DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Transposição: Estudo de um Novo Operador Genético Inspirado Biologicamente

Anabela Borges Simões

Coimbra

Maio de 1999

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Transposição: Estudo de um Novo Operador Genético Inspirado Biologicamente

Anabela Borges Simões

Maio de 1999

Dissertação submetida para satisfação parcial dos requisitos de programa de Mestrado em Engenharia Informática, Departamento de Engenharia Informática, Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Dissertação realizada sob a orientação do
Prof. Doutor Ernesto Jorge Fernandes Costa
Professor Associado com Agregação
do Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia
da Universidade de Coimbra

Para o Jorge

RESUMO

Desde os estudos pioneiros realizados por John Holland até aos trabalhos de investigação actuais, o Algoritmo Genético (AG) conheceu inúmeras variantes, quer a nível da representação, quer nas características dos operadores genéticos utilizados. Grande parte dos AG's implementados para a resolução de problemas específicos, "afastam-se" das ideias básicas da genética, utilizando operadores mais adequados à representação e dependentes do domínio sobre o qual operam. Sem criticar estes AG's, alguns autores alertaram para o facto de as novas descobertas da biologia molecular poderem fornecer ideias para *novos* algoritmos genéticos, mais próximos da biologia. Neste sentido, trabalhos recentes procuram aproximar o modelo computacional dos modelos biológicos, colocando mais genética na sua implementação.

Este trabalho segue esta linha de orientação e teve como objectivo encontrar nos sistemas biológicos operadores genéticos, responsáveis pela diversidade das populações, que pudessem ser adaptados e integrados no AG tradicional. Apesar dos sistemas biológicos nos fornecerem um grande número destes mecanismos, testes preliminares com um deles conduziram a resultados promissores que achámos que deveriam ser solidificados. Este mecanismo, objecto de estudo deste trabalho, designa-se por **transposição**. Foram propostas duas variantes do mecanismo de transposição. A primeira, denominada por **transposição simples,** envolve troca bidireccional de material genético de determinadas características (o transposão) entre dois indivíduos escolhidos aleatoriamente. A segunda designada por **transposição baseada em torneio**, caracteriza-se pela transferência unidireccional do transposão, de um indivíduo (o vencedor do torneio) para outro (o perdedor).

Para estudar as potencialidades deste operador genético, utilizámos o AG no domínio clássico de optimização, substituindo o operador de *crossover* tradicional (com 1 ponto de corte, 2 pontos de corte e uniforme) pelas duas variantes do mecanismo de transposição. Realizou-se um extenso estudo empírico envolvendo a optimização de dezoito funções, todas elas abrangendo diferentes características e já utilizadas por diversos autores como medida de eficiência do AG.

A transposição foi implementada variando um parâmetro, o tamanho das sequências flanqueadoras, cuja escolha se revelou de muita importância para a qualidade das soluções encontradas.

A análise dos resultados foca dois aspectos principais. O primeiro descreve a forma encontrada para escolher o valor para o tamanho das sequências flanqueadoras que conduz ao desempenho máximo do AG. O segundo aspecto foca a análise comparativa entre o mecanismo de transposição e os operadores de *crossover*.

Os resultados demonstraram que a transposição, no domínio de optimização de funções, permite ao AG obter melhores resultados do que os operadores clássicos de *crossover*. Além disso, a principal vantagem do mecanismo proposto é permitir que o AG, mesmo com populações pequenas, encontre melhores soluções do que os operadores de *crossover*, com populações de maiores dimensões.

ABSTRACT

Since John Holland's pioneering work, the Genetic Algorithm (GA) undergo several modifications, concerning representation issues and genetic operators. The majority of current GA's, applied to specific problems, "deviate" from the basic ideas of genetics and use domain dependent genetic operators, more suitable to the selected representation. Without criticizing these GA's, several authors emphasize the last discoveries of molecular biology as a good source of inspiration for *new* genetic algorithms more close to biology. Following these guidelines, recent works try to reduce the gap between natural biological systems and the corresponding computational models, putting more genetics in their implementation.

