

UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Representações Redundantes para Procura Evolutiva

Tese para a obtenção do grau de doutor no ramo de Engenharia Electrónica
e Computação, especialidade de Ciências da Computação

Marisol de Brito Correia

Orientador: Doutor Carlos Manuel Mira da Fonseca

Constituição do Júri:

Presidente: Doutor João Pinto Guerreiro, Reitor da Universidade do Algarve

Vogais:

Doutor Pedro João Valente Dias Guerreiro, FCT, Universidade do Algarve

Doutor Luís Miguel Parreira e Correia, FC, Universidade de Lisboa

Doutor Carlos Manuel Mira da Fonseca, FCT, Universidade do Algarve

Doutor Fernando Miguel Pais da Graça Lobo, FCT, Universidade do Algarve

Doutor Marco Paulo dos Santos Carrasco, ESGHT, Universidade do Algarve

Doutor Francisco José Baptista Pereira, ISE, Instituto Politécnico de Coimbra

FARO

2009

UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Representações Redundantes para Procura Evolutiva

Tese para a obtenção do grau de doutor no ramo de Engenharia Electrónica
e Computação, especialidade de Ciências da Computação

Marisol de Brito Correia

Orientador: Doutor Carlos Manuel Mira da Fonseca

Constituição do Júri:

Presidente: Doutor João Pinto Guerreiro, Reitor da Universidade do Algarve

Vogais:

Doutor Pedro João Valente Dias Guerreiro, FCT, Universidade do Algarve

Doutor Luís Miguel Parreira e Correia, FC, Universidade de Lisboa

Doutor Carlos Manuel Mira da Fonseca, FCT, Universidade do Algarve

Doutor Fernando Miguel Pais da Graça Lobo, FCT, Universidade do Algarve

Doutor Marco Paulo dos Santos Carrasco, ESGHT, Universidade do Algarve

Doutor Francisco José Baptista Pereira, ISE, Instituto Politécnico de Coimbra

FARO

2009

NOME: Marisol de Brito Correia

FACULDADE: Faculdade de Ciências e Tecnologia

ORIENTADOR: Carlos Manuel Mira da Fonseca

DATA: Julho de 2009

TÍTULO DA TESE: Representações Redundantes para Procura Evolutiva

Resumo

Existe actualmente um grande interesse na influência da redundância e da neutralidade no comportamento dos algoritmos evolutivos. Alguns autores consideram que a procura evolutiva pode ser influenciada de forma positiva pela utilização de representações redundantes, enquanto outros afirmam que a adição de redundância aleatória parece ser inútil na optimização. Em virtude desta falta de consenso e tendo em conta a existência de redundância e de neutralidade no código genético, são propostas nesta dissertação duas novas famílias de representações binárias redundantes.

A família de representações neutrais baseia-se na formulação matemática dos códigos de controlo de erros, utilizados em comunicações digitais, enquanto a família de representações não neutrais baseia-se em transformações lineares e permite definir a vizinhança fenotípica pretendida de uma forma simples e directa.

É apresentado um estudo experimental destinado a avaliar a influência da redundância e da neutralidade no desempenho de um algoritmo evolutivo simples, especificamente, de uma estratégia evolutiva $(1+1)$ -ES, modelada através de cadeias de Markov e aplicada a paisagens de aptidão NK .

Os resultados permitem concluir que a vizinhança fenotípica induzida pela representação redundante domina o comportamento do algoritmo evolutivo, afectando de forma mais profunda a procura do que a neutralidade, e que as melhores representações não apresentam valores extremos de nenhum dos indicadores de qualidade das representações habitualmente considerados na literatura.

Palavras-chave: Algoritmos Evolutivos, Representações Redundantes, Neutralidade, Códigos de Controlo de Erros, Paisagens de Aptidão *NK*

Redundant Representations for Evolutionary Search

Abstract

There is currently great interest in how redundancy and neutrality influence the behaviour of evolutionary algorithms. Some authors consider that evolutionary search may be positively influenced by the use of redundant representations, whereas others note that the addition of random redundancy to a representation may be useless in optimization. Given this lack of a consensus, and motivated by the presence of redundancy and neutrality in the natural genetic code, two new families of redundant binary representations, one neutral and another non-neutral, are proposed in this dissertation.

The family of neutral representations is based on the mathematical formulation of error-control codes, commonly used in telecommunications, while the family of non-neutral representations is based on linear transformations, and allows phenotypic neighbourhoods to be designed simply and effectively.

An experimental study aimed at assessing the influence of redundancy and neutrality on the performance of a simple evolutionary algorithm is presented. Specifically, an $(1+1)$ -ES evolution strategy applied to NK fitness landscapes is modeled via Markov chains.

From the results, it may be concluded that the phenotypic neighbourhood induced by a redundant representation dominates the behaviour of the evolutionary algorithm, affecting the search more strongly than neutrality, and that the best representations do not exhibit extreme values of any of the indicators of representation quality commonly adopted in the literature.

Keywords: Evolutionary Algorithms, Redundant Representations, Neutrality, Error Control Codes, NK Fitness Landscapes

Agradecimentos

Ao Prof. Carlos Fonseca pela orientação e conhecimentos transmitidos e pelo acolhimento no laboratório de Computação Evolutiva.

À Escola Superior de Gestão, Hotelaria e Turismo (ESGHT) da Universidade do Algarve por apoiar e incentivar a formação dos seus docentes, em particular na dispensa do serviço docente e no apoio financeiro para divulgação da investigação realizada.

Ao Prof. Paulo Martel pelas correcções à secção 2.2, ao Prof. Fernando Lobo e ao Prof. Paulo Carrasco pelo incentivo e apoio prestados, sempre que foi necessário.

Ao colega Paulo Águas por ter assegurado a direcção da ESGHT durante a fase final da escrita da dissertação.

Aos Prof. Graça Ruano, Prof. João Brandão e Prof. Hans du Buf pela disponibilização do equipamento informático para as simulações efectuadas.

À colega Célia Ramos pelo acréscimo do seu serviço docente, permitindo a minha dispensa e pelo apoio incondicional com que sempre me presenteou. Ao colega João Fontinha pelas conversas que me permitiram ter uma melhor visão social e humana do mundo, e aos restantes colegas do núcleo e da ESGHT por me terem sempre apoiado.

Aos colegas que trabalharam no laboratório de Computação Evolutiva, em

particular à Cristina Vieira, Marília Felismino, Luís Simões, Mónica Neves, Dilip Datta da Índia, Elizabeth Wanner, André Cançado, Rodrigo Cardoso e Eduardo Carrano do Brasil e à Iryna Yevseyeva da Finlândia.

Ao meu marido Roberto Laranja pelo seu amor e inestimável apoio emocional, aos meus filhos Sara e Roberto pelo carinho imenso que sempre me ofereceram, aos meus pais por terem contribuído e ajudado na minha formação, aos meus irmãos que, mesmo de longe, sempre me incentivaram, e aos restantes familiares e amigos que sempre me ajudaram.

Acrónimos

AE (EA)	Algoritmo Evolutivo	Evolutionary Algorithm
ADN (DNA)	Ácido Desoxirribonucleico	Deoxyribonucleic Acid
ARN (RNA)	Ácido Ribonucleico	Ribonucleic Acid
AC (CA)	Autómato Celular	Cellular Automaton
CE (EC)	Computação Evolutiva	Evolutionary Computation
EMD (MDS)	Escalonamento Multidimensional	Multi-Dimensional Scaling
EE (ES)	Estratégia Evolutiva	Evolution Strategy
NK	Paisagem de Aptidão NK	NK Fitness Landscape
RBA (RBN)	Rede Booleana Aleatória	Random Boolean Network

Notação

f_g	Mapeamento genótipo-fenótipo
f_f	Mapeamento fenótipo-aptidão
$d(u, v)$	Distância de Hamming entre u e v
Φ_g	Espaço genotípico
Φ_f	Espaço fenotípico
x_g	Genótipo
x_f ou $f_g(x_g)$	Fenótipo
$f_f(x_f)$	Aptidão do fenótipo
d_m	Localidade
$C(n, k)$	Código de bloco
$GF(2)$	Corpo de <i>Galois</i> de ordem 2
d_{min}	Distância mínima de um código de bloco $C(n, k)$
u	Dados de mensagem
v	Palavra-código
G	Matriz geradora
H	Matriz de verificação de paridade
s	Síndrome
C_i	‘Coset’ com síndrome i
$U(x)$	Polinómio de informação
$V(x)$	Polinómio de código
$g(x)$	Polinómio gerador
$S(x)$	Polinómio de síndrome
$NonNN_Z(\ell, k)$	Família de representações não neutrais
$NN_g(\ell, k)$	Família de representações neutrais
$R = \{z_0, z_1, \dots, z_{2^\ell - k - 1}\}$	Representação neutral

S_{in}	Medida de sinonímia
P	Matriz de transição de uma cadeia de Markov
p_{ij}	Elemento da matriz de transição de uma cadeia de Markov P
$VM_{x_{f_i}}$	Conjunto de vizinhos com aptidão maior ou igual a x_{f_i}

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Estrutura	2
1.3	Contribuições	3
2	Redundância e neutralidade em sistemas evolutivos naturais e artificiais	7
2.1	Introdução	7
2.2	Redundância no código genético	8
2.2.1	Teorias evolutivas	12
2.2.2	Conformação do ARN	14
2.3	Redundância em computação evolutiva	18
2.3.1	Recomendações para o desenvolvimento e utilização de representações	22
2.4	Representações redundantes	24
2.4.1	Métrica de Hamming	26
2.4.2	Mapeamento genótipo-fenótipo e mapeamento fenótipo-aptidão	27
2.4.3	Exemplos de mapeamentos genótipo-fenótipo binários .	29
	Mapeamento aleatório estático	29

	Mapeamento votação trivial	29
	Mapeamento votação standard	30
	Mapeamento autómato celular	31
	Mapeamento rede booleana aleatória	32
	Mapeamento maioria	33
	Mapeamento paridade	33
	Mapeamento tabela de verdade	33
2.5	Propriedades das representações	34
2.5.1	Uniformidade	34
2.5.2	Conectividade	34
2.5.3	Sinonímia	34
2.5.4	Localidade	35
2.6	Considerações finais	36
3	Códigos de controlo de erros	39
3.1	Introdução	39
3.2	Códigos de bloco lineares	41
3.2.1	Códigos de Hamming	44
	Matriz geradora	44
	Matriz de verificação de paridade	46
	Descodificação pela síndrome	47
	‘Cosets’ e matriz padrão	48
3.2.2	Códigos cíclicos	50
	Descodificação com códigos cíclicos	54
3.2.3	Propriedades dos códigos e dos ‘cosets’	57
3.3	Códigos equivalentes	59
3.4	Considerações finais	60

4	Desenvolvimento de famílias de representações binárias redundantes	61
4.1	Introdução	61
4.2	Representações redundantes baseadas em transformações lineares	62
4.2.1	Representações com genes não codificantes	63
4.2.2	Representações com poligenia	64
4.2.3	Representações com poligenia e pleiotropia	66
4.3	Representações redundantes neutrais inspiradas em códigos de controlo de erros	70
4.3.1	Neutralidade	70
4.3.2	Representações neutrais	73
	Codificação e decodificação	75
4.3.3	Representações equivalentes	76
4.4	Enumeração de representações neutrais	81
4.4.1	Fecho da família de representações neutrais sobre a relação de <i>Vizinhança</i>	83
4.5	Propriedades de $NN_g(\ell, k)$	85
4.5.1	Uniformidade	85
4.5.2	Conectividade e vizinhança fenotípica	86
4.5.3	Topologia	90
4.5.4	Sinonímia	91
4.5.5	Localidade	93
4.5.6	Relação entre as propriedades	94
4.6	Considerações finais	96
5	Desempenho das famílias de representações redundantes	99
5.1	Introdução	99

5.2	Paisagens de aptidão NK	100
5.2.1	Ótimo global e ótimos locais nas paisagens NK . . .	102
5.2.2	Representação não redundante versus representação neutral	103
5.2.3	Representação neutral versus representação não neutral	109
5.3	Cadeias de Markov	111
5.3.1	Representação não redundante	113
5.3.2	Representação neutral	114
5.3.3	Representação não neutral	115
5.3.4	Probabilidade de alcançar o ótimo global	117
5.4	Estudo experimental e resultados	120
5.5	Considerações finais	136
6	Conclusões	139
6.1	Perspectivas de trabalho futuro	142
	Bibliografia	145
A	Topologias das redes neutrais de $NN_g\{7,4\}$	155

Lista de Figuras

2.1	Uma estrutura primária do ARN e a respectiva estrutura secundária. Fonte:(Contrafold 2009)	15
2.2	Esquema de redes neutrais no espaço das sequências (A) e vizinhança fenotípica (B)	16
2.3	Hipergrafo representando o espaço dos genótipos com $\ell = 4$	25
3.1	Exemplo de um código $C(7, 4)$	45
3.2	Geração do código cíclico $C(7, 4)$ da figura 3.1	55
4.1	Espaço genotípico redundante dividido em 4 subespaços entrelaçados	72
4.2	Redes neutrais correspondentes aos 16 fenótipos diferentes	72
4.3	Exemplo de uma representação neutral de $NN_g(7, 4)$	80
4.4	Hipergrafo representando um espaço genotípico com $ \Phi_g = 2^\ell = 2^7$	86
4.5	Hipergrafo representando o espaço genotípico com $\ell = 4$ e os fenótipos relativos à representação neutral $\{0, 1, 2, 4\} \in NN_g(4, 2)$	87
4.6	MDS de algumas representações de $NN_g(7, 4)$	91
4.7	MDS de algumas representações de $NN_g(14, 11)$	91
4.8	MDS de algumas representações de $NN_g(6, 2)$	92

4.9	Relação entre algumas propriedades de $NN_g(14, 11)$	95
5.1	Hipergrafo representando uma paisagem de aptidão $NK(3, 2)$.	104
5.2	Hipergrafo representando um espaço genotípico binário com $\ell = 6$	105
5.3	Hipergrafo representando o espaço dos genótipos com $\ell = 6$ em decimal	106
5.4	Hipergrafo representando uma paisagem de aptidão $NK(3, 2)$ utilizando a representação neutral $\{0, 1, 2, 3, 4, 5, 6, 17\} \in$ $NN_g(6, 3)$	106
5.5	Hipergrafo representando uma instância de $NK(2, 1)$	109
5.6	Hipergrafo representando uma paisagem de aptidão $NK(2, 1)$ utilizando a representação neutral $\{0, 2\} \in NN_g(3, 2)$	110
5.7	Hipergrafo representando uma paisagem de aptidão $NK(2, 1)$ utilizando a representação não neutral $NonNN_Z(4, 2)$ com a mesma vizinhança fenotípica de $\{0, 2\} \in NN_g(3, 2)$	111
5.8	Probabilidade de alcançar o óptimo global numa instância da paisagem de aptidão $NK(11, 1)$ utilizando a representação neutral $\{0, 1, 2, 70, 74, 8192, 514, 66\} \in NN_g(14, 11)$ e a repre- sentação não neutral $NonNN_Z(98, 11)$ correspondente	121
5.9	Probabilidade de alcançar o óptimo global numa instância da paisagem de aptidão $NK(11, 1)$ utilizando a representação neu- tral $\{0, 1, 2, 3, 4, 5, 514, 515\} \in NN_g(14, 11)$ e a representação não neutral $NonNN_Z(92, 11)$ correspondente	122
5.10	Valores esperados da aptidão em $NK(11, K)$, utilizando $(1+1)$ - <i>ES</i> com representações neutrais e não neutrais, e procura aleatória	125
5.11	Rankings das representações neutrais seleccionadas versus ran- kings gerados aleatoriamente	128

5.12	Rankings médios das representações neutrais seleccionadas para $K = 0, 1, 2, 3, 4, 5, 6$ versus rankings gerados aleatoriamente	129
5.13	Rankings médios das representações neutrais	130
5.14	Desempenho versus características das representações neutrais	131
5.15	Características das representações neutrais que contêm a vizinhança fenotípica da representação não redundante	132
5.16	Desempenho das representações neutrais versus localidade . .	133
5.17	Desempenho das representações neutrais versus conectividade	134
5.18	Desempenho das representações neutrais versus sinonímia . . .	135

Lista de Tabelas

2.1	Tabela do código genético	11
3.1	Adição módulo-2	41
3.2	Multiplicação módulo-2	41
3.3	Matriz padrão de um código $C(n, k)$	49
3.4	Geração dos polinómios de código do código cíclico da figura 3.1 utilizando o polinómio gerador $g(x) = x^3 + x + 1 = 1011$	53
3.5	Tabela de síndromes e de erros do código cíclico da figura 3.2 utilizando o polinómio gerador $g(x) = x^3 + x + 1 = 1011$. . .	57
3.6	Matriz geradora de um código equivalente ao código da figura 3.1	59
4.1	Genótipos e respectivos fenótipos obtidos utilizando a representação com poligenia definida na expressão 4.5	65
4.2	Genótipos vizinhos e respectivos fenótipos dos genótipos que descodificam para o fenótipo 00 da representação definida na expressão 4.5	66
4.3	Genótipos e respectivos fenótipos obtidos utilizando a representação com a matriz definida em 4.7	68

4.4	Genótipos vizinhos e respectivos fenótipos dos genótipos que descodificam para o fenótipo 000 da representação definida em	
4.7		68
4.5	Polinómios geradores utilizados para definir $NN_g(\ell, k)$	75
4.6	Representações equivalentes a $\{0, 1, 9, 77, 4, 14, 6, 73\}$	79
4.7	Representações equivalentes a $\{0, 1, 9, 77, 4, 14, 6, 73\}$ com os <i>zeros</i> ordenados pela síndrome	81
4.8	Representações vizinhas de $\{0, 1, 2, 3\} \in NN_g(4, 2)$ e representação canónica das equivalentes	82
4.9	Número de representações neutrais de $NN_g(\ell, k)$ - Parte 1	84
4.10	Número de representações neutrais de $NN_g(\ell, k)$ - Parte 2	85
4.11	Genótipos e respectivos fenótipos da representação neutral $\{0, 1, 2, 4\} \in NN_g(4, 2)$	87
4.12	Genótipos vizinhos e respectivos fenótipos da rede neutral que representa o fenótipo 1 na representação $\{0, 1, 2, 4\} \in NN_g(4, 2)$	88
4.13	Conectividade mínima e máxima para cada $NN_g(\ell, k)$ - Parte 1	90
4.14	Conectividade mínima e máxima para cada $NN_g(\ell, k)$ - Parte 2	90
4.15	Sinonímia mínima e máxima para cada $NN_g(\ell, k)$ - Parte 1	92
4.16	Sinonímia mínima e máxima para cada $NN_g(\ell, k)$ - Parte 2	93
4.17	Diâmetro mínimo e máximo de $NN_g(\ell, k)$	93
4.18	Localidade mínima e máxima para cada $NN_g(\ell, k)$ - Parte 1	94
4.19	Localidade mínima e máxima para cada $NN_g(\ell, k)$ - Parte 2	94
5.1	Genótipos que mapeiam para cada fenótipo de $\{0, 1, 2, 3, 4, 5, 6, 17\} \in NN_g(6, 3)$	104
5.2	Vizinhos da rede neutral que representa o fenótipo 1 na representação $\{0, 1, 2, 3, 4, 5, 6, 17\} \in NN_g(6, 3)$ - Parte 1	107

5.3	Vizinhos da rede neutral que representa o fenótipo 1 na representação $\{0, 1, 2, 3, 4, 5, 6, 17\} \in NN_g(6, 3)$ - Parte 2	107
5.4	Matriz de transição da cadeia de Markov obtida para a paisagem de aptidão $NK(2, 1)$ da figura 5.5 utilizando a representação não redundante	114
5.5	Matriz de transição da cadeia de Markov obtida para a paisagem de aptidão $NK(2, 1)$ utilizando a representação neutral $\{0, 2\} \in NN_g(3, 2)$	115
5.6	Matriz de transição da cadeia de Markov obtida para a paisagem de aptidão $NK(2, 1)$ utilizando a representação não neutral $NonNN_Z(4, 2)$ que corresponde a $\{0, 2\} \in NN_g(3, 2)$	116
5.7	Probabilidades de atingir os óptimos locais na paisagem de aptidão NK da figura 5.6 utilizando a representação neutral $\{0, 2\} \in NN_g(3, 2)$	118
5.8	Probabilidades de alcançar os óptimos locais na paisagem de aptidão NK da figura 5.7 utilizando a representação não neutral $NonNN_Z(4, 2)$ correspondente a $\{0, 2\} \in NN_g(3, 2)$	119
5.9	Probabilidades de alcançar os óptimos locais na paisagem de aptidão NK da figura 5.1 utilizando a representação não redundante	119
5.10	Resultados da utilização das representações: neutral (NN), não neutral ($NonNN$) e não redundante (NR) em paisagens $NK(11, 1)$	122
5.11	Resultados da utilização das representações: neutral (NN), não neutral ($NonNN$) e não redundante (NR) em paisagens $NK(11, 3)$	123

5.12 Resultados da utilização das representações: neutral NN , não
neutral $NonNN$ e não redundante NR em paisagens $NK(11, 10)$ 123

Capítulo 1

Introdução

1.1 Motivação

A existência de redundância e as manifestações de neutralidade no código genético, juntamente com a teoria neutral da evolução molecular proposta por Kimura (1968) têm motivado o aparecimento de diversas propostas de representações redundantes na área da computação evolutiva (CE).

A utilidade destas representações, com e sem neutralidade, quando utilizadas em algoritmos evolutivos (AEs), tem sido alvo de estudo. A noção de neutralidade e de redes neutras tem atraído a atenção de inúmeros investigadores, pelo potencial de estabelecer caminhos alternativos na evolução da população em AEs, e desta forma melhorar a qualidade da procura.

Alguns estudos procuraram determinar o efeito da redundância, neutral ou não, no desempenho da procura evolutiva, mas os resultados foram divergentes. Enquanto uns consideram que um aumento do potencial evolutivo de uma população, ou seja, a capacidade de que mutações aleatórias possam contribuir para a melhoria do processo evolutivo, pode ser conseguido utilizando representações redundantes, outros notam que a adição de redundância

aleatória não parece aumentar esse potencial.

Em contrapartida, a introdução de redundância controlada e estruturada para permitir a detecção e/ou a correção de erros em comunicações e em armazenamento digital, sugere a utilização da formulação matemática bem estudada dos códigos de controlo de erros no desenvolvimento de representações redundantes para procura evolutiva. Idealmente, uma tal abordagem iria permitir um tratamento mais formal das questões associadas ao papel da redundância na evolução do que o que tem sido possível até à data.

A ausência de resultados conclusivos na literatura sobre as eventuais vantagens das representações redundantes, a existência de redundância e de neutralidade no código genético e o potencial da utilização dos códigos de controlo de erros conduziram ao desenvolvimento e à análise das duas famílias de representações binárias redundantes que se propõem nesta dissertação.

1.2 Estrutura

As formas de manifestação da redundância e da neutralidade no código genético e as representações redundantes com e sem neutralidade que têm vindo a ser propostas em CE são revistas no capítulo 2. As recomendações sobre o desenvolvimento de representações para procura evolutiva, bem como os mapeamentos genótipo-fenótipo redundantes propostos na literatura e as propriedades relevantes das representações redundantes são alvo de análise nesse capítulo.

Os códigos de controlo de erros, que permitem a detecção e a correção de erros ocorridos durante a transmissão de mensagens e a gravação e leitura em dispositivos de armazenamento, são apresentados no capítulo 3. São considerados alguns códigos de bloco lineares, como os códigos cíclicos, e em

particular, os códigos de Hamming. A definição destes códigos pela matriz geradora e a sua descodificação pela síndrome, bem como a definição de códigos cíclicos a partir de um polinómio gerador, são apresentados como base para o desenvolvimento de representações redundantes em computação evolutiva.

O capítulo 4 apresenta as duas famílias de representações redundantes propostas nesta dissertação. A abordagem incremental que foi utilizada para definir a família de representações redundantes não neutrais, partindo de uma representação redundante não codificante e acabando numa representação que incorpora as propriedades de poligenia e de pleiotropia, é indicada. A família de representações neutrais baseadas em códigos de controlo de erros é definida, e é apresentado um algoritmo para a enumeração de representações dessa família. As propriedades destas representações e as relações entre elas são também enunciadas.

Os efeitos da redundância e da neutralidade das representações no desempenho de um algoritmo evolutivo simples são estudados experimentalmente no capítulo 5. É considerada uma estratégia evolutiva do tipo $(1+1)$ -ES aplicada a paisagens de aptidão NK , e o seu comportamento é modelado usando cadeias de Markov. A discussão dos resultados é apresentada no final do capítulo.

Por último, a discussão das conclusões e as perspectivas de trabalho futuro são apresentados no capítulo 6.

1.3 Contribuições

As principais contribuições desta dissertação são:

1. O desenvolvimento de uma nova família de representações binárias re-

dundantes, com base em códigos de controlo de erros, que exibem neutralidade e que são definidas pela topologia da rede neutral pretendida. Esta família é constituída por inúmeras representações neutrais que apresentam uma vasta gama de características (Fonseca & Correia 2005);

2. O desenvolvimento de uma família de representações binárias redundantes, não neutrais, definidas com base em transformações lineares e que permitem a definição da vizinhança fenotípica desejada de uma forma simples e directa (Fonseca & Correia 2005);
3. O desenvolvimento de um algoritmo para a geração de representações de famílias de representações neutrais;
4. A caracterização das representações neutrais propostas quanto às propriedades referidas na literatura, como uniformidade, conectividade, sinonímia e localidade;
5. A realização de um estudo experimental para analisar a influência da redundância e o papel da neutralidade no desempenho das representações propostas, em combinação com um algoritmo evolutivo simples, aplicado a paisagens de aptidão NK , sendo o comportamento do algoritmo modelado através de cadeias de Markov (Correia & Fonseca 2007a, Correia & Fonseca 2007b).

São contribuições adicionais desta dissertação:

1. Uma discussão sobre as formas de manifestação de redundância e de neutralidade no código genético e sobre a utilização de representações redundantes, neutrais ou não, em computação evolutiva;

2. Uma discussão sobre a forma como as representações neutrais e não neutrais transformam as paisagens de aptidão NK (Correia & Fonseca 2007a, Correia & Fonseca 2007b).

Capítulo 2

Redundância e neutralidade em sistemas evolutivos naturais e artificiais

2.1 Introdução

Antes da teoria da selecção natural, proposta por Darwin em 1859, considerava-se que as formas vivas que se encontram na natureza teriam permanecido imutáveis desde a sua criação. A teoria de Darwin veio mudar de forma definitiva o pensamento científico e filosófico, ao afirmar que certas espécies podem descender umas de outras, com modificações que se produziram no decurso de grandes períodos de tempo. Após a teoria da selecção natural, surgiu a teoria neutral da evolução molecular, proposta pelo cientista japonês Kimura, que considera que uma fracção elevada de mutações é neutral e que a acumulação de mutações neutrais fornece, eventualmente, novos caminhos para a evolução.

A teoria neutral da evolução molecular serviu de inspiração a várias re-

representações redundantes e neutrais propostas na literatura. O facto da representação utilizada na implementação dos AEs ser um dos principais factores de sucesso nestes algoritmos (Michalewicz 1992, Mitchell 1996, Goldberg 1989), justifica o estudo e a análise da redundância ao nível do código genético e dos diferentes mapeamentos e características das representações utilizadas em CE.

Este capítulo começa por apresentar a existência de redundância e de neutralidade no código genético na secção 2.2, bem como as teorias evolutivas. A forma como a redundância e a neutralidade têm vindo a ser utilizadas em computação evolutiva e a indicação de algumas recomendações para o desenvolvimento de representações são apresentadas na secção 2.3. A noção de representação redundante, a métrica utilizada nas representações binárias a propôr e alguns exemplos de mapeamentos que foram utilizados na literatura, são estudados na secção 2.4. As propriedades que são indicadas na literatura para caracterizar as representações são indicadas na secção 2.5. Por último, são apresentadas algumas considerações finais, onde são explicitadas as diferenças entre as famílias de representações que se propõem nesta dissertação e as formas de manifestação da redundância e de neutralidade no código genético.

2.2 Redundância no código genético

Nos seres vivos a célula é a unidade estrutural e fisiológica. A auto-conservação, a auto-regulação e a auto-reprodução são mecanismos essenciais ao controlo da actividade celular e à herança da informação biológica. A célula pode ser vista como uma fábrica molecular, capaz de velar pela sua própria manutenção e fabricar os seus próprios produtos e operadores. O

corpo de qualquer ser vivo é constituído por células que contém instruções que lhes permitem produzir as substâncias essenciais de que precisam, as proteínas. Dentro do núcleo da célula, existe uma molécula chamada ADN (ácido desoxirribonucleico), que tem a forma de uma dupla hélice, constituída por uma sequência linear de grupos químicos chamados nucleótidos (de Rosnay 1989, Jones & Loon 1996).

No ADN, cada nucleótido é composto de desoxirribose, fosfato e uma base orgânica: Adenina (A), Guanina (G), Citosina (C) ou Timina (T). Estes quatro nucleótidos emparelham-se na dupla hélice do ADN de uma forma específica: a Adenina liga-se à Timina e a Citosina liga-se à Guanina.

Para produzir as proteínas, a célula utiliza blocos do ADN constituídos por sequências de 3 nucleótidos consecutivos, designados por tripletos, a fim de que ocorra a síntese de um aminoácido. Existem 20 aminoácidos diferentes, como a Leucina, Tirosina, Histidina, entre outros. Os aminoácidos são os blocos constituintes das proteínas. Embora estes aminoácidos existam na célula, é necessário um outro ácido, o ARN (ácido ribonucleico), para produzir as proteínas. Basicamente, o ARN difere do ADN em três aspectos: é constituído por uma só cadeia, é composto por ribose em vez de desoxirribose e apresenta o nucleótido Uracila (U) em vez da Timina, mas que também se liga à Adenina.

O ARN pode ocorrer em diversas formas: ARN mensageiro (ARNm), ARN ribossómico (ARNr) e ARN de transferência (ARNt). Por um processo denominado de transcrição, o ARNm transporta as instruções do ADN para o ARNr (ribossomas), que existem fora do núcleo da célula. Os ribossomas trabalham em cadeia, lendo sucessivamente a mensagem transportada pelo ARNm e fabricando a proteína, aminoácido por aminoácido. É o ARNt que garante o posicionamento de cada um dos aminoácidos na cadeia de proteína

em crescimento. Os aminoácidos, em determinado número e alinhados numa sequência correcta, permitem produzir determinada proteína. É a ordem pela qual se dispõem os aminoácidos que confere a cada proteína as suas características próprias. A hemoglobina, a queratina e a caseína são exemplos de proteínas.

No processo de transcrição, os tripletos de ADN são convertidos em codões do ARN. Estes codões são, à semelhança dos tripletos, conjuntos de três nucleótidos de uma cadeia de ARN. Como existem mais combinações de nucleótidos do que aminoácidos formados, como se pode verificar na tabela do código genético apresentada na tabela 2.1, o código genético é considerado redundante. Este fenómeno é designado por degenerescência do código genético. Por exemplo, os codões UCU, UCC, UCA e UCG codificam todos o mesmo aminoácido, a Serina. O código genético é redundante, mas não é ambíguo, ou seja, cada codão codifica um único aminoácido.

De notar que existem codões considerados de finalização, que indicam à célula que a sequência de aminoácidos destinada a determinar uma proteína acaba nesse ponto. Por outro lado, apenas os aminoácidos Triptófano e a Metionina (codão de iniciação que indica que a sequência de aminoácidos começa a ser codificada nesse ponto) são codificados por um único codão, os outros dezoito aminoácidos são codificados por vários codões.

A mudança mais pequena que pode ocorrer num codão é a mudança de um nucleótido por outro, ou seja, uma mutação num ponto. Nesta perspectiva dois codões são considerados vizinhos se diferirem num único nucleótido, ou seja, numa única base. Cada codão pode ter nove vizinhos, uma vez que cada base do codão pode ser alterada para uma das três restantes bases. Considerando, por exemplo, o codão ACG, uma mutação na primeira base origina CCG, GCG e UCG, que na tabela ocupam a mesma posição relativa

Tabela 2.1: Tabela do código genético

1	2	U	C	A	G	3
U		Fenilalanina (phe)	Serina (ser)	Tirosina (tyr)	Cisteína (cys)	U
U		Fenilalanina (phe)	Serina (ser)	Tirosina (tyr)	Cisteína (cys)	C
U		Leucina (leu)	Serina (ser)	STOP	STOP	A
U		Leucina (leu)	Serina (ser)	STOP	Triptófano (trp)	G
C		Leucina (leu)	Prolina (pro)	Histidina (his)	Arginina (arg)	U
C		Leucina (leu)	Prolina (pro)	Histidina (his)	Arginina (arg)	C
C		Leucina (leu)	Prolina (pro)	Glutamina (gln)	Arginina (arg)	A
C		Leucina (leu)	Prolina (pro)	Glutamina (gln)	Arginina (arg)	G
A		Isoleucina (ile)	Treonina (thr)	Asparagina (asn)	Serina (ser)	U
A		Isoleucina (ile)	Treonina (thr)	Asparagina (asn)	Serina (ser)	C
A		Isoleucina (ile)	Treonina (thr)	Lisina (lys)	Arginina (arg)	A
A		Metionina (met)	Treonina (thr)	Lisina (lys)	Arginina (arg)	G
G		Valina (val)	Alanina (ala)	Ácido Aspártico (asp)	Glicina (gly)	U
G		Valina (val)	Alanina (ala)	Ácido Aspártico (asp)	Glicina (gly)	C
G		Valina (val)	Alanina (ala)	Ácido Glutâmico (glu)	Glicina (gly)	A
G		Valina (val)	Alanina (ala)	Ácido Glutâmico (glu)	Glicina (gly)	G

que o codão ACG, na última linha de cada uma das restantes caixas da mesma coluna. Uma mutação na segunda base origina AAG, AGG e AUG, que aparece nas outras colunas da mesma linha. Por último, uma mutação na terceira base origina ACA, ACC e ACU, que estão na mesma caixa que ACG.

Devido à degenerescência do código genético, podem ocorrer mutações que não provocam qualquer mudança nas proteínas produzidas, visto que a alteração de uma base num codão nem sempre acarretar uma mudança do aminoácido codificado. Estas mutações designam-se por mutações neutrais.

Olhando para a tabela, é possível verificar que mutações na primeira base são ocasionalmente neutrais, como por exemplo, UUA e CUA, que codificam a Leucina. Mutações na segunda base só são neutrais num caso, UAA e UGA, que codificam ambos para um codão de finalização. Pelo contrário, mutações na terceira base são quase sempre neutrais. Portanto, os mesmos aminoácidos tendem a aparecer juntos na mesma caixa, indicando que codões que codificam o mesmo aminoácido (designados de codões sinónimos) são semelhantes

e que a diferença incide frequentemente na terceira base (degenerescência da terceira posição). Estes codões sinónimos são também designados por ‘family boxes’ e são os que começam por UC, CU, CC, CG, AC, GU, GC e GG e correspondem a metade dos codões.

Enquanto que o código genético indica a forma como deve ser interpretada a linguagem química dos genes e como é traduzida na linguagem química das proteínas, um gene é um segmento de ADN com um determinado número de nucleótidos, com uma ordem própria, que codifica uma proteína e que determina a hereditariedade de determinada característica, ou de um grupo delas. Nas células humanas o ADN existe em 46 cromossomas, 23 herdados da mãe e 23 do pai. Os genes estão localizados nos cromossomas, onde ocupam uma posição ou locus, e as várias formas alternativas do mesmo gene, ocupando um dado locus num cromossoma, designam-se por alelos. Cada cromossoma tem milhares de genes e cada espécie tem um número específico de cromossomas. O conjunto de todos os genes de um organismo é designado por complemento cromossómico.

2.2.1 Teorias evolutivas

Um indivíduo pode ser visto como uma dualidade entre o seu código genético e as suas características ou traços comportamentais, fisiológicas e morfológicas. O genótipo corresponde à informação guardada nos cromossomas, permitindo descrever os indivíduos ao nível dos genes, enquanto o fenótipo descreve a aparência do indivíduo. Na reprodução e durante a recombinação dos indivíduos, os descendentes não herdam directamente as características fenotípicas dos seus ascendentes, mas sim a informação genotípica que permite expressar essas características fenotípicas. O mapeamento genótipo-fenótipo, ou representação, utiliza a informação genotípica para construir o

fenótipo.

Na natureza, o mapeamento genótipo-fenótipo é complexo. A influência de um único gene não se encontra resumida a um único tipo de célula ou órgão, mas pode afectar simultaneamente diversos traços fenotípicos. Este efeito designa-se por pleiotropia. O efeito inverso, em que uma única característica fenotípica é determinada pela interacção de vários genes designa-se por poligenia (Fogel 1997).

A teoria da selecção natural de Darwin considera que as mudanças (mutações) genéticas benéficas resultam do processo de selecção combinado com operadores de variação, como recombinação e mutação, que podem algumas vezes resultar em indivíduos mais aptos que, gradualmente, substituem indivíduos menos aptos na população ao longo de várias gerações.

Por outro lado, a teoria neutral da evolução molecular considera que a maior parte da variação encontrada em sequências de ADN e proteínas é neutral em relação à selecção, e que boa parte dos diferentes alelos para um mesmo locus possui o mesmo valor adaptativo. A maioria das mudanças evolutivas resultam da fixação de mutações neutrais, que não têm efeitos imediatos na aptidão de um organismo, ou seja, não é a fixação de mutações benéficas resultante da selecção natural, mas a fixação aleatória de mutações neutrais, a principal força da evolução molecular.

Kimura demonstrou matematicamente que era possível haver evolução por deriva genética na ausência de selecção natural, mesmo em grandes populações e por longas escalas de tempo. A deriva genética é um processo estocástico que actua sobre as populações, modificando a frequência dos alelos e a predominância de certas características na população. A teoria da selecção natural e a teoria neutral da evolução molecular são consideradas hoje como teorias que se complementam, embora aquando do surgimento da

teoria neutral, tenham sido consideradas antagônicas (Pena 2006).

2.2.2 Conformação do ARN

Existem vários níveis de organização do ARN: a estrutura primária ou sequência, a estrutura secundária ou forma e a estrutura terciária. Estas estruturas correspondem a formas diferentes de descrever a estrutura do ARN em diferentes níveis de ‘resolução’.

A conformação (‘folding’ em inglês) do ARN (van Nimwegen, Crutchfield & Huynen 1999, Mount 2004) permite, dada a estrutura primária, prever a estrutura secundária. Basicamente, a estrutura secundária do ARN evolui para uma conformação de energia mínima, onde a estrutura é estabilizada pelos conjuntos emparelhados (‘ladders’) e é desestabilizada pelas regiões não emparelhadas (‘loops’). A figura 2.1 apresenta uma estrutura primária e a respectiva estrutura secundária de ARN, onde se podem visualizar os ‘ladders’ e os ‘loops’.

