ter permitido chegar a conclusões importantes acerca da viabilidade de cada uma das duas abordagens descritas, julgamos que é necessário continuar a realizar trabalho experimental. Nomeadamente, seria interessante verificar o comportamento da abordagem baseada em algoritmos genéticos perante peças de grandes dimensões. Isto, apesar de ao longo desta dissertação ter sido possível verificar que não é necessário que uma peça seja muito grande para levantar dificuldades a um processo de extracção de padrões (tome-se como exemplo a peça «Kalenda Maya», a mais pequena das cinco peças utilizadas).

As funções de avaliação que utilizámos nas abordagens descritas não têm em conta aspectos locais das peças como sejam a proximidade ou diferença entre eventos contíguos. No entanto, tal como o reflectem as regras *RP1*, *RP2*, *RP3* e *RP4* da estrutura de agrupamento da Teoria Geral da Música Tonal ou o Modelo de Detecção Local de Limites (MDDL) da Teoria Geral Computacional da Estrutura Musical (ver capítulo 2) esta informação pode ser bastante útil no processo de segmentação de uma peça musical. Em particular, julgamos que a função de avaliação poderia beneficiar com a inclusão de um termo que tivesse em conta os valores calculados com o MDLL. Se este modelo fosse utilizado, o desempenho de um indivíduo seria tanto melhor quanto maior fosse a soma dos VLS dos limites dos segmentos apresentados por um indivíduo.

Apesar de considerarmos que os tempos de execução do algoritmo utilizado na segunda abordagem são aceitáveis, julgamos que há ainda espaço para pelo menos

uma otimização importante. Ao longo da evolução é natural que existam segmentos que são comparados entre si mais do que uma vez, sendo que esta tendência tende a aumentar à medida que o algoritmo vai convergindo. Durante o decorrer do nosso trabalho tentámos utilizar uma estrutura de dados que armazenasse o resultado da comparação entre todos os pares de segmentos já comparados de modo a evitar a comparação repetida de segmentos. No entanto, devido ao elevado número de combinações possíveis entre segmentos, a estrutura de dados facilmente atingia dimensões inportáveis. A procura de uma estrutura de dados adequada a este problema parece-nos um desafio interessante que, se for ultrapassado, pode ajudar a diminuir ainda mais o tempo de execução do algoritmo.

Tal como referimos no capítulo 6, o algoritmo de classificação que é utilizado na abordagem baseada em algoritmos genéticos para escolher os segmentos a utilizar na segunda segmentação de uma peça é algo rudimentar, sendo um dos aspectos a melhorar nesta abordagem. Uma possibilidade de o fazer poderá passar eventualmente pela utilização do algoritmo *Centro de Estrela* descrito no capítulo 3.

O algoritmo de Wagner-Fischer que utilizámos em ambas as abordagens tem em conta apenas as três operações de edição clássicas: substituição, inserção e remoção. Uma forma de melhorar o desempenho dos algoritmos consistirá certamente em utilizar outras operações, como sejam, por exemplo, a fragmentação e a consolidação descritas no capítulo 3.

Finalmente, apesar de as abordagens que descrevemos nesta dissertação terem sido concebidas para o domínio musical, ambas podem ser facilmente adaptadas para trabalhar em outros domínios desde que o corpo de informação a analisar possa ser representado por uma sequência de símbolos. Consideramos, por isso, que seria interessante testar as abordagens descritas, em especial a abordagem baseada em algoritmos genéticos, noutros domínios como, por exemplo, a biologia molecular.

Bibliografia

[Anagnostopoulou, 1999] C. Anagnostopoulou, D. Hörnel, K. Höthker. Investigating the Influence of Representations and Algorithms in Music Classification. In *Focus Workshop on Pattern Processing in Music Analysis and Creation, AISB'99 Convention*, 1999.

[Anagnostopoulou, 1997] C. Anagnostopoulou, G. Westermann. Classification in Music: a Computational Model for Paradigmatic Analysis. In *Proceedings of the International Computer Music Conference*, pp. 125-128, 1997.

[Bäck, 1994] T. Bäck. Selective Pressure in Evolutionary Algorithms: a Characterization of Selection Mechanisms. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pp. 57-62, IEEE Press, 1994.

[Bäck, 1996] T. Bäck, H. Schwefel. Evolutionary Computation: an Overview. In *Proceedings of the Third IEEE Conference on Evolutionary Computation*, pp. 20-29, IEEE Press, 1996.