The main goal of this work was to look for genetic operators, present in the biological systems, responsible for genetic diversity of the populations, that could be adapted and integrated in the simple GA. Nature is a good source of inspiration and we found a large set of genetic operators, capable of rearranging the genetic material of the individuals. Preliminary experiments with one of those mechanisms produced promising results and we decided to study it thoroughly. This mechanism, and the main goal of this work, is called **transposition**. We proposed two variations of the mechanism of transposition. The first, called **simple transposition**, implies the bidireccional exchange of specific genetic material (the transposon) between to individuals randomly chosen. The second variation, called **tournament-based transposition**, characterizes itself by the unidirectional exchange of the transposon, from one individual (the winner of the tournament) to another individual (the loser).

To study the effectiveness of the proposed mechanism we used the GA in the classical domain of optimization, being the traditional crossover operator (with one cut point, 2 cut points and uniform) replaced by transposition.

We carried out an extensive empirical study using a set of eighteen test functions, covering a large set of characteristics and already used by several authors as benchmarks to evolutionary approaches.

The mechanism of transposition was implemented varying a parameter, the flanking sequence length, which must be selected very carefully in order to achieve good results.

The analysis of the results focus to main situations. First, we describe the heuristics that should be applied to chose the flanking sequence length leading the GA to the best solutions. Secondly, we present the comparative analysis of the results achieved with transposition and crossover.

The results showed that, in the selected domain - function optimization -, the modified GA with transposition can obtain better results than the standard GA using crossover. Moreover, one clear advantage of using the GA with transposition is that, even with smaller populations, it can obtain much better results than when used with crossover, with larger populations.

AGRADECIMENTOS

Gostaria de agradecer a um conjunto de pessoas que permitiram que este trabalho fosse não só possível, mas também muito compensador.

Um agradecimento especial ao meu orientador, o Prof. Doutor Ernesto Costa, que acompanhou diariamente a realização deste trabalho, dando-me todo o apoio nos momentos de angústia e partilhando comigo as horas mais felizes da minha vida. Além disso, só com a sua grande paciência e esforço este texto conseguiu chegar à sua versão final, depois de muitas revisões do texto original.

Agradeço aos meus colegas e amigos do Grupo de Inteligência Artificial pelo seu apoio e opiniões. Em particular gostaria de agradecer à Ana Paula Neves e ao Arlindo Silva a amizade e encorajamento que sempre me deram e que me ajudaram a superar alguns momentos difíceis.

À minha família um agradecimento especial, em particular à minha mãe, porque sem a sua ajuda e alguns sacrifícios pessoais não teria sido possível percorrer todo o caminho que culminou com a conclusão deste trabalho.

Um agradecimento muito especial à minha irmã Carolina Simões pela paciência e disponibilidade que sempre demonstrou na revisão do (mau) inglês dos meus artigos.

O trabalho de investigação descrito nesta teses foi desenvolvido no Laboratório de Inteligência Artificial do Centro de Informática e Sistemas da Universidade de Coimbra.

Este trabalho foi financiado parcialmente pelo Ministério Português da Ciência e Tecnologia, sob o programa Praxis XXI.

Índice

CAPÍTULO 1 - INTRODUÇÃO **1**

SUMÁRIO 1
1.1. ALGORITMOS GENÉTICOS: GENERALIDADES 2
1.2. OS ALGORITMOS GENÉTICOS E A BIOLOGIA 3
1.3. ORGANIZAÇÃO DO TRABALHO 5

CAPÍTULO 2 - ALGORITMOS GENÉTICOS **7**

SUMÁRIO 7
2.1. INTRODUÇÃO 8
2.1.1. Terminologia *8*
2.1.2. Elementos Básicos de um AG *10*
2.1.3. Funcionamento do Algoritmo *10*
2.1.4. Representação *12*
2.1.5. Métodos de Selecção *16*
2.1.6. Operadores de Crossover *20*
2.1.7. Operador de Mutação *23*
2.1.8. Função de avaliação *24*
2.1.9. População *24*
2.2. ASPECTOS TEÓRICOS 25
2.3. APLICAÇÕES DOS ALGORITMOS GENÉTICOS 29
2.3.1. Modelos Científicos *31*
2.3.2. Resolução de Problemas *34*
2.4. ALTERAÇÕES AO AG CLÁSSICO DE INSPIRAÇÃO BIOLÓGICA 41
2.4.1. Operador de inversão *41*
2.4.2. Operadores de duplicação e deleção *43*
2.4.3. Operador de Transdução *43*
2.4.4. Operador de Conjugação *44*
2.4.5. Representações Poliplóides *47*