Um número elevado de estruturas secundárias são possíveis para uma determinada estrutura primária. No entanto, uma ou poucas destas estruturas conseguem encontrar o nível mínimo de energia. Devido à dificuldade computacional em determinar a estrutura correcta, as estruturas secundárias são utilizadas quando se analisa a conformação.

Alterar um conjunto de nucleótidos numa determinada posição numa sequência pode provocar possibilidades de ligação entre os nucleótidos, podendo ocorrer uma mudança na estrutura secundária do ARN. No entanto, a consequência mais usual corresponde a que mutações em várias posições na sequência, não alterem a estrutura secundária do ARN (Fontana 2002).

Se a distância entre duas sequências corresponder ao número de mutações necessárias para converter uma sequência em outra, os vizinhos de uma

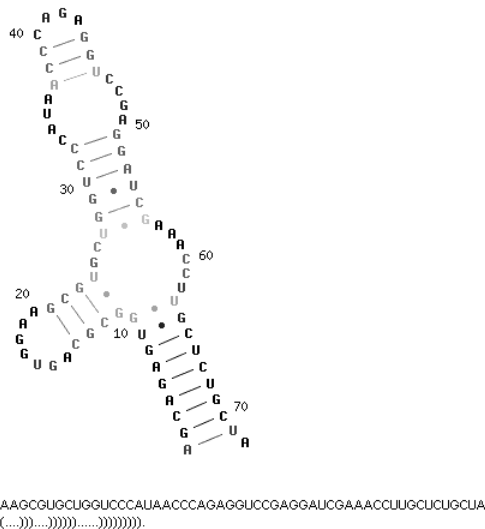


Figura 2.1: Uma estrutura primária do ARN e a respectiva estrutura secundária. Fonte:(Contrafold 2009)

sequência serão todas as sequências que estão separadas desta por uma mutação. Este espaço é multi-dimensional, uma vez que cada posição constitui uma dimensão, sendo possível saltar de uma sequência para outra, mantendo a mesma estrutura secundária.

Ao nível molecular, estas sequências de ARN que mapeiam para a mesma estrutura secundária formam redes neutrais. Neste contexto, uma rede neutra corresponde ao conjunto de sequências que formam a mesma estrutura secundária (Schuster, Fontana, Stadler & Hofacker 1994). Considerando as sequências de ARN como genótipos e as estruturas secundárias como fenótipos, uma rede neutra pode ser definida como o conjunto de todos os genótipos que mapeiam para o mesmo fenótipo (Barnett 2003) ou, mais especificamente, como uma rede conectada de genótipos que mapeiam para o mesmo fenótipo, onde dois genótipos estão conectados se diferirem numa

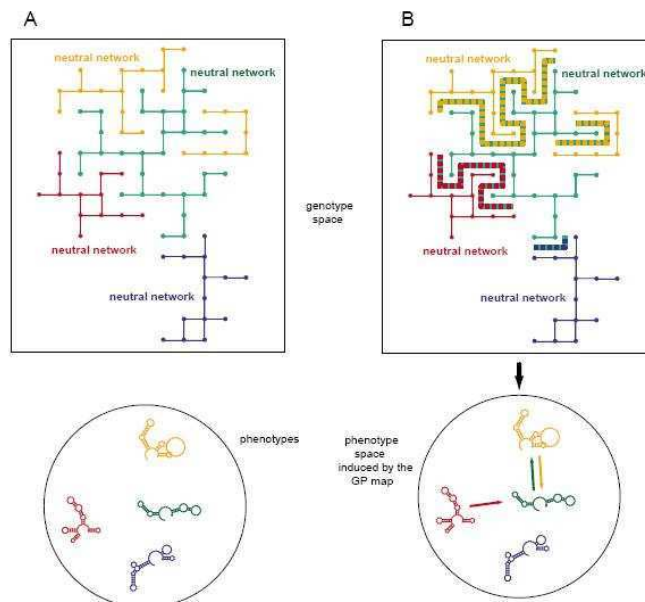


Figura 2.2: Esquema de redes neutrais no espaço das seqüências (A) e vizinhança fenotípica (B)

in W. Fontana, *The Topology of the Possible (Understanding Change: Models, Methodologies and Metaphors)*, 2005, Palgrave Macmillan. Reprodução autorizada por Palgrave Macmillan. Este material não pode ser copiado nem reproduzido sem a permissão de Palgrave Macmillan.

ou num pequeno número de mutações (Barnett 2001).

Um exemplo de um esquema de redes neutrais no espaço das seqüências é apresentado na figura 2.2 (Fontana 2005). Esta figura mostra apenas uma projecção a duas dimensões que não corresponde à realidade, uma vez que no espaço das seqüências, o número de dimensões pode ser na ordem das centenas. Considerando que a cor da rede neutral indica o fenótipo que cada rede representa (parte inferior da figura), consegue-se visualizar que os genótipos no canto superior direito da rede neutral verde não conseguem, por mutação, gerar indivíduos cujo fenótipo seja azul. No entanto, esses indivíduos conseguem gerar outros indivíduos pertencentes ao fenótipo verde, conseguindo atravessar toda a rede neutral até ser atingida a rede neutral que corresponde ao fenótipo azul.

Em Schuster & Fontana (1999) e Fontana (2005) o espaço das estruturas secundárias é definido como um espaço de formas, utilizando uma relação de acessibilidade entre duas formas, por exemplo, α e β , que depende da adjacência das redes neutrais correspondentes no espaço das sequências. A fronteira da rede neutral consiste em todas as sequências que diferem numa mutação da rede neutral em questão. A acessibilidade é uma medida da probabilidade de, estando em qualquer ponto da rede neutral que representa a forma (fenótipo) α , ser possível saltar para uma rede neutral que exhibe a forma β . Como a probabilidade de saltar de α para β pode não ser a mesma que a de saltar de β para α , é possível que a forma β seja considerada próxima (na vizinhança de) α , mas α não seja próxima de β . Na figura 2.2, a rede verde está próxima da rede vermelha, porque de um ponto qualquer da rede vermelha existe uma forte probabilidade de cair na verde. No entanto, a rede vermelha não está próxima da verde, porque partindo de um ponto qualquer da rede verde, existe uma fraca probabilidade de cair na vermelha, já que a superfície de contacto desta com a rede verde é reduzido. Este efeito acontece, porque as redes neutrais apresentam formas e tamanhos diferentes.

Alguns estudos revelam propriedades interessantes das estruturas secundárias de ARN com base nas sequências de nucleótidos (Schuster et al. 1994, Schuster 1997, Schuster & Fontana 1999). A este nível, detectou-se que uma única estrutura secundária é, em geral, representada por muitas sequências diferentes, indicando um mapeamento genótipo-fenótipo altamente redundante e neutral, uma vez que mutações em posições diferentes nas sequências não afectam a estrutura secundária. Estes autores concluíram que o espaço das sequências é ‘percolado’ por redes neutrais compridas que mapeiam para estruturas secundárias idênticas e que existem muitas sequências diferentes que conformam num pequeno número de estruturas (es-

truturas comuns) e poucas sequências que conformam num número elevado de estruturas (estruturas raras).

Outros estudos realizados por Huynen (1996) e em (Huynen 1996, Huynen, Stadler & Fontana 1996) sobre os efeitos da redundância no contexto da conformação do ARN concluíram que a existência de redes neutrais no espaço das sequências permite encontrar um número elevado de novas estruturas secundárias em cada passo.

As redes neutrais e as estruturas no contexto da conformação do ARN, apresentam diferenças substanciais em relação às existentes na família de representações neutrais propostas no capítulo 4. Uma das diferenças prende-se com a existência de estruturas de ARN comuns e raras, indicando que algumas estruturas estão sobre-representadas e outras sub-representadas, enquanto na família de representações neutrais esta distribuição é uniforme. Por outro lado, a nível molecular, as redes neutrais podem estar desconectadas e terem tamanhos e formas diferentes, enquanto na família que se propõe, para cada representação, todos os fenótipos apresentam redes neutrais com a mesma forma e tamanho.

De realçar que, o facto das redes neutrais e das estruturas a nível molecular já serem o resultado da evolução ao longo de milhões de anos, indicia que o próprio código genético é o resultado de um processo de evolução e portanto, de aperfeiçoamento.

2.3 Redundância em computação evolutiva

Existem diferentes opiniões sobre a influência das representações redundantes no desempenho dos AEs. Representações redundantes diversas foram utilizadas e, com algumas destas, o desempenho dos AEs melhorou (Rothlauf 2006),

enquanto com outras piorou (Ronald, Asenstorfer & Vincent 1995). A perda da diversidade da população, o aumento do espaço de procura ao nível do genótipo e o facto de diferentes genótipos que descodificam para o mesmo fenótipo competirem entre si, são algumas das razões apontadas para a deterioração do desempenho. Pelo contrário, o aumento da diversidade, que permite atrasar a convergência prematura, é uma das razões apontadas para o aumento do desempenho dos AEs.

As investigações levadas a cabo por Rothlauf (2006) indicaram que a utilização de representações redundantes sinónimas (em que os genótipos que descodificam para o mesmo fenótipo estão próximos uns dos outros no espaço genotípico) e uniformes (todos os fenótipos são representados pelo mesmo número de genótipos) não altera o comportamento dos AEs. Neste contexto, só a sobre-representação da solução óptima permitirá que o desempenho dos AEs aumente. Este autor também refere que a conectividade entre fenótipos não aumenta utilizando representações sinónimas. No entanto, esta conclusão não é consistente com os resultados obtidos no capítulo 4, onde se apresentam representações sinónimas que permite o aumento da conectividade entre fenótipos, quando comparadas com a representação não redundante.

Outros estudos centram-se nas características da procura, como a alcançabilidade dos fenótipos, o potencial evolutivo das populações e a conectividade dos espaços de procura. Alguns autores sugerem que a utilização de representações altamente redundantes permite melhorar a confiabilidade da procura, a aptidão alcançada, a capacidade de lidar com ambientes mutáveis e a robustez a elevadas taxas de mutação (Shipman, Shackleton, Ebner & Watson 2000, Shackleton, Shipman & Ebner 2000, Ebner, Langguth, Albert, Shackleton & Shipman 2001, Ebner, Shackleton & Shipman 2001). Estes autores utilizaram vários mapeamentos genótipo-fenótipo altamente redun-

dantes, que serão apresentados na subsecção 2.4.3, e concluíram que a adição de redundância é vantajosa porque, permite aumentar a conectividade entre os fenótipos, e os indivíduos são capazes de manter os valores de aptidão mesmo na presença de elevadas taxas de mutação, devido a uma elevada fracção das mutações serem neutrais.

Pelo contrário, Knowles & Watson (2002) utilizaram vários problemas de optimização, entre os quais, o H-IFF (Hierarchical If and Only If), o MAX-SAT (Maximum Satisfiability problem) e o problema das paisagens de aptidão NK (NK fitness landscapes) e concluíram que a adição de redundância aleatória não revelou ser útil na optimização.

Por outro lado, Harvey (1997a) (1997b) considerou que nem todos os tipos de representações redundantes são úteis. A evolução artificial foi discutida, nestes estudos, no contexto de uma experiência envolvendo uma configuração em hardware para um chip de silício utilizando a tecnologia FPGA (Field Programmable Gate Array). Foi utilizado um algoritmo genético denominado SAGA (Species Adaptation Genetic Algorithms) para otimizar as funções e o padrão de conexões das células com vista ao desempenho pelo circuito de uma determinada tarefa. Este autor considerou que a convergência prematura da população pode ser evitada utilizando uma representação redundante que apresente redes neutrais no espaço dos genótipos. Estas redes neutrais podem ser úteis se forem utilizadas para escapar dos óptimos locais, desde que seja num intervalo de tempo muito inferior ao da procura aleatória.

Harvey distinguiu entre o ‘junk DNA’, que poderá ser ‘lixo inútil’ se, mesmo apresentando redes neutrais, não for utilizado para determinar o fenótipo, do ‘lixo potencialmente útil’ que permite criar redes neutrais que transformem o potencial evolutivo da população. De facto, as redes neutrais podem transformar uma paisagem de aptidão com muitos óptimos locais

numa paisagem com apenas o óptimo global, como se poderá verificar na subsecção 5.2.2, onde a família de representações neutrais a propor nesta dissertação, consegue transformar paisagens de aptidão NK com rugosidade elevada, que apresentam muitos óptimos locais, em paisagens de aptidão com poucos óptimos locais ou com apenas o óptimo global.

Para poder controlar a neutralidade em paisagens de aptidão NK , foram propostas as paisagens adaptativas NKp (Barnett 1998, Barnett 2003), que correspondem a paisagens onde não é apenas a rugosidade (modalidade) da paisagem que é ajustável, através do parâmetro K , mas também a neutralidade utilizando o parâmetro p . Este autor concluiu que, para a classe de paisagens de aptidão que apresentam neutralidade e que podem aparecer em problemas de optimização reais, o operador genético a utilizar deverá ser a mutação e não a recombinação e o processo de procura evolutiva deverá ser uma população de um indivíduo, ou seja, um ‘hill-climber’ estocástico, em vez de uma população de vários indivíduos. Nesta dissertação serão estes o operador genético e o processo de procura a utilizar.

Em Yu & Miller (2001) foi utilizado um mecanismo de medida da neutralidade durante a evolução, tendo os autores concluído que a neutralidade melhora o potencial evolutivo. O efeito de ‘buffer’ da neutralidade protege o genótipo de mutações destrutivas durante o estágio final da evolução. Em problemas de ‘agulha no palheiro’, Yu & Miller (2002) demonstrou que na maioria dos casos, a neutralidade ajuda os AEs a encontrar a solução e nas situações onde a neutralidade não melhora o desempenho, pelo menos, não o prejudica.

Smith, Husband & O’Shea (2001a) (2001b) observaram o efeito da presença de redes neutrais no processo evolutivo, tendo concluído que a neutralidade afecta a dinâmica da população durante o processo de evolução, mas

também que esta não fornece necessariamente qualquer vantagem.

Recentemente, a relação entre o espaço das soluções (espaço fenotípico) e o espaço da procura (espaço genotípico) na presença de neutralidade Bit-Wise, foi analisada em (Galván-López & Poli 2006, Galván-López & Poli 2007, Poli & Galván-López 2007). Nestes estudos, a neutralidade é introduzida por um mapeamento genótipo-fenótipo em que cada bit do fenótipo é obtido pela transformação de um grupo de bits do genótipo, utilizando uma função de codificação. Os seus autores analisaram, também, a dinâmica da população, para perceber como a movimentação da população no espaço da procura é afectada pela presença da neutralidade e em que circunstâncias a neutralidade pode melhorar o desempenho do processo evolutivo. Estudaram a forma como taxas de mutação ao nível do fenótipo sofrem alterações em função das taxas de mutação ao nível do genótipo. Por último, concluíram que o desempenho dos algoritmos genéticos pode ser alterado radicalmente utilizando diferentes tipos de neutralidade e diferentes parâmetros de procura. A utilização desta representação foi estudada por Friedrich & Neumann (2008) de um ponto de vista teórico, tendo chegado à conclusão que este mecanismo só melhora o desempenho dos AEs baseados na mutação, se não for utilizado o mesmo número de bits do genótipo que influenciam cada bit do fenótipo.

2.3.1 Recomendações para o desenvolvimento e utilização de representações

A representação utilizada influencia o desempenho dos algoritmos genéticos (Goldberg 1989, Liepins & Vose 1990) e permite alterar a dificuldade do problema a ser optimizado. Não existe uma formulação teórica da influência da representação no desempenho dos AEs. Em virtude da importância deste assunto, algumas recomendações foram apresentadas para o desenvolvimento

e a utilização de representações (Ronald 1997):

1. As representações devem utilizar um processo de mapeamento genótipo-fenótipo apropriado, se um mapeamento simples para o fenótipo não for possível;
2. As representações que incluam apenas soluções viáveis devem ser preferíveis a representações que possam originar soluções inviáveis;
3. O problema deve ser representado no nível correcto de abstracção;
4. As representações devem minimizar a epistasia (interacção entre os genes);
5. Formas isomórficas, onde o fenótipo de um indivíduo é codificado por mais do que um genótipo, não devem ser utilizadas.

Estas representações parecem incentivar a utilização de representações não redundantes e, na opinião de Rothlauf (2006), são tão gerais que não se tornam úteis na concepção e na avaliação de representações.

Outro autor, Palmer (1994), analisou as propriedades de representações em árvore, mas que podem ser consideradas em outros tipos de representações. As recomendações foram:

1. Uma representação deve contemplar todos os fenótipos possíveis;
2. Uma representação deve considerar, de forma equitativa, todos os fenótipos;
3. Uma representação deve codificar apenas soluções factíveis;
4. A descodificação do genótipo para obter o fenótipo deve ser um processo simples;

5. Um representação deve exibir localidade, ou seja, distâncias pequenas no espaço dos genótipos devem resultar em distâncias pequenas no espaço dos fenótipos.

De salientar que, algumas destas recomendações são contempladas na família de representações neutrais que se propõe nesta dissertação. Por exemplo, a codificação de todos os fenótipos possíveis, onde todos os fenótipos estão representados de forma equitativa (uniformidade), é considerada. No entanto, também viola outras, porque apresenta redundância e nem todas as representações exibem localidade.

2.4 Representações redundantes

Uma representação é não redundante quando um fenótipo é representado por apenas um genótipo e vice-versa. Quando se utiliza uma representação binária, onde os genótipos e os fenótipos são codificados como *strings* com comprimento ℓ , o mapeamento genótipo-fenótipo f_g de uma representação não redundante é definido como:

$$f_g : \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell$$

onde:

$$\forall (x_0, \dots, x_{\ell-1}) \in \{0, 1\}^\ell, \forall (y_0, \dots, y_{\ell-1}) \in \{0, 1\}^\ell,$$

$$(x_0, \dots, x_{\ell-1}) \neq (y_0, \dots, y_{\ell-1}) \Rightarrow f_g(x_0, \dots, x_{\ell-1}) \neq f_g(y_0, \dots, y_{\ell-1})$$

Segundo Barreau (2002) existem duas formas de produzir um código redundante. Uma é reduzir o espaço de procura dos fenótipos, a outra é aumentar o espaço de procura dos genótipos. A primeira opção reduz o poder de expressão da representação, pois existirão fenótipos que não podem ser

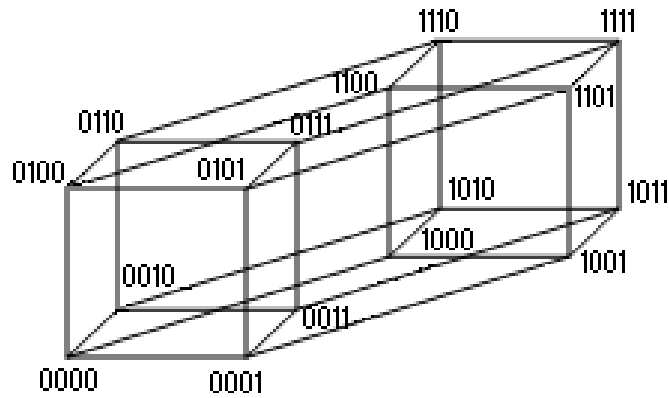


Figura 2.3: Hipergrafo representando o espaço dos genótipos com $\ell = 4$

representados. A segunda opção permite introduzir redundância de forma controlada. As representações redundantes que se propõem nesta dissertação são obtidas através do aumento do espaço de procura dos genótipos.

Um exemplo de uma representação gráfica do espaço dos genótipos/fenótipos com $\ell = 3$ é o cubo de Hamming, que permite representar 8 genótipos/fenótipos. Quando se acrescenta mais um bit ao genótipo, mantendo o mesmo número de bits no fenótipo, o espaço de procura dos genótipos duplica, passando a ser representado pelo hipergrafo da figura 2.3. Se a representação for uniforme, cada fenótipo será representado por dois genótipos distintos.

Para utilizar uma representação redundante com vista à resolução de um problema de otimização, é necessário definir um mapeamento genótipo-fenótipo f_g , para além do mapeamento fenótipo-aptidão, f_f . Para cada espaço de procura utilizado pelos mapeamentos, uma métrica tem de ser definida. Como a métrica de Hamming é a mais comum, e é a utilizada nesta dissertação, será apresentada de seguida.

2.4.1 Métrica de Hamming

Quando se utilizam algoritmos de procura, uma métrica tem de ser definida para cada um dos espaços de procura existentes. Baseado numa métrica, a distância $d(u, v)$ entre dois indivíduos u e v , pertencentes a um espaço de procura Φ , mede quão similares são os dois indivíduos (Rothlauf 2006). Quanto maior a distância, maior é a diferença entre os dois indivíduos. Em geral, podem ser definidas métricas diferentes para o mesmo espaço de procura, onde métricas diferentes resultam em distâncias diferentes e, portanto, medidas diferentes da similaridade das soluções.

Para compreender a métrica de Hamming (Hamming 1980), é necessário definir a noção de distância de Hamming e de peso.

Definição 2.1 (Peso de um vector binário). *O peso $w(v)$ de um vector binário v corresponde ao número de ‘1’ que estão presentes em v .*

Definição 2.2 (Distância de Hamming). *A distância de Hamming d entre dois vectores binários v e u com o mesmo comprimento, é igual ao número de elementos diferentes entre eles, i.e.,*

$$\forall v, u \in \{0, 1\}^\ell \quad d(v, u) = w(v + u) \quad (2.1)$$

onde $\{0, 1\}^\ell$ é o conjunto de todos os ℓ -tuplos binários e a adição corresponde à adição módulo-2.

Enquanto o peso de um vector corresponde ao número de coordenadas não nulas desse vector, a distância de Hamming entre dois vectores binários corresponde ao peso da adição módulo-2 dos dois vectores. Sendo u , v e z vectores binários com o mesmo comprimento, a distância de Hamming satisfaz as seguintes propriedades:

1. $d(u, u) = 0 \wedge d(u, v) > 0$ se $u \neq v$ (*positividade*)
2. $d(u, v) = d(v, u)$ (*simetria*)
3. $d(u, v) + d(v, z) \geq d(u, z)$ (*desigualdade triangular*)

2.4.2 Mapeamento genótipo-fenótipo e mapeamento fenótipo-aptidão

Considerando que Φ_g corresponde ao espaço dos genótipos e Φ_f ao espaço dos fenótipos, um mapeamento genótipo-fenótipo f_g determina quais os fenótipos que resultam da descodificação de determinados genótipos e pode ser definido como:

$$f_g(x_g) : \Phi_g \rightarrow \Phi_f$$

$$x_g \mapsto x_f$$

Sendo x_g um genótipo, o fenótipo correspondente é $x_f = f_g(x_g)$. O mapeamento genótipo-fenótipo f_g apenas permite determinar quais os genótipos que representam determinado fenótipo, não dando qualquer informação sobre a similaridade entre as soluções. O mapeamento fenótipo-aptidão atribui um valor de aptidão $f_f(x_f)$ a cada fenótipo $x_f \in \Phi_f$:

$$f_f(x_f) : \Phi_f \rightarrow \mathbb{R}$$

As métricas utilizadas podem ser diferentes no espaço de procura genotípico e no espaço de procura fenotípico. A métrica do espaço de procura dos fenótipos Φ_f é determinada pelo problema a ser resolvido, ou seja, depende da natureza das variáveis de decisão. Por exemplo, se o problema a otimizar for a paisagem de aptidão NK , a métrica de Hamming pode ser utilizada no espaço de procura fenotípico.

Pelo contrário, a métrica a considerar no espaço dos genótipos depende não só do espaço, mas também do operador de procura que é utilizado sobre os genótipos. Se, por exemplo, o operador utilizado for a mutação de um bit, a métrica de Hamming será a mais apropriada no espaço genotípico. O operador de procura e a métrica utilizada no espaço dos genótipos determinam-se mutuamente, não podendo ser escolhidos de forma independente um do outro.

Considerando espaços binários de genótipos com comprimento ℓ e de fenótipos com comprimento k , a cardinalidade do espaço genotípico é $|\Phi_g| = 2^\ell$, enquanto a cardinalidade do espaço fenotípico é $|\Phi_f| = 2^k$. Uma medida de redundância que é possível definir é a razão entre o número de genótipos e o número de fenótipos (Ebner, Langguth, Albert, Shackleton & Shipman 2001, Ebner, Shackleton & Shipman 2001). Por exemplo, para $|\Phi_g| = 2^\ell$ e $|\Phi_f| = 2^k$ e sendo a representação uniforme, a redundância é $2^{\ell-k} : 1$ que corresponde ao número de genótipos que representam cada fenótipo.

O termo neutralidade pode ser utilizado sob duas perspectivas (Lehre & Haddow 2005). Uma considera que a neutralidade ocorre quando dois fenótipos diferentes têm a mesma aptidão; a outra quando genótipos diferentes mapeiam para o mesmo fenótipo. Portanto, o termo rede neutral, na literatura, tanto pode ser utilizado para descrever o conjunto de genótipos com a mesma aptidão, como o conjunto de genótipos que mapeiam para o mesmo fenótipo. Pela primeira perspectiva:

Definição 2.3 (Genótipos neutrais). *Dois genótipos x_{g_1} e x_{g_2} com comprimento ℓ , onde $x_{g_1} \in \{0, 1\}^\ell$, $x_{g_2} \in \{0, 1\}^\ell$, são neutrais se e só se $f_f(f_g(x_{g_2})) = f_f(f_g(x_{g_1}))$, onde $f_f(f_g(x_{g_2}))$ e $f_f(f_g(x_{g_1}))$ correspondem à aptidão dos genótipos x_{g_2} e x_{g_1} , respectivamente.*

A outra perspectiva, utilizada nas representações neutrais que se propõem, considera que:

Definição 2.4 (Genótipos neutrais). *Dois genótipos x_{g_1} e x_{g_2} com comprimento ℓ , onde $x_{g_1} \in \{0, 1\}^\ell$, $x_{g_2} \in \{0, 1\}^\ell$, são neutrais se e só se $x_{f_2} = x_{f_1}$, onde $x_{f_2} = f_g(x_{g_2})$ e $x_{f_1} = f_g(x_{g_1})$.*

A seguir são apresentados alguns exemplos de mapeamentos genótipo-fenótipo binários que foram propostos na literatura.

2.4.3 Exemplos de mapeamentos genótipo-fenótipo binários

Na literatura existe uma variedade de mapeamentos genótipo-fenótipo que exibem diferentes graus de redundância. A seguir, são apresentados alguns desses mapeamentos, onde os espaços de procura genotípico e fenotípico são binários e se considera um espaço fenotípico com cardinalidade $|\Phi_f| = 2^{16}$.

Mapeamento aleatório estático

No mapeamento aleatório estático (Static random mapping), proposto por Shackleton et al. (2000), o genótipo tem comprimento 30 e a redundância é de $2^{30-16} = 2^{14} : 1$. O mapeamento é inicializado aleatoriamente e continua estático, onde o mesmo genótipo mapeia sempre para o fenótipo correspondente.

Mapeamento votação trivial

O mapeamento votação trivial (Trivial voting mapping), da autoria de Shackleton et al. (2000), apresenta genótipos com comprimento 48, sendo a redundância igual a $2^{48-16} = 2^{32} : 1$. Cada bit do fenótipo é determinado por

um conjunto de 3 bits do genótipo, onde os 3 primeiros bits do genótipo determinam o primeiro bit do fenótipo e assim sucessivamente. Os conjuntos de 3 bits do genótipo não partilham bits entre si, não existindo sobreposição entre os conjuntos, fazendo com que cada bit do genótipo afecte apenas um bit do fenótipo. Neste caso, existe o efeito de poligenia, mas não existe o efeito de pleiotropia. Por último, um bit do fenótipo é considerado a 1 se a maioria dos 3 bits do genótipo estiver a 1, caso contrário é considerado a 0.

Mapeamento votação standard

No mapeamento votação standard (Standard voting mapping), também proposto por Shackleton et al. (2000), o genótipo tem comprimento 32 e a redundância é $2^{32-16} = 2^{16} : 1$. Neste mapeamento cada bit do fenótipo é determinado pela interacção dos bits do genótipo com os quais está conectado e para além da poligenia, existe também o efeito de pleiotropia, permitindo este último que vários bits do fenótipo possam mudar simultaneamente com uma única mutação no genótipo. Neste caso, existe uma sobreposição considerável entre os vários conjuntos de bits do genótipo. As conexões entre os bits do genótipo e do fenótipo são realizadas da seguinte forma: um número ímpar e constante de bits do genótipo, neste caso 21, são escolhidos aleatoriamente para votar em determinado bit do fenótipo. O valor do voto de cada bit é gerado aleatoriamente, podendo ser favorável ao bit do fenótipo correspondente ficar a 1 ou desfavorável. Os votos favoráveis são considerados positivos, os desfavoráveis negativos. Quando a soma dos votos dos bits do genótipo que contribuem para um determinado bit do fenótipo é positiva, o bit do fenótipo é activado a 1, caso contrário é 0.

Mapeamento autómato celular

Um autómato celular (AC) (Kauffman 1995) é um sistema dinâmico nos quais o espaço e o tempo são discretos. Consiste numa grelha de células onde cada célula pode estar em um número finito de estados que variam de acordo com regras determinísticas. Quando o AC é booleano, cada célula pode ter um de dois estados: ‘on’ ou ‘off’. A grelha pode ser em qualquer número finito de dimensões e o estado da célula no tempo t é uma função do estado no tempo $t - 1$ de um número finito de células na sua vizinhança, podendo a própria célula estar incluída. As conexões são estabelecidas com as células que pertencem à vizinhança local. Por exemplo, se o autómato tiver 2 dimensões, a vizinhança local pode consistir nas células da esquerda ou da direita, as células acima ou abaixo, ou outras. As células evoluem segundo regras de actualização baseadas nos valores das suas células vizinhas. O autómato é não uniforme quando as tabelas de regras de actualização são diferentes para as várias células, caso contrário é uniforme. De cada vez que as regras são aplicadas à grelha completa, uma nova iteração é produzida.

O AC proposto por Shackleton et al. (2000) e analisado em (Shipman et al. 2000, Ebner, Langguth, Albert, Shackleton & Shipman 2001, Ebner, Shackleton & Shipman 2001), é booleano, unidimensional, não uniforme e com vizinhança adjacente igual a 2 (célula da esquerda e da direita).

A tabela de regras para uma célula i e para as suas vizinhas $i - 1$ e $i + 1$, onde o estado das 3 células especifica um endereço da tabela de regras da célula i , permite determinar o estado da célula i no instante seguinte. Neste caso como são necessárias 2^3 entradas para cada célula, mais a própria célula, e tendo o fenótipo comprimento 16, o genótipo apresenta comprimento $(8 + 1) \times 16 = 144$ e a redundância é $2^{144-16} = 2^{128}$.

Para determinar o fenótipo para um dado genótipo, o autómato celular é

inicializado com os estados iniciais de cada célula que constam no genótipo e este é iterado um número constante de vezes. O estado final do autómato celular é interpretado como o fenótipo.

Mapeamento rede booleana aleatória

Uma rede booleana aleatória (RBA) (Kauffman 1995) é uma generalização do autómato celular. Enquanto o AC apresenta células e as conexões são realizadas com células geograficamente próximas, no RBA existem nós e as conexões são estabelecidas entre nós possivelmente distantes.

No mapeamento RBA proposto por Shackleton et al. (2000) e alvo de análise em (Shipman et al. 2000, Ebner, Langguth, Albert, Shackleton & Shipman 2001, Ebner, Shackleton & Shipman 2001), a conectividade é igual a 3, onde o estado seguinte de cada nó depende do estado de 3 nós (o próprio mais dois vizinhos remotos escolhidos aleatoriamente) dos $2^4 = 16$ nós que compõem o RBA. Neste mapeamento o genótipo tem comprimento $(1 + (3 \times 4) + 8) \times 16 = 336$, que corresponde ao estado inicial (1), aos endereços (4 bits para representar um espaço de 16) dos (3) nós, tabelas de regras de cada nó (8), para um fenótipo de comprimento (16). A redundância é $2^{336-16} = 2^{320} : 1$.

Tal como o AC, o RBA é executado um número fixo de iterações e o estado final da rede é interpretado como o fenótipo.

Os mapeamentos que a seguir se indicam, utilizam a neutralidade Bit-Wise, já referida na secção 2.3, onde cada fenótipo é obtido pela transformação de um grupo de bits do genótipo, neste caso n bits do genótipo para cada um dos bits do fenótipo, utilizando uma função de codificação (Galván-López & Poli 2006, Galván-López & Poli 2007, Poli & Galván-López 2007):

Mapeamento maioria

Para este mapeamento, o utilizador define um intervalo T , $0 \leq T \leq n$. Se em cada grupo de n bits do genótipo existirem T ou mais bits a 1, o bit correspondente no fenótipo é definido a 1, caso contrário é considerado 0. Para evitar o enviesamento do sistema, $T = n/2$ e n é um número ímpar, permitindo que 0s e 1s sejam tratados de forma equitativa.

Mapeamento paridade

Neste mapeamento, se o número de 1s em cada grupo de n bits do genótipo é par, o bit correspondente no fenótipo é definido a 1, caso contrário é definido a 0.

Mapeamento tabela de verdade

Neste caso, uma tabela de verdade é gerada e a saída de cada combinação é gerada aleatoriamente, sendo que metade das saídas da tabela são definidas a 1s e a outra metade a 0s. Os n bits do genótipo são as entradas da tabela e a sua saída é definida no bit correspondente do fenótipo.

Como é possível verificar, os mapeamentos estudados em (Shipman et al. 2000, Shackleton et al. 2000, Ebner, Langguth, Albert, Shackleton & Shipman 2001, Ebner, Shackleton & Shipman 2001) apresentam uma redundância que varia entre $2^{14} : 1$ e $2^{336-16} = 2^{320} : 1$, ou seja, variam entre 16 a 320 bits redundantes. Na família de representações neutrais propostas no capítulo 4, é possível definir representações com características interessantes, com um número reduzido de bits redundantes, 3 ou 4.

2.5 Propriedades das representações

Na literatura são apresentadas algumas propriedades que permitem caracterizar as representações. Algumas destas características são (Rothlauf 2006):

2.5.1 Uniformidade

Uma representação é uniforme quando todos os fenótipos são representados pelo mesmo número de genótipos.

2.5.2 Conectividade

Enquanto a vizinhança fenotípica de um fenótipo corresponde aos fenótipos que estão acessíveis a partir de um dado fenótipo por mutação de um bit dos respectivos genótipos, a conectividade corresponde ao número de fenótipos diferentes que constituem a vizinhança fenotípica de um fenótipo. Para uma representação binária não redundante, em que o fenótipo tem comprimento ℓ e em que os fenótipos acessíveis a partir de qualquer fenótipo são todos diferentes, a conectividade é ℓ . Uma representação binária redundante pode permitir o aumento da conectividade, se a representação for desenhada para tal.

2.5.3 Sinonímia

Uma representação é altamente sinónima se os genótipos que representam o mesmo fenótipo são semelhantes, ou seja, se estão próximos uns dos outros no espaço genotípico. A sinonímia mede quão similares são os genótipos que representam determinado fenótipo. Segundo Rothlauf (2006), se para todos os fenótipos, a soma das distâncias entre todos os genótipos que representam

determinado fenótipo é razoavelmente pequena, a representação é considerada altamente sinónima. A sinonímia de uma representação depende da métrica definida em Φ_g , que, por sua vez, depende do operador de procura utilizado, e de Φ_f , que depende das propriedades do problema. Uma medida da sinonímia é apresentada na expressão 2.2.

$$\sum_{x_f \in \Phi_f} \frac{1}{2} \left(\sum_{x_{g_1} \in \Phi_g^{x_f}} \sum_{x_{g_2} \in \Phi_g^{x_f}} d(x_{g_1}, x_{g_2}) \right), \quad x_{g_1} \neq x_{g_2} \quad (2.2)$$

onde $d(x_{g_1}, x_{g_2})$ corresponde à distância entre o genótipo x_{g_1} e o genótipo x_{g_2} e $\Phi_g^{x_f}$ denota o conjunto de genótipos que representam o fenótipo x_f . A fracção $\frac{1}{2}$ surge em 2.2 porque a distância de Hamming é simétrica.

2.5.4 Localidade

A localidade de uma representação descreve quão aproximadamente a vizinhança genotípica corresponde à vizinhança fenotípica. O objectivo é verificar se a vizinhança é preservada, ou seja, se para genótipos que são vizinhos no espaço genotípico, os respectivos fenótipos serão também vizinhos no espaço fenotípico. A localidade de uma representação é uma propriedade que permite caracterizar tanto representações redundantes como não redundantes. A localidade depende da representação utilizada f_g , da métrica definida em Φ_g e da métrica definida em Φ_f . f_g indica os genótipos que representam cada fenótipo, a métrica utilizada em Φ_g descreve a similaridade entre genótipos, enquanto a métrica utilizada em Φ_f descreve a similaridade entre fenótipos.

Rothlauf (2006) propõe a equação 2.3 para determinar a localidade d_m :

$$d_m = \sum_{d(x_{g_1}, x_{g_2})=d^g_{min}} \left| d(x_{f_1}, x_{f_2}) - d^f_{min} \right| \quad (2.3)$$

onde $d(x_{f_1}, x_{f_2})$ corresponde à distância entre os fenótipos x_{f_1} e x_{f_2} , $d(x_{g_1}, x_{g_2})$ descreve a distância entre os respectivos genótipos x_{g_1} e x_{g_2} , d^g_{min} é a distância mínima entre dois genótipos vizinhos e d^f_{min} indica a distância mínima entre dois fenótipos vizinhos. Quando a métrica de Hamming é utilizada no espaço fenotípico e no espaço genotípico, $d^g_{min} = 1$ e $d^f_{min} = 1$, respectivamente. Por outro lado, $d_m = 0$ indica que, para todos os genótipos que são vizinhos, os respectivos fenótipos também são vizinhos e a representação é perfeita quanto à localidade.

Uma representação tem localidade elevada se os vizinhos dos genótipos que representam cada fenótipo correspondem a fenótipos próximos do fenótipo em questão, segundo a métrica utilizada. Neste caso, o valor de d_m é reduzido e pequenas mudanças no espaço dos genótipos correspondem a pequenas mudanças no espaço dos fenótipos. Pelo contrário, quando d_m é elevado, a representação tem localidade baixa e pequenas mudanças no espaço dos genótipos correspondem a grandes mudanças no espaço dos fenótipos.

De realçar que a sinonímia e a uniformidade são propriedades que só estão definidas para representações redundantes. As restantes propriedades aplicam-se tanto a representações redundantes como não redundantes.