[Bäck, 1996a] T. Bäck, Evolution Strategies: an Alternative Evolutionary Algorithm. In J. M. Alliot, E. Lutton, E. Ronald (Eds.), *Artificial Evolution*, pp. 3-20, Springer-Verlag, 1996.

[Banzhaf, 1998] W. Banzhaf, P. Nordin, R. E. Keller, F. D. Francone. *Genetic Programming An Introduction. On the Automatic Evolution of Computer Programs and Its Applications*. Morgan Kaufmann, 1998.

[Bent, 1987] I. Bent, W. Drabkin. *Analysis*. W. W. Norton & Company, 1987.

[Blickle, 1995] T. Blickle, L. Thiele. *A Comparison of Selection Schemes Used in Genetic Algorithms*. TIK-Report Nr. 11, TI, Institut für Technische Informatik und Kommunikationsnetze, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, 1995.

[Cambouropoulos, 1999a] E. Cambouropoulos, M. Crochemore, C. S. Iliopoulos, L.

Mouchard, Y. J. Pinzon. Algorithms for Computing Approximate Repetitions in Musical Sequences. In *Proceedings of the AWOCA'99 Workshop (Australasian Workshop on Combinatorial Algorithms)*, 1999.

[Camboropoulos, 1999b] E. Camboropoulos, T. Crawford, C. S. Iliopoulos. Pattern Processing in Melodic Sequences: Challenges, Caveats & Prospects. In *Proceedings of AISB'99, Symposium on Musical Creativity*, pp. 42-47, 1999.

[Camboropoulos, 1998a] E. Camboropoulos. *Towards a General Computational Theory of Musical Structure*. PhD Thesis, University of Edinburgh, 1998.

[Camboropoulos, 1998b] E. Camboropoulos. Musical Parallelism and Melodic Segmentation. In *Proceedings of the XII Colloquio de Informatica Musicale*, 1998.

[Camilleri, 1992] L. Camilleri. Computational Theories of Music: Theoretical and Applicative Issues. In A. Marsden *et al* (Eds.), *Computer Representations and Models in Music*, pp. 171-185, Academic Press, 1992.

[Conte, 1997] M. Conte, G. Trautteur, I. de Falco, A. Della Cioppa, E. Tarantino. Genetic Programming Estimates of Kolmogorov Complexity. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pp. 743-750, Morgan Kaufmann, 1997.

[Cook, 1987] N. Cook. *A Guide to Musical Analysis*. Oxford University Press, 1987.

[Cope, 1991] D. Cope. *Computers and Musical Style*. Oxford University Press, 1991.

[Cope, 1996] D. Cope. *Experiments in Musical Intelligence*. A-R Editions, 1996.

[Crawford, 1998] T. Crawford, C.S. Iliopoulos, R. Raman. String Matching Techniques for Musical Similarity and Melodic Recognition. In W. B. Hewlett, E. Selfridge-Field (Eds.), *Melodic Similarity: Concepts, Procedures and Applications*, vol. 11 of *Computing in Musicology*, pp. 71-100, MIT Press, 1998.

[Crochemore, 1998] M. Crochemore, C. Hancart. Pattern Matching in Strings. In M. J. Atallah (Ed.), *Algorithms and Theory of Computation Handbook*, cap. 11, CRS Press, 1998.

- [Dannenberg, 2000] R. Dannenberg. Dynamic Programming for Interactive Music Systems. In E. M. Miranda (Ed.), *Readings in Music and Artificial Intelligence*, pp. 189-206, Harwood Academic Publishers, 2000.
- [De Jong, 1993] K. A. De Jong, J. Sarma. Generation Gaps Revisited. In L. D. Whitley (Ed.), *Foundations of Genetic Algorithms 2*, Morgan Kaufmann, 1993.
- [Fogel, 1995] G. B. Fogel, D. B. Fogel. Continuous Evolutionary Programming: Analysis and Experiments. In *Cybernetics and Systems*, vol. 26, pp. 79-90, 1995.
- [Forte, 1966] A. Forte. A Program for the Analytical Reading of Scores. *Journal of Music Theory*, 1966. Reproduzido em *Machine Models of Music*, pp. 55-81, MIT Press, 1993.
- [Goldberg, 1989] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [Grilo, 2001] C. Grilo, F. Machado, A. Cardoso. Paradigmatic Analysis Using Genetic Programming. In *Proceedings of the Artificial Intelligence and the Simulation of Behavior Convention (AISB'2001), Symposium on Artificial Intelligence and Creativity in Arts and Science*, pp. 51-58, 2001.
- [Harp, 1991] S. A. Harp, T. Samad, A. Guha. Towards the Genetic Synthesis of Neural Networks. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 360-369, Morgan Kaufmann, 1991.
- [Hart, 1994] W. E. Hart, T. Kammeyer e R. Belew. The Role of Development in Genetic Algorithms. In D. Whitley, M. Vose (Eds.), *Foundations of Genetic Algorithms 3*, pp. 315-332, Morgan Kaufmann, 1994.
- [Huelsenbergen, 1996] L. Huelsenbergen. Toward Simulated Evolution of Machine Language Iteration. In *Genetic Programming 1996: Proceedings of the First Annual Conference*, pp. 315-320, MIT Press, 1996.
- [Huron, 1992] D. Huron. Design Principles in Computer-Based Music Representation. In A. Marsden, A. Pople (Eds.), *Computer Representations and Models in Music*, pp. 5-39, Academic Press, 1992.