CAPÍTULO 3 - INSPIRAÇÕES DA BIOLOGIA **51**

SUMÁRIO 51
3.1. BIOLOGIA: ALGUMAS NOÇÕES 52
3.1.1. Terminologia *52*
3.1.2. O Caso das Bactérias *56*
3.1.3. Para além das Bactérias *57*
3.2. DIVERSIDADE GENÉTICA 59
3.2.1. Transdução *59*
3.2.2. Conjugação *61*
3.2.3. Transformação *64*
3.2.4. Deleção *65*
3.2.5. Duplicação *65*
3.2.6. Crossover desigual *67*
3.2.7. Quebra e Fusão *69*
3.2.8. Inversão *70*
3.2.9. Translocação *71*
3.2.10. Transposição *72*
3.3. INCORPORAÇÃO DOS MECANISMOS DESCRITOS NO ALGORITMO GENÉTICO 72

CAPÍTULO 4 - O MECANISMO DE TRANSPOSIÇÃO **75**

SUMÁRIO 75
4.1. A TRANSPOSIÇÃO NOS SERES VIVOS 76
4.1.1. Diferentes tipos de transposões *80*
4.1.2. Como se movimentam os diferentes tipos de transposões *81*

4.2. TRANSPOSIÇÃO COMPUTACIONAL 82
4.2.1. Como se forma o transposão 83
4.2.2. Como se define o ponto de inserção 85
4.2.3. Como se movimenta o transposão 85
4.2.5. Casos particulares 87
4.3. O ALGORITMO 87
4.4. UM EXEMPLO 88
4.4.1. Mecanismo de transposição simples 89
4.4.2. Mecanismo de transposição baseada em torneio 90

CAPÍTULO 5 - CARACTERÍSTICAS DO AMBIENTE DE EXPERIMENTAÇÃO 91

SUMÁRIO 91
5.1. PROBLEMA DE MAXIMIZAÇÃO 93
5.1.1. Funções de teste de K. De Jong 93
5.1.2. Conjunto de teste suplementar 97
5.2. PROBLEMA DE MINIMIZAÇÃO 101
5.2.1. Conjunto de teste de Koon 101
5.3. RESUMO DAS CARACTERÍSTICAS DAS FUNÇÕES 107
5.4. O ALGORITMO GENÉTICO UTILIZADO 108
5.4.1. O AG e o mecanismo de transposição 108
5.4.2. Mecanismo de selecção 109
5.4.3. Probabilidades dos operadores genéticos 110
5.4.4. Tamanho da população 111
5.4.5. Método de inserção 111
5.4.6. Avaliação do desempenho do AG 111
5.4.7. Tamanho das sequências flanqueadoras 112
5.5. PLANO DA EXPERIMENTAÇÃO REALIZADA 113
5.5.1. Série 1 - Operador de crossover com 1 ponto de corte 114
5.5.2. Série 2 - Operador de crossover com 2 pontos de corte 114
5.5.3. Série 3 - Operador de crossover uniforme 114
5.5.4. Série 4 – Mecanismo de transposição simples 114
5.5.5. Série 5 – Mecanismo de transposição baseado em torneio 114

CAPÍTULO 6 - ANÁLISE DOS RESULTADOS 117

SUMÁRIO 117
6.1. O MECANISMO DE TRANSPOSIÇÃO 118
6.1.1. Influência do Tamanho da População 118
6.1.2. Influência do Tamanho das Sequências Flanqueadoras 119
6.1.3. Escolha do Tamanho das Sequências Flanqueadoras 120
6.2. COMPARAÇÃO DA TRANSPOSIÇÃO COM OS OPERADORES DE *CROSSOVER* (1 PONTO DE CORTE, 2 PONTOS DE CORTE E UNIFORME) 126
6.2.1. Funções de teste de De Jong (maximização) 126
6.2.2. Funções de teste suplementar (maximização) 131
6.2.3. Funções de teste sugeridas por Koon (minimização) 137
6.3. SÍNTESE DOS RESULTADOS 147