2.6 Considerações finais

São várias as diferenças que existirão entre as famílias de representações redundantes que se propõem nesta dissertação e a forma como a redundância e a neutralidade se manifestam no código genético. Enquanto no ADN, os nucleótidos $\{A, G, C, T\}$ ou os nucleótidos $\{A, G, C, U\}$ no ARN, constituem

o alfabeto utilizado no espaço das sequências, o alfabeto binário $\{0, 1\}$ será utilizado nas famílias de representações que se propõem.

Por outro lado, no ARN, a frequência das estruturas secundárias não é uniforme, existindo poucas estruturas comuns e muitas estruturas raras, enquanto as representações a propôr serão uniformes, por forma a garantir que não existe sub-representação ou sobre-representação de alguns fenótipos.

As outras diferenças estão relacionadas com as redes neutrais. Enquanto no espaço das sequências, as redes neutrais podem estar desconectadas e apresentam tamanhos (diâmetros) e formas (topologias) diferentes, na representação neutral a propôr, as redes estão conectadas e apresentam a mesma forma e o mesmo tamanho para todos os fenótipos. No entanto, devido às inúmeras representações neutrais que se poderão definir, cada representação neutral poderá exibir redes neutrais com um diâmetro e uma topologia específicos.

Como foi possível verificar, alguns mapeamentos genótipos-fenótipos propostos na literatura são altamente redundantes e apresentam estruturas aleatórias e sem uma base matemática clara. No próximo capítulo, será apresentada a formulação matemática dos códigos de controlo de erros, que será usada no capítulo 4 para implementar representações redundantes e neutrais de forma estruturada.

Capítulo 3

Códigos de controlo de erros

3.1 Introdução

A família de representações neutrais que se propõe nesta dissertação permite definir redes conectadas de genótipos que descodificam para o mesmo fenótipo, onde dois genótipos estão conectados se diferirem numa mutação. O mapeamento genótipo-fenótipo proposto baseia-se nos códigos de controlo de erros (Carlson 2002, Blahut 1990, Abrantes n.d.), que são utilizados principalmente na área das comunicações digitais.

Em comunicações e armazenamento, os erros podem acontecer em variadíssimas situações, durante a transmissão de dados num canal (linha telefónica, ligação via satélite, etc.) ou, por exemplo, na leitura e gravação de dados num dispositivo de armazenamento como o CD ou o DVD. Basicamente, existem dois tipos de técnicas de controlo de erros. Uma das técnicas, ARQ (Automatic Repeat Request), baseia-se na retransmissão dos blocos de dados onde tenham sido detectados erros de transmissão. Estes tipos de códigos são geralmente menos complexos e envolvem normalmente uma utilização menos eficiente do canal de transmissão. A outra técnica FEC

(Forward Error Correction), utilizada quando a retransmissão é impossível ou impraticável, baseia-se nos códigos de controlo de erros.

A ideia base dos códigos de controlo de erros é acrescentar redundância à mensagem a ser transmitida, de forma a permitir que o receptor seja capaz de detectar e, se possível, corrigir eventuais erros ocorridos durante a transmissão. Esta redundância é inserida de forma controlada para que a mensagem original possa ser recuperada a partir dos dados recebidos, se o número e tipo de erros estiver dentro dos limites para o qual o código foi desenhado. Os tipos de códigos normalmente utilizados agrupam-se em duas categorias: os códigos de bloco e os códigos convolucionais. Enquanto os códigos de bloco codificam um bloco de k símbolos de mensagem num bloco de palavras-código com n símbolos, formando um espaço vectorial, nos códigos convolucionais, a sequência de informação que se deseja codificar não está seccionada em blocos, mas espalhada ao longo da palavra-código. Cada categoria de código apresenta propriedades, técnicas de descodificação e modos de representação diferentes. Os códigos de bloco lineares serão os códigos utilizados nesta dissertação.

Na secção 3.2 são estudados os códigos de bloco lineares. Em particular, são apresentados os códigos de Hamming, utilizando as noções de matriz geradora, matriz de verificação de paridade, descodificação pela síndrome e a noção de ‘cosets’. Os códigos cíclicos são também estudados nesta secção, bem como as propriedades dos códigos e dos respectivos ‘cosets’. Por último, os códigos equivalentes são estudados na secção 3.3, enquanto um resumo do capítulo é apresentado na secção 3.4.

Tabela 3.1: Adição módulo-2

+	0	1
0	0	1
1	1	0

Tabela 3.2: Multiplicação módulo-2

*	0	1
0	0	0
1	0	1

3.2 Códigos de bloco lineares

Um código de bloco $C(n, k)$ definido sobre um alfabeto com q símbolos consiste num conjunto de palavras-código com comprimento fixo n , que resultam da codificação de k símbolos de dados de mensagem, onde são acrescentados $n - k$ símbolos de verificação e onde as palavras-código constituem um sub-espço vectorial de dimensão k . A taxa de código é $R_c = k/n$ e quando $q = 2$, o código é binário. Como as representações redundantes a propor são binárias, será dado destaque aos códigos binários. Os códigos lineares (Lin & Costello 1983) constituem uma importante família de códigos de bloco:

Definição 3.1 (Código de bloco linear). *Um código de bloco de comprimento n e com 2^k palavras-código é dito um código (n, k) linear se e só se as 2^k palavras-código formarem um sub-espço de dimensão k do espaço vectorial de todos os n -tuplos sobre $GF(2)$.*

$GF(2)$ corresponde ao corpo de Galois de ordem 2 formado pelos elementos $\{0, 1\}$, com aritmética módulo-2 que corresponde ao ou-exclusivo, tal como indicado na tabela 3.1, e a multiplicação corresponde à operação e-lógico, como apresentado na tabela 3.2.

De salientar que, devido à linearidade do código, é garantido que:

1. O vector zero é uma palavra-código;
2. A adição de duas palavras-código origina outra palavra-código.

Um dos códigos de bloco mais simples é o código de verificação de paridade que não tem capacidade para corrigir erros, mas permite detectar a sua ocorrência se esta se verificar em número ímpar. Neste código é adicionado um bit extra a cada bloco de dados de mensagem, de forma a que o número total de bits com valor 1 seja, por exemplo, par. Neste caso, se um erro simples ocorrer, a palavra recebida será inválida, e o receptor saberá que ocorreu um erro, embora não tenha informação suficiente para saber qual o bit errado e, conseqüentemente, para o corrigir. Neste código, se for recebida uma palavra com um número par de bits errados (dois, quatro, etc.), os erros não poderão ser detectados.

A separação entre as palavras-código pode ser medida pela distância de Hamming d , definida na secção 2.1. O número de erros que um código de bloco pode detectar e/ou corrigir depende da sua distância mínima d_{min} , tal como é definido em 3.1:

Definição 3.2 (Distância mínima de um código de bloco). *A distância mínima d_{min} de um código de bloco C corresponde à menor distância de Hamming entre quaisquer duas palavras-código do código, i.e.*

$$d_{min} = \min\{d(v, u) : v, u \in C, v \neq u\} \quad (3.1)$$

É fácil provar que a d_{min} de um código de bloco linear C é igual ao peso da palavra-código não nula com menor peso. Tendo em conta a definição 3.1 e sabendo que a distância de Hamming entre dois vectores binários corresponde

ao peso da adição módulo-2 dos mesmos, tal como indicado em 2.1, então:

$$d_{min} = \min\{w(v + u) : v, u \in C, v \neq u\} \quad (3.2)$$

Logo,

$$d_{min} = \min\{w(x) : x \in C, x \neq 0\} \quad (3.3)$$

A detecção de erros é sempre possível quando o número de erros numa palavra-código é menor do que d_{min} , de tal forma que a palavra recebida não corresponde a uma palavra-código. Quando o número de erros é igual ou excede d_{min} , a palavra com erros recebida pode corresponder a outra palavra-código e os erros poderão não ser detectados. Por isso, para detectar até ℓ erros por palavra, a distância mínima d_{min} deverá ser:

$$d_{min} \geq \ell + 1 \quad (3.4)$$

Por outro lado, para corrigir até t erros por palavra, d_{min} deverá ser:

$$d_{min} \geq 2t + 1 \quad (3.5)$$

Se um código corrige até t erros por palavra e detecta $\ell > t$ erros por palavra, d_{min} deverá ser:

$$d_{min} \geq t + \ell + 1 \quad (3.6)$$

A distância mínima de um código de blocos $C(n, k)$ é limitada superiormente pelo limite de ‘Singleton’:

$$d_{min} \leq n - k + 1 \quad (3.7)$$

Os códigos cíclicos, os códigos de verificação de paridade e os códigos de

repetição são alguns exemplos de códigos de bloco lineares. Os códigos de Hamming e os códigos cíclicos serviram de inspiração para a definição da representação neutra e, por isso, serão abordados neste capítulo.

3.2.1 Códigos de Hamming

Os códigos de Hamming podem ser definidos da seguinte forma (Carlson 2002):

Definição 3.3 (Código de Hamming). *Um código de Hamming é um código de bloco linear $C(n, k)$ com $n - k \geq 3$ bits redundantes (bits de paridade), $n = 2^{n-k} - 1$, onde, independentemente do número de bits de paridade, a distância mínima é sempre igual a 3, $d_{\min} = 3$, permitindo corrigir apenas um erro e detectar até dois erros.*

Exemplos de códigos de Hamming com 3, 4 e 5 bits redundantes são $C(7, 4)$, $C(15, 11)$ e $C(31, 26)$. A figura 3.1 mostra as palavras-código de um dos possíveis códigos $C(7, 4)$, que está na forma sistemática porque os bits de dados aparecem inalterados num bloco da palavra-código.

Matriz geradora

As palavras-código v de um código de Hamming $C(n, k)$, definido sobre o Corpo de Galois $GF(2)$, podem ser obtidas a partir dos dados u , utilizando uma matriz geradora G :

$$v = uG \tag{3.8}$$

onde G é uma matriz $k \times n$. Se a matriz geradora G estiver numa das formas apresentadas em 3.9 ou 3.10, o código de Hamming $C(n, k)$ está na forma

Matriz Geradora G Matriz de Verificação de Paridade H Síndrome s	Cosets
<p>$C(n,k)$ em $GF(2)$ $n=7$ $k=4$</p> <p>v – palavra-código u - mensagem</p> <p>$v=uG$</p> <p>$G=[I_k:P_{k \times (n-k)}]$</p> $G = \left[\begin{array}{cccc cccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array} \right]$ <p>$H=[(P_{k \times (n-k)})^T : I_{n-k}]$</p> $H = \left[\begin{array}{cccc cccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right]$ <p>$s=vH^T$</p>	<p>Coset 0 - Palavras-código - Síndrome 0</p> <p><u>0 0 0 0 0 0 0</u></p> <p>0 1 1 0 0 0 1 1 1 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 0 1 0 0 0 1 0 1 0 0 1 0 1 1 0 0 1 0 0 1 1 1 1 0 1 1 0 0 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 0 0 1 0 1 1 0 1 0 1 1 0 0 0 0 1 1 1 0 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1</p> <p>←→←→←→ dados bits redundantes</p>
	<p>Coset 1 – Síndrome 1</p> <p><u>0 0 0 0 0 0 1</u></p> <p>0 1 1 0 0 0 0 1 1 0 0 0 1 1 1 0 1 0 0 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 0 1 0 1 1 1 0 1 0 0 1 1 0 1 0 1 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0 1 1 0 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 0</p>
	<p>.</p> <p>.</p>
	<p>Coset $2^{n-k}-1$ – Síndrome $2^{n-k}-1$</p> <p><u>1 0 0 0 0 0 0</u></p> <p>1 1 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 1 1 1 0 0 1 1 0 0 0 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 0 0 1 0 1 1 1 1 0 1 1 0 0 1 0 1 1 1 0 1 0 0 0 1 1 1 0 0 1 1 1 1 1 1</p>

Figura 3.1: Exemplo de um código $C(7,4)$

sistemática, onde os k bits de dados aparecem inalterados num bloco da palavra-código e as k linhas da submatriz P devem ser as combinações com $n - k$ bits com maior número de ‘1’ arranjados em qualquer ordem, sendo a submatriz I a matriz identidade $k \times k$.

$$G = [I_k : P_{k \times (n-k)}] \quad (3.9)$$

ou

$$G = [P_{k \times (n-k)} : I_k] \quad (3.10)$$

Enquanto no primeiro exemplo os bits de paridade das palavras-código geradas pela matriz geradora aparecerão no fim, no segundo exemplo, os bits de paridade aparecerão no início. O processo de codificação é simples, bastando para tal utilizar a equação 3.8 com uma das formas de matriz geradora apresentadas.

De notar que as k linhas da matriz geradora G são linearmente independentes, portanto, nenhuma delas consegue ser obtida por adição das outras linhas. Por outro lado, cada linha da matriz geradora é uma palavra-código.

Na figura 3.1 é apresentada a matriz geradora para um código de Hamming $C(7, 4)$ e as palavras-código obtidas. Os k bits de dados aparecem no início de cada palavra-código, enquanto os $n - k$ bits de paridade aparecem no fim.

Matriz de verificação de paridade

Para o processo de descodificação é necessário utilizar a matriz de verificação de paridade H que é composta pela submatriz P transposta.

$$H = [(P_{k \times (n-k)})^T : I_{n-k}] \quad (3.11)$$

A submatriz P de dimensão $k \times (n - k)$ determina as propriedades do código (ela está presente tanto em G como em H^T). De notar que se multiplicarmos qualquer palavra-código por H^T obtemos um vector nulo, tal como indicado a seguir:

$$vH^T = uGH^T = 0 \quad (3.12)$$

onde o produto da matriz geradora G pela transposta da matriz de verificação de paridade H é uma matriz nula de dimensão $k \times (n - k)$:

$$GH^T = [I_k : P_{k \times (n-k)}] \begin{bmatrix} P_{k \times (n-k)} \\ I_{n-k} \end{bmatrix} = P + P = 0 \quad (3.13)$$

Resumindo, se o vector resultante da multiplicação da palavra recebida por H^T for nulo, podemos concluir que a palavra recebida é uma palavra-código. Pelo contrário, se não for nulo, podemos concluir que a palavra recebida não é uma palavra-código e que houve erros na transmissão.

A figura 3.1 mostra a matriz de verificação de paridade referente à matriz geradora exibida no exemplo.

Descodificação pela síndrome

Aproveitando o facto do produto de uma palavra-código pela transposta da matriz de verificação de paridade H^T ser nulo, a verificação da existência de erros numa palavra recebida pode ser realizada calculando a síndrome s :

$$s = vH^T \quad (3.14)$$

Quando todos os elementos da síndrome são zero, uma de duas situações pode ter acontecido: a palavra recebida corresponde à palavra-código enviada e não houve erros durante a transmissão ou, a palavra recebida corresponde

a outra palavra-código e os erros não foram detectados. Quando acontece apenas um erro durante a transmissão, a síndrome é diferente do vector zero.

Considerando que a palavra recebida z que se pretende descodificar é igual à soma da palavra enviada u com um vector de erro e (que pode ser nulo ou não), tal que:

$$z = u + e \quad (3.15)$$

A síndrome dessa palavra fica:

$$s = (u + e)H^T = uH^T + eH^T = eH^T \quad (3.16)$$

Como $uH^T = 0$, é possível concluir que a síndrome depende do erro e não da palavra transmitida.

Num código $C(n, k)$ existem 2^{n-k} síndromes possíveis e 2^n palavras de n bits possíveis. Logo, várias palavras produzem a mesma síndrome e só é possível corrigir 2^{n-k} padrões de erros. Como só é possível corrigir aquele número de padrões de erros, a descodificação deverá tirar partido do facto de alguns padrões de erros terem maior probabilidade de acontecer do que outros. Admitindo que os padrões de erro com menor peso são mais prováveis, no caso de erros independentes, uma vez que um erro único é mais provável de acontecer do que um erro duplo e assim sucessivamente, a descodificação utiliza as síndromes e os padrões de erros com menor peso para descodificar a palavra recebida. Esta técnica denomina-se descodificação de máxima verosimilhança.

‘Cosets’ e matriz padrão

A matriz padrão (Standard Decoding Array ou Slepian Array) de um código $C(n, k)$ pode ser utilizada no processo de descodificação e é constituída pelas

Tabela 3.3: Matriz padrão de um código $C(n, k)$

$$\left[\begin{array}{c|cccc} v_1 = 0 & v_2 & v_3 & \dots & v_{2^k} \\ \hline e_1 & v_2 + e_1 & v_3 + e_1 & \dots & v_{2^k} + e_1 \\ e_2 & v_2 + e_2 & v_3 + e_2 & \dots & v_{2^k} + e_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ e_{2^{n-k}-1} & v_2 + e_{2^{n-k}-1} & v_3 + e_{2^{n-k}-1} & \dots & v_{2^k} + e_{2^{n-k}-1} \end{array} \right]$$

2^n palavras possíveis do código dispostas da seguinte forma:

1. Primeira linha da matriz constituída por todas as 2^k palavras-código, ficando o vector zero mais à esquerda;
2. Escolher das restantes palavras do espaço $\{0, 1\}^n$ que não estejam na matriz, aquela com menor peso. Adicionar a palavra escolhida a todas as palavras-código da primeira linha para obter a linha seguinte. Continuar o processo até a matriz ficar com 2^{n-k} linhas.

A matriz padrão resultante ficará com 2^{n-k} linhas e 2^k colunas. Se as palavras com menor peso escolhidas forem vistas como os padrões de erro mais prováveis, a matriz padrão corresponderá à tabela 3.3.

Cada linha da matriz padrão pode ser vista como ‘translações’ das palavras-código, ou seja, classes equivalentes ao código C . Na teoria de grupos (Humphreys 1996), estas classes são denominadas ‘cosets’ do código linear C e os padrões de erros que as originaram (primeira coluna) são designados de ‘coset leaders’ ou ‘coset representatives’. Qualquer palavra do ‘coset’ poderia ser utilizada como o seu ‘coset leader’, no entanto, para minimizar a probabilidade de erro da descodificação, devem ser escolhidos para ‘coset leaders’ os padrões de erros mais prováveis.

Seja $v + e$ um elemento do ‘coset’ resultante da palavra-código v com o ‘coset leader’ e . A síndrome deste elemento será:

$$s = (v + e)H^T = vH^T + eH^T = eH^T \quad (3.17)$$

Em conclusão, a síndrome de qualquer elemento de um ‘coset’ é igual à síndrome do seu ‘coset leader’, ou seja, todas as palavras pertencentes a um ‘coset’ têm a mesma síndrome. Num código $C(n, k)$, 2^{n-k} ‘cosets’ C_i podem ser definidos, para $i = 0, \dots, 2^{n-k} - 1$, onde o ‘coset’ C_i designa o ‘coset’ com síndrome i . Cada ‘coset’ contém 2^k palavras e o ‘coset’ C_i corresponde ao código. Na figura 3.1, os ‘coset leaders’ aparecem indicados na primeira linha de cada ‘coset’.

Quando os ‘coset leaders’ escolhidos correspondem a todos os padrões de erro com peso menor ou igual a t (número de erros que consegue corrigir), então o código é considerado perfeito. Os códigos de Hamming são perfeitos porque não conseguem corrigir nenhuma palavra recebida com mais do que t erros, tal como indicado na expressão 3.6.

3.2.2 Códigos cíclicos

Os códigos cíclicos são uma subclasse importante dos códigos de bloco lineares. Os códigos de Hamming, os códigos CRC (Cyclic Redundancy Check Codes), os códigos Reed Solomon, os códigos BCH (Bose Chaudhuri Hocquenghem) e os códigos Golay são alguns exemplos de códigos cíclicos. O código da figura 3.1 é também um código cíclico.

Um código cíclico obedece às seguintes propriedades:

1. A soma de duas palavras-código é uma palavra-código;
2. O deslocamento de uma palavra-código é uma palavra-código.

A primeira propriedade é necessária para que o código seja linear e a segunda é a que caracteriza um código cíclico. Se, por exemplo, $[v_{n-1}v_{n-2}\dots v_1v_0]$ for uma palavra-código de um código cíclico então $[v_{n-2}\dots v_0v_{n-1}]$ também será.

Os códigos cíclicos são geralmente definidos utilizando a notação polinomial. Neste caso, um código cíclico $C(n, k)$ pode ser definido na forma sistemática da seguinte maneira:

$$\begin{aligned} V(x) &= u_{k-1}x^{n-1} + u_{k-2}x^{n-2} \dots + u_0x^{n-k} + c_{n-k-1}x^{n-k-1} + \dots + c_0 \\ &= x^{n-k}U(x) + C(x) \end{aligned} \quad (3.18)$$

onde $V(x)$ corresponde ao polinómio de código, $U(x)$ é o polinómio de informação composto pelos k bits da mensagem a transmitir e o polinómio $C(x)$ corresponde aos $n - k$ bits de paridade. O polinómio $U(x)$ tem grau, no máximo, $k - 1$ e por se tratar de um código binário, os seus coeficientes são 0 ou 1. O grau máximo do polinómio $V(x)$ é $n - 1$.

Em analogia ao conceito de matriz geradora dos códigos de bloco, os códigos cíclicos podem ser definidos utilizando o polinómio gerador $g(x)$ que deverá ter as seguintes propriedades:

1. $g(x)$ é um factor de $x^n + 1$;
2. O grau de $g(x)$ é igual a $n - k$, o número de bits de paridade.

Como consequência destas propriedades, qualquer polinómio de código é múltiplo do polinómio gerador e o polinómio gerador é também um polinómio de código (o de menor grau). Por outro lado, qualquer polinómio que seja factor de $x^n + 1$ com grau $n - k$ poderia ser utilizado como polinómio gerador de um código cíclico, no entanto, nem todos os polinómios

nestas condições parecem produzir códigos com as características desejáveis. Assim, os polinómios geradores devem ser polinómios primitivos, isto é, tais que o menor grau n de um polinómio $X^n + 1$ divisível por $g(x)$ é $n = 2^m - 1$, onde m representa o grau do polinómio gerador (Lin & Costello 1983, p. 41). Em Carlson (2002) são referidos alguns exemplos de polinómios geradores recomendados. O polinómio gerador $g(x) = x^3 + x + 1 = 1011$, utilizado no código da figura 3.1, corresponde ao polinómio de código de menor grau e é um dos polinómios recomendados.

Quando se pretende um código não sistemático, os polinómios de código são obtidos multiplicando os polinómios de informação $U(x)$ pelo polinómio gerador $g(x)$, ou seja:

$$V(x) = U(x)g(x) \quad (3.19)$$

Quando se pretende um código cíclico na forma sistemática, as seguintes operações devem ser efectuadas:

1. Multiplicar $U(x)$ por x^{n-k} ;
2. Dividir $x^{n-k}U(x)$ por $g(x)$ e calcular o resto, $C(x) = x^{n-k}U(x) \bmod g(x)$;
3. Acrescentar $C(x)$ a $x^{n-k}U(x)$, tal que: $V(x) = x^{n-k}U(x) + C(x)$.

A tabela 3.4 mostra a geração dos polinómios de código do código cíclico da figura 3.1.

Outra forma de gerar os polinómios de código do código cíclico da figura 3.1 é utilizar os polinómios de código que constituem a matriz geradora. Isto é possível porque todos os códigos cíclicos são lineares (embora nem todos os códigos de bloco lineares sejam cíclicos), logo é possível determinar a matriz geradora a partir do polinómio gerador. Para um código sistemático, em que os bits de paridade surgem depois dos bits de informação, a linha de

Tabela 3.4: Geração dos polinómios de código do código cíclico da figura 3.1 utilizando o polinómio gerador $g(x) = x^3 + x + 1 = 1011$

U	$U(x)$	$x^3U(x)$	$C(x) = x^3U(x) \text{ mod } g(x)$	V
0000	0	0	0	0000000
0001	1	x^3	$x + 1$	0001011
0010	x	x^4	$x^2 + x$	0010110
0011	$x + 1$	$x^4 + x^3$	$x^2 + 1$	0011101
0100	x^2	x^5	$x^2 + x + 1$	0100111
0101	$x^2 + 1$	$x^5 + x^3$	x^2	0101100
0110	$x^2 + x$	$x^5 + x^4$	1	0110001
0111	$x^2 + x + 1$	$x^5 + x^4 + x^3$	x	0111010
1000	x^3	x^6	$x^2 + 1$	1000101
1001	$x^3 + 1$	$x^6 + x^3$	$x^2 + x$	1001110
1010	$x^3 + x$	$x^6 + x^4$	$x + 1$	1010011
1011	$x^3 + x + 1$	$x^6 + x^4 + x^3$	0	1011000
1100	$x^3 + x^2$	$x^6 + x^5$	x	1100010
1101	$x^3 + x^2 + 1$	$x^6 + x^5 + x^3$	1	1101001
1110	$x^3 + x^2 + x$	$x^6 + x^5 + x^4$	x^2	1110100
1111	$x^3 + x^2 + x + 1$	$x^6 + x^5 + x^4 + x^3$	$x^2 + x + 1$	1111111

ordem i da submatriz P corresponde ao resto da divisão do polinómio x^{n-i} pelo polinómio gerador, tal como se pode verificar na figura 3.2. Por outro lado, como todos os códigos cíclicos são lineares, o polinómio de código nulo pertence ao código e os outros polinómios de código resultam da soma das várias combinações dos polinómios de código que constituem a matriz geradora. A figura 3.2 mostra a matriz geradora definida em função do polinómio gerador e a geração dos restantes polinómios de código.

De salientar que os códigos cíclicos, para além de serem utilizados actualmente na correcção de erros em discos compactos e na transmissão de televisão digital, foram utilizados pela NASA nas naves espaciais Voyager (Abrantes n.d.).

Descodificação com códigos cíclicos

Com base na equação 3.19, é possível verificar que apenas os polinómios de código são divisíveis pelo polinómio gerador. Ao dividir o polinómio correspondente à palavra recebida pelo polinómio gerador, se o resto for nulo, o polinómio recebido corresponde a um polinómio de código, caso contrário, o polinómio recebido não corresponde a um polinómio de código e aconteceu um erro durante a transmissão.

Nos códigos cíclicos é possível definir o polinómio de síndrome como:

$$S(x) = Z(x) \text{ mod } g(x) \quad (3.20)$$

em que $Z(x)$ corresponde ao polinómio recebido, que pode não ser igual a algum dos polinómios de código $V(x)$, caso tenha ocorrido algum erro durante a transmissão. Neste caso:

$$Z(x) = V(x) + e(x) \quad (3.21)$$

Geração Polinômios de Códigos	Cosets
$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{array}{l} x^6 \text{ mod } g(x) \\ x^5 \text{ mod } g(x) \\ x^4 \text{ mod } g(x) \\ x^3 \text{ mod } g(x) \end{array} = \begin{array}{l} y_1 \\ y_2 \\ y_3 \\ y_4 \end{array}$ $G = \begin{bmatrix} 1 & 0 & 0 & 0 & & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & & 0 & 1 & 1 \end{bmatrix} = \begin{array}{l} y_1 \\ y_2 \\ y_3 \\ y_4 \end{array}$	<p>Polinômio Gerador $g(x)=x^3+x+1=1011$</p> <p>Coset 0 – Polinômios de Códigos - Síndrome 0</p> <p><u>0000</u>000 = y_0 0110001 = $y_2 + y_3$ 1100010 = $y_1 + y_2$ 1010011 = $y_1 + y_3$ 1110100 = $y_1 + y_2 + y_3$ 1000101 = y_1 0010110 = y_3 0100111 = y_2 1011000 = $y_1 + y_3 + y_4$ 1101001 = $y_1 + y_2 + y_4$ 0111010 = $y_2 + y_3 + y_4$ 0001011 = y_4 0101100 = $y_2 + y_4$ 0011101 = $y_3 + y_4$ 1001110 = $y_1 + y_4$ 1111111 = $y_1 + y_2 + y_3 + y_4$</p> <p>←←←←→→→→ dados bits redundantes</p>
	<p>Coset 1 – Síndrome 1</p> <p><u>000000</u>1 0110000 1100011 1010010 1110101 1000100 0010111 0100110 1011001 1101000 0111011 0001010 0101101 0011100 1001111 1111110</p>
	<p>.</p>
	<p>Coset $2^{rk}-1$ – Síndrome x^2+x+1</p> <p><u>100000</u>0 1110001 0100010 0010011 0110100 0000101 1010110 1100111 0011000 0101001 1111010 1001011 1101100 1011101 0001110 0111111</p>

Figura 3.2: Geração do código cíclico $C(7, 4)$ da figura 3.1

em que $e(x)$ é um polinómio de erro. O polinómio de síndrome pode ser escrito como:

$$S(x) = [V(x) + e(x)] \text{ mod } g(x) \quad (3.22)$$

Quando $Z(x)$ corresponde a um polinómio de código, o polinómio de erro é nulo, $e(x) = 0$. Como $V(x) \text{ mod } g(x) = 0$, o polinómio de síndrome é igual ao resto da divisão do polinómio de erro $e(x)$ por $g(x)$:

$$\begin{aligned} S(x) &= [V(x) + e(x)] \text{ mod } g(x) = V(x) \text{ mod } g(x) + e(x) \text{ mod } g(x) \\ &= e(x) \text{ mod } g(x) \end{aligned} \quad (3.23)$$

Tal como com os códigos de Hamming, a síndrome dos códigos cíclicos só depende do padrão de erro e não do polinómio recebido $Z(x)$.

Como $g(x)$ tem grau $n - k$, o grau da síndrome é, no máximo, $n - k - 1$, e esta pode ser definida como:

$$S(x) = s_{n-k-1}x^{n-k-1} + s_{n-k-2}x^{n-k-2} \dots + s_1x + s_0 \quad (3.24)$$

Tal como foi referido em 3.2.1, também nos códigos cíclicos é possível fazer a descodificação utilizando uma tabela de síndromes e de polinómios de erros. A tabela de síndromes e de erros para o código $C(7, 4)$ da figura 3.2 é apresentada na tabela 3.5.

Para efectuar a descodificação utilizando a tabela dos erros e síndromes, basta calcular a síndrome do polinómio recebido $Z(x)$, tal como indicado na expressão 3.20 e a partir da síndrome obtida e consultando a tabela, determinar o polinómio de erro $e(x)$. Utilizando a expressão 3.21, é possível determinar o polinómio de código $V(x)$. Se o código estiver na sua forma sistemática basta truncar os bits de paridade para obter o polinómio de

Tabela 3.5: Tabela de síndromes e de erros do código cíclico da figura 3.2 utilizando o polinómio gerador $g(x) = x^3 + x + 1 = 1011$

e	$e(x)$	s	$S(x)$
1000000	x^6	101	$x^2 + 1$
0100000	x^5	111	$x^2 + x + 1$
0010000	x^4	110	$x^2 + x$
0001000	x^3	011	$x + 1$
0000100	x^2	100	x^2
0000010	x	010	x
0000001	1	001	1

informação.

Um exemplo é apresentado. Considerando o polinómio recebido $Z(x) = x^5 + x^4 = 0110000$, a sua síndrome é $S(x) = 1 = 001$. Consultando a tabela 3.5 é possível verificar que o erro é $e(x) = 1 = 0000001$. Somando o polinómio de erro com o polinómio recebido, o polinómio código obtido é $V(x) = x^5 + x^4 + 1 = 0110001$ e truncando os $n - k$ bits de paridade, permite obter o polinómio de informação $U(x) = x^2 + x = 0110$.

De notar que se os erros apenas ocorrerem nos bits de paridade, o polinómio de erro $e(x)$ terá um grau que é, no máximo, $n - k - 1$, como se pode constatar na tabela 3.5. Neste caso, a síndrome é igual ao polinómio de erro porque o resto da divisão de um polinómio de grau $n - k - 1$ pelo polinómio gerador $g(x)$ de grau $n - k$ é o próprio dividendo.

3.2.3 Propriedades dos códigos e dos ‘cosets’

Existem propriedades interessantes nos códigos utilizados e nos ‘cosets’ definidos que resultam da linearidade dos códigos. Estas propriedades são:

1. Como o código (‘coset’ com síndrome 0) é linear, a adição de duas

palavras-código origina outra palavra-código. Considerando que v e q são palavras-código, as suas síndromes são $s(v) = 0$ e $s(q) = 0$, a soma $v + q = a$ origina uma palavra-código e $s(v) + s(q) = 0 + 0 = 0 = s(a)$;

2. A adição de uma palavra-código com uma palavra de um ‘coset’ com síndrome i origina uma palavra do ‘coset’ com síndrome i . Considerando que v é uma palavra-código, que p corresponde a uma palavra do ‘coset’ com síndrome i , a soma $v + p = y$ origina uma palavra com síndrome i porque $s(v) + s(p) = 0 + i = i = s(y)$;
3. A adição de duas palavras que pertencem ao mesmo ‘coset’ origina uma palavra-código. Considerando que y e p são palavras do ‘coset’ com síndrome i , a soma $y + p = a$ origina uma palavra-código porque $s(y) + s(p) = i + i = 0 = s(a)$;
4. A distância d_{min} a que se encontram as palavras que pertencem ao mesmo ‘coset’, é igual à distância mínima a que se encontram as palavras-código do código em questão quando o código é linear. Considerando as palavras-código v e q , que por definição estão, no mínimo, a uma distância d_{min} , tal que $d(v, q) = peso(v + q) \geq d_{min}$. Adicionando um ‘coset leader’ c , a distância entre as palavras resultantes, $d(v + c, q + c) = peso(v + c + q + c) = peso(v + q) \geq d_{min}$ não é alterada porque $c + c = 0$;
5. Uma mutação numa palavra de um ‘coset’ origina uma palavra de outro ‘coset’. Considerando que uma mutação numa palavra corresponde à adição da palavra com um vector com peso 1, tal que, $y + e = q$, onde $s(y) + s(e) = i + j = m = s(q)$. Como a $d_{min} \geq 2$ e o vector nulo tem de pertencer ao código, não existem palavras-código com peso 1, assim, $s(e) \neq 0$, indicando que, $i + j \neq i$.

Tabela 3.6: Matriz geradora de um código equivalente ao código da figura 3.1

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

3.3 Códigos equivalentes

Dois códigos lineares $C(n, k)$ em $GF(2)$ são considerados equivalentes se as matrizes geradoras correspondentes diferirem por (Bose 2002):

1. Substituição de uma linha pela soma dela com um múltiplo de uma outra;
2. Permutação de linhas;
3. Permutação de colunas.

Enquanto as duas primeiras operações são relativas a linhas e preservam a independência das linhas da matriz geradora, a última operação é referente às colunas e permite converter a matriz numa outra que produzirá um código equivalente.

Um exemplo é apresentado. Considerando a matriz geradora da tabela 3.6 que corresponde a um código $C(7, 4)$, é possível obter a matriz geradora do código da figura 3.1 por aplicação de várias operações. Sendo l_1, \dots, l_4 as linhas e c_1, \dots, c_7 as colunas da matriz geradora, é possível obter a matriz geradora da figura 3.1 por aplicação das seguintes operações: $c_3 \leftrightarrow c_1$, $c_5 \leftrightarrow c_2$, $c_7 \leftrightarrow c_3$, $l_2 = l_2 + l_4$, $c_5 \leftrightarrow c_7$.

3.4 Considerações finais

Os códigos de controlo de erros, em particular, os códigos de bloco lineares, foram estudados neste capítulo. Os códigos de Hamming e os códigos cíclicos foram apresentados, por serem estes os códigos em que se baseia o desenvolvimento da família de representações neutrais que se propõe nesta dissertação. A formulação matemática destes códigos permitirá definir representações que exibem vários níveis de neutralidade e características diversas.

Outra família de representações redundantes, que não exibem neutralidade, será também apresentada. Esta família tem a particularidade de ser definida por transformações lineares idênticas às usadas neste capítulo.

De realçar que, os códigos correctores de erros apresentados podem ser considerados representações neutrais. Se o número de erros durante a transmissão de uma palavra for inferior à distância mínima do código, a palavra recebida será associada à palavra-código correcta. Considerando que os erros podem ocorrer em diferentes posições na palavra transmitida, palavras diferentes são corrigidas de modo a corresponderem à mensagem transmitida, ou seja, as mutações nas palavras-código podem ser consideradas neutrais porque descodificam para a mesma mensagem.

Capítulo 4

Desenvolvimento de famílias de representações binárias redundantes

4.1 Introdução

Neste capítulo, são propostas várias representações binárias construídas de modo a exibirem as propriedades de redundância, poligenia, pleiotropia e neutralidade, existentes no código genético natural.

O desenvolvimento das primeiras representações redundantes que se propõem baseia-se em transformações lineares e é realizada de forma incremental, começando por uma representação redundante não codificante simples, e incorporando em cada passo uma propriedade, primeiro a poligenia, depois a pleiotropia.

Em virtude destas transformações lineares não permitirem a incorporação de neutralidade, a não ser pela utilização de genes não codificantes, é adotada uma segunda abordagem, baseada em códigos de controlo de erros,

para definir uma família de representações redundantes neutrais. A forma como a representação é construída, os processos de codificação e de decodificação, a noção de representação canónica e um algoritmo para a geração de representações de uma dada família são apresentados. O estudo das características utilizadas na literatura sobre representações redundantes, como a conectividade, a localidade e a sinonímia, é realizada para esta família de representações redundantes.

Este capítulo começa por apresentar um modelo para construir representações redundantes que não exibem neutralidade. Na secção 4.3 é apresentada uma forma de construir representações redundantes neutrais inspirada na formulação matemática dos códigos de controlo de erros, que permite obter uma família de representações que apresentam características interessantes, mesmo utilizando um número reduzido de bits redundantes. A definição de representação neutral e os processos de codificação e de decodificação desta representação são também estudados nesta secção. O algoritmo para a geração de representações neutrais é desenvolvido na secção 4.4. As propriedades e as relações entre as propriedades das representações neutrais estão indicadas na secção 4.5. Por último, são apresentadas algumas considerações finais.

4.2 Representações redundantes baseadas em transformações lineares

Considerando que os genótipos v e os fenótipos u são vectores binários, a forma mais simples de mapear genótipos em fenótipos é considerar o mapeamento identidade:

$$u = v \cdot I_k \tag{4.1}$$

onde $u \in \{0, 1\}^k$ e $v \in \{0, 1\}^\ell$ são elementos de um espaço vectorial sobre o corpo $GF(2)$, $\ell = k$ e I_k é a matriz identidade de ordem k . Esta transformação entre genótipo e fenótipo é uma transformação linear, à semelhança da relação linear que permite construir as palavras-código v de um código de controlo de erros, com base na matriz geradora G , a partir dos dados de mensagem u , tal como foi explicado na subsecção 3.2.1.

Com base em transformações lineares deste tipo, é ainda possível definir representações redundantes, como a seguir se descreve.