[Johanson, 1997] B. E. Johanson. *Automated Fitness Raters for the GP-Music System*. Masters Degree Final Project, University of Birmingham, 1997.

[Keller, 1996] E. Keller, W. Banzhaf. Genetic Programming Using Genotype-Phenotype Mapping From Linear Genomes into Linear Phenotypes. In *Genetic Programming 1996: Proceedings of the First Annual Conference*, pp. 116-122, MIT Press, 1996.

[Kinnear Jr., 1993] K. E. Kinnear Jr.. Generality and Difficulty in Genetic Programming: Evolving a Sort. In S. Forrest (Ed.), *Proceedings of the fifth International Conference on Genetic Algorithms, ICGA-93*, pp. 287-294, Morgan Kaufmann, 1993.

[Koza, 1992] J. R. Koza. *Genetic Programming: Automatic Discovery of Reusable Programs*. MIT Press, 1992.

[Koza, 1993] J. R. Koza. Hierarchical Automatic Function Definition in Genetic Programming. In *Proceedings of the Second Workshop on foundations of Genetic Algorithms*, pp. 297-318, Morgan Kaufmann, 1993.

[Koza, 1994] J. R. Koza. *Genetic Programming II*. MIT Press, 1994.

[Langdon, 1995] W.B. Langdon. Evolving Data Structures Using Genetic Programming. In *Genetic Algorithms: Proceedings of the Sixth International conference (ICGA95)*, pp. 295-302, Morgan Kaufmann, 1995.

[Laske, 1992] O. Laske. Artificial Intelligence and Music: A Cornerstone of Cognitive Musicology. In M. Balaban *et al.* (Eds.), *Understanding Music with AI*, MIT Press, 1992.

[Lemström, 2000] K. Lemström. *String Matching Techniques for Musical Retrieval*. PhD Thesis, University of Helsinki, 2000.

[Lerdahl & Jackendoff, 1983] F. Lerdahl, R. Jackendoff. *A Generative Theory of Tonal Music*, MIT Press, 1983.

[Mayley, 1996] G. Mayley. Landscapes, Learning Costs and Genetic Assimilation. In *Evolutionary Computation*, vol. 4(3), pp. 213-234, 1996.

[Meredith, 2001] D. Meredith, G. A. Wiggins, K. Lemström. Pattern Induction and

Matching in Polyphonic Music and other Multidimensional Datasets. In the 5th *World Multi-Conference on Systemics, Cybernetics and Informatics (SCI'2001)*, vol. X, pp. 61-66, 2001.

[Michalewicz, 1996] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1996.

[Michalewicz, 1997] Z. Michalewicz, R. Hinterding, M. Michalewicz. Evolutionary Algorithms. In W. Pedrycz (Ed.), *Fuzzy Evolutionary Computation*, cap. 2, Kluwer Academic, 1997.

[Mitchell, 1996] M. Mitchell. *An Introduction to Genetic Algorithms*, MIT Press, 1996.

[Mitchell, 1999] M. Mitchell, C. E. Taylor. Evolutionary Computation: an Overview. In *Annual Review of Ecology and Systematics*, 20: 593-616, 1999.

[Mitchell, 1997] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[Mongeau & Sankoff, 1990] M. Mongeau, D. Sankoff. Comparison of Musical Sequences. In *Computer and the Humanities*, 24: 161-175, 1990.

[Montana, 1994] D. J. Montana. *Strongly Typed Genetic Programming*. BBN Technical Report #7866, Bolt Beranek and Newman, 1994.

[Music Theory Online] <http://www.dolmetsch.com/theoryintro.htm>.