CAPÍTULO 7 - CONCLUSÕES E TRABALHO FUTURO 151

SUMÁRIO 151
7.1. CONCLUSÕES 152
7.2. TRABALHO FUTURO 154

BIBLIOGRAFIA 157

GLOSSÁRIO 177

ANEXOS 180

Índice de Figuras

Figura 2.1 - O Algoritmo Genético clássico 11
Figura 2.2 - Representação em Árvore 15
Figura 2.3 - Funcionamento do Operador de Crossover com 1 ponto de corte 21
Figura 2.4 - Funcionamento do Operador de Crossover com 2 pontos de corte 22
Figura 2.5 - Funcionamento do Operador de Crossover Uniforme 22
Figura 2.6 - Funcionamento do Operador de Mutação 23
Figura 2.7 - Efeito dos genes recessivos e dominantes na descendência de dois indivíduos 48
Figura 3.1 – Par de cromossomas homólogos 53
Figura 3.2 – Células homozigóticas, heterozigóticas, com genes dominantes e recessivos 54
Figura 3.3 - O Processo de Crossover 55
Figura 3.4 – Mecanismo de transdução generalizada 60
Figura 3.5 – Mecanismo de transdução especializada 61
Figura 3.6 - Mecanismo de conjugação entre duas células 62
Figura 3.7 – Mecanismo de conjugação: formação de células F+ e F' a partir de células Hfr ... 63
Figura 3.8 – Mecanismo de Transformação 64
Figura 3.9– Mecanismo de deleção de um segmento de um cromossoma 65
Figura 3.10 - Mecanismo de duplicação de um segmento de um cromossoma 66
Figura 3.11 – Formas de duplicação de segmentos: tandem, tandem inversa e tandem terminal .. 66
Figura 3.12 – Mecanismo de crossover desigual 68
Figura 3.13 – Mecanismo de crossover desigual a partir de cromossomas mutantes 68
Figura 3.14 – Mecanismo de quebra e fusão 69
Figura 3.15 – Mecanismo de inversão – inversão pericêntrica e inversão paricêntrica 70
Figura 3.16 – Mecanismo de translocação: intracromossómica, intercromossómica não-recíproca e recíproca. 71
Figura 4.1– Transposão formado por sequências flanqueadoras iguais ou inversas 77
Figura 4.2 – processo de transposição: EXCISÃO + INTEGRAÇÃO 79
Figura 4.3 – Processo de formação do transposão 84
Figura 4.4 – Representação do cromossoma em circulo 84
Figura 4.5 – Processo de pesquisa do ponto de inserção no cromossoma receptor 85
Figura 4.6 - Representação do funcionamento das duas variantes propostas para o mecanismo de transposição 86
Figura 4.7 – Algoritmo do mecanismo de transposição 88

Índice de Tabelas

Tabela 3.1 – Genes dominantes e recessivos para a cor dos olhos 55
Tabela 3.2 – Síntese dos mecanismos que provocam a alteração da estrutura do material genético presentes nos seres vivos 58
Tabela 6.1.1- Efeitos da variação do tamanho dos flanqueadores na função de teste F1 de De Jong 119
Tabela 6.1.2 – Síntese dos sequências flanqueadoras que permitiram obter os melhores resultados para o mecanismo de transposição 123
Tabela 6.1.3 – Síntese das sequências flanqueadoras que permitiram obter os melhores resultados para o mecanismo de transposição/torneio 125
Tabela 6.2.1 – Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função De Jong – F1 128
Tabela 6.2.2 – Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função De Jong – F2 129
Tabela 6.2.3 – Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função De Jong – F3 130
Tabela 6.2.4 – Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função De Jong – F4 131
Tabela 6.2.5 – Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função De Jong – F5 131
Tabela 6.2.6 – Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função de Michalewicz. 132
Tabela 6.2.7 – Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bohachevsky (I) 133
Tabela 6.2.8 – Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bohachevsky (II) 134
Tabela 6.2.9 – Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função de Griewangk (N=10). 135
Tabela 6.2.10 – Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função de Rastrigin (N=20) 136
Tabela 6.2.11 – Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função de Schwefel (N=10). 137
Tabela 6.2.12 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função 6-HumpCamel Back. 138
Tabela 6.2.13 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função de Shubert. 139
Tabela 6.2.14 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, transposição e transposição/torneio para a função N-dimensional (n=1). 140

Tabela 6.2.15 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função N-dimensional (n=2)..... *140*
Tabela 6.2.16 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função N-dimensional (n=3)..... *141*
Tabela 6.2.17 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função N-dimensional (n=4)..... *141*
Tabela 6.2.18 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional I. *142*
Tabela 6.2.19 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional II (n=1). *143*
Tabela 6.2.20 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional II (n=2). *143*
Tabela 6.2.21 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional II (n=3). *144*
Tabela 6.2.22 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional II (n=4). *144*
Tabela 6.2.23 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional de Rastrigin. *145*
Tabela 6.2.24 – Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Uni-dimensional. *146*