4.2.1 Representações com genes não codificantes

Adicionar genes não codificantes ou inactivos é a forma mais simples de introduzir redundância numa representação. Uma das várias formas de construir uma representação com redundância não codificante é considerar que o fenótipo $u \in \{0, 1\}^k$, é obtido a partir de um genótipo $v \in \{0, 1\}^\ell$, onde $\ell > k$, segundo:

$$u = v \cdot X_{\ell \times k} \quad (4.2)$$

Na matriz X só deverá existir um bit activo (bit a 1) por coluna e a sua localização na coluna indica o bit do genótipo que codifica para o fenótipo. Por exemplo, considerando uma representação onde os primeiros k bits do genótipo correspondem aos k bits do fenótipo e onde os restantes $(\ell - k)$ bits do genótipo são não codificantes, a matriz X deverá ter a seguinte forma:

$$X_{\ell \times k} = \begin{bmatrix} I_k \\ O_{(\ell-k) \times k} \end{bmatrix} \quad (4.3)$$

onde I_k é a matriz identidade e $O_{(\ell-k) \times k}$ representa a matriz nula.

Esta representação é uniforme porque todos os fenótipos são representa-

dos pelo mesmo número de genótipos. Neste caso, cada fenótipo é representado por $2^{\ell-k}$ genótipos. Por outro lado, este tipo de representação exhibe neutralidade, porque os genótipos que diferem apenas nos bits não codificantes descodificam para o mesmo fenótipo.

No contexto da computação evolutiva, a utilidade deste tipo de representação é reduzida. Como os bits não codificantes ficam sujeitos à mutação da mesma forma que os bits codificantes, mas não produzem alterações no fenótipo, se o número de bits redundantes não for explicitamente tido em conta na determinação da taxa de mutação por indivíduo, os bits codificantes do genótipo terão menor probabilidade de sofrerem uma mutação. Isso seria equivalente a usar uma representação não redundante com uma taxa de mutação mais baixa, o que, em princípio, tornaria o processo de evolução mais lento.

4.2.2 Representações com poligenia

Num mapeamento genótipo-fenótipo redundante sem bits não codificantes, pelo menos um traço fenotípico será necessariamente determinado pela interação de dois ou mais bits do genótipo. Este efeito corresponde ao efeito de poligenia, já explicado no capítulo 2. A implementação deste mecanismo pode ser realizada considerando um mapeamento genótipo-fenótipo definido da seguinte forma:

$$u = v \cdot Y_{\ell \times k} \tag{4.4}$$

Na matriz Y os vários bits activos que existam em cada coluna indicam a localização dos bits do genótipo que interagem e contribuem para o correspondente bit do fenótipo. Como cada bit do genótipo só pode influenciar um

Tabela 4.1: Genótipos e respectivos fenótipos obtidos utilizando a representação com poligenia definida na expressão 4.5

v	u	v	u	v	u	v	u
0000	00	0001	01	0010	10	0011	11
0100	01	0101	00	0110	11	0111	10
1000	10	1001	11	1010	00	1011	01
1100	11	1101	10	1110	01	1111	00

bit do fenótipo, em cada linha da matriz Y só poderá estar um bit activo. Por exemplo, para uma representação com $\ell = 4$ e $k = 2$, onde o 1º bit do fenótipo corresponde à interacção do 1º e 3º bits do genótipo e o 2º bit do fenótipo corresponde à interacção do 3º e 4º bits do genótipo, a matriz Y deverá ter a seguinte forma:

$$Y_{4 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.5)$$

De realçar que quantos mais bits activos estiverem numa coluna de Y , maior a probabilidade do bit do fenótipo a que essa coluna corresponde sofrer alterações devido a uma mutação simples.

A tabela 4.1 mostra os fenótipos obtidos para cada um dos genótipos possíveis utilizando a expressão 4.5. A representação é uniforme como se pode constatar na tabela 4.1. Por outro lado, a tabela 4.2 mostra um exemplo em que se pode constatar que os genótipos que descodificam para o fenótipo 00 não alcançam quaisquer genótipos que descodifiquem também para 00, ou seja, a representação não apresenta neutralidade.

É de notar ainda que os fenótipos alcançáveis a partir do fenótipo 00

Tabela 4.2: Genótipos vizinhos e respectivos fenótipos dos genótipos que descodificam para o fenótipo 00 da representação definida na expressão 4.5

	1	2	3	4
0000	(1)[1]	(2)[2]	(4)[1]	(8)[2]
0101	(4)[1]	(7)[2]	(1)[1]	(13)[2]
1010	(11)[1]	(8)[2]	(14)[1]	(2)[2]
1111	(14)[1]	(13)[2]	(11)[1]	(7)[2]

correspondem às linhas da matriz Y .

4.2.3 Representações com poligenia e pleiotropia

As representações da subsecção anterior podem ser generalizadas de modo a permitir que o mesmo bit do genótipo afecte mais do que um bit do fenótipo, implementando assim o efeito de pleiotropia. A transformação linear entre genótipo e fenótipo é semelhante à anterior:

$$u = v \cdot Z_{\ell \times k} \quad (4.6)$$

mas agora as linhas da matriz $Z_{\ell \times k}$ poderão ter qualquer número de bits activos (excepto o 0 para não introduzir genes não codificantes).

Uma vez que as linhas da matriz Z determinam a vizinhança fenotípica induzida pela representação, torna-se possível especificar qualquer vizinhança pretendida. Por outro lado, a ordem das linhas da matriz Z define a posição do bit do genótipo que permite alcançar determinado fenótipo.

Por exemplo, quando se pretende construir uma representação redundante onde a partir do fenótipo 000 seja possível alcançar os fenótipos 001, 010,

100, 011 e 110, a matriz Z poderá ter a seguinte forma:

$$Z_{\ell \times k} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad (4.7)$$

É fácil verificar que a representação definida por esta matriz exhibe pleiotropia, porque pelo menos um dos bits do genótipo afecta mais do que um fenótipo. Por exemplo, o 4º bit do genótipo afecta o 2º e 3º bits do fenótipo (4ª linha da matriz Z). A representação também apresenta poligenia, porque pelo menos um bit do fenótipo é determinado pela interacção de mais do que um bit do genótipo. Por exemplo, o 1º bit do fenótipo resulta da interacção dos 3º e 5º bits do genótipo (1ª coluna da matriz Z). Desta forma, é possível desenhar qualquer vizinhança fenotípica pretendida. Por outro lado, tal como no caso da representação anterior, quanto mais bits activos estiverem numa coluna de Z , maior a probabilidade do bit do fenótipo a que essa coluna corresponde sofrer alterações devido a uma mutação simples. Em particular, assumindo uma probabilidade de mutação igual para todos os bits do genótipo, o 2º bit do fenótipo tenderá a sofrer alterações mais frequentemente que o 1º e o 3º.

A tabela 4.3 apresenta os fenótipos obtidos para cada um dos genótipos possíveis utilizando a matriz da equação 4.7, podendo verificar-se que a representação é uniforme. Por outro lado, a tabela 4.4 mostra que o fenótipo vizinho do fenótipo 000 resultante da mutação do 1º bit dos genótipos que representam o fenótipo 000 é o fenótipo 001, para todos os genótipos que representam o fenótipo 000; assim como o único fenótipo alcançável é o 110,

Tabela 4.3: Genótipos e respectivos fenótipos obtidos utilizando a representação com a matriz definida em 4.7

v	u	v	u	v	u	v	u
00000	000	00001	110	00010	011	00011	101
00100	100	00101	010	00110	111	00111	001
01000	010	01001	100	01010	001	01011	111
01100	110	01101	000	01110	101	01111	011
10000	001	10001	111	10010	010	10011	100
10100	101	10101	011	10110	110	10111	000
11000	011	11001	101	11010	000	11011	110
11100	111	11101	001	11110	100	11111	010

Tabela 4.4: Genótipos vizinhos e respectivos fenótipos dos genótipos que descodificam para o fenótipo 000 da representação definida em 4.7

	1	2	3	4	5
00000	(1)[6]	(2)[3]	(4)[4]	(8)[2]	(16)[1]
10111	(22)[6]	(21)[3]	(19)[4]	(31)[2]	(7)[1]
11010	(27)[6]	(24)[3]	(30)[4]	(18)[2]	(10)[1]
01101	(12)[6]	(15)[3]	(9)[4]	(5)[2]	(29)[1]

quando se considera a vizinhança resultante da mutação do último bit de cada genótipo que descodifica para o fenótipo 000. De facto, todos os genótipos que descodificam para determinado fenótipo alcançam o mesmo fenótipo pela mutação do mesmo bit. Esta é uma das limitações resultantes da transformação desta representação ser linear, levando a que a conectividade só possa crescer linearmente com o comprimento ℓ do genótipo.

Esta limitação pode ser verificada formalmente. Considerando que $Z = [z_0; z_1; \dots; z_{\ell-1}]$, onde cada z_i é uma linha de Z , $0 \leq i < \ell$, e representando o efeito de uma mutação no bit i do genótipo v por $v + e_i$, onde e_i corresponde a um vector com comprimento ℓ com apenas um bit a 1 na posição i e v

descodifica para o fenótipo u , verifica-se que:

$$(v + e_i) \cdot Z = v \cdot Z + e_i \cdot Z = u + z_i \quad (4.8)$$

ou seja, que o fenótipo alcançável a partir do fenótipo u por mutação do bit i só depende do próprio fenótipo u e da linha z_i da matriz Z . Da expressão 4.8 conclui-se que o efeito de uma mutação, qualquer que seja o fenótipo considerado, não depende do(s) genótipo(s) que o representam, mas apenas do bit mutado.

Como já foi referido, só é possível introduzir neutralidade nesta representação pela introdução de bits não codificantes, o que não acrescenta qualquer mais valia à representação. De realçar, no entanto, que esta representação permite o aumento da conectividade entre fenótipos quando comparada com as representações anteriores.

Em resumo, as representações redundantes desta família:

1. São uniformes e só permitem introduzir neutralidade através de genes não codificantes;
2. Permitem desenhar de forma simples a vizinhança pretendida no espaço de procura fenotípico;
3. Permitem que a conectividade cresça linearmente com o comprimento do genótipo;
4. Não permitem alcançar fenótipos diferentes a partir dos diferentes genótipos que descodificam para o mesmo fenótipo.

A notação $NonNN_Z(\ell, k)$ (*Non Neutral Network*) será utilizada para designar a família de representações não neutrais definidas com base na matriz Z .

4.3 Representações redundantes neutrais inspiradas em códigos de controlo de erros

4.3.1 Neutralidade

Embora as representações lineares propostas na secção anterior permitam a especificação do conjunto de fenótipos alcançáveis a partir de determinado fenótipo por mutações simples, todos os genótipos que representam o mesmo fenótipo alcançam o mesmo conjunto de fenótipos. Por outro lado, segundo a teoria da evolução neutral, a acumulação de mutações neutrais deverá permitir encontrar novos caminhos para a evolução, o que requer que:

1. os genótipos que representam o mesmo fenótipo formem uma rede neutral conectada;
2. os genótipos que representam o mesmo fenótipo possam alcançar fenótipos diferentes.

Nenhuma das representações apresentadas anteriormente possui estas duas propriedades. Assim, é necessária uma abordagem diferente para conseguir introduzir estas características na representação, tal como a seguir se descreve (Fonseca & Correia 2005):

1. Dividir o espaço redundante dos genótipos, de cardinalidade 2^ℓ , em $2^{\ell-k}$ subespaços (classes), cada um com cardinalidade 2^k , de tal modo que mutações de um bit no genótipo permitam passar de um subespaço para outro;
2. Definir mapeamentos bijectivos entre cada subespaço do espaço dos genótipos e o espaço dos fenótipos, de tal modo que os diferentes

genótipos que representam o mesmo fenótipo formem redes neutrais conectadas.

As figuras 4.1 e 4.2 permitem ilustrar esta abordagem. A figura 4.1 mostra um espaço com 64 elementos dividido em 4 subespaços diferentes representados pelos símbolos \circ , \bullet , \square e \blacksquare , com 16 elementos cada um. Movendo tanto na vertical como na horizontal a partir de qualquer um dos 64 pontos, qualquer dos pontos acessíveis pertence a um subespaço diferente do subespaço do ponto de origem. Considera-se que os vizinhos dos pontos nos lados do quadrado incluem os pontos correspondentes no lado oposto.

Na figura 4.2, os fenótipos foram atribuídos de tal modo que cada genótipo que codifica determinado fenótipo consegue aceder a outro genótipo que codifica o mesmo fenótipo. Nesta figura, as linhas sólidas que unem vários genótipos representam as redes neutrais que correspondem a determinado fenótipo, e as linhas a tracejado indicam a ligação das redes com os outros lados do quadrado. O espaço fenotípico tem cardinalidade $|\Phi_f| = 2^4 = 16$ e a cada rede neutral corresponde um fenótipo diferente. Considerando que a vizinhança é definida pelos movimentos norte, sul, este e oeste, cada fenótipo pode alcançar 6 fenótipos diferentes e cada genótipo de uma rede neutral consegue alcançar fenótipos vizinhos diferentes dos restantes genótipos da mesma rede.

A mesma abordagem pode ser adoptada no caso binário, desde que seja possível dividir o espaço dos genótipos em classes, de tal modo que a distância de Hamming mínima entre genótipos da mesma classe seja pelo menos 2. Como foi visto no capítulo 3, este é um aspecto essencial dos códigos de controlo de erros, em geral, e dos códigos de bloco lineares, em particular.

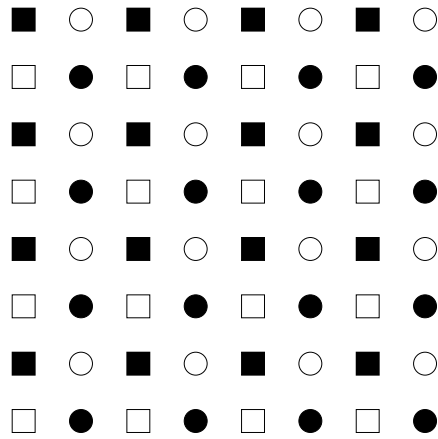


Figura 4.1: Espaço genotípico redundante dividido em 4 subespaços entrelaçados

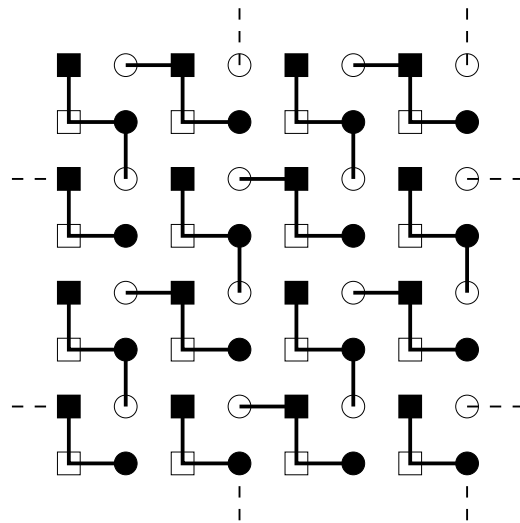


Figura 4.2: Redes neutrais correspondentes aos 16 fenótipos diferentes

4.3.2 Representações neutrais

Um código linear $C(\ell, k)$ gerado por um determinado polinómio gerador $g(x)$, onde as palavras-código constituem um subespaço com a propriedade de distância mínima, fornece a ferramenta necessária para construir uma família de representações redundantes e neutrais, que será designada por $NN_g(\ell, k)$ (*Neutral Network*).

Se um genótipo for visto como a palavra a ser transmitida v , de comprimento ℓ , ou alternativamente, como o polinómio de código $V(x)$, de grau até $\ell - 1$, de um código linear $C(\ell, k)$ gerado com o polinómio gerador $g(x)$, e o fenótipo como a palavra u de comprimento k , ou como o polinómio de informação $U(x)$, de grau até $k - 1$, a expressão seguinte, já apresentada no capítulo 3, permite definir o genótipo em função do fenótipo:

$$V(x) = x^{\ell-k}U(x) + x^{\ell-k}U(x) \text{ mod } g(x) \quad (4.9)$$

O genótipo $V(x)$ assim calculado é uma palavra-código e, portanto, pertence apenas ao ‘coset’ 0 definido pelo polinómio gerador $g(x)$. Em geral, a representação de $U(x)$ em cada um dos $2^{\ell-k}$ ‘cosets’ pode ser definida seleccionando $2^{\ell-k}$ ‘coset leaders’ $z_i(x)$, e calculando:

$$V_i(x) = V(x) + z_i(x) \quad (4.10)$$

Para que os genótipos $V_i(x)$ que representam qualquer $U(x)$ formem uma rede neutral conectada, basta que os ‘coset leaders’ $z_i(x)$ formem eles próprios um grafo conectado por ligações correspondentes a alterações de um bit. Quando $U(x)$ corresponde ao fenótipo 0, também $V(x)$ é igual a zero e os $2^{\ell-k}$ polinómios $V_i(x)$ correspondem aos $2^{\ell-k}$ ‘coset leaders’ $z_i(x)$. Neste contexto e por esta razão, os ‘coset leaders’ passam a ser designados por

zeros da representação e a rede neutral formada pelos ‘coset leaders’ define uma representação neutral da família $NN_g(\ell, k)$:

Definição 4.1. *Uma representação neutral pertencente à família $NN_g(\ell, k)$ consiste num mapeamento do conjunto de strings binárias de comprimento ℓ (o espaço dos genótipos) no conjunto das strings binárias de comprimento k (o espaço dos fenótipos), com $k < \ell$, que é caracterizado pelo polinómio gerador $g(x)$, de grau $\ell - k$, e pelo conjunto de $2^{\ell-k}$ genótipos:*

$$R = \{z_0, z_1, \dots, z_{2^{\ell-k}-1}\}, z_i \in C_i, \forall i = 0, \dots, 2^{\ell-k} - 1$$

onde cada C_i denota o conjunto de genótipos com síndrome i , e cada z_i é considerado o ‘coset leader’ de C_i . Para cada par (z_i, z_j) , ou $d(z_i, z_j) = 1$, ou $d(z_i, z_j) > 1$ e existirá um conjunto de genótipos $\{w_1, \dots, w_p\} \subset R$, $p = d(z_i, z_j) - 1$, tal que:

$$\begin{aligned} d(z_i, w_1) &= 1 \\ d(w_p, z_j) &= 1 \\ d(w_t, w_{t+1}) &= 1, \quad \forall t, 1 \leq t < p \end{aligned}$$

Cada representação neutral de $NN_g(\ell, k)$ permite desenhar a estrutura da rede neutral, pela indicação dos seus zeros, sendo a vizinhança fenotípica definida implicitamente em consequência dessa escolha. Em contrapartida, a representação não neutral proposta na secção anterior permite definir explicitamente a vizinhança desejada.

Os ‘coset leaders’ utilizados na representação neutral apresentam uma diferença em relação aos indicados na secção 3.2.1, uma vez que estes podem não ser as palavras de menor peso de cada ‘coset’ C_i . Por uma questão de

Tabela 4.5: Polinómios geradores utilizados para definir $NN_g(\ell, k)$

$\ell - k$	$g(x)$
1	$x + 1$
2	$x^2 + x + 1$
3	$x^3 + x + 1$
4	$x^4 + x + 1$
5	$x^5 + x^2 + 1$

simplicidade, a designação rede neutral referir-se-á sempre a redes neutrais conectadas.

À semelhança dos polinómios geradores dos códigos referidos no capítulo 3, os polinómios geradores $g(x)$ são polinómios primitivos, e portanto, irredutíveis (não podem ser factorizados). A tabela 4.5 apresenta os polinómios geradores escolhidos para definir cada família $NN_g(\ell, k)$, que correspondem também aos polinómios geradores propostos na literatura dos códigos de controlo de erros (Carlson 2002, Lin & Costello 1983). Neste trabalho, sempre que ℓ e k sejam especificados, o polinómio $g(x)$ corresponderá a um dos polinómios geradores indicados na tabela 4.5.

Tendo em conta esta definição de representação neutral, os processos de codificação e de decodificação são apresentados seguidamente.

Codificação e decodificação

A codificação do fenótipo $U(x)$ nos genótipos $V_i(x)$, onde $0 \leq i \leq 2^{\ell-k} - 1$, dado o polinómio gerador $g(x)$ e a representação neutral $\{z_0, \dots, z_{2^{\ell-k}-1}\}$ de uma família $NN_g(\ell, k)$ na forma sistemática, está sintetizada no algoritmo 4.1.

O processo de decodificação do genótipo $V(x)$ no fenótipo $U(x)$, dado o polinómio gerador $g(x)$ e a representação neutral $\{z_0, \dots, z_{2^{\ell-k}-1}\}$ de

Algoritmo 4.1 Codificação

- 1: Entrada: $U(x), g(x), \{z_0, \dots, z_{2^{\ell-k}-1}\} \in NN_g(\ell, k)$
 - 2: Saída: Todos os $V_i(x)$
 - 3: Calcula $V_0(x) = x^{\ell-k}U(x) + x^{\ell-k}U(x) \bmod g(x)$ que corresponde a um polinómio de código ('coset' 0 com síndrome 0).
 - 4: **for** $i = 0$ to $2^{\ell-k} - 1$ **do**
 - 5: Calcula $V_i(x)$ que resulta da adição do zero $z_i(x)$ com o polinómio $V_0(x)$
 - 6: **end for**
-

$NN_g(\ell, k)$ na forma sistemática, é realizado pelo algoritmo 4.2.

Algoritmo 4.2 Descodificação

- 1: Entrada: $V(x), g(x), \{z_0, \dots, z_{2^{\ell-k}-1}\} \in NN_g(\ell, k)$
 - 2: Saída: $U(x)$
 - 3: Determina a síndrome $S(x) = V(x) \bmod g(x)$
 - 4: Adiciona $V(x)$ com o zero $z_i(x)$ da representação neutral que tem síndrome $S(x)$
 - 5: Calcula o quociente da divisão de $V(x) + z_i(x)$ por $x^{\ell-k}$ para obter o fenótipo $U(x)$ e descarta o resto
-

Como exemplo é apresentada a descodificação do genótipo $V(x) = x^5 + x^4 + x^2$ ou $v = 0110100 = 52$, utilizando a representação neutral $\{0, 1, 9, 77, 4, 14, 6, 73\}$ da família $NN_g(7, 4)$, gerada pelo polinómio gerador correspondente indicado na tabela 4.5, $g(x) = x^3 + x + 1$. A síndrome $S(x) = V(x) \bmod g(x) = x^2 + 1$, e o zero com a mesma síndrome é $z_i(x) = x^3 + x^2 + x$. A adição $V(x) + z_i(x) = (x^5 + x^4 + x^2) + (x^3 + x^2 + x) = x^5 + x^4 + x^3 + x$ e o quociente da divisão de $V(x) + z_i(x)$ por $x^{\ell-k}$ permite obter o fenótipo $U(x) = x^2 + x + 1$.

4.3.3 Representações equivalentes

Tendo em conta que os valores dos genótipos não são avaliados directamente, admite-se que o valor de cada gene (0 ou 1) não seja intrinsecamente impor-

tante. Neste sentido, interessa explorar a relação que pode existir entre uma representação neutral $\{z_0, z_1, \dots, z_{2^{\ell-k}-1}\}$ pertencente a $NN_g(\ell, k)$ e a representação neutral $\{z_0 + c, z_1 + c, \dots, z_{2^{\ell-k}-1} + c\}$, também pertencente a $NN_g(\ell, k)$, resultante da adição de uma constante c a todos os zeros z_i .

Teorema 4.3.1. *Dados:*

(i) *Uma constante $c \in \{0, 1\}^\ell$.*

(ii) *Uma representação neutral $R_1 = \{z_0, z_1, \dots, z_{2^{\ell-k}-1}\} \in NN_g(\ell, k)$ que define um mapeamento genótipo-fenótipo f_{g_1} .*

(iii) *Uma representação neutral $R_2 = \{z_0 + c, z_1 + c, \dots, z_{2^{\ell-k}-1} + c\} \in NN_g(\ell, k)$ que define um mapeamento genótipo-fenótipo f_{g_2} .*

Verifica-se que $f_{g_1}(x_{g_1}) = f_{g_2}(x_{g_2})$ para quaisquer genótipos x_{g_1} e x_{g_2} , tais que $x_{g_2} = x_{g_1} + c$.

Demonstração. Considerando que a síndrome de x_{g_1} é igual à síndrome do zero z_{i_1} , para algum $i_1 \in \{0, \dots, z_{2^{\ell-k}-1}\}$, i.e.:

$$s(x_{g_1}) = s(z_{i_1}), \quad (4.11)$$

a síndrome de x_{g_2} será igual à síndrome de $z_{i_1} + c = z_{i_2} \in R_2$, que é um zero da segunda representação, i.e.:

$$s(x_{g_2}) = s(x_{g_1} + c) = s(z_{i_1} + c) = s(z_{i_2}) \quad (4.12)$$

O valor do fenótipo x_{f_1} , calculado utilizando o algoritmo 4.2, corresponde aos k bits mais significativos da soma do genótipo x_{g_1} com o zero que apresenta a mesma síndrome que ele, i.e., z_{i_1} .

Do mesmo modo, o valor do fenótipo x_{f_2} corresponde aos k bits mais significativos da soma do genótipo x_{g_2} com o *zero* que apresenta a mesma síndrome que x_{g_2} , i.e., $z_{i_2} = z_{i_1} + c$. Como $x_{g_2} = x_{g_1} + c$, verifica-se que:

$$x_{g_2} + z_{i_2} = (x_{g_1} + c) + (z_{i_1} + c) = x_{g_1} + z_{i_1} \quad (4.13)$$

pelo que $f_{g_1}(x_{g_1}) = f_{g_2}(x_{g_2})$, como queríamos demonstrar. \square

Corolário 4.3.2. *Nas condições do teorema 4.3.1, dado um vector $e_i \in \{0, 1\}^\ell$, $0 \leq i < \ell$, com apenas um bit a 1 na posição i , $f_{g_1}(x_{g_1} + e_i) = f_{g_2}(x_{g_2} + e_i)$.*

Demonstração. Como, pelo teorema 4.3.1, $f_{g_1}(x_{g_1}) = f_{g_2}(x_{g_1} + c)$, qualquer que seja $x_{g_1} \in \{0, 1\}^\ell$, a igualdade também se verifica para $x_{g_1} + e_i \in \{0, 1\}^\ell$.

\square

O teorema 4.3.1 e o seu corolário estabelecem uma relação de equivalência entre as representações definidas por $R_1 = \{z_0, z_1, \dots, z_{2^\ell - k - 1}\} \in NN_g(\ell, k)$ e $R_2 = \{z_0 + c, z_1 + c, \dots, z_{2^\ell - k - 1} + c\} \in NN_g(\ell, k)$, para qualquer constante c . De facto, para cada genótipo x_{g_1} que representa $x_f = f_{g_1}(x_{g_1})$, existe um genótipo correspondente x_{g_2} , tal que $x_f = f_{g_2}(x_{g_2})$, e qualquer mutação de x_{g_1} corresponde ao mesmo fenótipo que a mesma mutação aplicada a x_{g_2} , o que significa que a vizinhança fenotípica das duas representações é a mesma.

Para representar o conjunto das representações cujos *zeros* diferem apenas de uma constante c , decidiu-se escolher uma representação canónica.

Definição 4.2 (Representação canónica). *A representação definida pelo conjunto de zeros $R = \{z_0, z_1, \dots, z_{2^\ell - k - 1}\}$ onde o índice i corresponde à síndrome de cada z_i , é uma representação canónica de $NN_g(\ell, k)$ se e só se o vector $(z_0, z_1, \dots, z_{2^\ell - k - 1})$ é o mínimo lexicográfico do conjunto de vectores correspondentes a todas as representações equivalentes $\{z_0 + c, z_1 +$*

Tabela 4.6: Representações equivalentes a $\{0, 1, 9, 77, 4, 14, 6, 73\}$

	$\{0, 1, 9, 77, 4, 14, 6, 73\}$	z_i
0	$\{0, 1, 9, 77, 4, 14, 6, 73\}$	z_0
1	$\{1, 0, 8, 76, 5, 15, 7, 72\}$	z_1
9	$\{9, 8, 0, 68, 13, 7, 15, 64\}$	z_2
77	$\{77, 76, 68, 0, 73, 67, 75, 4\}$	z_3
4	$\{4, 5, 13, 73, 0, 10, 2, 77\}$	z_4
14	$\{14, 15, 7, 67, 10, 0, 8, 71\}$	z_5
6	$\{6, 7, 15, 75, 2, 8, 0, 79\}$	z_6
73	$\{73, 72, 64, 4, 77, 71, 79, 0\}$	z_7

$c, \dots, z_{2^{\ell-k-1}} + c\}$, $c \in \{0, 1\}^\ell$. De notar que a síndrome de $z_i + c$ pode ser diferente da síndrome de z_i e que se entende que as componentes dos referidos vectores se encontram ordenadas pela síndrome de $z_i + c$.

Dada uma representação $R = \{z_0, z_1, \dots, z_{2^{\ell-k-1}}\}$, a representação canónica equivalente pode ser facilmente determinada de entre as $2^{\ell-k}$ representações construídas como $R_i = \{z_0 + z_i, z_1 + z_i, \dots, z_{2^{\ell-k-1}} + z_i\}$, para cada $z_i \in R$. Por exemplo, na tabela 4.6 apresentam-se as representações equivalentes a $\{0, 1, 9, 77, 4, 14, 6, 73\}$ e na tabela 4.7 apresentam-se as mesmas representações equivalentes com os *zeros* ordenados pela síndrome, o que permite verificar que a representação dada já é uma representação canónica.

A figura 4.3 apresenta o código gerado pelo polinómio gerador indicado, juntamente com os *zeros* da representação neutral referida anteriormente, e que é baseada no mesmo polinómio gerador. São apresentados também dois dos ‘cosets’ desta representação. Cada *zero* da representação neutral aparece em binário e em decimal dentro de parênteses curvos e as palavras do código e dos cosets são apresentadas em binário. O fenótipo que cada genótipo representa surge dentro de parênteses rectos, enquanto os ‘cosets leaders’ aparecem dentro de um rectângulo.

$C(\ell, k)$	Cosets
$\ell=7$ $k=4$ $G = \left[\begin{array}{cccc cccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & \end{array} \right]$ $H = \left[\begin{array}{cccc cccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 & \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & \end{array} \right]$	Polinómio Gerador - $g(x)=x^3+x+1=1011$ Coset 0 - Palavras-código - Síndrome 0 $\boxed{0000000}$ [0] 0110001 [6] 1100010 [12] 1010011 [10] 1110100 [14] 1000101 [8] 0010110 [2] 0100111 [4] 1011000 [11] 1101001 [13] 0111010 [7] 0001011 [1] 0101100 [5] 0011101 [3] 1001110 [9] 1111111 [15] \longleftrightarrow dados bits redundantes
$\{0,1,9,7,4,14,6,73\} \in \text{NNG}(7,4)$ $ \Phi_0 =2^l, \Phi_1 =2^k$ 0000000 (0) [0] 0000001 (1) [0] 0001001 (9) [0] 1001101 (77) [0] 0000100 (4) [0] 0001110 (14) [0] 0000110 (6) [0] 1001001 (73) [0]	Coset 1 – Síndrome 1 $\boxed{0000001}$ [0] 0110000 [6] 1100011 [12] 1010010 [10] 1110101 [14] 1000100 [8] 0010111 [2] 0100110 [4] 1011001 [11] 1101000 [13] 0111011 [7] 0001010 [1] 0101101 [5] 0011100 [3] 1001111 [9] 1111110 [15]
	. ..
	Coset 2^{k-1} – Síndrome 2^{k-1} $\boxed{1001001}$ [0] 1111000 [6] 0101011 [12] 0011010 [10] 0111101 [14] 0001100 [8] 1011111 [2] 1101110 [4] 0010001 [11] 0100000 [13] 1110011 [7] 1000010 [1] 1100101 [5] 1010100 [3] 0000111 [9] 0110110 [15]

Figura 4.3: Exemplo de uma representação neutral de $\text{NNG}(7, 4)$

Tabela 4.7: Representações equivalentes a $\{0, 1, 9, 77, 4, 14, 6, 73\}$ com os zeros ordenados pela síndrome

R_{eq_1}	$\{0, 1, 9, 77, 4, 14, 6, 73\}$
R_{eq_2}	$\{0, 1, 76, 8, 15, 5, 72, 7\}$
R_{eq_3}	$\{0, 68, 9, 8, 15, 64, 13, 7\}$
R_{eq_4}	$\{0, 68, 76, 77, 4, 75, 67, 73\}$
R_{eq_5}	$\{0, 10, 2, 77, 4, 5, 13, 73\}$
R_{eq_6}	$\{0, 10, 71, 8, 15, 14, 67, 7\}$
R_{eq_7}	$\{0, 79, 2, 8, 15, 75, 6, 7\}$
R_{eq_8}	$\{0, 79, 71, 77, 4, 64, 72, 73\}$

4.4 Enumeração de representações neutrais

Embora cada representação da família $NN_g(\ell, k)$ fique definida pela escolha do respectivo conjunto de *zeros*, é importante não esquecer que os elementos deste conjunto obedecem a restrições de síndrome e conectividade. Por esse motivo, é importante considerar a forma como as representações de uma tal família podem ser geradas.

A abordagem proposta consiste em definir uma relação de *vizinhança* entre representações da mesma família e em calcular um conjunto de representações canónicas como o fecho (Lewis & Papadimitriou 1998) de uma dada representação canónica inicial sob essa relação.

Definição 4.3 (Representações neutrais vizinhas). *Duas representações neutrais de uma família $NN_g(\ell, k)$ definidas pelos conjuntos de zeros $R_1 = \{z_{1,0}, z_{1,1}, \dots, z_{1,2^{\ell-k}-1}\}$ e $R_2 = \{z_{2,0}, z_{2,1}, \dots, z_{2,2^{\ell-k}-1}\}$ dizem-se vizinhas se e só se os conjuntos R_1 e R_2 diferem apenas num elemento, ou seja, $\exists i \in \{0, \dots, 2^{\ell-k} - 1\}$, tal que, $\forall j \in \{0, \dots, 2^{\ell-k} - 1\}$, $j \neq i \Leftrightarrow z_{1,j} = z_{2,j}$.*

Por outro lado, é importante notar que as representações vizinhas de uma representação canónica não serão necessariamente representações canónicas.

Tabela 4.8: Representações vizinhas de $\{0, 1, 2, 3\} \in NN_g(4, 2)$ e representação canónica das equivalentes

R_1	$\{0, 1, 2, 4\} \Leftrightarrow \{0, 1, 2, 4\}$
R_2	$\{0, 8, 2, 3\} \Leftrightarrow \{0, 1, 2, 10\}$
R_3	$\{0, 1, 5, 3\} \Leftrightarrow \{0, 1, 2, 4\}$
R_4	$\{9, 1, 2, 3\} \Leftrightarrow \{0, 1, 2, 10\}$
R_5	$\{0, 6, 2, 3\} \Leftrightarrow \{0, 1, 2, 4\}$
R_6	$\{0, 1, 2, 10\} \Leftrightarrow \{0, 1, 2, 10\}$
R_7	$\{7, 1, 2, 3\} \Leftrightarrow \{0, 1, 2, 4\}$
R_8	$\{0, 1, 11, 3\} \Leftrightarrow \{0, 1, 2, 10\}$

No entanto, é possível adoptar uma definição de vizinhança alternativa, aplicável apenas ao conjunto de representações canónicas. Note-se que a adição de uma mesma constante c aos *zeros* de duas representações vizinhas origina duas outras representações, também elas vizinhas uma da outra.

Definição 4.4 (Representações neutras canónicas vizinhas). *Dois representações neutras canónicas de $NN_g(\ell, k)$ definidas pelos conjuntos de zeros $R_1 = \{z_{1,0}, z_{1,1}, \dots, z_{1,2^{\ell-k}-1}\}$ e $R_2 = \{z_{2,0}, z_{2,1}, \dots, z_{2,2^{\ell-k}-1}\}$ dizem-se vizinhas se e só se existir uma terceira representação R_3 , não necessariamente canónica, vizinha de R_1 segundo a definição 4.3, que seja equivalente a R_2 .*

O algoritmo 4.3 calcula as representações vizinhas de uma representação neutra canónica. Uma representação neutra de $NN_g(\ell, k)$ terá, no máximo, $2^{\ell-k}\ell$ representações vizinhas, podendo algumas ser iguais à própria representação por questões de simetria. De salientar que uma representação vizinha só é válida se a rede neutra for conectada e só é analisada depois de encontrada a sua representação canónica equivalente.

As representações vizinhas de $\{0, 1, 2, 3\} \in NN_g(4, 2)$ e as representações canónicas que lhes são equivalentes estão indicadas na tabela 4.8.

Algoritmo 4.3 Calcula representações vizinhas *RepViz*

- 1: Entrada: Representação neutra $\{0, z_1, \dots, z_{2^{\ell-k}-1}\} \in NN_g(\ell, k)$
 - 2: Saída: Representações vizinhas da representação neutra *RepViz*
 - 3: $R = \{z_0, z_1, \dots, z_{2^{\ell-k}-1}\}$
 - 4: Conjunto de representações vizinhas está vazio $RepViz = \emptyset$
 - 5: **while** houver zeros $z_i \in R$ **do**
 - 6: Calcula vizinhos de zero z_i , tal que $Viz(z_i) = \{viz_1(z_i), viz_2(z_i), \dots, viz_\ell(z_i)\}$
 - 7: **while** houver vizinhos $viz_m \in Viz(z_i)$ **do**
 - 8: Calcula síndrome $s(viz_m)$
 - 9: Procura zero com mesma síndrome $s(z_j) = s(viz_m)$
 - 10: Substitui o zero encontrado z_j por viz_m
 - 11: **if** representação R é conectada **then**
 - 12: Determina a representação canónica equivalente a R
 - 13: Acrescenta a *RepViz*
 - 14: **end if**
 - 15: **end while**
 - 16: **end while**
-

4.4.1 Fecho da família de representações neutrais sobre a relação de *Vizinhança*

A determinação de representações canónicas da família $NN_g(\ell, k)$ pode ser realizada com base no cálculo do fecho de uma representação canónica inicial sob a relação de *Vizinhança* entre representações canónicas (definição 4.4). Admitindo que o conjunto de representações canónicas é fechado sob a relação de vizinhança, este cálculo permitirá enumerar todas essas representações. Na ausência de demonstração da validade deste pressuposto, pode-se mesmo assim afirmar que será encontrada, pelo menos, uma parte das representações pretendidas.