[Nattiez, 1975] J. J. Nattiez. *Fondements d'une Semiologie de la Musique*. Union Generale d'Editions, 1975.

[Neves, 1999] A. Neves, A. Silva, E. Costa, Evolutionary Path Planning for Nonholonomic Robots. In *Proceeding of the Genetic and Evolutionary Computation Conference*, vol. 1, pp. 466-472, Morgan Kaufmann, 1999.

[Ó Maidín, 1992] D. Ó Maidín. Representation of Music Scores for Analysis. In A. Marsden, A. Pople (Eds.), *Computer Representations and Models in Music*, pp. 66-93, Academic Press, 1992.

[Ó Maidín, 1998] D. Ó Maidín. A Geometrical Algorithm for Melodic Difference. In

W. B. Hewlett, E. Selfridge-Field (Eds.), *Melodic Similarity: Concepts, Procedures and Applications*, vol. 11 of *Computing in Musicology*, pp. 65-72, MIT Press, 1998.

[Pennycook, 1993] B. Pennycook, D.R. Stammen, D. Reynolds. Toward a computer Model of a Jazz Improviser. In *Proceedings of the 1993 International Computer Music Conference*, pp. 228-231, International Computer Music Association, 1993.

[Pereira, 1997] F. C. Pereira, C. Grilo, L. Macedo, A. Cardoso. Composing Music with Case-Based Reasoning. In T. Veale (Ed.), *Proceedings of the International Conference on Computational Models of Creative Cognition*, MIND II, 1997.

[Pereira, 1999] F. B. Pereira, P. Machado, E. Costa, A. Cardoso. Graph Based Crossover - a Case Study With the Busy Beaver Problem. *Gecco'99*, Morgan Kaufmann, 1999.

[Pereira, 2002] F. B. Pereira. Estudo das *Interações entre Evolução e Aprendizagem em Ambientes de Computação Evolucionária*. Tese de Doutoramento, Universidade de Coimbra, 2002.

[Perttu, 2000] S. Perttu. *Combinatorial Pattern Matching in Musical Sequences*. Master's Thesis, University of Helsinki, Department of Computer Science, Series of Publications, C-2000-38, 2000.

[Poli, 1997] R. Poli. Evolution of Graph Like Programs With Parallel Distributed Genetic Programming. In *Genetic Algorithms: Proceedings of the Seventh International Conference*, pp. 346-353, Morgan Kaufmann, 1997.

[Porto Editora] <http://www.portoeditora.pt>.

[Rolland, 2000] P.Y. Rolland, J.G. Ganascia. Musical Pattern Extraction and Similarity Assessment. In E. M. Miranda (Ed.), *Readings in Music and Artificial Intelligence*, pp. 115-144, Harwood Academic, 2000.

[Rolland, 1999] P.Y. Rolland. Discovering Patterns in Musical Sequences. In *Journal of New Music Research*, 28, n° 4, pp. 334-350, 1999.

- [Rowe, 1993] R. Rowe. *Interactive Music Systems: Machine Listening and Composing*. MIT Press, 1993.
- [Ruwet, 1972] N. Ruwet. *Langage, musique, poésie*. Editions du Seuil, 1972.
- [Selfridge-Field, 1998] E. Selfridge-Field. Conceptual and Representational Issues in Melodic Comparison. In W. B. Hewlett, E. Selfridge-Field (Eds.), *Melodic Similarity: Concepts, Procedures and Applications*, vol. 11 of *Computing in Musicology*, pp. 3-64, MIT Press, 1998.
- [Silva, 2000], A. Silva. *Redes Programadas Geneticamente: Uma Abordagem Evolucionária à Síntese de Agentes Autómatos*. Tese de Mestrado, Universidade de Coimbra, 2000.
- [Smail, 1993] A. Smail, G. Wiggins, M. Harris. Hierarchical Music Representation to Composition and Analysis. In *Computer and the Humanities*, vol. 27, pp. 7-27, 1993.
- [Smith, 1998] L. A. Smith, R. J. McNab, I. H. Witten. Sequence-Based Melodic Comparison: A Dynamic-Programming Approach. In W. B. Hewlett, E. Selfridge-Field (Eds.), *Melodic Similarity: Concepts, Procedures and Applications*, vol. 11 of *Computing in Musicology*, pp. 101-117, MIT Press, 1998.
- [Spears, 1993] W. Spears, K. A. De Jong, T. Bäck, D. B. Fogel, H. de Garris. An Overview of Evolutionary Computation. In *Proceedings of the European Conference on Machine Learning (ECML-93)*, LNAI, vol. 667, pp. 442-459, Springer-Verlag, 1993.
- [Spears, 1998] W. Spears. *The Role of Mutation and Recombination in Evolutionary Algorithms*. PhD Thesis, George Mason University, 1998.
- [Stephen, 1992] Graham G. Stephen. *String Search*. Technical Report, School of Electronic Engineering Science, University College of North Wales, 1992.
- [Sundberg, 1976] J. Sundberg, B. Lindblom. Generative Theories in Language and Music Description. In *Cognition*, pp. 99-122, 1976.
- [Whitley, 1989] D. Whitley, T. Starkweather, D. Fuquay. Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator. In *Proceedings of the*