CAPÍTULO 1
INTRODUÇÃO

Sumário

Os Algoritmos Genéticos (AG) são métodos de procura estocásticos que permitem a pesquisa paralela de espaços de procura complexos, normalmente intratáveis pelos métodos tradicionais. Um AG clássico inspira-se nos processos biológicos que explicam a progressão dos seres vivos através do mecanismo de selecção natural e da acção dos operadores genéticos.

Desde o aparecimento dos AG, em 1975 [Holland 1992], foram propostas várias alterações ao AG tradicional, quer na forma de representação do problema, quer no tipo de operadores utilizados.

O trabalho desenvolvido nesta tese de mestrado constitui um novo contributo para esta área de investigação. Procurando inspirações na biologia conclui-se que a variabilidade genética das populações resulta de inúmeros factores que vão muito além da recombinação e da mutação. Um desses factores consiste no mecanismo de transposição, objecto de estudo deste trabalho.

Ao longo desta tese abordaremos as características do AG tradicional, as inúmeras variantes que foram propostas desde o seu aparecimento e apresentaremos uma alteração particular: substituir o operador de *crossover* pelo mecanismo de transposição.

Os resultados obtidos através da análise de um conjunto vasto de funções de teste mostra que, em determinadas circunstâncias, a qualidade das soluções obtidas pelo mecanismo de transposição é superior à obtida pelos operadores tradicionais de recombinação.

1.1. Algoritmos Genéticos: Generalidades

Os anos 60 foram os anos do aparecimento dos AG e com eles uma das mais fascinantes áreas de investigação.

Os AG foram introduzidos por John Holland e desenvolvidos por Holland e o seu grupo da Universidade do Michigan nos anos 60 e 70 [Holland 1992]. Inicialmente, Holland propôs os AG como uma ferramenta para estudar o fenómeno da adaptação dos seres vivos tal como ocorre na natureza. Posteriormente, os AG encontraram como um grande potencial a área de resolução de problemas.

Os AG são métodos de procura adaptativos utilizados na pesquisa de soluções e na resolução de problemas. Têm a sua inspiração nos processos biológicos dos organismos vivos, nomeadamente na "sobrevivência do mais forte" de acordo com o princípio da evolução (Charles Darwin).

A ideia subjacente aos AG é "imitar" aquilo que a natureza faz. Assim, através de um processo iterativo, ao longo de várias gerações, o algoritmo evolui uma população de soluções candidatas para um dado problema. A população inicial consiste, em geral, num conjunto de soluções potenciais distribuídas aleatoriamente no espaço de procura. Através dos operadores de selecção, recombinação e mutação, os indivíduos vão sendo alterados evoluindo para zonas mais promissoras do espaço de procura até, espera-se, à solução óptima.

Tal como nos processos biológicos reais, os indivíduos de cada geração são obtidos por recombinação do material genético dos seus progenitores e podem ser também sujeitos a mutações. Estes dois operadores genéticos permitem manter a diversidade da população e evitar que o AG convirja prematuramente para um máximo local.

Se o problema for codificado correctamente a população evoluirá em gerações sucessivas de tal forma que a qualidade[1] do melhor indivíduo e a qualidade média da população em cada geração aumentará em direcção a um máximo. Esta conclusão é fundamentada pelo estudo matemático desenvolvido por John Holland, denominado

[1] do inglês, *fitness*

teorema do esquema [Holland 1992]. Este teorema baseia-se na noção de **esquema**[2] e mostra como um AG utilizando representação binária e com uma população de indivíduos sujeita aos efeitos da selecção, *crossover* e mutação, permite o aumento exponencial de esquemas de elevada qualidade ao longo das gerações. O teorema do esquema será abordado com mais detalhe no capítulo 2.

Existe uma vasta classe de problemas interessantes para os quais não existe um algoritmo "suficientemente" rápido e eficaz capaz de os resolver. Muitos destes problemas são **problemas de optimização** que surgem frequentemente em inúmeras aplicações. Dado um determinado problema de optimização, com um elevado grau de dificuldade, é possível encontrar um processo eficiente que encontre soluções próximas do óptimo. Por exemplo, para algumas tarefas de optimização, podem ser utilizados algoritmos probabilísticos. No entanto, nada garante que estes algoritmos encontrem a solução desejada.