O algoritmo começa com a representação $\{0, 1, \dots, 2^{\ell-k} - 1\}$. Esta é uma representação canónica de $NN_g(\ell, k)$, qualquer que seja ℓ e k e corresponde à representação com redundância não codificante. Assim, o conjunto de partida

Tabela 4.9: Número de representações neutrais de $NN_g(\ell, k)$ - Parte 1

	1	2	3	4	5	6	k
$x+1$	2	3	4	5	6	7	
x^2+x+1	4	9	18	32	50	75	
x^3+x+1	19	266	1828	8128	25100	66750	
x^4+x+1	1489	1845628					
$g(x)$							

é:

$$A = \{\{0, z_1, \dots, z_{2^{\ell-k}-1}\}\}$$

O conjunto D das representações vizinhas de cada representação é calculado pelo algoritmo 4.3. O algoritmo 4.4 realiza o cálculo do fecho A sob a relação de *Vizinhança*.

Algoritmo 4.4 Calcula fecho A sob a relação *Vizinhança*

- 1: Entrada: $A = \{0, 1, \dots, 2^{\ell-k} - 1\} \in NN_g(\ell, k)$
 - 2: Saída: $FechoA$
 - 3: $FechoA = A$
 - 4: **while** houver representações para analisar em $FechoA$ **do**
 - 5: Calcula D o conjunto das representações canônicas vizinhas da representação em análise
 - 6: **for** cada representação vizinha em D **do**
 - 7: **if** representação vizinha não pertence a $FechoA$ **then**
 - 8: Acrescenta a $FechoA$
 - 9: **end if**
 - 10: **end for**
 - 11: **end while**
-

As tabelas 4.9 e 4.10 apresentam o número de representações canônicas neutrais de $NN_g(\ell, k)$ encontradas para $0 < k \leq 11$, quando o número de bits redundantes se situa em $0 < \ell - k \leq 4$ (polinômios geradores com grau $0 < \ell - k \leq 4$).

Como se pode verificar, o número de representações neutrais cresce rapidamente com k e muito rapidamente com o número de bits redundantes $\ell - k$. Este número elevadíssimo cria dificuldades não só ao nível do tempo de

Tabela 4.10: Número de representações neutrais de $NN_g(\ell, k)$ - Parte 2

	7	8	9	10	11	k
$x+1$	8	9	10	11	12	
x^2+x+1	108	147	196	256	324	
x^3+x+1	158784	342701	690748	1307528	2350336	
x^4+x+1						
$g(x)$						

execução, mas também da capacidade de armazenamento das representações geradas. Por exemplo, para um espaço $|\Phi_f| = 2^2 = 4$, ao serem acrescentados 3 bits redundantes, são obtidas 266 representações neutrais. Quando são acrescentados 4 bits redundantes, o número de representações neutrais atinge 1845628.

Como o número de representações neutrais obtido para alguns $NN_g(\ell, k)$ é muito elevado, decidiu-se utilizar uma biblioteca freeware, a Oracle Berkely DB, que proporciona um alto desempenho em bases de dados embebidas e pode ser utilizada com as linguagens de programação C, C++, Java, Perl, Python, entre outras. Neste caso, foi utilizada a linguagem C. Esta base de dados permite manipular dados na ordem dos terabytes em vários sistemas, incluindo UNIX e Microsoft Windows.

4.5 Propriedades de $NN_g(\ell, k)$

As propriedades usadas na literatura para caracterizar representações redundantes foram explicadas na secção 2.5. Estas propriedades serão agora discutidas no âmbito da família $NN_g(\ell, k)$.

4.5.1 Uniformidade

A família $NN_g(\ell, k)$ é uniforme porque todos os fenótipos são representados pelo mesmo número de genótipos. Em cada representação de $NN_g(\ell, k)$, cada

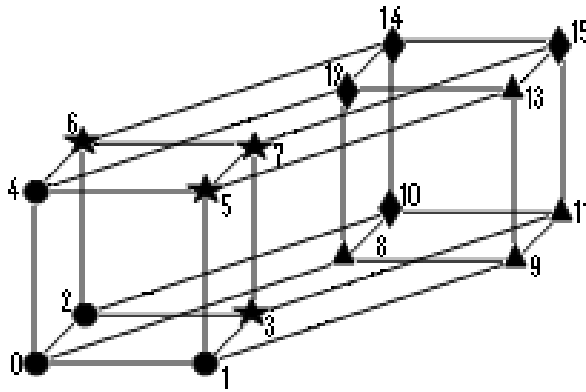


Figura 4.5: Hipergrafo representando o espaço genotípico com $\ell = 4$ e os fenótipos relativos à representação neutral $\{0, 1, 2, 4\} \in NN_g(4, 2)$

Tabela 4.11: Genótipos e respectivos fenótipos da representação neutral $\{0, 1, 2, 4\} \in NN_g(4, 2)$

síndrome 0	síndrome 1	síndrome 2	síndrome 3
0000 (0) [0]	0001 (1) [0]	0010 (2) [0]	0100 (4) [0]
0111 (7) [1]	0110 (6) [1]	0101 (5) [1]	0011 (3) [1]
1001 (9) [2]	1000 (8) [2]	1011 (11) [2]	1101 (13) [2]
1110 (14) [3]	1111 (15) [3]	1100 (12) [3]	1010 (10) [3]

indicada. Sendo a representação uniforme, cada fenótipo é representado por $2^{\ell-k} = 2^2 = 4$ genótipos, o que corresponde ao número de *zeros* da representação neutral.

Como cada fenótipo é representado por uma rede neutral com $2^{\ell-k}$ genótipos e cada um desses genótipos tem ℓ genótipos vizinhos, incluindo, pelo menos, um genótipo da mesma rede, a vizinhança fenotípica de x_f terá, no máximo, $2^{\ell-k}(\ell - 1)$ fenótipos diferentes. O número exacto de fenótipos diferentes nesta vizinhança determina a conectividade exibida pela representação. Na figura 4.5, partindo de qualquer fenótipo é possível alcançar

Tabela 4.12: Genótipos vizinhos e respectivos fenótipos da rede neutral que representa o fenótipo 1 na representação $\{0, 1, 2, 4\} \in NN_g(4, 2)$

	1	2	3	4
0111	(6)[1]	(5)[1]	(3)[1]	(15)[3]
0110	(7)[1]	(4)[0]	(2)[0]	(14)[3]
0101	(4)[0]	(7)[1]	(1)[0]	(13)[2]
0011	(2)[0]	(1)[0]	(7)[1]	(11)[2]

os outros três fenótipos. Portanto, a conectividade desta representação é 3, porque é possível aceder aos 3 fenótipos possíveis (excluindo o próprio).

Como a topologia da rede neutral de uma representação é igual para todos os fenótipos, a conectividade também é igual para todos os outros fenótipos, embora os fenótipos acessíveis sejam, naturalmente, diferentes. Por exemplo, a tabela 4.12 mostra os genótipos alcançáveis a partir da rede neutral que representa o fenótipo 1, e os fenótipos correspondentes a esses genótipos.

Comparando com uma representação não redundante com fenótipo com comprimento 2, onde cada fenótipo só consegue alcançar 2 fenótipos diferentes, a representação neutral $\{0, 1, 2, 4\} \in NN_g(4, 2)$ consegue alcançar um número maior de fenótipos diferentes, neste caso 3 em vez de apenas 2. De facto, todas as representações neutrais $NN_g(\ell, k)$ têm conectividade superior ou igual à conectividade da representação não redundante correspondente, ou seja, todas têm conectividade superior ou igual a k .

Outro aspecto importante a destacar é o grau de neutralidade de uma representação neutral, que corresponde à percentagem de fenótipos vizinhos que são neutrais. Por exemplo, o grau neutral de $\{0, 1, 2, 4\} \in NN_g(4, 2)$ é $6/16 = 0.375$, porque dos 16 genótipos vizinhos dos *zeros* desta representação, 6 correspondem ao fenótipo 0. Para os outros fenótipos da mesma representação neutral, o grau de neutralidade é o mesmo. Por último, o

grau de neutralidade varia entre as representações neutrais que pertencem à mesma família $NN_g(\ell, k)$.

Em conclusão, a forma como as redes neutrais destas representações estão estruturadas é tal que:

1. A conectividade de qualquer representação neutral pertencente à família $NN_g(\ell, k)$ é maior ou igual à conectividade da representação não redundante correspondente, ou seja, maior ou igual a k ;
2. A conectividade de uma representação neutral não varia com o fenótipo. O número de fenótipos diferentes acessíveis é igual para todos os fenótipos de uma mesma representação neutral;
3. Os diferentes genótipos de uma rede neutral podem alcançar fenótipos diferentes;
4. O grau de neutralidade varia de representação para representação.

Por último, as tabelas 4.13 e 4.14 apresentam a conectividade mínima e a conectividade máxima encontradas para cada família de $NN_g(\ell, k)$. Por exemplo, a conectividade mínima e a conectividade máxima de $NN_g(7, 4)$ são 4 e 14, respectivamente. Para os conjuntos de representações neutrais cuja enumeração não foi possível, um limite inferior para a conectividade máxima foi obtido gerando as representações neutrais a partir da representação neutral com maior conectividade até ao momento, e deixando o algoritmo executar por um longo período de tempo. Estes valores aparecem dentro de parêntesis curvos.

De realçar que a conectividade máxima parece aumentar exponencialmente com o número de bits redundantes $\ell - k$. Quando se compara a conectividade máxima alcançada por uma representação neutral e por uma

Tabela 4.13: Conectividade mínima e máxima para cada $NN_g(\ell, k)$ - Parte 1

	1	2	3	4	5	6	k
$x + 1$	1-1	2-2	3-3	4-4	5-5	6-6	
$x^2 + x + 1$	1-1	2-3	3-5	4-7	5-9	6-11	
$x^3 + x + 1$	1-1	2-3	3-7	4-14	5-18	6-22	
$x^4 + x + 1$	1-1	2-3	3-(7)	4-(15)	5-(30)	6-(43)	
$x^5 + x^2 + 1$	1-1	-(3)	-(7)	-(15)	-(31)	-(59)	
$g(x)$							

Tabela 4.14: Conectividade mínima e máxima para cada $NN_g(\ell, k)$ - Parte 2

	7	8	9	10	11	k
$x + 1$	7-7	8-8	9-9	10-10	11-11	
$x^2 + x + 1$	7-13	8-15	9-17	10-19	11-21	
$x^3 + x + 1$	7-26	8-30	9-34	10-38	11-42	
$x^4 + x + 1$	7-(55)	8-(63)	9-	10-	11-(91)	
$x^5 + x^2 + 1$	-(98)	-(125)				
$g(x)$						

representação não neutral com igual número de bits do genótipo, a primeira permite um aumento substancial da conectividade entre fenótipos.

4.5.3 Topologia

A topologia ou forma de uma rede neutral pode ser visualizada utilizando uma técnica denominada MDS (Multi-Dimensional Scaling) (de Leeuw 2000). Esta técnica consiste em determinar posições para um conjunto de pontos no espaço Euclidiano, tais que as distâncias entre eles aproximem uma medida de dissemelhança dos objectos que se pretendem visualizar. Deste modo, uma imagem da topologia de uma rede neutral pode ser produzida a partir das distâncias de Hamming entre os genótipos da rede.

A figura 4.6 mostra as topologias das redes neutrais em algumas representações de $NN_g(7, 4)$. Enquanto a representação não codificante, onde a redundância é considerada ‘morta’ apresenta a topologia de um cubo, a representação correspondente ao código corrector de erros $C(7, 4)$ apresenta uma topologia em estrela, onde todos os *zeros*, excepto o primeiro, estão

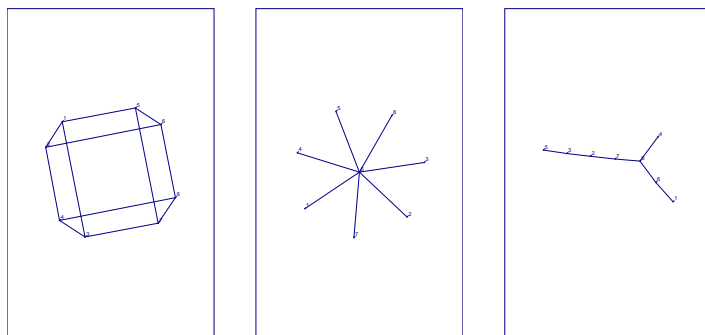


Figura 4.6: MDS de algumas representações de $NN_g(7, 4)$

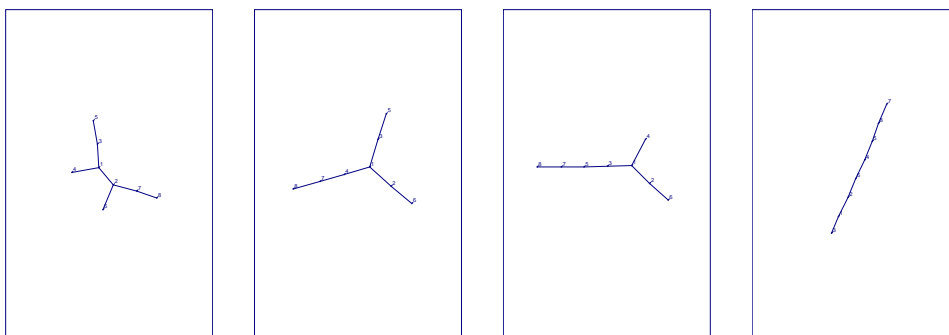


Figura 4.7: MDS de algumas representações de $NN_g(14, 11)$

equidistantes deste. A representação com topologia em Y corresponde a uma representação com a conectividade máxima (14) da família $NN_g(7, 4)$. O anexo A apresenta as 69 topologias diferentes encontradas para a família $NN_g(7, 4)$.

Na figura 4.7 apresentam-se as topologias de $NN_g(14, 11)$ que apresentam maior conectividade (42) e na figura 4.8 mostram-se as topologias de algumas representações de $NN_g(6, 2)$.

4.5.4 Sinonímia

O cálculo da sinonímia pode ser simplificado na representação neutral, utilizando a expressão 4.14, porque as distâncias entre todos os genótipos da

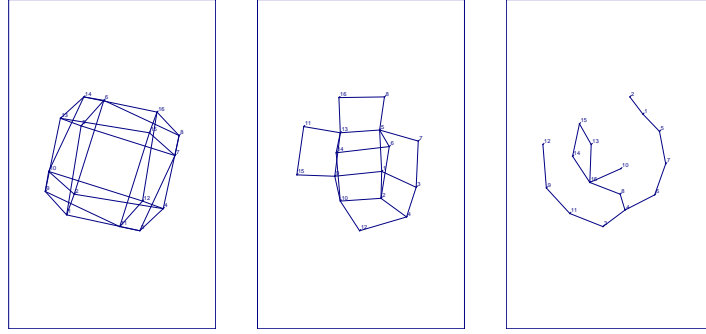


Figura 4.8: MDS de algumas representações de $NN_g(6, 2)$

Tabela 4.15: Sinonímia mínima e máxima para cada $NN_g(\ell, k)$ - Parte 1

	1	2	3	4	5	6	k
$x + 1$	1	1	1	1	1	1	
$x^2 + x + 1$	1.33-1.5	1.33-1.67	1.33-1.67	1.33-1.67	1.33-1.67	1.33-1.67	
$x^3 + x + 1$	1.71-2.18	1.71-2.71	1.71-2.86	1.71-2.86	1.71-3	1.71-3	
$x^4 + x + 1$	2.13-2.67						
$g(x)$							

rede neutra que representa determinado fenótipo são iguais para todos os fenótipos. Assim:

$$S_{in} = \frac{\sum_{i=0}^{2^{\ell-k}-1} \sum_{j=0}^{2^{\ell-k}-1} d(z_i, z_j)}{(2^{\ell-k}) \times (2^{\ell-k} - 1)} \quad (4.14)$$

onde $d(z_i, z_j)$ corresponde à distância de Hamming entre o zero z_i e o zero z_j . Uma representação é considerada altamente sinónima quando o valor de S_{in} é baixo e pouco sinónima quando este valor é elevado.

Como a sinonímia é uma medida de quão próximos estão os genótipos que descodificam para o mesmo fenótipo, a sinonímia está relacionada com a topologia das redes neutrais. As tabelas 4.15 e 4.16 apresentam a sinonímia mínima e máxima determinadas para diversos $NN_g(\ell, k)$.

Outra medida de sinonímia é a maior distância entre cada par de zeros da representação (diâmetro). Por exemplo, a rede com topologia em estrela tem diâmetro 2 enquanto a rede com topologia em cubo tem diâmetro 3.

Tabela 4.16: Sinonímia mínima e máxima para cada $NN_g(\ell, k)$ - Parte 2

	7	8	9	10	11	k
$x+1$	1	1	1	1	1	
x^2+x+1	1.33-1.67	1.33-1.67	1.33-1.67	1.33-1.67	1.33-1.67	
x^3+x+1	1.71-3	1.71-3	1.71-3	1.71-3	1.71-3	
x^4+x+1						
$g(x)$						

Tabela 4.17: Diâmetro mínimo e máximo de $NN_g(\ell, k)$

	1	2	3	4	5	6	7	8	9	10	11	k
$x+1$	1	1	1	1	1	1	1	1	1	1		
x^2+x+1	2	2-3	2-3	2-3	2-3	2-3	2-3	2-3	2-3	2-3	2-3	
x^3+x+1	3-4	3-5	3-6	2-6	2-7	2-7	2-7	2-7	2-7	2-7	2-7	
x^4+x+1	4-5	4-6									(4)-(14)	
$g(x)$												

Também é possível verificar que, sendo o número de *zeros* igual a $2^{\ell-k}$ e pelo facto da rede ser conectada, o diâmetro máximo é simultaneamente menor que o número de *zeros* e menor ou igual ao comprimento do genótipo, ou seja, o diâmetro máximo é menor ou igual a $\min(2^{\ell-k} - 1, \ell)$.

Na tabela 4.17 são apresentados os diâmetros mínimo e máximo encontrados para diversas $NN_g(\ell, k)$.

4.5.5 Localidade

Na representação neutral, o valor da localidade não varia com os diferentes fenótipos, pelo que pode ser calculada com base num único fenótipo, o que conduz à simplificação da fórmula apresentada na secção 2.5. O cálculo da localidade para $NN_g(\ell, k)$ pode ser efectuado considerando a média dos pesos dos fenótipos correspondentes aos genótipos vizinhos dos *zeros* da representação. As tabelas 4.18 e 4.19 apresentam a localidade mínima e a localidade máxima calculadas para diversas $NN_g(\ell, k)$.

Tabela 4.18: Localidade mínima e máxima para cada $NN_g(\ell, k)$ - Parte 1

	1	2	3	4	5	6	k
$x + 1$	0.5	0.67-1.0	0.75-1.25	0.8-1.4	0.83-1.5	0.86-1.57	
$x^2 + x + 1$	0.33-0.5	0.5-1.0	0.6-1.5	0.67-1.67	0.71-1.86	0.75-2.0	
$x^3 + x + 1$	0.25-0.50	0.4-1.0	0.5-1.58	0.57-2.0	0.62-2.38	0.67-2.78	
$x^4 + x + 1$	0.2-0.57						
$g(x)$							

Tabela 4.19: Localidade mínima e máxima para cada $NN_g(\ell, k)$ - Parte 2

	7	8	9	10	11	k
$x + 1$	0.88-1.62	0.89-1.67	0.9-1.7	0.91-1.73	0.92-1.75	
$x^2 + x + 1$	0.78-2.0	0.8-2.1	0.82-2.18	0.83-2.17	0.85-2.23	
$x^3 + x + 1$	0.7-2.95	0.73-3.14	0.75-3.25	0.77-3.27	0.79-3.36	
$x^4 + x + 1$						
$g(x)$						

4.5.6 Relação entre as propriedades

A figura 4.9 apresenta os gráficos de dispersão que mostram a relação entre a conectividade, a localidade e a sinonímia para a família $NN_g(14, 11)$. Embora apenas se apresentem as características para a família $NN_g(14, 11)$, os resultados são similares para as restantes famílias de $NN_g(\ell, k)$ calculadas.

Estes gráficos de dispersão demonstram que existem representações com localidade relativamente elevada, mesmo para valores de conectividade elevados. O segundo gráfico da figura contradiz a ideia apresentada em Rothlauf (2006), de que representações redundantes altamente sinónimas não permitem o aumento da conectividade. Como é possível verificar pelo gráfico, existem representações neutrais altamente sinónimas que apresentam uma conectividade elevada.

Em suma, a família $NN_g(\ell, k)$ apresenta um leque variado de representações com características diversas, o que irá permitir estudar, no capítulo 5, até que ponto essas características podem influenciar o desempenho de um algoritmo evolutivo.

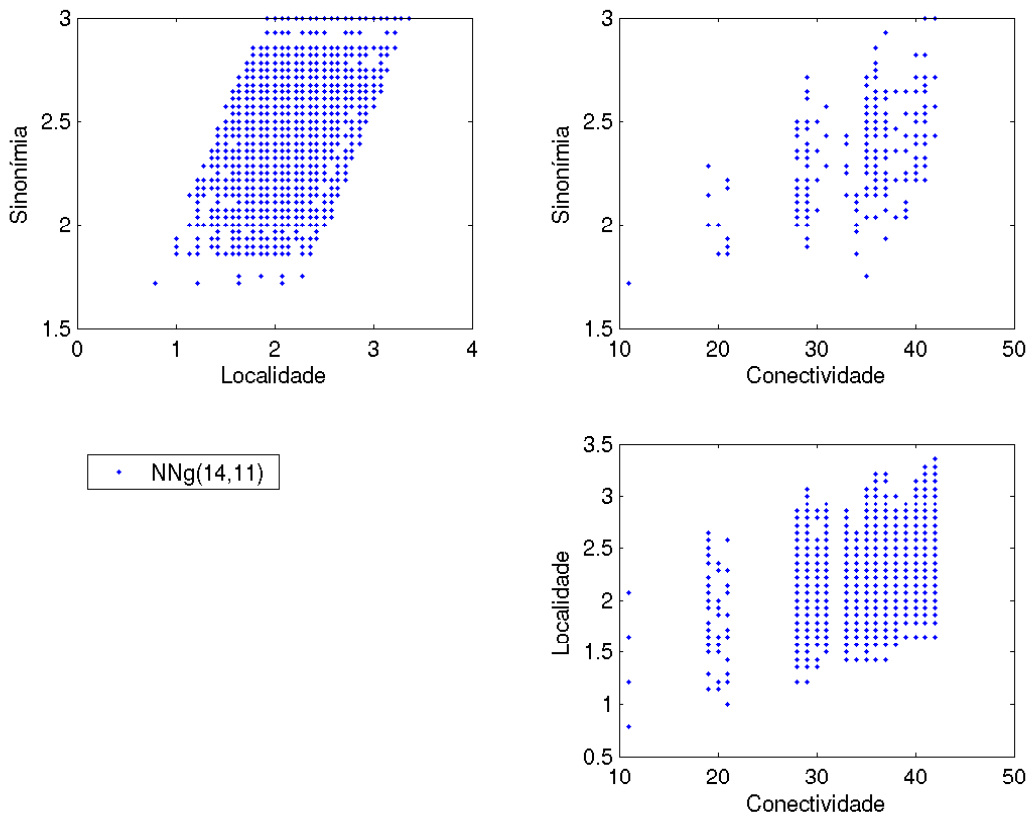


Figura 4.9: Relação entre algumas propriedades de $NN_g(14, 11)$

4.6 Considerações finais

Neste capítulo foram apresentadas duas novas famílias de representações binárias redundantes, a primeira exibindo as propriedades de redundância, poligenia e pleiotropia, a segunda acrescentando a neutralidade.

A família de representações não neutras $NonNN_Z(\ell', k)$ baseia-se em transformações lineares e permite a definição da vizinhança fenotípica desejada de modo simples e directo.

Os códigos de controlo de erros foram utilizados para definir uma família de representações neutras $NN_g(\ell, k)$, especificadas pela rede que representa o fenótipo e que determina as restantes características da representação. Para cada família $NN_g(\ell, k)$, as representações neutras foram geradas utilizando o algoritmo do fecho.

Foram analisadas as principais características das representações geradas, nomeadamente, a conectividade, a localidade e a sinonímia, bem como as relações entre elas. A família $NN_g(\ell, k)$ revelou-se muito rica em representações com características diversas, fornecendo uma base importante para o estudo das relações entre as características das representações e a sua influência no desempenho de algoritmos evolutivos.

De referir que a família de representações neutras proposta permite um aumento substancial da conectividade entre fenótipos, quando comparada com as representações não neutras. O aumento da conectividade em $NonNN_Z(\ell', k)$ é conseguido à custa da adição de redundância na representação, sendo necessários mais bits redundantes para implementar a mesma vizinhança fenotípica da representação neutral correspondente. Para além disso, na representação não neutral, todos os genótipos que descodificam para o mesmo fenótipo são equivalentes em termos da vizinhança fenotípica.

No capítulo seguinte, é estudado o desempenho das famílias $NN_g(\ell, k)$ e

$NonNN_Z(\ell', k)$, quando utilizadas num AE simples, aplicado em paisagens de aptidão NK .

Capítulo 5

Desempenho das famílias de representações redundantes

5.1 Introdução

Neste capítulo, pretende-se estudar a influência das representações das famílias $NN_g(\ell, k)$ e $NonNN_Z(\ell', k)$ no desempenho de um AE simples. Para o efeito, é considerada uma estratégia evolutiva $(1+1)$ -ES (Rudolph 2000) aplicada a paisagens de aptidão NK com diferentes graus de rugosidade, sendo o seu comportamento modelado através de cadeias de Markov.

Especificamente, pretende-se analisar o efeito da redundância e da neutralidade das representações no desempenho de um AE. Este desempenho pode ser entendido, quer em termos da probabilidade de atingir o óptimo global, quer em termos do valor esperado da aptidão da melhor solução encontrada, sendo de interesse tanto a velocidade de convergência do algoritmo como o seu comportamento a longo prazo. Estas questões foram abordadas através de um estudo experimental baseado nas famílias de representações neutrais e não neutrais calculadas anteriormente.

O capítulo começa por apresentar as paisagens de aptidão NK , as noções de óptimo local e de óptimo global e a definição de rede neutral localmente óptima. A secção 5.3 explica a modelação da $(1+1)$ - ES utilizando as cadeias de Markov. Na secção 5.4 são apresentados os resultados do estudo experimental realizado. Por último, são discutidas as implicações dos resultados obtidos e as conclusões.

5.2 Paisagens de aptidão NK

As paisagens de aptidão NK , propostas por Kauffman (1995), baseiam-se nas paisagens de aptidão propostas por Wright (1932). Wright verificou que quando existem interacções entre os genes, a composição genética de uma população pode evoluir para múltiplos domínios de atracção (Altenberg 1997). Os genes podem não só depender de outros genes para expressar as características do fenótipo, como podem também suprimir ou activar os efeitos de outros genes. A interacção entre genes denomina-se epistasia. Wright encontrou uma ligação conceptual entre essa propriedade microscópica dos organismos (epistasia) e uma propriedade macroscópica da dinâmica evolutiva (múltiplos atractores da população no espaço dos genótipos), e utilizou a analogia com uma paisagem com múltiplos picos para ilustrar esta situação. As paisagens de aptidão podem ser definidas como (Verel 2005):

Definição 5.1 (Paisagem de aptidão). *Uma paisagem é um triplete (S, V, f) onde S é um conjunto de soluções potenciais que corresponde ao espaço de procura, $V : S \rightarrow 2^S$ é uma função que atribui a cada $s \in S$ um conjunto de vizinhos $V(s) \subset S$, definindo a estrutura de vizinhança, e $f : S \rightarrow \mathbb{R}$ é uma função de aptidão que atribui um valor de aptidão a cada solução s .*

Kauffman utilizou as paisagens de aptidão de Wright ao desenvolver as

paisagens de aptidão NK , a fim de poder explorar a forma como a epistasia controla a rugosidade da paisagem. Para o efeito, desenvolveu uma família de funções de aptidão cuja rugosidade poderia ser moldada por um único parâmetro. As paisagens de aptidão NK são definidas como funções de aptidão geradas estocasticamente, aplicadas a cadeias de bits com N genes e interações entre grupos de $K + 1$ genes.

O modelo NK é um método estocástico para gerar funções de aptidão $F : \{0, 1\}^N \mapsto \mathbb{R}^+$, onde o genótipo $x \in \{0, 1\}^N$ possui N loci, com dois alelos possíveis em cada locus x_i . Os parâmetros principais do modelo são N , o comprimento das strings binárias, ou cromossomas, que correspondem aos pontos na paisagem, e K , o número de outros genes que interagem com cada gene em particular. A aptidão de cada gene depende do seu valor e dos valores dos genes com que interage. Para cada instância da paisagem, os valores de aptidão associados a cada gene são realizações de uma variável aleatória com distribuição uniforme entre 0.0 e 1.0, sendo a aptidão de cada cromossoma a média das aptidões dos genes para todos os N loci.

A rugosidade da paisagem é controlada pelo parâmetro K e é máxima quando $K = N - 1$ e mínima quando $K = 0$. Quando $K = 0$, não há epistasia, a paisagem apresenta um único óptimo local (o óptimo global), e a aptidão dos diferentes cromossomas está altamente correlacionada com a distância de Hamming entre eles. Quando $K = N - 1$, o número de interações entre os genes é máximo e a função é puramente aleatória, pelo que existem muitos óptimos locais na paisagem e a aptidão dos cromossomas não está correlacionada com a distância de Hamming entre eles.

Existem duas alternativas quanto à escolha dos K genes: a vizinhança adjacente, onde os K genes correspondem aos K genes mais próximos do locus que está sendo avaliado, e a vizinhança aleatória, onde os K genes são

escolhidos aleatoriamente no cromossoma.

A complexidade computacional do problema de otimização das paisagens NK baseadas na vizinhança adjacente é polinomial em N , enquanto o problema baseado na vizinhança aleatória é NP-difícil para $K > 1$ e polinomial para $K = 1$ (Wright, Thompson & Zhang 2000).

As paisagens de aptidão NK são uma estrutura de análise extremamente flexível que tem sido aplicada em diversas áreas, como a biologia evolutiva, a química, a física, a computação e a investigação operacional, entre outras (Kimura, Basso, Kayo & Suen 2008). Têm sido aplicadas em problemas na área da gestão, como referido em Rivkin & Siggelkow (2002), onde se destacam os estudos sobre a modelação do comportamento e evolução das organizações (Levinthal 1997, Levitan, Lobo, Schuler & Kauffman 2002, Rivkin 2000), a adopção de tecnologias nas organizações (Kauffman 2000, Lobo, Miller & Fontana 2004) e os mecanismos de organização social e económica (Kauffman 1995).

5.2.1 Óptimo global e óptimos locais nas paisagens NK

Um óptimo local corresponde a um ponto da paisagem de aptidão que não tem vizinhos com aptidão superior. Um óptimo local será o óptimo global se for a melhor solução do problema. Considerando um problema de maximização (Verel 2005):

Definição 5.2 (Óptimo local). *Dada uma paisagem de aptidão (S, V, f) , uma solução $s^* \in S$ é um óptimo local se e só se:*

$$\forall s \in V(s^*), f(s) \leq f(s^*) \quad (5.1)$$

Definição 5.3 (Óptimo global). *Dada uma paisagem de aptidão (S, V, f) ,*

uma solução $s^* \in S$ é um *ótimo global* se e só se:

$$\forall s \in S, f(s) \leq f(s^*) \quad (5.2)$$

A subsecção seguinte apresenta a forma como a escolha da representação afecta as paisagens de aptidão.

5.2.2 Representação não redundante versus representação neutral

As paisagens de aptidão NK , por poderem ir gradualmente da mais suave até à mais rugosa, são um bom modelo para estudar o efeito das representações neutrais propostas. Uma característica importante das paisagens de aptidão é a sua rugosidade, que pode ser avaliada em termos do número e da distribuição dos ótimos locais, ou seja, da sua topologia, que é induzida pelo operador que é utilizado para definir a vizinhança (Reeves 1999).

Como as paisagens NK são definidas sobre o espaço binário, a métrica de Hamming é utilizada e a relação de vizinhança pode ser representada por um grafo. A figura 5.1 representa uma possível paisagem de aptidão $NK(3, 2)$, onde $N = 3$ e $K = 2$. Nesta paisagem, o *ótimo global* corresponde ao genótipo 100 com aptidão 0.640, enquanto 001, 010 e 111 são *ótimos locais* com aptidões 0.589, 0.621 e 0.487, respectivamente. Graficamente, o *ótimo global* na figura 5.1 está representado como um rectângulo, enquanto os restantes *ótimos locais* estão representados por círculos.

As representações neutrais do tipo $NN_g(\ell, k)$, uma vez que podem apresentar valores elevados de conectividade, deverão permitir reduzir o número de *ótimos locais* nestas paisagens. As representações neutrais de $NN_g(\ell, k)$ podem ser utilizadas em paisagens de aptidão $NK(k, K)$ com

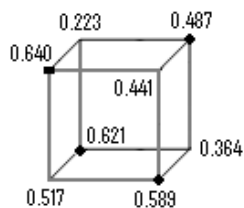


Figura 5.1: Hipergrafo representando uma paisagem de aptidão $NK(3,2)$

Tabela 5.1: Genótipos que mapeiam para cada fenótipo de $\{0, 1, 2, 3, 4, 5, 6, 17\} \in NN_g(6, 3)$

síndrome	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	17
1	11	10	9	8	15	14	13	26
2	22	23	20	21	18	19	16	7
3	29	28	31	30	25	24	27	12
4	39	38	37	36	35	34	33	54
5	44	45	46	47	40	41	42	61
6	49	48	51	50	53	52	55	32
7	58	59	56	57	62	63	60	43
fenótipo								

$K = 0, 1, 2, \dots, k - 1$.

Um exemplo é apresentado a seguir, utilizando a representação neutral $\{0, 1, 2, 3, 4, 5, 6, 17\} \in NN_g(6, 3)$ e a paisagem de aptidão da figura 5.1. O mapeamento genótipo-fenótipo da representação neutral é apresentado na tabela 5.1, onde os genótipos que descodificam para o mesmo fenótipo aparecem na mesma linha, ordenados pela síndrome. De modo a tornar a tabela mais compacta, todos os valores são apresentados em decimal.

A figura 5.2 mostra um hipergrafo representando o espaço dos genótipos quando se utiliza a referida representação. Por uma questão de clareza, apenas alguns genótipos são indicados na figura. Para os restantes, apenas

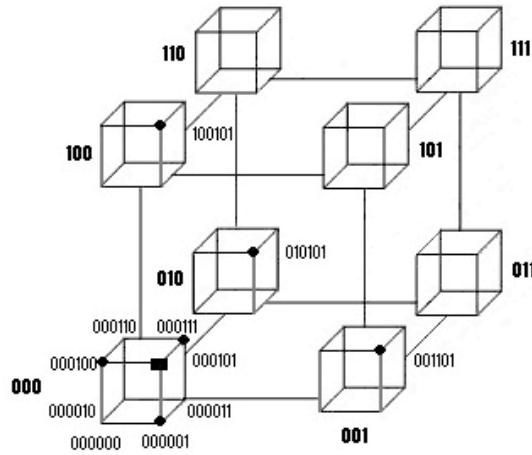


Figura 5.2: Hipergrafo representando um espaço genotípico binário com $\ell = 6$ os 3 primeiros bits mais significativos são apresentados. Como $\ell = 6$, cada genótipo tem 6 vizinhos. A título de exemplo, encontram-se indicados na figura 5.2, um genótipo (■) e os seus vizinhos (●). Para melhor perceber o exemplo, o hipergrafo da figura 5.2 é reproduzido na figura 5.3 com a notação em decimal.

Atribuindo a cada genótipo da representação neutral considerada a aptidão do fenótipo correspondente, a paisagem de aptidão da figura 5.1, que apresenta 4 ótimos locais, é transformada numa paisagem com apenas um ótimo (global), como se pode verificar na figura 5.4.

Tomemos como exemplo o fenótipo 1, com aptidão 0.589, que corresponde a um ótimo local quando se utiliza a representação não redundante. Nas tabelas 5.2 e 5.3 estão representados os genótipos (entre parêntesis curvos), os respectivos fenótipos (entre parêntesis rectos) e a aptidão dos vizinhos da rede neutral que representa o fenótipo 1. Como é possível verificar, este fenótipo deixa de ser um ótimo local porque existem vizinhos, neste caso os

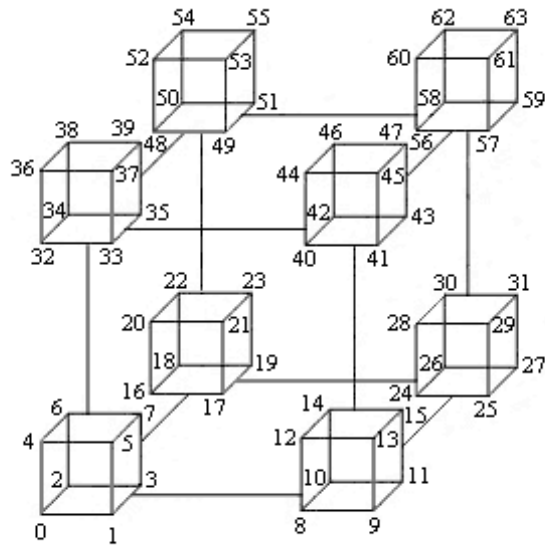


Figura 5.3: Hipergrafo representando o espaço dos genótipos com $\ell = 6$ em decimal

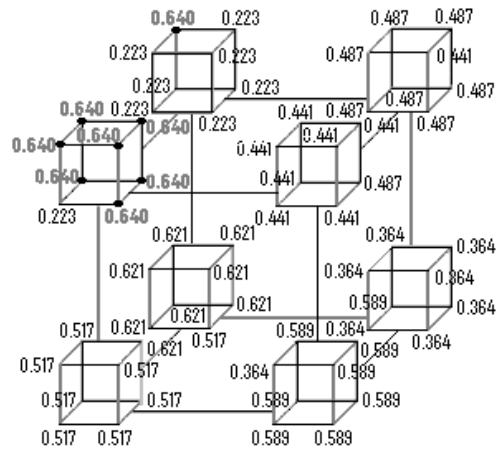


Figura 5.4: Hipergrafo representando uma paisagem de aptidão $NK(3,2)$ utilizando a representação neutra $\{0, 1, 2, 3, 4, 5, 6, 17\} \in NN_g(6, 3)$

Tabela 5.2: Vizinhos da rede neutral que representa o fenótipo 1 na representação $\{0, 1, 2, 3, 4, 5, 6, 17\} \in NN_g(6, 3)$ - Parte 1

síndrome	0	1	2	3
11		(10)[1] 0.589	(9)[1] 0.589	(3)[1] 0.517
10	(11)[1] 0.589		(2)[0] 0.517	(8)[1] 0.589
9	(11)[1] 0.589	(1)[0] 0.517		(8)[1] 0.589
8	(0)[0] 0.517	(10)[1] 0.589	(9)[1] 0.589	
15	(11)[1] 0.589		(31)[3] 0.364	(47)[5] 0.441
14		(10)[1] 0.589	(46)[5] 0.441	(30)[3] 0.364
13	(29)[3] 0.364	(45)[5] 0.441	(9)[1] 0.589	
26	(58)[7] 0.487	(10)[1] 0.589		(30)[3] 0.364
genótipo				

Tabela 5.3: Vizinhos da rede neutral que representa o fenótipo 1 na representação $\{0, 1, 2, 3, 4, 5, 6, 17\} \in NN_g(6, 3)$ - Parte 2

síndrome	4	5	6	7
11	(15)[1] 0.589		(27)[3] 0.364	(43)[7] 0.487
10		(14)[1] 0.589	(42)[5] 0.589	(26)[1] 0.589
9	(25)[3] 0.364	(41)[5] 0.441	(13)[1] 0.589	
8	(40)[5] 0.0441	(24)[3] 0.364		(12)[3] 0.364
15		(14)[1] 0.589	(13)[1] 0.589	(7)[2] 0.621
14	(15)[1] 0.589		(6)[0] 0.517	(12)[3] 0.364
13	(15)[1] 0.589	(5)[0] 0.517		(12)[3] 0.364
26	(18)[2] 0.621	(24)[3] 0.364	(27)[3] 0.364	
genótipo				

genótipos 7 e 18, que descodificam para o fenótipo 2, que tem maior aptidão (0.621). Os outros genótipos do fenótipo 1 não têm como vizinhos quaisquer genótipos com maior aptidão, mas podem alcançar um destes dois genótipos através de mutações neutrais.