Third International Conference on Genetic Algorithms, pp. 133-140, Morgan Kaufmann, 1989.

[Xiao, 1997] J. Xiao, Z. Michalewicz, L. Zhang, K. Trojanowski. Adaptive Evolutionary Planner/Navigator for Mobile Robots. In *IEEE Transactions on Evolutionary Computation*, vol. 1, No. 1, pp. 18-28, 1997.

[Zongker, 1996] D. Zongker, B. Punch, B. Rand. *lil-gp1.01*. Michigan State University, 1996.

Be m'aperdut...

Bernard de Ventadour

0

42

Al.1

Al.2

Al.3

84

141

X

Anexo I

Maria muoter reinû maît

The musical score is written in a single system with four systems of staves. The first system consists of two staves: the top staff is in treble clef and the bottom staff is in bass clef. The second system also consists of two staves in the same clefs. The third system consists of two staves, both in bass clef. The fourth system also consists of two staves, both in bass clef. The score is divided into sections labeled A, A', B, and B'. Section A (measures 0-27) includes sub-sections a (0-8), b (8-18), c (18-27), and b' (27-36). Section A' (measures 36-62) includes sub-sections a (36-44) and b' (44-62). Section B (measures 71-81) includes sub-sections dl (71-76) and d (76-81). Section B' (measures 90-108) includes sub-sections dl (90-95) and d (95-108). The notation includes various note values, rests, and dynamic markings.

Kalenda Maya

Raïmbaut de Vaqueiras

The musical score is written in treble clef and consists of ten staves. The first two staves are labeled 'A' and start at measure 0. The third staff is labeled 'B' and starts at measure 44. The fourth staff is labeled 'B'' and starts at measure 54. The fifth staff is labeled 'c' and starts at measure 64. The sixth staff is labeled 'c' and starts at measure 67. The seventh staff is labeled 'D' and starts at measure 70. The eighth staff is labeled 'c' and starts at measure 79. The ninth staff is labeled 'c' and starts at measure 82. The tenth staff is labeled 'D'' and starts at measure 85, ending at measure 95. The music features a mix of quarter, eighth, and sixteenth notes, with some rests and phrasing slurs.

Varição 4 das «Seis variaciones sobre um tema de Milán»

J. Nin-Culmell

Allegro

g a a b

*f*⁰

3 3 3

18

b b c g d b

30 36 42 51

b c'

63

e e

70 88

b' f a'

106 124

f d

139

f

151

220

A

X

A

Nota: Além dos segmentos {A, B, a, b, c, d, e, f} existem outros segmentos como, por exemplo, o segmento g que fazem parte de diferentes segmentos (neste caso os segmentos a e d). À exceção de duas das ocorrências do segmento g, não assinalámos estes segmentos para não dificultar a leitura da pauta.

Syrinx

Claude Debussy

Très modéré

mf

A

14

24

p

B

36

C

p

Un peu mouvementé (mais très peu)

A'

D

52

68

D

D'

80

E

E'

E''

F

92

96

100

p 104

F

F

G

F'

H

110

p 113

p

p 119

F'

H

I

C'

p 131

p

143

Syrinx (continuação)

The musical score for Syrinx (continuation) is written in G-flat major (three flats) and 4/4 time. It consists of five staves of music. The first staff features a series of eighth-note triplets, with some marked with a 'J' (jazz) articulation. The second staff includes trills marked '(tr)' and a dynamic marking of *mf* at measure 187. The third staff has a dynamic marking of *dim.* at measure 201. The fourth staff contains two measures of eighth-note triplets, both marked *p* (piano), with measure numbers 211 and 221. The fifth staff begins with the instruction *Em retenant jusqu'à la fin* and features a triplet of eighth notes followed by a half note, then a *Très retenu* section with a five-note descending scale marked *p marqué* and *perdendosi*.