De uma forma geral, qualquer tarefa de optimização pode ser vista como um problema que se pretende solucionar, o qual, por sua vez, pode ser descrito como uma procura através de um espaço de soluções possíveis. Quando o espaço de procura é pequeno, qualquer algoritmo que proceda a uma procura exaustiva é suficiente para encontrar a solução desejada. No entanto, se o espaço de procura for grande estes métodos exaustivos têm que ser postos de parte e recorre-se à utilização de técnicas de **Inteligência Artificial** (IA). Os AG enquadram-se nestas técnicas inteligentes e provaram ser ferramentas poderosas quando aplicados à classe de problemas referidos. Exemplos disso são problemas de escalonamento, controlo adaptativo, jogos, optimização de funções, o problema do caixeiro viajante, etc. [Chambers 1995a], [Chambers 1995b], [Davis 1991], [Bäck et al. 1997].

1.2. Os Algoritmos Genéticos e a Biologia

Dependendo do problema que se pretende abordar, o AG tradicional é frequentemente sujeito a alterações, nomeadamente no que respeita à representação e às características dos operadores genéticos utilizados.

[2] do inglês, *schema*

A literatura científica aponta como uma possível área de investigação a procura de novas alterações aos AG que possam trazer vantagens quer na convergência do algoritmo, quer na qualidade das soluções. Por exemplo, novas ideias inspiradas na biologia e na genética são frequentemente referidas como uma forte área de investigação [Mitchell 1996].

[Burke et al. 1998] partilham esta opinião e salientam a importância da biologia para o sucesso dos AG. Os autores referem que os AG actuais são implementados focando principalmente objectivos de engenharia e, sem criticar estes AG, apresentam um projecto que pretende aproximar o AG da biologia. Este trabalho pretende diminuir a lacuna entre as abordagens actuais e os sistemas biológicos e tem como objectivo a construção de um modelo computacional de evolução de sistemas virais.

Focando-se nos operadores genéticos, [Mitchell et al. 1994] apontam a importância do estudo das novas descobertas da genética molecular, particularmente dos mecanismos de rearranjo do material genético (transposição[3], deleção[4] e duplicação[5] de genes, intrões[6] e exões[7]) de forma a saber se alguns destes mecanismos pode trazer vantagens ao AG.

Também [Banzhaf et al. 1998] apontam no mesmo sentido, salientando a inexistência de um estudo exaustivo de outros mecanismos, como a transposição, a transdução[8] e a conjugação[9]. Estes autores acreditam que esta é uma forte área para investigar.

De facto, na natureza constata-se que os organismos evoluem devido à existência de diferenças entre eles. Estas diferenças genéticas surgem, não apenas devido aos processos de *crossover* e mutação[10]. Existe um conjunto de mecanismos responsáveis pela variação genética das populações de indivíduos, mesmo em organismos assexuados. Por exemplo, um desses mecanismos - a transposição - é responsável pelo

[3] do inglês, *transposition*
[4] do inglês, *deletion*
[5] do inglês, *duplication*
[6] do inglês, *introns*
[7] do inglês, *exons*
[8] do inglês, *transduction*
[9] do inglês, *conjugation*
[10] do inglês, *mutation*

aparecimento de grandes vantagens evolutivas em populações de bactérias, como sejam a resistência aos antibióticos [Russell 1998], [Gould et al. 1996].

A ideia subjacente a esta tese de mestrado foi precisamente o estudo de mecanismos biológicos presentes nos seres vivos, capazes de serem adaptados e incorporados no AG. Existe uma grande variedade desses mecanismos que serão descritos no capítulo 3. Em vez de procedermos a um estudo alargado e pouco exaustivo de um vasto conjunto de mecanismos, optámos pela análise detalhada de apenas um deles: a transposição. A escolha deste mecanismo deveu-se ao facto de um estudo preliminar com apenas uma função de teste ter conduzido a resultados muito promissores [Simões et al. 1999a] que achámos que deveriam ser aprofundados.