Analisando os outros fenótipos que também são ótimos locais quando se utiliza a representação não redundante, verifica-se que a paisagem deixou de ser multimodal. Partindo de qualquer genótipo é possível aceder ao fenótipo 4 com aptidão 0.640. Por exemplo, do genótipo 15, saltar para 7, acabando no 39, que possui aptidão 0.640. Outro exemplo é apresentado em Correia & Fonseca (2007a).

Experiências realizadas com a família $NN_g(14, 11)$ permitiram transformar paisagens de aptidão $NK(11, 10)$ com centenas de ótimos locais em

paisagens com apenas alguns óptimos locais, apesar do número de bits redundantes introduzido ser apenas 3. Segundo Altenberg (1997), o valor esperado do número de óptimos locais numa paisagem de aptidão com $K = N - 1$ é dado por $2^N / (N + 1)$. Para as paisagens de aptidão $NK(11, 10)$, este número é $2^{11} / (11 + 1) = 170.67$. Numa instância típica de $NK(11, 10)$ com 170 óptimos, algumas representações da família $NN_g(14, 11)$ permitiram reduzir o número de óptimos locais a 31, embora outras elevassem esse número para 191.

Claramente, nem todas as representações neutrais transformam paisagens multimodais em paisagens unimodais. O contrário também pode acontecer, ou seja, existem representações que transformam paisagens de aptidão com poucos óptimos locais em paisagens de aptidão com muitos óptimos locais. Em conclusão, o número de óptimos locais pode diminuir, manter-se ou aumentar quando uma representação neutral é utilizada.

Dada a definição de rede neutral adoptada (rede conectada), torna-se necessário rever a noção de óptimo local no espaço dos genótipos. Como todos os genótipos que representam determinado fenótipo estão conectados, em vez de um óptimo local, passa a ser possível pensar numa rede neutral localmente óptima. Tendo em conta a definição de óptimo local considerada na subsecção 5.2.1, todos os genótipos da rede neutral que não alcançam directamente genótipos com melhor aptidão são óptimos locais. No entanto, basta que um dos genótipos da rede neutral não seja um óptimo local, para que seja possível atravessá-la e escapar dela através do genótipo que tem ao seu alcance genótipos de aptidão superior. Esta estrutura pode ser entendida como uma ‘ponte’ na paisagem de aptidão.

Os genótipos indicados com um círculo na figura 5.4 correspondem à rede neutral globalmente óptima.

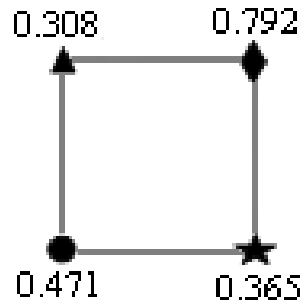


Figura 5.5: Hipergrafo representando uma instância de $NK(2, 1)$

5.2.3 Representação neutral versus representação não neutral

A seguir é apresentado um exemplo onde se ilustra a forma como uma representação não neutral transforma as paisagens de aptidão NK , em comparação com uma representação neutral. Escolhendo ambas as representações de modo que a vizinhança fenotípica seja a mesma, a paisagem no espaço dos fenótipos será igual tanto para a representação neutral como para a não neutral correspondente. No entanto, no espaço dos genótipos as paisagens serão bastante diferentes.

Como a representação não neutral é altamente redundante quando comparada com a representação neutral, será utilizada a paisagem de aptidão $NK(2, 1)$ da figura 5.5, que apresenta dois óptimos locais: o óptimo global no ponto 3 com aptidão 0.792 e o óptimo local no ponto 0 com aptidão 0.471.

Aplicando a representação neutral $\{0, 2\} \in NN_g(3, 2)$, o hipergrafo da figura 5.5 transforma-se no hipergrafo da figura 5.6, que apresenta apenas um óptimo local. O fenótipo 0 deixou de ser um óptimo local, porque, pelo menos, um dos seus genótipos apresenta um genótipo vizinho com maior aptidão. A representação neutral em questão transformou uma paisagem

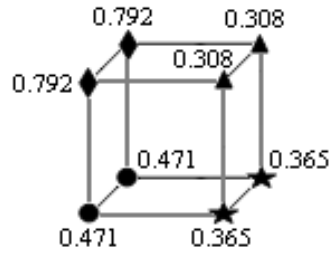


Figura 5.6: Hipergrafo representando uma paisagem de aptidão $NK(2, 1)$ utilizando a representação neutral $\{0, 2\} \in NN_g(3, 2)$

multimodal numa paisagem unimodal.

A figura 5.7 apresenta a paisagem de aptidão da figura 5.5 utilizando a representação não neutral $NonNN_Z(4, 2)$ definida pela matriz:

$$Z = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \quad (5.3)$$

construída de modo a obter a mesma vizinhança fenotípica da representação neutral $\{0, 2\} \in NN_g(3, 2)$. Os fenótipos acessíveis a partir dos fenótipos 1, 2 e 3 são $\{0, 2, 0, 2\}$, $\{3, 1, 3, 1\}$ e $\{2, 0, 2, 0\}$, respectivamente. De notar que, na representação não neutral, não podem existir genótipos vizinhos que descodifiquem para o mesmo fenótipo. Verifica-se que esta paisagem também apresenta apenas um óptimo local, o fenótipo 3, como acontece com a representação neutral. No entanto, no espaço dos genótipos, as paisagens são bastante diferentes. No caso da representação não neutral, os genótipos que correspondem ao óptimo global encontram-se afastados uns dos outros, mas como qualquer outro genótipo se encontra na bacia de atracção de um

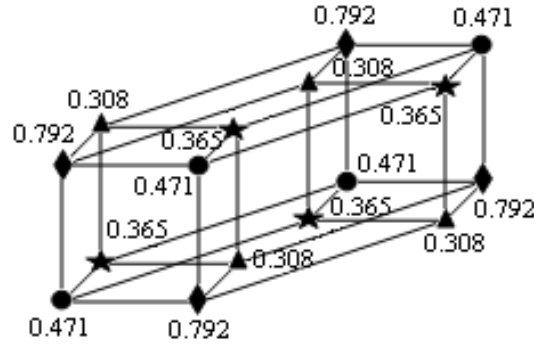


Figura 5.7: Hipergrafo representando uma paisagem de aptidão $NK(2, 1)$ utilizando a representação não neutral $NonNN_Z(4, 2)$ com a mesma vizinhança fenotípica de $\{0, 2\} \in NN_g(3, 2)$

dos genótipos óptimos, o resultado é um ótimo global único no espaços dos fenótipos.

5.3 Cadeias de Markov

Uma vez que só foi possível enumerar famílias de representações neutrais para $k \leq 11$, o espaço de procura resultante é relativamente pequeno, fazendo com que seja difícil detectar quaisquer diferenças de desempenho através da simples aplicação de um algoritmo genético a paisagens de aptidão NK . Como alternativa, foi considerada a modelação de um algoritmo evolutivo simples, uma estratégia evolutiva $(1+1)$ -ES, através de cadeias de Markov (Toonen 1994). Esta estratégia corresponde a um ‘hill climbing’ estocástico (Mitchell 1996), onde um indivíduo pai gera um indivíduo filho e o melhor dos dois torna-se o pai na próxima iteração.

Definição 5.4 (Cadeia de Markov). *Uma cadeia de Markov é uma sequência de variáveis aleatórias discretas $\{X_0, X_1, X_2, \dots\}$, que satisfaz a propriedade*

de Markov, i.e.:

$$\begin{aligned} \Pr(X_{n+1} = i_{n+1} | X_0 = i_0, X_1 = i_1, \dots, X_n = i_n) = \\ \Pr(X_{n+1} = i_{n+1} | X_n = i_n) \end{aligned} \quad (5.4)$$

A propriedade de Markov significa que o sistema não tem memória, ou seja, que o sistema apenas conhece o estado presente e é apenas desse estado que depende o valor do estado seguinte. À passagem de um estado para outro dá-se o nome de transição.

Nesta dissertação, são considerados apenas espaços de estados finitos e cadeias de Markov homogéneas, em que as probabilidades de transição são constantes no tempo. A matriz de transição é definida como $P = [p_{ij}]$, com:

$$p_{ij} = \Pr(X_{n+1} = j | X_n = i) \quad (5.5)$$

e $\sum_{j=1}^M p_{ij} = 1$, onde M corresponde ao número de estados.

A probabilidade de uma cadeia com M estados se encontrar no estado j num dado instante $n + 1$ pode ser calculada como:

$$\begin{aligned} \Pr(X_{n+1} = j) &= \sum_{i=1}^M \Pr(X_{n+1} = j \wedge X_n = i) \\ &= \sum_{i=1}^M \Pr(X_{n+1} = j | X_n = i) \Pr(X_n = i) \end{aligned} \quad (5.6)$$

desde que seja conhecida a distribuição de probabilidade do estado X no instante anterior, ou seja, $\Pr(X_n = i)$, para $i = 1, \dots, M$.

Fazendo $v_{n,i} = \Pr(X_n = i)$ e descrevendo a distribuição de X_n pelo vector $\vec{v}_n = (v_{n,1}, \dots, v_{n,M})$, a distribuição de X_{n+1} pode ser calculada directamente

a partir da matriz de transição P :

$$\vec{v}_{n+1} = \vec{v}_n P \quad (5.7)$$

Se for conhecida a distribuição do estado inicial \vec{v}_1 , a distribuição de X_n será dada por:

$$\vec{v}_n = \vec{v}_1 P^{n-1} \quad (5.8)$$

No caso da estratégia $(1+1)$ -ES, o estado actual corresponde ao indivíduo pai em cada iteração e o espaço de estados corresponde ao espaço de procura. Na estratégia utilizada, considerou-se que o filho é gerado por mutação de um bit do genótipo pai, substituindo-o na iteração seguinte se a sua aptidão não for inferior à dele.

Como as probabilidades de transição dependem do operador de mutação e da função de aptidão e sendo a vizinhança finita, a matriz de transição P é esparsa. Este facto é muito útil para a modelação porque facilita o armazenamento da matriz P em memória e reduz o tempo de cálculo da distribuição dos estados segundo a expressão 5.7.

A seguir, são apresentadas as matrizes de transição para as representações consideradas.

5.3.1 Representação não redundante

Uma representação não redundante com genótipos de comprimento k , onde $f_f(x_{f_i})$ corresponde à aptidão do indivíduo x_{f_i} e o conjunto de vizinhos de x_{f_i} com aptidão maior ou igual a $f_f(x_{f_i})$ é definido como (Toonen 1994):

$$VM_{x_{f_i}} = \{x_{f_j} | d(x_{f_i}, x_{f_j}) = 1 \wedge f_f(x_{f_j}) \geq f_f(x_{f_i})\} \quad (5.9)$$

Tabela 5.4: Matriz de transição da cadeia de Markov obtida para a paisagem de aptidão $NK(2, 1)$ da figura 5.5 utilizando a representação não redundante

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cada elemento da matriz de transição da cadeia de Markov P é definido como:

$$p_{ij} = \begin{cases} \frac{1}{k} & i \neq j \wedge x_{f_j} \in VM_{x_{f_i}} \\ 1 - \frac{|VM_{x_{f_i}}|}{k} & i = j \\ 0 & \text{caso contrário} \end{cases} \quad (5.10)$$

Se um indivíduo x_{f_i} só tiver vizinhos com menor aptidão, $VM_{x_{f_i}} = \emptyset$. Neste caso, a cadeia permanecerá neste estado sem poder encontrar qualquer indivíduo melhor, e x_{f_i} é denominado um estado absorvente. Na matriz de transição, todos os valores na linha correspondente a cada óptimo local são nulos, excepto o valor que se encontra na diagonal da matriz, que é 1.

A matriz de transição para o exemplo da paisagem de aptidão $NK(2, 1)$ da figura 5.5 é apresentada na tabela 5.4. Os fenótipos 0 e 3 são facilmente identificáveis como óptimos locais por corresponderem às linhas com o valor 1 na diagonal da matriz.

5.3.2 Representação neutral

Considere-se uma representação neutral $\{z_0, z_1, \dots, z_{2^\ell - k - 1}\} \in NN_g(\ell, k)$, onde ℓ corresponde ao número de bits dos genótipos x_{g_i} , f_g denota o mapeamento genótipo-fenótipo, f_f denota o mapeamento fenótipo-aptidão, o genótipo x_{g_i} corresponde ao fenótipo $x_{f_i} = f_f(x_{g_i})$ e o conjunto de vizinhos

Tabela 5.5: Matriz de transição da cadeia de Markov obtida para a paisagem de aptidão $NK(2, 1)$ utilizando a representação neutral $\{0, 2\} \in NN_g(3, 2)$

$$\mathbf{P} = \begin{bmatrix} 1/3 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 & 0 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2/3 & 0 & 1/3 & 0 \\ 0 & 1/3 & 0 & 0 & 1/3 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 2/3 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 1/3 & 1/3 & 0 \end{bmatrix}$$

de x_{g_i} com aptidão maior ou igual a $f_f(x_{f_i})$ é definido como:

$$VM_{x_{g_i}} = \{x_{g_j} | d(x_{g_i}, x_{g_j}) = 1 \wedge f_f(x_{f_j}) \geq f_f(x_{f_i})\} \quad (5.11)$$

Cada elemento da matriz de transição P de dimensão $2^\ell \times 2^\ell$ é definido como:

$$p_{ij} = \begin{cases} \frac{1}{\ell} & i \neq j \wedge x_{g_j} \in VM_{x_{g_i}} \\ 1 - \frac{|VM_{x_{g_i}}|}{\ell} & i = j \\ 0 & \text{caso contrário} \end{cases} \quad (5.12)$$

A matriz de transição P da cadeia de Markov obtida utilizando a representação neutral $\{0, 2\} \in NN_g(3, 2)$ está apresentada na tabela 5.5.

Como era de esperar, não existe nenhuma linha com o valor 1 porque a probabilidade de atingir um óptimo está distribuída pelos genótipos 100 e 110, da rede neutral que representa o fenótipo óptimo.

5.3.3 Representação não neutral

Considere-se uma representação não neutral $NonNN_Z(\ell', k)$ definida por uma dada matriz de transformação Z , de dimensão $\ell' \times k$. Como foi visto anteriormente, as linhas desta matriz definem os vizinhos do fenótipo 0, e os fenótipos alcançáveis a partir de qualquer fenótipo são os mesmos, qualquer

Tabela 5.6: Matriz de transição da cadeia de Markov obtida para a paisagem de aptidão $NK(2, 1)$ utilizando a representação não neutral $NonNN_Z(4, 2)$ que corresponde a $\{0, 2\} \in NN_g(3, 2)$

$$\mathbf{P} = \begin{bmatrix} 2/4 & 0 & 0 & 2/4 \\ 2/4 & 2/4 & 0 & 0 \\ 0 & 2/4 & 0 & 2/4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

que seja o genótipo que o representa. Por este motivo, é possível construir a matriz de transição P_i de dimensão $2^k \times 2^k$, fazendo referência apenas ao espaço dos fenótipos.

Seja $freq(x_{f_i}, x_{f_j})$ o número de linhas da matriz Z que são iguais a $x_{f_i} + x_{f_j}$. Sendo o conjunto de vizinhos com aptidão melhor ou igual a $f_f(x_{f_i})$ definido como:

$$VM_{x_{f_i}} = \{x_{f_j} | freq(x_{f_i}, x_{f_j}) > 0 \wedge f_f(x_{f_j}) \geq f_f(x_{f_i})\} \quad (5.13)$$

Cada elemento da matriz de transição P é definido como:

$$p_{ij} = \begin{cases} \frac{freq(x_{f_i}, x_{f_j})}{\ell'} & i \neq j \wedge x_{f_j} \in VM_{x_{f_i}} \\ 1 - \frac{\sum_{x_{f_j} \in VM_{x_{f_i}}} freq(x_{f_i}, x_{f_j})}{\ell'} & i = j \\ 0 & \text{caso contrário} \end{cases} \quad (5.14)$$

A matriz de transição P da cadeia de Markov obtida utilizando a representação não neutral $NonNN_Z(4, 2)$ que apresenta a mesma vizinhança fenotípica de $\{0, 2\} \in NN_g(3, 2)$ está representada na tabela 5.6.

A primeira linha da matriz indica que a probabilidade do indivíduo transitar do fenótipo 0 para o fenótipo 3 é 0.5, porque o fenótipo 3 tem maior aptidão que o fenótipo 0, e é alcançado duas vezes, portanto, com frequência igual a 2, num total de $\ell' = 4$ fenótipos acessíveis. Por outro lado, a proba-

bilidade de ficar retido no fenótipo 0 é também 0.5. A última linha indica que a probabilidade do indivíduo ficar no fenótipo 3 é 1, indicando que este fenótipo é um ótimo local.

5.3.4 Probabilidade de alcançar o ótimo global

O algoritmo 5.1 permite calcular a probabilidade de alcançar o ótimo global em função do número de iterações, tendo como parâmetros de entrada o ótimo global, a matriz de transição P e o número de iterações.

Algoritmo 5.1 Calcula a probabilidade de alcançar o ótimo global

```

1: Entrada: ótimo global,  $P$ , número de iterações
2: Saída: Probabilidades
3:  $y =$  ótimo global
4:  $v[1, :] = \frac{1}{\dim(P)}$ 
5: for  $t$  from 2 to número de iterações do
6:    $v[t, :] = v[t - 1, :] \times P$ 
7:    $p[t] = \text{soma}(v[t, y])$ 
8: end for

```

Este algoritmo será explicado utilizando o exemplo da representação neutral. No início, todos os estados têm igual probabilidade, pelo que todas as componentes do vector $v[1, :]$ são iguais a $\frac{1}{\dim(P)}$, sendo $\dim(P)$ o número de linhas da matriz P . Para a paisagem de aptidão NK em análise, esta probabilidade é $\frac{1}{23}$. Para a representação neutral, o ótimo global é representado por uma rede neutral. A probabilidade de o ótimo global y ser atingido em cada instante t é a soma das probabilidades do estado ser igual a cada um dos genótipos que o representam, ou seja, $p[t] = \text{soma}(v[t, y])$.

A tabela 5.7 apresenta a evolução destas probabilidades ao longo de 30 iterações. Neste caso, o fenótipo 3 corresponde ao ótimo global, e a soma das probabilidades dos seus genótipos 4 e 6 é igual a 1, indicando que a paisagem é unimodal. Obviamente, $\forall t, \sum_{i=1}^{\dim(P)} p_{ti} = 1$.

Tabela 5.7: Probabilidades de atingir os ótimos locais na paisagem de aptidão NK da figura 5.6 utilizando a representação neutral $\{0, 2\} \in NN_g(3, 2)$

$t/genótipo$	0	1	2	3	4	5	6	7
1	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250
2	0.1250	0.1250	0.1250	0.1250	0.2083	0.0417	0.2083	0.0417
3	0.1250	0.0972	0.1250	0.0972	0.2639	0.0139	0.2639	0.0139
4	0.1157	0.0694	0.1157	0.0694	0.3102	0.0046	0.3102	0.0046
5	0.1003	0.0478	0.1003	0.0478	0.3503	0.0015	0.3503	0.0015
6	0.0828	0.0324	0.0828	0.0324	0.3843	0.0005	0.3843	0.0005
7	0.0660	0.0218	0.0660	0.0218	0.4120	0.0002	0.4120	0.0002
8	0.0513	0.0146	0.0513	0.0146	0.4341	0.0001	0.4341	0.0001
9	0.0390	0.0097	0.0390	0.0097	0.4512	0.0000	0.4512	0.0000
10	0.0293	0.0065	0.0293	0.0065	0.4642	0.0000	0.4642	0.0000
11	0.0217	0.0043	0.0217	0.0043	0.4740	0.0000	0.4740	0.0000
12	0.0159	0.0029	0.0159	0.0029	0.4812	0.0000	0.4812	0.0000
13	0.0116	0.0019	0.0116	0.0019	0.4865	0.0000	0.4865	0.0000
14	0.0083	0.0013	0.0083	0.0013	0.4904	0.0000	0.4904	0.0000
15	0.0060	0.0009	0.0060	0.0009	0.4931	0.0000	0.4931	0.0000
16	0.0043	0.0006	0.0043	0.0006	0.4951	0.0000	0.4951	0.0000
17	0.0030	0.0004	0.0030	0.0004	0.4966	0.0000	0.4966	0.0000
18	0.0022	0.0003	0.0022	0.0003	0.4976	0.0000	0.4976	0.0000
19	0.0015	0.0002	0.0015	0.0002	0.4983	0.0000	0.4983	0.0000
20	0.0011	0.0001	0.0011	0.0001	0.4988	0.0000	0.4988	0.0000
21	0.0008	0.0001	0.0008	0.0001	0.4992	0.0000	0.4992	0.0000
22	0.0005	0.0001	0.0005	0.0001	0.4994	0.0000	0.4994	0.0000
23	0.0004	0.0000	0.0004	0.0000	0.4996	0.0000	0.4996	0.0000
24	0.0003	0.0000	0.0003	0.0000	0.4997	0.0000	0.4997	0.0000
25	0.0002	0.0000	0.0002	0.0000	0.4998	0.0000	0.4998	0.0000
26	0.0001	0.0000	0.0001	0.0000	0.4999	0.0000	0.4999	0.0000
27	0.0001	0.0000	0.0001	0.0000	0.4999	0.0000	0.4999	0.0000
28	0.0001	0.0000	0.0001	0.0000	0.4999	0.0000	0.4999	0.0000
29	0.0000	0.0000	0.0000	0.0000	0.5000	0.0000	0.5000	0.0000
30	0.0000	0.0000	0.0000	0.0000	0.5000	0.0000	0.5000	0.0000

Tabela 5.8: Probabilidades de alcançar os óptimos locais na paisagem de aptidão NK da figura 5.7 utilizando a representação não neutral $NonNN_Z(4, 2)$ correspondente a $\{0, 2\} \in NN_g(3, 2)$

$t/genótipo$	0	1	2	3
1	0.2500	0.2500	0.2500	0.2500
2	0.2500	0.2500	0	0.5000
3	0.2500	0.1250	0	0.6250
4	0.1875	0.0625	0	0.7500
5	0.1250	0.0312	0	0.8438
6	0.0781	0.0156	0	0.9062
7	0.0469	0.0078	0	0.9453
8	0.0273	0.0039	0	0.9688
9	0.0156	0.0020	0	0.9824
10	0.0088	0.0010	0	0.9902
11	0.0049	0.0005	0	0.9946
12	0.0027	0.0002	0	0.9971
13	0.0015	0.0001	0	0.9984
14	0.0008	0.0001	0	0.9991
15	0.0004	0.0000	0	0.9995
16	0.0002	0.0000	0	0.9998
17	0.0001	0.0000	0	0.9999
18	0.0001	0.0000	0	0.9999
19	0.0000	0.0000	0	1.0000
20	0.0000	0.0000	0	1.0000

Tabela 5.9: Probabilidades de alcançar os óptimos locais na paisagem de aptidão NK da figura 5.1 utilizando a representação não redundante

$t/genótipo$	0	1	2	3
1	0.25	0.25	0.25	0.25
2	0.5	0	0	0.5
3	0.5	0	0	0.5

A tabela 5.8 apresenta as probabilidades de atingir os óptimos locais utilizando a representação não neutral $NonNN_Z(4, 2)$ correspondente a $\{0, 2\} \in NN_g(3, 2)$. Como essa matriz de transição é calculada em função do espaço fenotípico, as probabilidades apresentadas apenas dizem respeito aos fenótipos. Por último, utilizando a representação não redundante, a paisagem apresenta dois óptimos locais, os fenótipos 0 e 3, e as probabilidades são as indicadas na tabela 5.9.

5.4 Estudo experimental e resultados

O estudo da influência da redundância e da neutralidade das representações propostas é realizado para as famílias $NN_g(14, 11)$, $NonNN_Z(\ell', 11)$ e para as paisagens $NK(11, K)$ com a variante vizinhança aleatória. O caso de $\ell = 14$ e $k = 11$ é estudado, porque corresponde à maior das famílias que foi enumerada e cujas características puderam ser determinadas.

Nas figuras 5.8 e 5.9 são apresentados exemplos ilustrativos da probabilidade de alcançar o óptimo global em duas instâncias de $NK(11, 1)$. As figuras mostram que esta probabilidade aumenta ao longo do tempo até se fixar num dado valor. Constatou-se que, a longo prazo, as representações de $NN_g(\ell, k)$ tanto podem alcançar o óptimo global com maior probabilidade que as representações não neutras correspondentes, como se pode verificar o contrário. Para além disso, as velocidades de convergência são também diferentes. Na figura 5.8, a representação não neutral converge mais rapidamente que a representação neutral.

Estes resultados indicam que as representações neutras e não neutras não são propriamente ‘equivalentes’. Como a vizinhança fenotípica de cada representação não neutral foi construída para ser igual à de uma representação neutral, seria de esperar que a probabilidade de atingir o óptimo global fosse a mesma em ambos os casos. De facto, a probabilidade de alcançar o óptimo global depende da probabilidade do primeiro indivíduo se encontrar na sua bacia de atracção. No entanto, também é possível que o primeiro indivíduo se encontre na fronteira de duas bacias de atracção diferentes. Nesse caso, a probabilidade de convergência para cada um dos óptimos dependerá não só da vizinhança fenotípica, mas também de a representação ser neutral ou não e, caso o seja, da topologia da rede neutral. A convergência mais lenta das representações neutras é menos surpreendente, já que a exploração completa

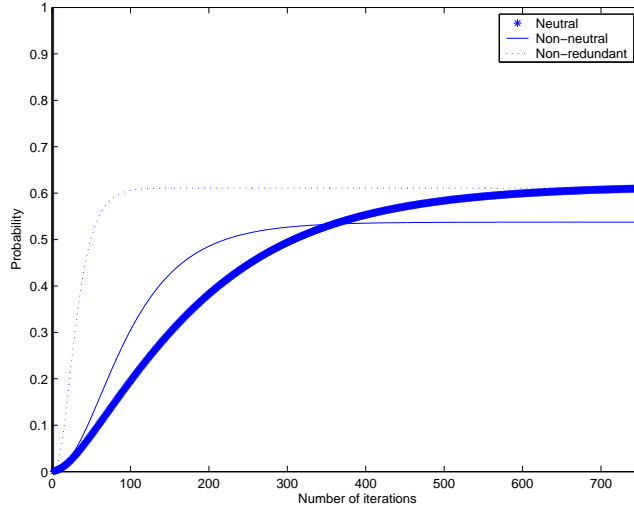


Figura 5.8: Probabilidade de alcançar o óptimo global numa instância da paisagem de aptidão $NK(11, 1)$ utilizando a representação neutral $\{0, 1, 2, 70, 74, 8192, 514, 66\} \in NN_g(14, 11)$ e a representação não neutral $NonNN_Z(98, 11)$ correspondente

da vizinhança de uma rede neutral obriga a que esta seja atravessada à custa de mutações neutrais.

Apesar disso, e a fim de tentar encontrar alguma relação entre a natureza das representações e o seu desempenho em procura evolutiva, foi considerada uma amostra aleatória de 20000 representações das 2350336 representações da família $NN_g(14, 11)$, bem como as 20000 correspondentes da família $NonNN_Z(\ell', 11)$, para servirem de termo de comparação. Cada uma destas representações foi aplicada a 8 instâncias de paisagens de aptidão $NK(11, K)$, com $K = 1, 3, 10$, com vista à determinação da probabilidade de atingir o óptimo global ao fim de 750 iterações. O número de representações considerado foi escolhido de modo a permitir que o tempo de execução e o espaço em disco não constituíssem um problema na obtenção dos resultados.

As tabelas 5.10, 5.11 e 5.12 apresentam os resultados obtidos. Cada uma das três colunas compara duas das três representações. Em cada coluna, o

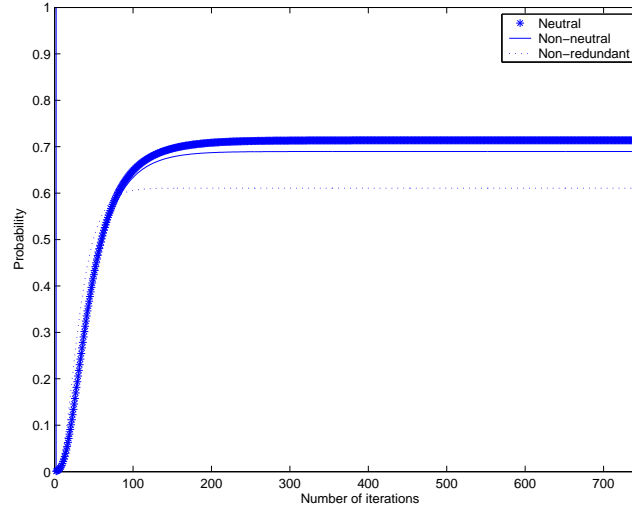


Figura 5.9: Probabilidade de alcançar o óptimo global numa instância da paisagem de aptidão $NK(11, 1)$ utilizando a representação neutral $\{0, 1, 2, 3, 4, 5, 514, 515\} \in NN_g(14, 11)$ e a representação não neutral $NonNN_Z(92, 11)$ correspondente

primeiro valor indica a percentagem de representações do primeiro tipo que alcançam o óptimo global com maior probabilidade que as representações do outro tipo, enquanto o segundo valor indica a percentagem de representações do primeiro tipo que o atingem com uma probabilidade igual ou inferior às representações do outro tipo (NN designa a representação neutral, $NonNN$ a representação não neutral e NR a representação não redundante).

Tabela 5.10: Resultados da utilização das representações: neutral (NN), não neutral ($NonNN$) e não redundante (NR) em paisagens $NK(11, 1)$

Instâncias NK	%NN melhor NonNN %NN igual pior NonNN	%NN melhor NR %NN igual pior NR	%NonNN melhor NR %NonNN igual pior NR
1	50.51 - 49.49	9.87 - 90.13	9.99 - 90.01
2	50.30 - 49.70	64.84 - 35.16	64.91 - 35.09
3	56.11 - 43.89	0 - 100	0 - 100
4	53.52 - 46.48	48.99 - 51.01	49.02 - 50.98
5	37.72 - 62.28	0.79 - 99.21	0.77 - 99.23
6	66.86 - 33.14	0 - 100	0 - 100
7	66.96 - 33.03	14.32 - 85.68	14.19 - 85.80
8	61.55 - 38.45	35.84 - 64.16	35.33 - 64.67

Tabela 5.11: Resultados da utilização das representações: neutral (NN), não neutral ($NonNN$) e não redundante (NR) em paisagens $NK(11, 3)$

Instâncias NK	%NN melhor NonNN %NN igual pior NonNN	%NN melhor NR %NN igual pior NR	%NonNN melhor NR %NonNN igual pior NR
1	44.82 - 51.18	27.22 - 72.78	27.33 - 72.67
2	46.00 - 54.00	5.85 - 94.15	5.82 - 94.18
3	52.34 - 47.65	34.48 - 65.52	34.40 - 65.60
4	46.74 - 53.26	50.40 - 49.60	50.30 - 49.70
5	46.57 - 53.43	12.42 - 87.58	12.90 - 87.10
6	57.83 - 42.17	26.22 - 73.78	25.19 - 74.81
7	51.58 - 48.42	68.84 - 31.16	68.58 - 31.42
8	50.27 - 49.74	07.07 - 92.93	7.27 - 92.73

Tabela 5.12: Resultados da utilização das representações: neutral NN , não neutral $NonNN$ e não redundante NR em paisagens $NK(11, 10)$

Instâncias NK	%NN melhor NonNN %NN igual pior NonNN	%NN melhor NR %NN igual pior NR	%NonNN melhor NR %NonNN igual pior NR
1	48.02 - 51.99	74.82 - 25.17	74.78 - 25.22
2	39.25 - 57.24	98.67 - 1.33	98.81 - 1.18
3	47.02 - 52.99	99.78 - 0.22	99.88 - 0.12
4	41.37 - 58.63	54.65 - 45.40	55.04 - 44.96
5	40.71 - 59.29	84.28 - 15.72	84.82 - 15.18
6	44.19 - 55.82	96.75 - 3.26	96.90 - 3.10
7	39.87 - 60.13	85.59 - 14.41	86.30 - 13.70
8	43.89 - 56.12	100 - 0	100 - 0

Embora os resultados não sejam muito conclusivos, manifestam um comportamento previsível das representações. Quando K é pequeno, a representação não redundante apresenta um melhor comportamento que as representações redundantes, já que qualquer alteração à vizinhança pode introduzir mais óptimos locais do que os existentes na paisagem original. Em contrapartida, quando K é grande, as representações redundantes apresentam um melhor comportamento que a representação não redundante. Neste caso, o aumento da conectividade deverá contribuir para a redução do número de óptimos locais.

Face a estes resultados, a probabilidade de atingir o óptimo global não parece ser um indicador adequado do desempenho das representações. Como alternativa, considerou-se a evolução do valor esperado da aptidão em função do tempo, valor este que também pode ser facilmente calculado a partir das distribuições de probabilidade fornecidas pela cadeia de Markov. Assim:

1. Determinou-se o valor esperado da aptidão para cada par representação-instância da paisagem NK ;
2. Seriou-se o conjunto das representações com base nesse valor separadamente para cada instância das paisagens NK , uma vez que a aptidão não é comparável entre instâncias distintas.

Para garantir que o número de óptimos locais das paisagens de aptidão não aumentaria com a utilização das representações redundantes, consideraram-se apenas representações cuja vizinhança fenotípica contém a vizinhança da representação não redundante. Este conjunto de representações corresponde a 12819 representações da família $NN_g(14, 11)$.

A figura 5.10 apresenta os gráficos de dispersão dos valores esperados da aptidão obtidos com as representações neutrais e não neutrais seleccionadas,

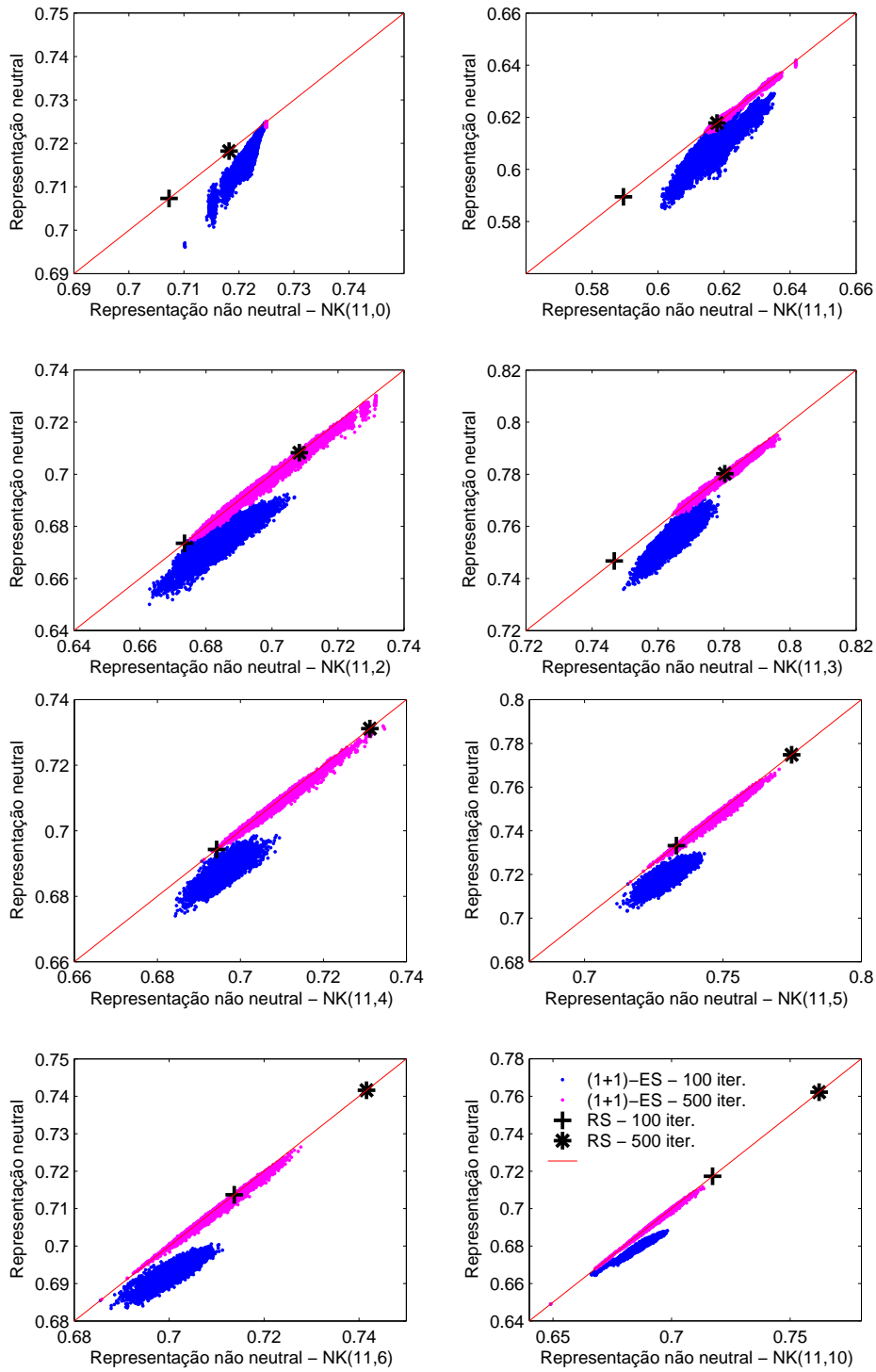


Figura 5.10: Valores esperados da aptidão em $NK(11, K)$, utilizando $(1+1)$ -ES com representações neutrais e não neutrais, e procura aleatória

numa instância das paisagens de aptidão NK para cada um dos valores de K indicados. As manchas azuis correspondem aos valores esperados ao fim de 100 iterações, enquanto as manchas rosa representam os valores esperados obtidos ao fim de 500 iterações. Para efeitos de comparação, é indicado ainda o valor esperado da aptidão para o caso da procura puramente aleatória, determinado utilizando a representação não redundante. Os sinais ‘+’ e ‘*’ indicam este valor para 100 e 500 iterações, respectivamente.