Seleccionado o mecanismo, realizou-se um extenso estudo comparativo com os operadores de *crossover* tradicionais: um ponto de corte, dois pontos de corte e uniforme. Para este estudo comparativo utilizou-se um conjunto de dezoito funções, que serviram como medida de desempenho do AG. No capítulo 5 apresentam-se detalhadamente todas as características das funções utilizadas. A escolha deste conjunto de funções resultou de uma análise cuidada de um vasto número de funções de teste, disponibilizadas quer na Internet, quer na literatura cientifica. As funções foram seleccionadas segundo dois critérios fundamentais: por um lado, abrangerem características distintas (continuidade, dimensão, número de máximos/mínimos, etc.) e, por outro, serem já conjuntos de teste estudados por diversos autores, com garantias de serem boas medidas de desempenho para o problema a analisar.

Os resultados obtidos através da optimização das funções estudadas mostram que, substituindo o operador de *crossover* pelo mecanismo de transposição, se obtêm soluções de melhor qualidade.

1.3. Organização do trabalho

Esta tese encontra-se organizada da seguinte forma: no capítulo 2, aprofundaremos as características de um algoritmo genético clássico, nomeadamente o funcionamento dos diversos mecanismos de selecção, de recombinação e de mutação. No capítulo 3, apresenta-se um conjunto de mecanismos inspirados nos processos biológicos e que permitem obter variação genética nos seres vivos. Nestes mecanismos

inclui-se o de transposição. No capítulo 4, aborda-se com detalhe a forma utilizada para implementar o mecanismo de transposição, bem como todas as simplificações assumidas. O ambiente de estudo para a análise dos resultados é apresentado no capítulo 5. Todas as funções, particularmente as suas características, serão aqui pormenorizadas.

No capítulo 6, apresenta-se o estudo comparativo do mecanismo de transposição com os operadores de *crossover* com um ponto de corte, dois pontos de corte e uniforme. Finalmente, no capítulo 7, serão apresentadas as conclusões e algumas indicações para trabalho futuro. Em anexo, fornece-se um vasto conjunto de gráficos que permitem ilustrar de forma mais clara as características das funções abordadas no capítulo 5 e os resultados obtidos e analisados no capítulo 6.

CAPÍTULO 2.
ALGORITMOS GENÉTICOS

Sumário

Os Algoritmos Genéticos (AG) são técnicas de optimização estocásticas propostas por John Holland [Holland 1992] e inspiradas no processo de evolução através de selecção natural sugerida por Charles Darwin na sua obra "A Origem das Espécies" e complementada pelos princípios da genética inicialmente propostos por G. Mendel.

Os AG utilizam normalmente uma população de soluções candidatas para um dado problema. Os indivíduos dessa população são seleccionados para reprodução de acordo com a sua qualidade. O processo de reprodução consiste na troca de informação entre os progenitores e é designado por *crossover*. Os novos indivíduos assim gerados podem ser alterados de forma localizada através dos efeitos de um operador conhecido por mutação.

Neste capítulo analisaremos detalhadamente o funcionamento do AG clássico, nomeadamente o funcionamento dos seus elementos constituintes básicos: representação, função de avaliação, selecção, *crossover* e mutação.

Abordaremos as áreas de aplicação dos AG, com particular destaque para a área de optimização, utilizada neste trabalho.

Para finalizar apresentaremos os trabalhos já propostos por diversos autores no âmbito da alteração do AG clássico com inspiração nos processos biológicos.

2.1. Introdução

Nesta secção abordaremos com pormenor o funcionamento do AG tradicional. Começaremos por descrever a terminologia normalmente utilizada e que tem a sua inspiração na biologia. De seguida abordaremos quais os elementos constituintes de um AG simples. De seguida, analisaremos a estrutura e funcionamento do algoritmo. Finalmente, será feita uma descrição pormenorizada dos seus elementos.

Para leitores menos familiarizados com esta área recomendam-se os seguintes textos introdutórios: [Beasley et al. 1993a], [Beasley et al. 1993b], [De Jong et al. 1993a], [Crutchfield et al. 1995], [Fogel 1995], [Forrest 1996], [Goldberg 1989], [Haupt et al. 1998], [Mitchell 1996], [Mühlenbein 1993], [Spears et al. 1993] e [Whitley 1993].

2.1.1. Terminologia

Inspirados nos processos biológicos da genética das populações e da evolução através da selecção natural, os AG utilizam uma terminologia biológica que passamos a descrever.