Analisando o primeiro gráfico, que corresponde a uma paisagem $NK(11, 0)$ com apenas um óptimo global, todas as representações neutrais e não neutrais parecem atingir o óptimo global ao fim das 500 iterações, o que pode ser facilmente atestado pela mancha rosa concentrada. No entanto, ao fim de 100 iterações, existem muitas representações que ainda não o alcançaram. Verifica-se também que a procura aleatória, tanto ao fim de 100 iterações como de 500 iterações, apresenta um pior comportamento do que a estratégia evolutiva. Este comportamento é, gradualmente, invertido com o aumento do valor de K , como se pode comprovar pela visualização dos restantes gráficos da figura 5.10. Como era de esperar, quanto mais difícil o problema, pior o comportamento da estratégia evolutiva em comparação com a procura aleatória.

Por outro lado, o facto de, em todos os gráficos da figura 5.10, a mancha azul estar sempre situada abaixo da diagonal indica que as representações neutrais convergem mais lentamente do que as representações não neutrais correspondentes. No entanto, ao fim das 500 iterações, pode-se constatar que os comportamentos dos dois tipos de representações são semelhantes, uma vez que as manchas rosa se aproximam bastante da diagonal do gráfico (linha vermelha).

Para averiguar se algumas das representações consideradas apresentariam

sistematicamente melhor comportamento do que as outras, determinou-se a média dos seus rankings nas seriações realizadas com base em conjuntos de 10 instâncias das paisagens NK para diversos valores de K . De notar que, se houvesse concordância total entre esses rankings, o ranking médio corresponderia simplesmente aos números inteiros de 1 a 12819. Por outro lado, se não existisse qualquer relação entre esses rankings, a distribuição de valores do ranking médio seria idêntica à que se obtém através do cálculo da média de igual número de rankings aleatórios. Esta é, essencialmente, a hipótese nula deste estudo.

Na figura 5.11 verifica-se que existe, de facto, alguma concordância entre os rankings para cada conjunto de instâncias da paisagem $NK(11, K)$, onde $K = 0, 1, 2, 3, 4, 5, 6, 10$, uma vez que a distribuição dos rankings médios se encontra afastada da distribuição sob a hipótese nula, na direcção da diagonal do gráfico. Embora não seja aqui determinado qualquer valor de significância, é de referir que as diferenças observadas são muito superiores à variação entre diferentes estimativas da distribuição dos rankings sob a hipótese nula.

Por outro lado, os resultados variam com o valor de K . Quando $K = 0, 4, 5, 6, 10$ as diferenças em relação ao caso aleatório parecem ser mais acentuadas do que quando $K = 1, 2, 3$. Excluindo o caso $K = 0$, quanto maior o K e, portanto, maior o número de óptimos locais, maior a diferença em relação ao caso aleatório.

A figura 5.12 apresenta a distribuição da média agregada de todos os rankings obtidos para $K = 0, 1, 2, 3, 4, 5, 6$, constatando-se que a concordância entre os rankings se mantém mesmo entre paisagens com diferentes valores de K .

A figura 5.13 mostra os rankings médios obtidos pelas representações neutrais para sete pares dos valores de K considerados. Os pontos vermelhos

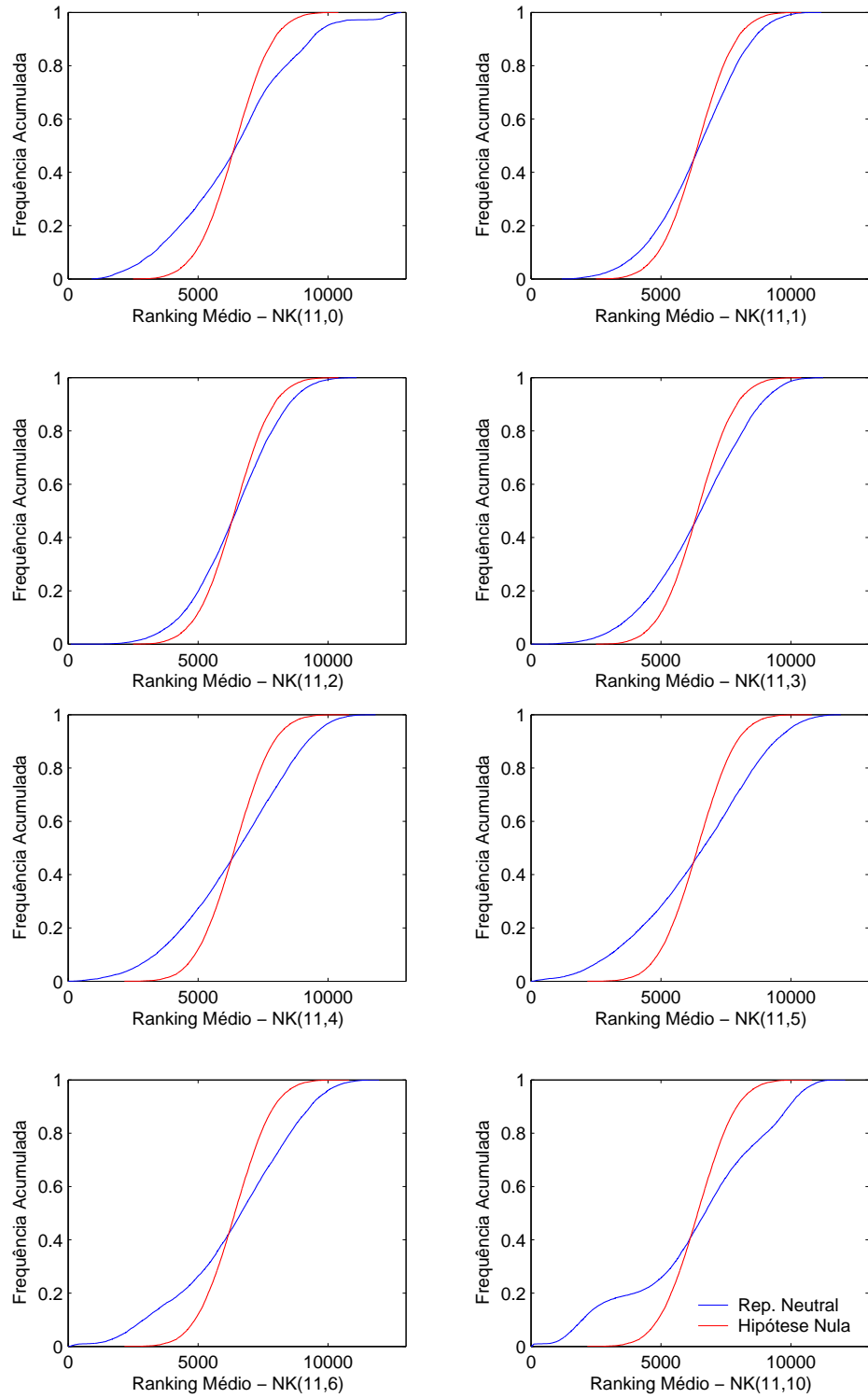


Figura 5.11: Rankings das representações neutras seleccionadas versus rankings gerados aleatoriamente

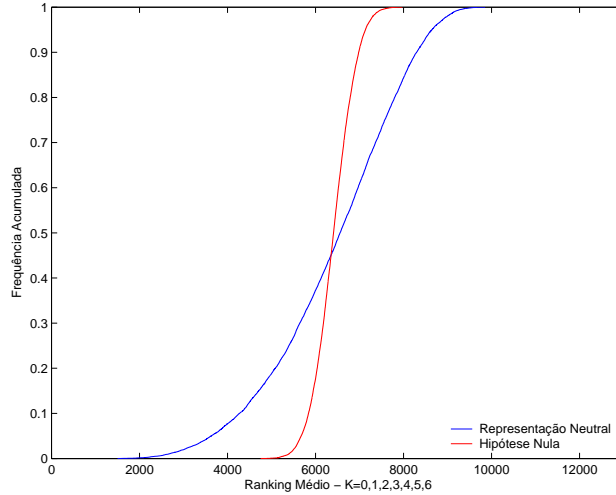


Figura 5.12: Rankings médios das representações neutras seleccionadas para $K = 0, 1, 2, 3, 4, 5, 6$ versus rankings gerados aleatoriamente

correspondem às 20 representações neutras com melhores rankings médios agregados, calculados para a figura 5.12.

Em geral, é visível uma correlação positiva entre os rankings médios obtidos por cada representação em paisagens com valores de K semelhantes. Em particular, as 20 melhores representações apresentam bom desempenho para todos os valores de K considerados, à excepção dos casos extremos, $K = 0$ e $K = 10$. Para $K = 0$, existe um conjunto de representações com desempenho claramente superior às 20 representações assinaladas. Este conjunto é constituído por representações com conectividade baixa que, em paisagens unimodais, conduzem a uma convergência rápida para o óptimo. As restantes representações também levam à convergência para o óptimo, mas mais lentamente. Para $K = 10$, a função é puramente aleatória, e as melhores representações são as que exibem conectividade mais elevada.

A figura 5.14 mostra como a conectividade, a localidade e a sinonímia das representações utilizadas se relacionam com o seu desempenho. Os resultados mostram que as 20 melhores representações apresentam valores elevados,

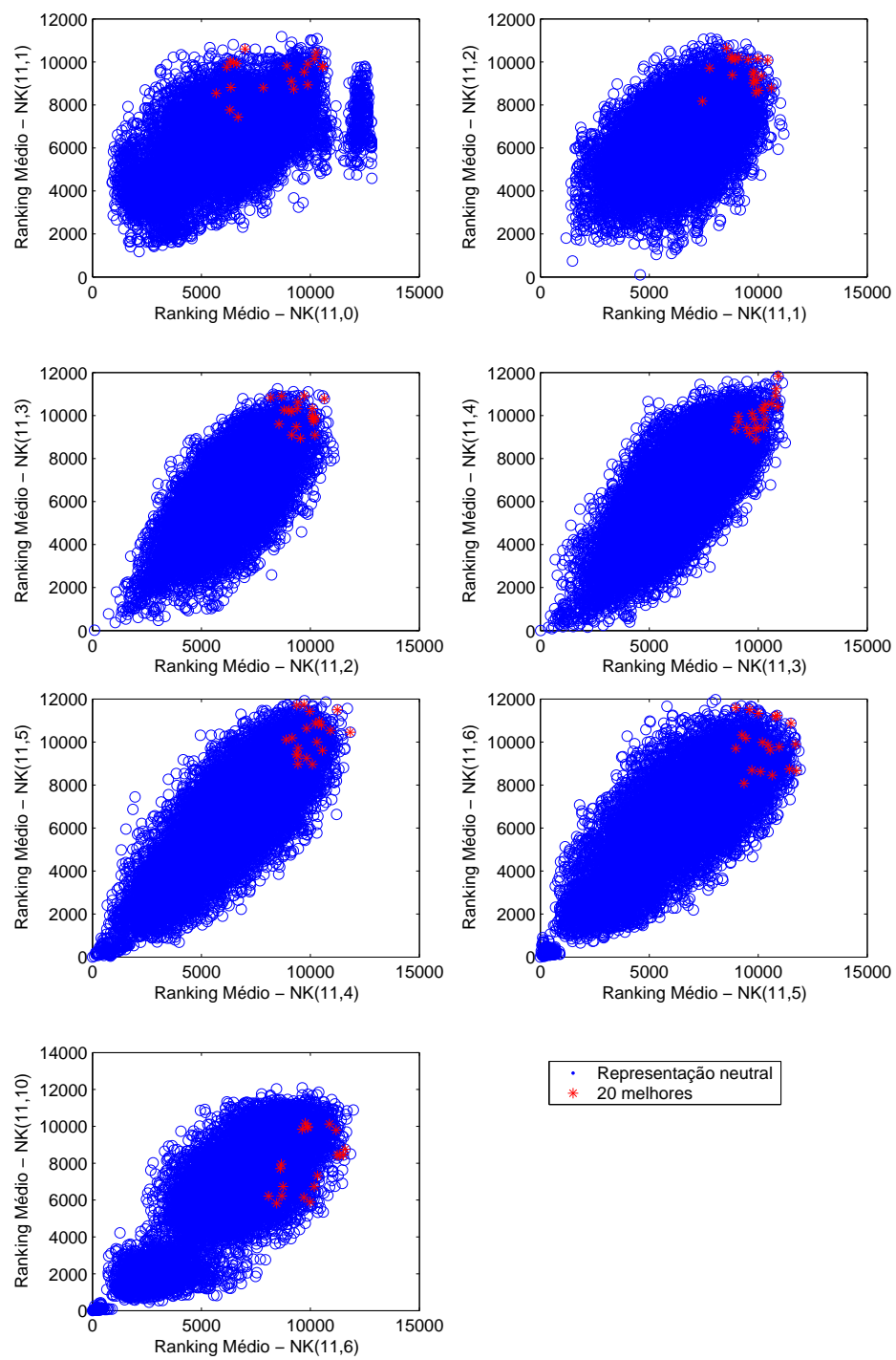


Figura 5.13: Rankings médios das representações neutrais

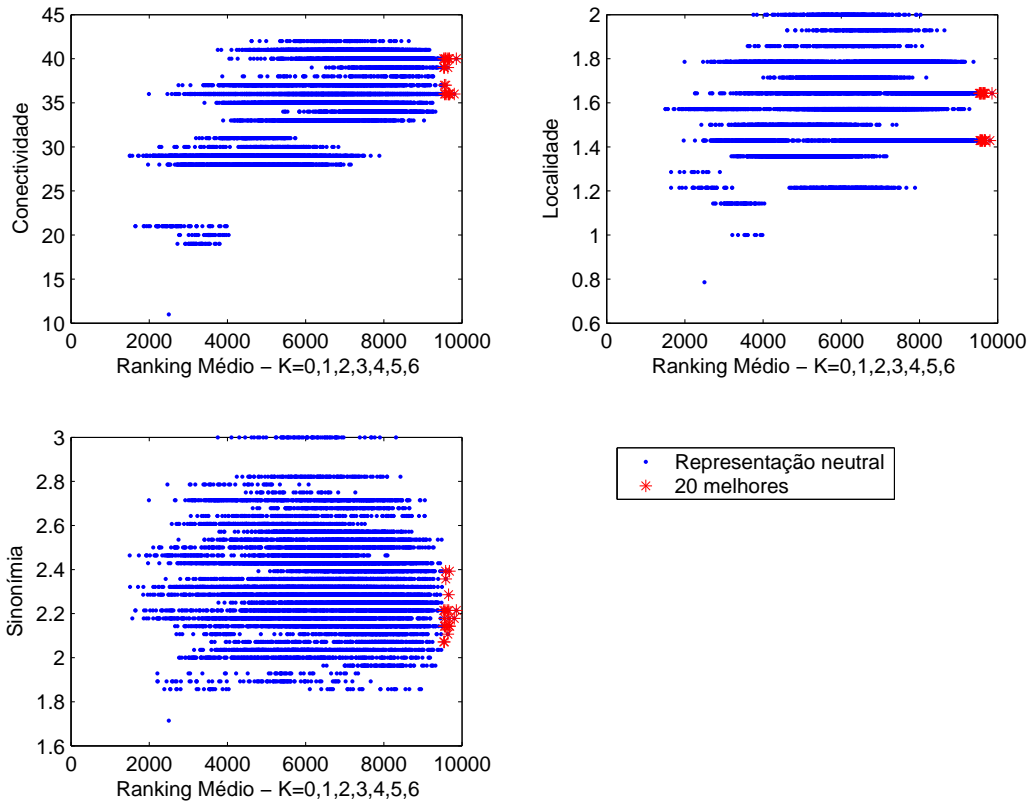


Figura 5.14: Desempenho versus características das representações neutrais

mas não extremos, de conectividade e valores intermédios de localidade e sinonímia. O mesmo pode ser observado na figura 5.15, que ilustra também o facto de, no conjunto de representações utilizadas, as medidas de conectividade, localidade e sinonímia estarem correlacionadas positivamente.

A representação neutral com a topologia em cubo, que corresponde à redundância não codificante e que apresenta conectividade 11, medida de sinonímia $S_{in} = 1,71$ e medida de localidade $d_m = 0,79$, sendo uma representação com localidade elevada (possui o valor mais baixo de d_m), apresentou um dos piores desempenhos no ranking médio dos K analisados. No entanto, para $K = 0$, esta representação apresentou o melhor desempenho, confirmando a conclusão transmitida por Rothlauf (2006), de que representações

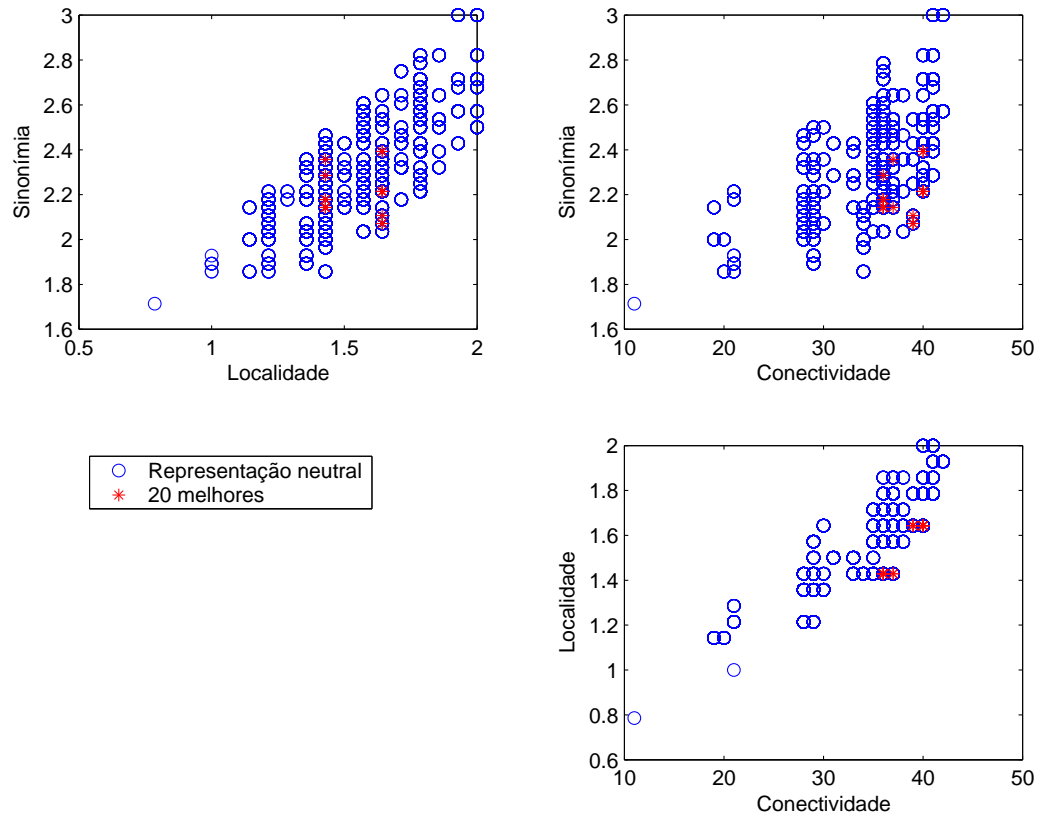


Figura 5.15: Características das representações neutrais que contêm a vizinhança fenotípica da representação não redundante

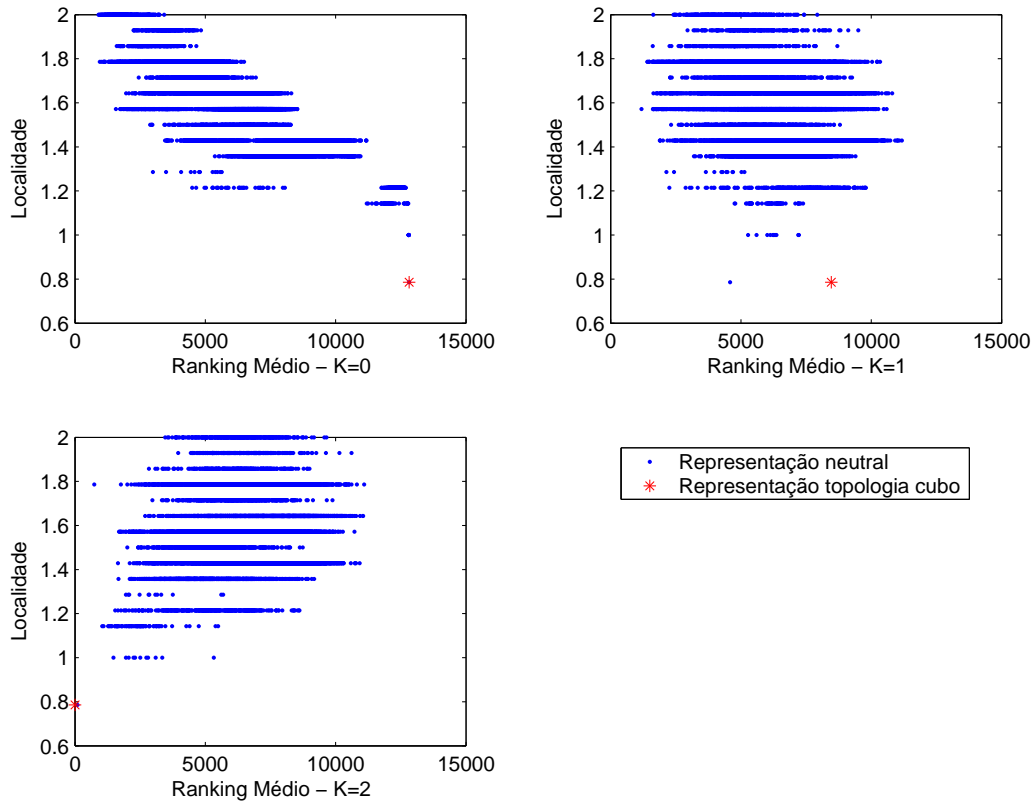


Figura 5.16: Desempenho das representações neutrais versus localidade

com localidade elevada são as mais indicadas para uma procura evolutiva eficiente em problemas fáceis. Como se pode verificar nos gráficos da figura 5.16, onde aparece indicada a vermelho a representação neutral em causa, o desempenho desta representação vai piorando com o aumento do valor de K . Comparando o desempenho das representações com a conectividade, a figura 5.17 mostra que nos problemas fáceis não é necessário que a conectividade seja elevada para obter um bom desempenho, uma vez que representações com baixa conectividade atingiram bons resultados. Quando a dificuldade do problema cresce, as representações que apresentam melhores desempenhos possuem conectividade de intermédia a elevada, embora tal característica não seja, por si só, garantia de um bom desempenho.

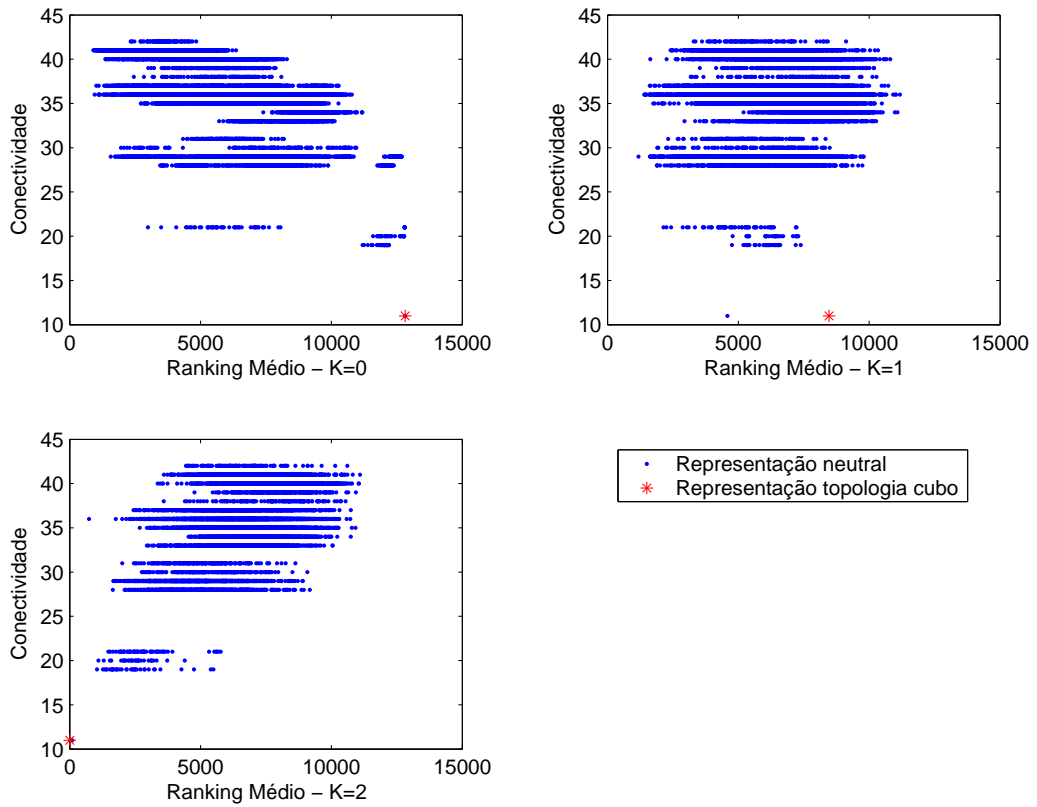


Figura 5.17: Desempenho das representações neutrais versus conectividade

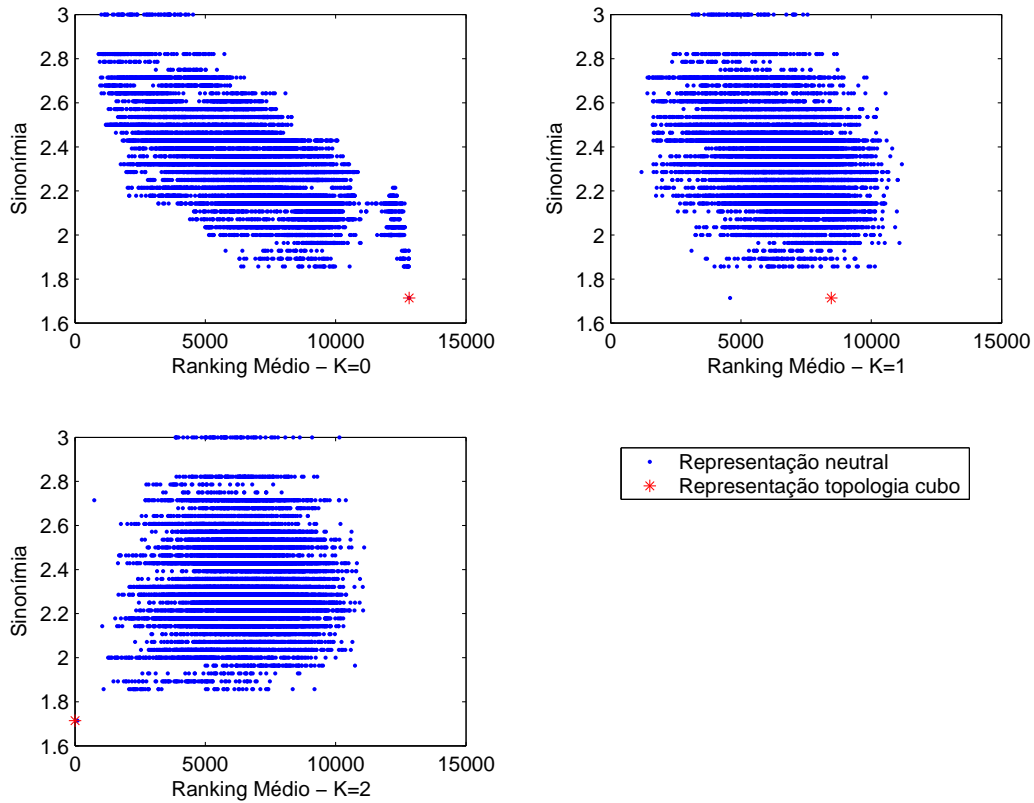


Figura 5.18: Desempenho das representações neutrais versus sinonímia

Quanto à sinonímia, a observação da figura 5.18 parece indicar que quando o problema é fácil, as representações altamente sinónimas apresentam um bom desempenho. Com o aumento da dificuldade do problema, as representações que apresentam melhores resultados possuem valores intermédios de sinonímia. No entanto, há que considerar ainda o facto referido anteriormente destas três medidas, no conjunto das representações utilizadas, estarem relacionadas umas com as outras, o que dificulta a identificação de relações de causa e efeito entre cada uma delas individualmente e o desempenho a que conduzem.

5.5 Considerações finais

Neste capítulo foi estudada a influência das representações das famílias $NN_g(\ell, k)$ e $NonNN_Z(\ell', k)$ no desempenho de uma $(1+1)$ -ES aplicada a paisagens de aptidão NK . O comportamento da estratégia foi modelado através de cadeias de Markov.

Foram realizadas várias experiências para avaliar o desempenho das representações neutrais e não neutrais propostas, juntamente com a representação não redundante. A forma como as representações das famílias $NN_g(\ell, k)$ e $NonNN_Z(\ell', k)$ transformam as paisagens de aptidão foi analisada. No caso da família $NN_g(\ell, k)$, a noção de redes neutrais localmente óptimas foi apresentada.

O desempenho da estratégia evolutiva foi considerado tanto quanto à probabilidade de atingir o óptimo global como relativamente ao valor esperado da aptidão da melhor solução encontrada. A velocidade de convergência do algoritmo e o seu comportamento a longo prazo foram analisados. A utilização da estratégia $(1+1)$ -ES considerando o valor esperado permitiu diminuir a variabilidade dos resultados e detectar de forma objectiva a influência da vizinhança e o papel da neutralidade no desempenho do algoritmo.

O estudo efectuado com as paisagens de aptidão NK corroborou a ideia de que as representações não redundantes apresentam um melhor comportamento do que as representações redundantes quando o problema é fácil, e que o contrário se passa quando o problema é considerado difícil.

Por outro lado, também foi possível observar como a vizinhança fenotípica induzida por uma representação influencia o desempenho de um algoritmo evolutivo. Utilizando a estratégia $(1+1)$ -ES e considerando o valor esperado da aptidão, concluiu-se que a vizinhança fenotípica induzida pela representação parece dominar o comportamento do algoritmo, afectando de forma

mais significativa a procura do que a neutralidade. Em geral, a neutralidade atrasou a convergência da estratégia, mas não pareceu afectar muito o comportamento do algoritmo a longo prazo.

Também foi possível observar que, entre as representações neutrais cuja vizinhança contém a vizinhança fenotípica da representação não redundante, algumas apresentaram um desempenho sistematicamente superior às outras, o que sugere que existam, de facto, representações mais adequadas que outras à optimização de paisagens do tipo NK , mesmo com rugosidades muito diferentes.

Verificou-se ainda que as representações com melhor desempenho não apresentaram valores extremos de nenhum dos indicadores de qualidade das representações habitualmente considerados na literatura, o que talvez ajude a compreender a actual dificuldade em construir representações redundantes comprovadamente bem sucedidas em computação evolutiva.

Finalmente, a obtenção dos resultados apresentados só foi possível graças à utilização de famílias diversificadas de representações redundantes bem definidas e de carácter complementar, o que vem também justificar o interesse do seu desenvolvimento.

Capítulo 6

Conclusões

A forma como a redundância e a neutralidade se manifestam no código genético foi alvo de estudo nesta dissertação. O facto de vários codões codificarem para o mesmo aminoácido e de um número elevado de estruturas primárias de ARN determinarem as mesmas estruturas secundárias de ARN, são formas de redundância no código genético. Devido à degenerescência do código genético, podem ocorrer mutações que não provoquem qualquer mudança nas proteínas produzidas. Do mesmo modo, mutações numa sequência de ARN podem não alterar a estrutura secundária da mesma.

A existência de redundância e neutralidade no código genético e a teoria neutral da evolução molecular proposta por Kimura têm motivado a utilização de representações redundantes e neutrais na computação evolutiva. A falta de consenso sobre o desempenho destas representações quando utilizadas em algoritmos evolutivos, e a verificação de que a maioria das representações propostas na literatura são altamente redundantes e utilizam mapeamentos genótipo-fenótipo pouco estruturados, foram as razões para o desenvolvimento da nova família de representações neutrais aqui proposta. do propostas na literatura. Esta família permite o desenho da vizinhança

fenotípica pretendida, utilizando para o efeito a matriz que a define. Esta forma simples e directa de definir a vizinhança desejada é uma mais valia que compensa o alto nível de redundância. O aumento da conectividade entre fenótipos é conseguido à custa do aumento correspondente do espaço de procura genotípico.

A enumeração de todas as representações neutrais para cada família $NN_g(\ell, k)$, dados ℓ e k , foi conseguida. O resultado foram famílias de representações diversificadas, que apresentam um vasto leque de características diferentes, o que permitiu corroborar alguns resultados existentes na literatura e refutar outros. Assim, foi possível mostrar que mesmo representações redundantes A formulação matemática dos códigos de controlo de erros serviu de base à definição da família de representações neutrais $NN_g(\ell, k)$. Foram estudados os códigos de bloco lineares, em particular, os códigos de Hamming e a sua formulação como códigos cíclicos. As representações de $NN_g(\ell, k)$, para $0 < k \leq 11$, quando o número de bits redundantes se situa no intervalo $0 < \ell - k \leq 4$, foram geradas. Cada representação desta família é definida pela topologia da rede neutral e apresenta propriedades diversas. A conectividade, a localidade, a sinonímia e a topologia foram algumas das características analisadas.

Uma família de representações não neutrais $NonNN_Z(\ell', k)$ com base em transformações lineares foi também apresentada. O desenvolvimento desta família foi realizado de forma incremental, começando com uma representação redundante não codificante, acrescentando a poligenia, seguida da pleiotropia. Estas representações revelaram-se altamente redundantes, quando comparadas com $NN_g(\ell, k)$, mas mesmo assim menos redundantes do que algumas das representações que têm siantes altamente sinónimas permitem o aumento da conectividade em comparação com a representação não

redundante, contradizendo a posição de alguns autores, e corroborar que as representações com localidade elevada permitem uma procura evolutiva eficiente em problemas fáceis.

A fim de estudar o papel da redundância e da neutralidade no comportamento de um algoritmo evolutivo simples utilizando as representações propostas, um estudo experimental foi realizado recorrendo a um conjunto de representações com características diversas. Para o efeito, foi utilizada uma estratégia evolutiva $(1+1)$ -*ES*, aplicada a paisagens de aptidão NK e o seu comportamento foi modelado através de cadeias de Markov. O desempenho da $(1+1)$ -*ES* foi considerado tanto quanto à probabilidade de atingir o óptimo global como relativamente ao valor esperado da aptidão da melhor solução encontrada.

Das experiências realizadas com as paisagens de aptidão NK , confirmou-se que a introdução de redundância, neutral ou não, é mais útil quando o problema a otimizar é difícil. O aumento da conectividade entre fenótipos, que poderá ser induzido pela representação redundante, permite a redução da rugosidade da paisagem, resultando em problemas mais fáceis.

Foi ainda possível concluir que, no conjunto de representações neutrais considerado, existem representações mais adequadas do que outras para a optimização das paisagens de aptidão NK . Por outro lado, verificou-se que estas representações não apresentavam valores extremos da conectividade, sinonímia ou localidade, ao contrário do que seria de esperar tendo em conta as recomendações feitas na literatura.

Para cada representação neutral considerada, foi construída uma representação não neutral da família $NonNN_Z(\ell', k)$ com a mesma vizinhança fenotípica, o que permitiu separar a influência da neutralidade dos efeitos da vizinhança comuns a ambas as representações. Assim, a neutralidade pa-

rece afectar de forma menos significativa a procura do que a vizinhança fenotípica, atrasando a convergência do algoritmo, em comparação com uma representação não neutral correspondente, mas não alterando significativamente o seu comportamento a longo prazo.

6.1 Perspectivas de trabalho futuro

Os resultados obtidos nesta dissertação abrem diversos caminhos para a investigação futura:

1. *Topologia das redes neutrais.* Tendo-se verificado que existem muitas representações neutrais com vizinhanças fenotípicas diferentes mas que apresentam a mesma topologia, seria interessante classificar as representações neutrais com base na topologia das redes neutrais. Esta classificação poderá permitir uma enumeração mais rápida, facilitando a geração de famílias com um espaço fenotípico maior;
2. *Extensão do estudo experimental a outros problemas.* Problemas como o H-IFF (Hierarchical If and Only If) e o problema de Programação Quadrática 0-1 poderiam alargar o leque de problemas a utilizar, incluindo a utilização das paisagens de aptidão NK na variante de vizinhança aleatória. Sendo os problemas de optimização baseados neste tipo de vizinhança de NP-difíceis, para $K > 1$, seria interessante saber, se para este caso, também existem representações neutrais que apresentem um bom desempenho para diversos valores de K ;
3. *Evolução das representações neutrais.* Uma vez que o valor esperado da aptidão revelou ser uma medida fiável do comportamento das representações neutrais na procura evolutiva, deverá ser possível evoluir

directamente representações da família $NN_g(\ell, k)$ com base nessa medida, utilizando para o efeito algoritmos evolutivos e um operador de mutação baseado na relação de vizinhança entre representações como descrito no capítulo 4.

Bibliografia

- Abrantes, S. A. (n.d.). *Códigos Correctores de Erros para Comunicações Digitais*, FEUP Edições. Aguarda publicação.
- Altenberg, L. (1997). NK fitness landscapes, in T. Bäck, D. Fogel & Z. Michalewicz (eds), *The Handbook of Evolutionary Computation*, Oxford University Press.
- Barnett, L. (1998). Ruggedness and neutrality - the NK_p family of fitness landscape, *Proceedings of the Sixth International Conference on Artificial Life*, pp. 18–27.
- Barnett, L. (2001). Netcrawling - optimal evolutionary search with neutral networks, *Proceedings of the 2001 Congress on Evolutionary Computation (CEC01)*, Piscataway, NJ, IEEE Press, pp. 30–37.
- Barnett, L. (2003). *Evolutionary Search on Fitness Landscapes with Neutral Networks*, PhD thesis, University of Sussex.
- Barreau, G. (2002). *The Evolutionary Consequences of Redundancy in Natural and Artificial Genetic Codes*, PhD thesis, University of Sussex.
- Blahut, R. E. (1990). *Digital Transmission of Information*, Addison Wesley.

- Bose, R. (2002). *Information Theory, Coding and Cryptography*, Tata McGraw-Hill.
- Carlson, A. B. (2002). *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*, 4th edn, McGraw-Hill College.
- Contrafold (2009). Conditional training for rna secondary structure prediction, <http://contra.stanford.edu/contrafold/server.html>.
- Correia, M. B. & Fonseca, C. M. (2007a). How redundancy and neutrality may affect evolution on NK fitness landscapes, *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC07)*, pp. 2842–2849.
- Correia, M. B. & Fonseca, C. M. (2007b). On the roles of redundancy and neutrality in evolutionary optimization: An experimental study, *Proceedings of the 2007 Genetic and Evolutionary Computation Conference (GECCO07)*, Vol. 2, p. 1504.
- de Leeuw, J. (2000). Multidimensional scaling, UCLA Statistics Preprint no. 274.
- de Rosnay, J. (1989). *A Aventura da Vida*, 1st edn, Livraria Almedina.
- Ebner, M., Langguth, P., Albert, J., Shackleton, M. & Shipman, R. (2001). On neutral networks and evolvability, *Proceedings of the 2001 IEEE Congress on Evolutionary Computation (CEC01)*, Vol. 1, pp. 1–8.
- Ebner, M., Shackleton, M. & Shipman, R. (2001). How neutral networks influence evolvability, *Complexity* **7**: 19–33.
- Fogel, D. (1997). Principles of evolutionary processes, in T. Bäck, D. B. Fogel & Z. Michalewicz (eds), *Handbook of Evolutionary Computation*, IOP Publishing and Oxford University Press.

- Fonseca, C. M. & Correia, M. B. (2005). Developing redundant binary representations for genetic search, *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC05)*, Vol. 2, pp. 1675–1682.
- Fontana, W. (2002). Modelling 'evo-devo' with rna, *BioEssays* **24**: 1164–1177.
- Fontana, W. (2005). The topology of the possible, in A. Wimmer & R. Kössler (eds), *Understanding Change: Models, Methodologies and Metaphors*, Palgrave Macmillan.
- Friedrich, T. & Neumann, F. (2008). When to use bit-wise neutrality, *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, pp. 997–1003.
- Galván-López, E. & Poli, R. (2006). An empirical investigation of how and why neutrality affects evolutionary search, *Proceedings of the 2006 Genetic and Evolutionary Computation Conference (GECCO06)*, pp. 1149–1156.
- Galván-López, E. & Poli, R. (2007). How and why a bit-wise neutrality with and without locality affects evolutionary search, *Proceedings of the 2007 Genetic and Evolutionary Computation Conference (GECCO07)*, p. 1508.
- Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.
- Hamming, R. (1980). *Coding and Information Theory*, Prentice-Hall, London.

- Harvey, I. (1997a). Artificial evolution for real problems, *Invited paper in Evolutionary Robotics: From Intelligent Robots to Artificial Life (ER97)*, pp. 127–149.
- Harvey, I. (1997b). Through the labyrinth evolution finds a way: A silicon ridge, *Lecture Notes in Computer Science* **1259**: 406–422.
- Humphreys, J. F. (1996). *A Course in Group Theory*, Oxford Science Publications.
- Huynen, M. A. (1996). Exploring phenotype space through neutral evolution, *Journal of Molecular Evolution* **43**: 165–169.
- Huynen, M. A., Stadler, P. F. & Fontana, W. (1996). Smoothness within ruggedness: The role of neutrality in adaptation, *Proceedings of National Academic Science (USA)*, Vol. 93, pp. 397–401.
- Jones, S. & Loon, B. V. (1996). *Genética para Principiantes*, Publicações Dom Quixote.
- Kauffman, S. A. (1995). *At Home in the Universe - The Search for the Laws of Self-Organization and Complexity*, Oxford University Press, New York.
- Kauffman, S. A. (2000). Optimal search on a technology landscape, *Journal of Economic Behavior & Organization* **43**: 141–166.
- Kimura, H., Basso, L., Kayo, E. K. & Suen, A. (2008). Strategic positioning and risk-return behavior, Disponível em SSRN: <http://ssrn.com/abstract=1084007>.
- Kimura, M. (1968). Evolutionary rate at the molecular level, *Nature* **217**: 624–626.

- Knowles, J. D. & Watson, R. A. (2002). On the utility of redundant encodings in mutation-based evolutionary search, *Proceedings of Parallel Problem Solving From Nature (PPSN VII)*, pp. 88–98.
- Lehre, P. K. & Haddow, P. C. (2005). Accessibility between neutral networks in indirect genotype-phenotype mappings, *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC05)*, Vol. 1, pp. 419–426.
- Levinthal, D. A. (1997). Adaptation on rugged landscapes, *Management Science* **43**: 934–950.
- Levitan, B., Lobo, J., Schuler, R. & Kauffman, S. A. (2002). Evolution of organizational performance and stability in a stochastic environment, *Computational & Mathematical Organization Theory* **8**: 281–313.
- Lewis, H. & Papadimitriou, C. (1998). *Elements of the Theory of Computation*, 2nd edn, Prentice-Hall.
- Liepins, G. E. & Vose, M. D. (1990). Representational issues in genetic optimization, *Journal of Experimental and Theoretical Artificial Intelligence* **2** pp. 101–115.
- Lin, S. & Costello, D. (1983). *Error Control Coding: Fundamental and Application*, Prentice-Hall Series in Computer Application in Electrical Engineering.
- Lobo, J., Miller, J. H. & Fontana, W. (2004). Neutrality in technological landscapes, Working Paper, Santa Fe Institute.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*, New York: Springer-Verlag.

- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*, Cambridge: MIT Press.
- Mount, D. (2004). *Bioinformatics: Sequence and Genome Analysis*, University of Arizona, Tucson.
- Palmer, C. C. (1994). *An Approach to a Problem in Network Design Using Genetic Algorithms*, PhD thesis, Polytechnic University, Troy, New York.
- Pena, S. D. (2006). O genoma joga dados? neutralismo chocou-se com o paradigma causal predominante na teoria evolucionária até então, <http://cienciahoje.uol.com.br/4258>.
- Poli, R. & Galván-López, E. (2007). On the effects of bit-wise neutrality on fitness distance correlation, phenotypic mutation rates and problem hardness, *Proceedings of the 2007 Conference on Foundations of Genetic Algorithms (FOGA07)*, pp. 138–164.
- Reeves, C. R. (1999). Landscapes, operators and heuristic search, *Annals of Operational Research* **86**: 473–490.
- Rivkin, J. W. (2000). Imitation of complex strategies, *Management Science* **46**: 824–844.
- Rivkin, J. W. & Siggelkow, N. (2002). Organizational sticking points on NK landscapes, *Complexity* **7**: 31–43.
- Ronald, S. (1997). Robust encodings in genetic algorithms: A survey of encodings issues, *Proceedings of the Forth International Conference on Evolutionary Computation*, pp. 43–48.

- Ronald, S., Asenstorfer, J. & Vincent, M. (1995). Representational redundancy in evolutionary algorithms, *Proceedings of the 1995 IEEE International Conference on Evolutionary Computation*, Vol. 2, pp. 631–636.
- Rothlauf, F. (2006). *Representations for Genetic and Evolutionary Algorithms*, 2nd edn, Springer.
- Rudolph, G. (2000). Evolution strategies, in T. Bäck, D. Fogel & T. Michalewicz (eds), *Evolutionary Computation 1: Basic Algorithms and Operators*, Institute of Physics Publishing.
- Schuster, P. (1997). Landscapes and molecular evolution, *Physica D.* **107**: 351–365.
- Schuster, P. & Fontana, W. (1999). Chance and necessity in evolution: Lessons from rna, *Physica D.* **133**: 427–452.
- Schuster, P., Fontana, W., Stadler, P. & Hofacker, I. (1994). From sequences to shapes and back: A case study in rna secondary structures, *Proceedings of Biological Sciences - Royal Society (London) B*, Vol. 255, pp. 279–284.
- Shackleton, M., Shipman, R. & Ebner, M. (2000). An investigation of redundant genotype-phenotype mappings and their role in evolutionary search, *Proceedings of the 2000 Congress on Evolutionary Computation (CEC00)*, Vol. 1, pp. 493–500.
- Sharpe, O. (2000). *Towards a Rational Methodology for Using Evolutionary Search Algorithms*, PhD thesis, University of Sussex.

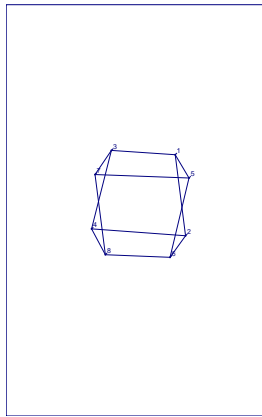
- Shipman, R., Shackleton, M., Ebner, M. & Watson, R. (2000). Neutral search spaces for artificial evolution: A lesson from life, *Proceedings of the Seventh International Conference on Artificial Life*.
- Smith, T., Husbands, P. & O'Shea, M. (2001a). Neutral networks and evolvability with complex genotype-phenotype mapping, *Proceedings of the European Conference on Artificial Life (ECAL2001)*, pp. 272–281.
- Smith, T., Husbands, P. & O'Shea, M. (2001b). Neutral networks in an evolutionary robotics search space, *Proceedings of the 2001 Congress on Evolutionary Computation (CEC01)*, Vol. 1, pp. 136–143.
- Toonen, P. G. H. (1994). Markov chain analysis of evolutionary database optimization.
- van Nimwegen, E., Crutchfield, J. P. & Huynen, M. (1999). Neutral evolution of mutational robustness, *Proceedings of National Academic Science (USA)*, pp. 9716–9720.
- Verel, S. (2005). *Étude et Exploitation des Réseaux de Neutralité dans les Paysages Adaptatifs pour l'Optimisation Difficile*, PhD thesis, l'Université de Nice-Sophia Antipolis.
- Wright, A. H., Thompson, R. K. & Zhang, J. (2000). The computational complexity of n-k fitness functions, *IEEE Transactions on Evolutionary Computation* **4**: 373–379.
- Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution, *Proceedings of the Sixth International Congress on Genetics*, Vol. 1, pp. 356–366.

Yu, T. & Miller, J. (2001). Neutrality and the evolvability of boolean function landscape, *Proceedings of the Fourth European Conference on Genetic Programming*, pp. 204–211.

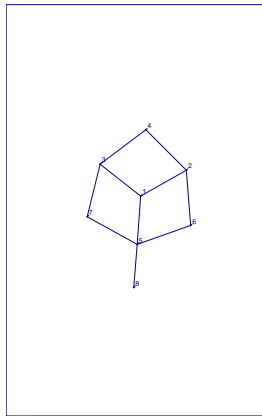
Yu, T. & Miller, J. (2002). Finding needles in haystacks is not hard with neutrality, *Proceedings of the Fifth European Conference on Genetic Programming*, pp. 13–25.

Apêndice A

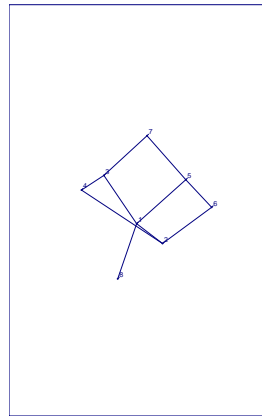
Topologias das redes neutrais de $NN_g\{7, 4\}$



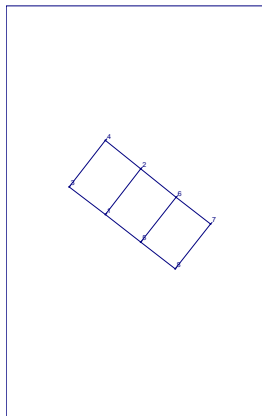
{0,1,2,3,4,5,6,7}



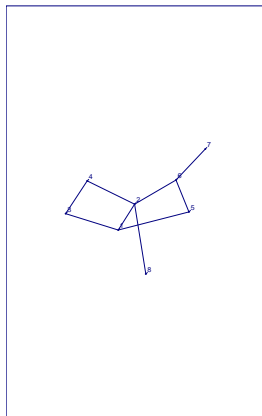
{0,1,2,3,4,5,6,12}



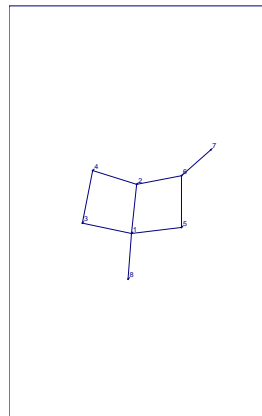
{0,1,2,3,4,5,6,32}



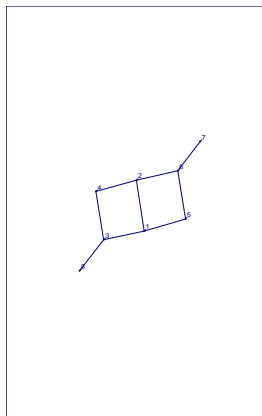
{0,1,2,3,4,5,13,12}



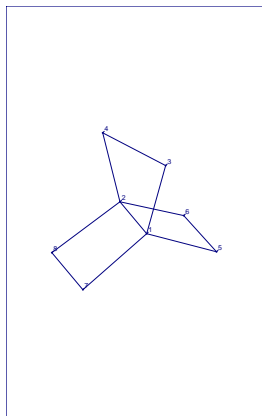
{0,1,2,3,4,5,13,17}



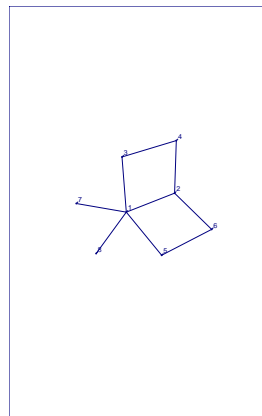
{0,1,2,3,4,5,13,32}



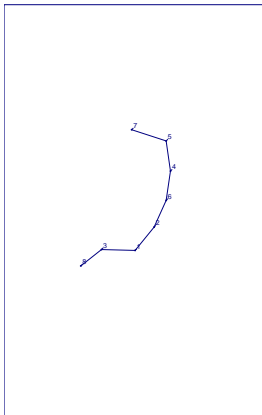
{0,1,2,3,4,5,13,66}



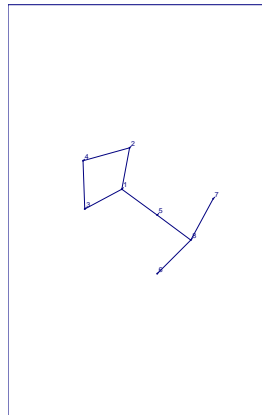
{0,1,2,3,4,5,16,17}



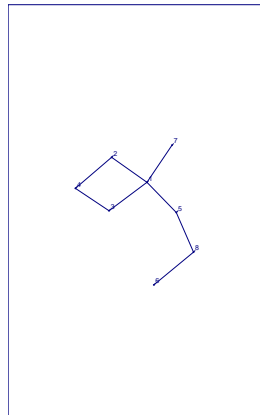
{0,1,2,3,4,5,16,32}



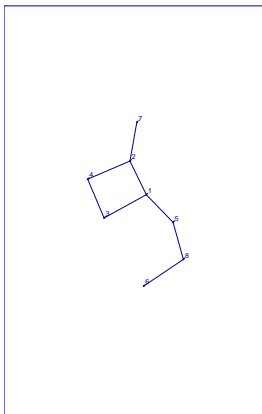
{0,1,2,21,53,5,55,66}



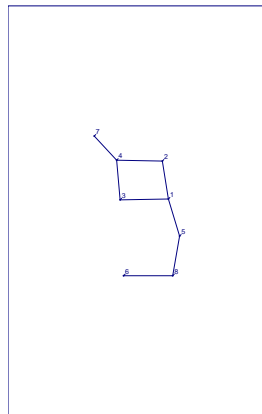
{0,1,2,3,4,14,13,12}



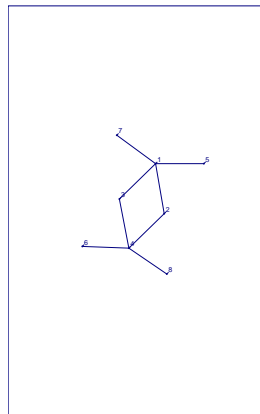
{0,1,2,3,4,14,16,12}



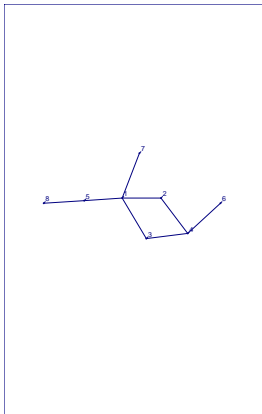
{0,1,2,3,4,14,33,12}



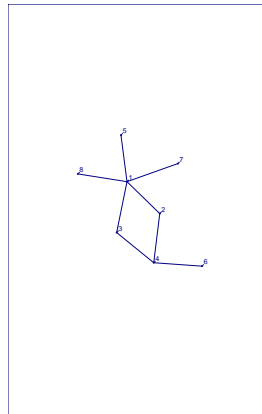
{0,1,2,3,4,14,67,12}



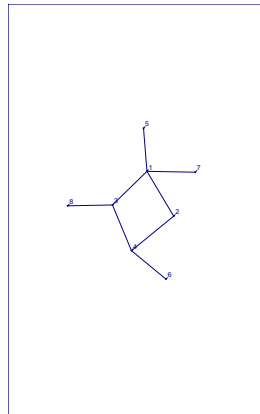
{0,1,2,3,4,19,16,7}



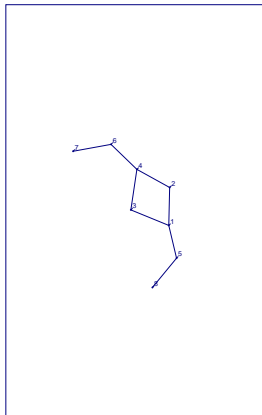
{0,1,2,3,4,19,16,12}



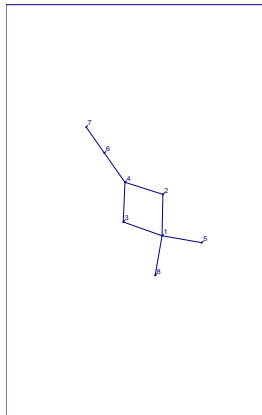
{0,1,2,3,4,19,16,32}



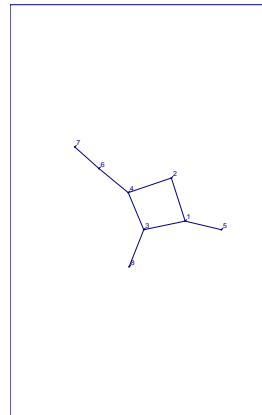
{0,1,2,3,4,19,16,66}



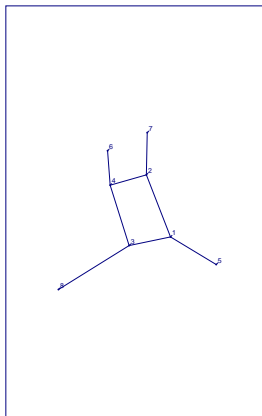
{0,1,2,3,4,19,27,12}



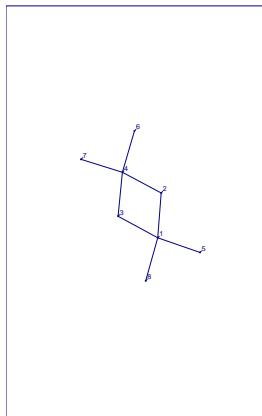
{0,1,2,3,4,19,27,32}



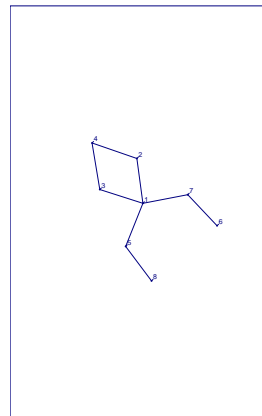
{0,1,2,3,4,19,27,66}



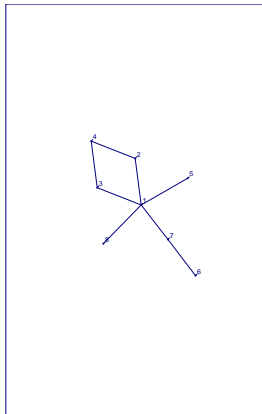
{0,1,2,3,4,19,33,66}



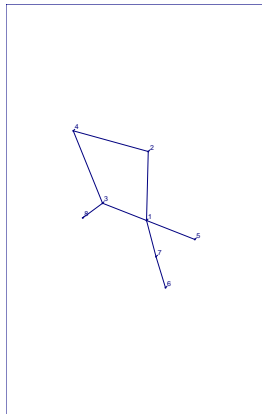
{0,1,2,3,4,19,67,32}



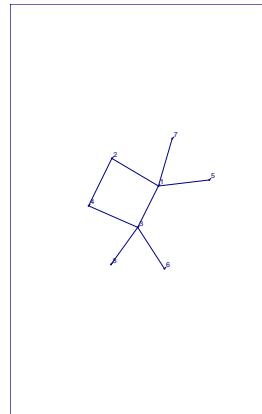
{0,1,2,3,4,24,16,12}



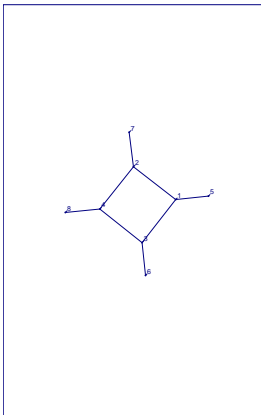
{0,1,2,3,4,24,16,32}



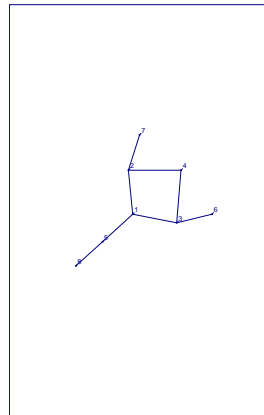
{0,1,2,3,4,24,16,66}



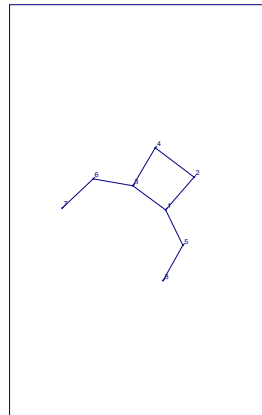
{0,1,2,3,4,34,16,66}



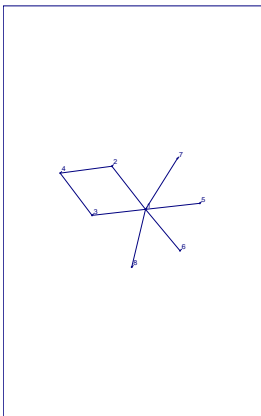
{0,1,2,3,4,34,33,7}



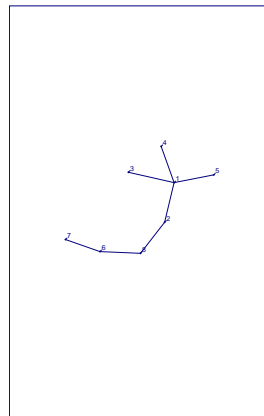
{0,1,2,3,4,34,33,12}



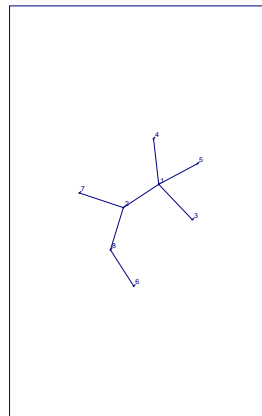
{0,1,2,3,4,34,42,12}



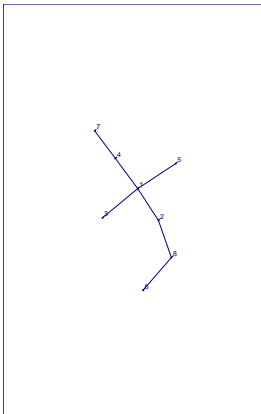
{0,1,2,3,4,64,16,32}



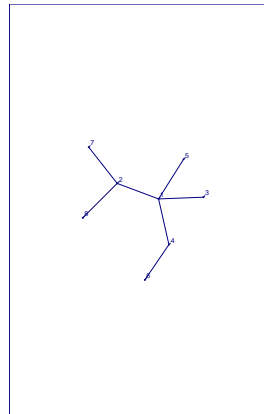
{0,1,2,8,4,19,27,17}



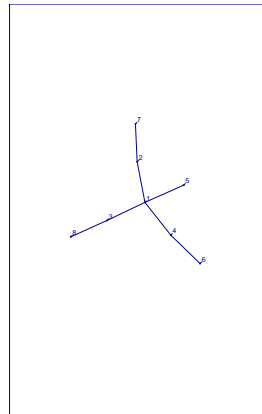
{0,1,2,8,4,19,33,17}



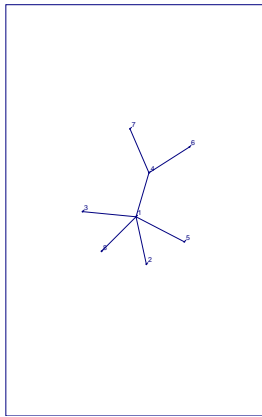
{0,1,2,8,4,19,72,17}



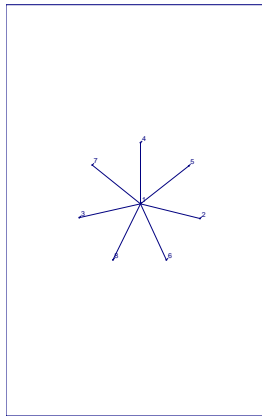
{0,1,2,8,4,24,33,17}



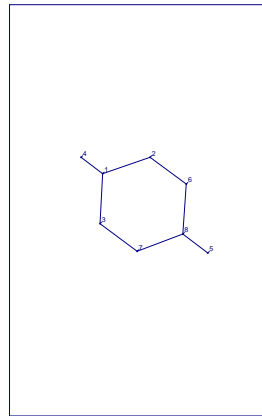
{0,1,2,8,4,24,33,66}



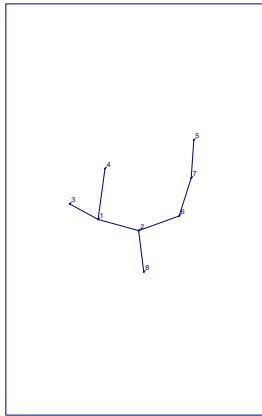
{0,1,2,8,4,24,72,32}



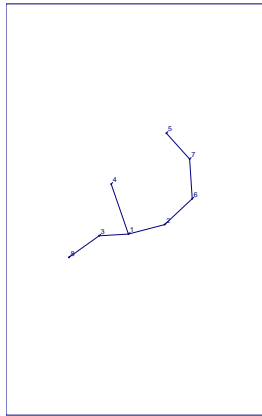
{0,1,2,8,4,64,16,32}



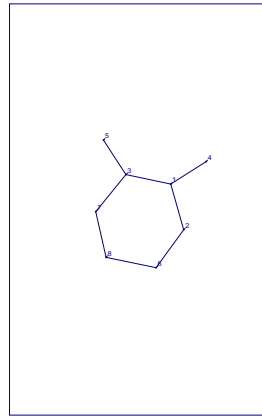
{0,1,2,8,15,5,6,7}



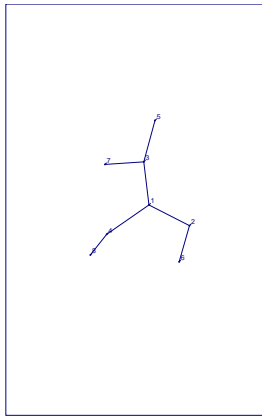
{0,1,2,8,15,5,13,17}



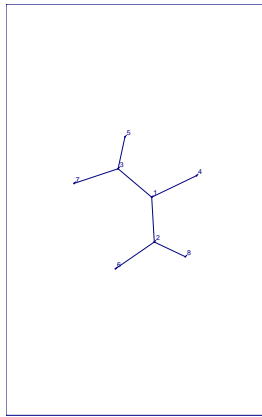
{0,1,2,8,15,5,13,66}



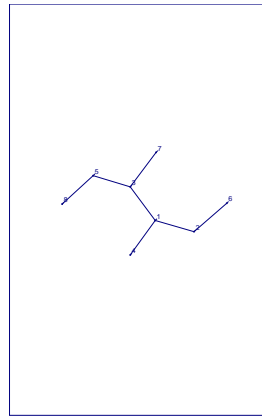
{0,1,2,8,18,5,6,7}



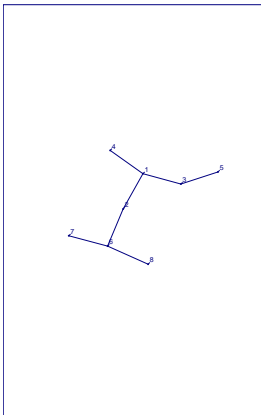
{0,1,2,8,18,5,6,12}



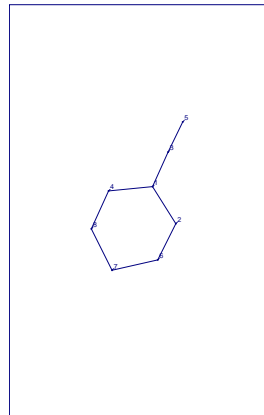
{0,1,2,8,18,5,6,17}



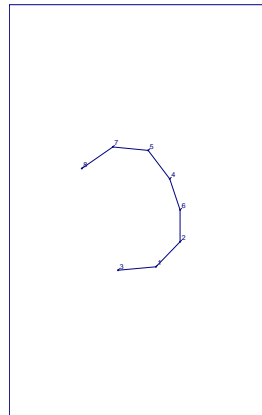
{0,1,2,8,18,5,6,26}



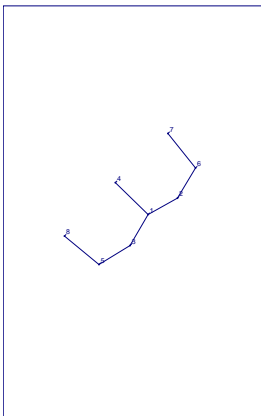
{0,1,2,8,18,5,13,7}



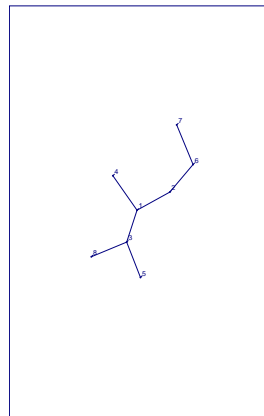
{0,1,2,8,18,5,13,12}



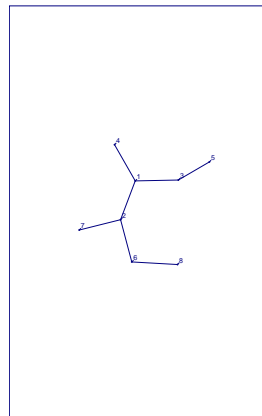
{0,1,2,21,53,5,55,54}



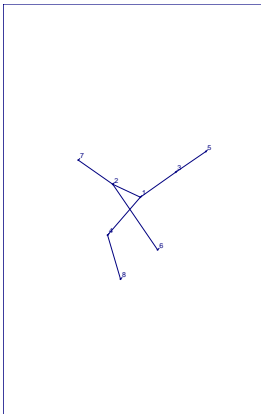
{0,1,2,8,18,5,13,26}



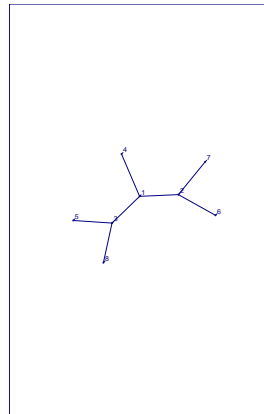
{0,1,2,8,18,5,13,66}



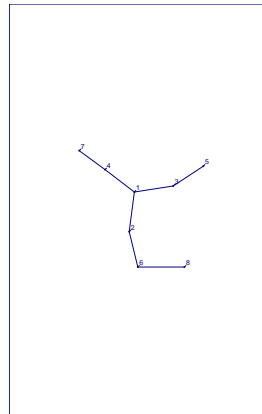
{0,1,2,8,18,5,33,7}



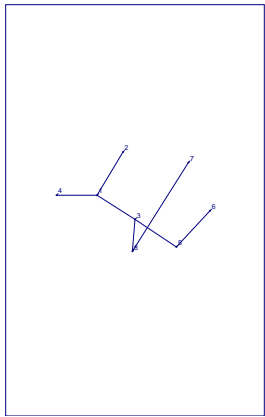
{0,1,2,8,18,5,33,12}



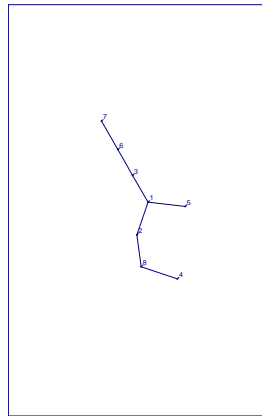
{0,1,2,8,18,5,33,66}



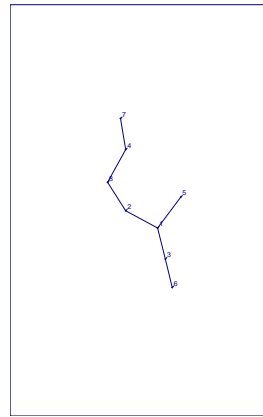
{0,1,2,8,18,5,72,7}



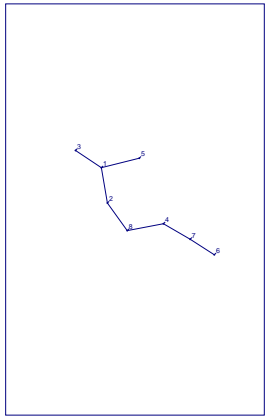
{0,1,2,8,18,19,67,66}



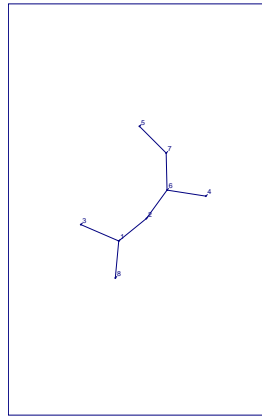
{0,1,2,21,4,34,42,17}



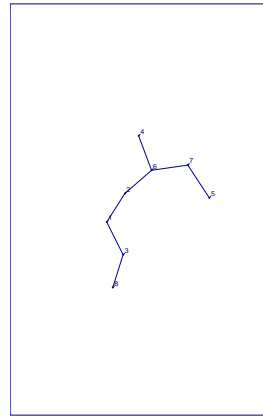
{0,1,2,21,4,34,85,17}



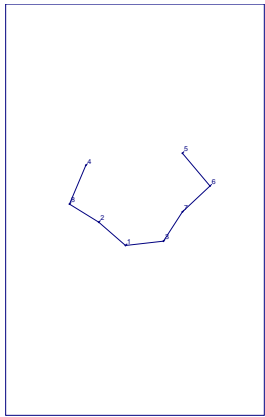
{0,1,2,21,4,93,85,17}



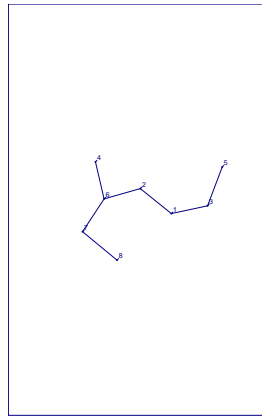
{0,1,2,21,15,5,13,32}



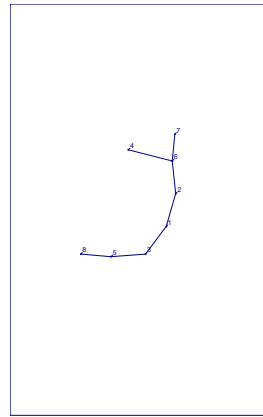
{0,1,2,21,15,5,13,66}



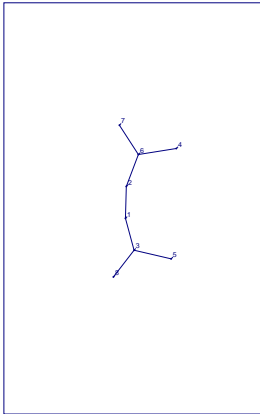
{0,1,2,21,15,14,6,17}



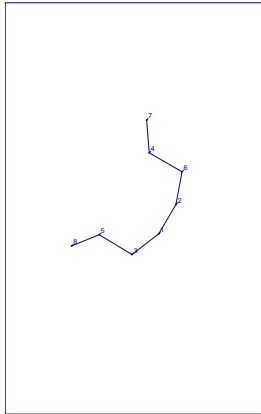
{0,1,2,21,18,5,13,12}



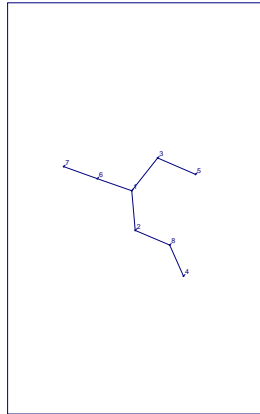
{0,1,2,21,18,5,13,26}



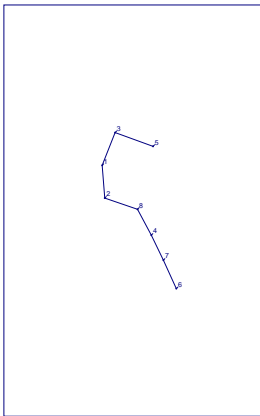
{0,1,2,21,18,5,13,66}



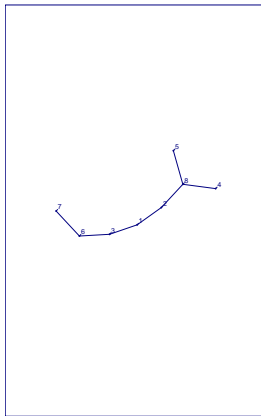
{0,1,2,21,18,5,85,26}



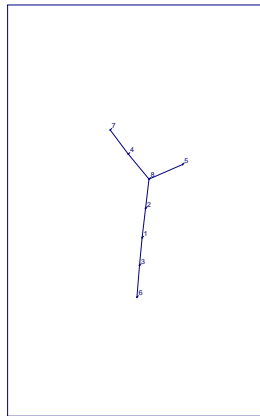
{0,1,2,21,18,64,72,17}



{0,1,2,21,18,93,85,17}



{0,1,2,21,25,34,42,17}



{0,1,2,21,25,34,85,17}