Todos os organismos vivos são constituídos por células que contêm o mesmo conjunto de um ou mais **cromossomas**. Os cromossomas consistem em cadeias de **ADN** (Ácido Desoxirribonucleico[11]) constituídas por **genes** que codificam determinadas características dos indivíduos. Os diferentes valores que cada gene pode tomar designam-se por **alelos**. A posição de um gene na cadeia de **ADN** chama-se ***locus***.

A todo o conjunto de material genético de um organismo chama-se **genoma**. O conjunto particular de genes contido no genoma designa-se por **genótipo**. O genótipo codifica o **fenótipo** de um indivíduo, isto é, o conjunto das suas características visíveis.

Uma célula diz-se **diplóide** se os seus cromossomas se encontram aos pares (pares de cromossomas homólogos), é este o caso do Homem. Caso contrário o organismo diz-se **haplóide**.

Os organismos diplóides utilizam a reprodução sexuada e ao haplóides a reprodução assexuada.

[11] do inglês, *DNA (Deoxyribonucleic acid)*

O processo de reprodução sexuada, envolve a união de duas células sexuais (gâmetas) haplóides, provenientes de cada um dos progenitores. A fusão dos gâmetas forma uma nova célula diplóide que dará origem a um novo indivíduo,

Os organismos haplóides reproduzem-se através de **fissão binária** de uma célula-mãe que dará origem a duas células-filhas contendo o mesmo material genético da sua progenitora.

Em qualquer dos casos, a nova descendência gerada pode ser sujeita a mutações que provocam alterações no material genético e, consequentemente, no seu fenótipo.

O mérito ou qualidade de um indivíduo é quantificado por uma função de avaliação[12] e traduz a capacidade que ele tem em viver e produzir descendência. Indivíduos de melhor qualidade terão mais oportunidades para se reproduzir e, portanto, impor as suas características na população.

Nos AG não existe uma simulação exacta dos fenómenos atrás descritos. Em vez disso, inspiram-se nesses modelos biológicos utilizando operadores e representações que podemos encontrar na natureza.

Um AG trabalha sobre uma população de indivíduos cujo tamanho permanece, em geral, constante ao longo das gerações. Estes indivíduos, designados **cromossomas**, representam as soluções candidatas para um dado problema. Os cromossomas são constituídos por um conjunto de **genes** (por exemplo uma sequência de *bits*), que podem tomar diferentes valores, por exemplo, *0* ou *1*. Aos diferentes valores que um gene pode tomar dá-se o nome de **alelo**.

O mecanismo de ***crossover*** consiste na troca de material genético de dois progenitores, seleccionados de acordo com a sua função de avaliação.

Os novos indivíduos podem ser sujeitos a **mutações** que provocam trocas dos valores de um ou mais genes.

A grande maioria das aplicações em AG utilizam indivíduos **haplóides** (com um único cromossoma).

O **genótipo** de um indivíduo num AG consiste na sequência de genes do(s) seu(s) cromossoma(s).

[12] do inglês, *fitness function*

2.1.2. Elementos Básicos de um AG

Como já foi referido, um AG permite a procura de espaços de procura complexos, normalmente intratáveis por métodos tradicionais. Para que o algoritmo permita a resolução do problema em causa, é necessário que este codifique correctamente as especificações do problema a resolver. Assim, torna-se necessária a definição de alguns aspectos muito importantes:

- Representação a utilizar pelos indivíduos da população;
- Método de selecção utilizado que garanta que os melhores indivíduos da população tenham maior probabilidade de sobreviver e de se reproduzir.
- Operadores a aplicar para a obtenção de novos indivíduos e quais as probabilidades com que estes actuarão;
- Função de avaliação que permita medir o mérito de cada indivíduo;
- Características da população: tamanho, geração da população inicial, etc.

2.1.3. Funcionamento do Algoritmo

Depois de definidos todos os aspectos abordados na secção anterior, o funcionamento de um AG pode descrever-se da seguinte forma:

A população inicial é, normalmente, gerada aleatoriamente. Esta população inicial evoluirá ao longo de um número de gerações através da actuação dos mecanismos de selecção, *crossover* e mutação. Os melhores indivíduos da população são seleccionados de acordo com a sua qualidade (medida pela função de avaliação) para que se possam reproduzir. A reprodução consiste na troca de material genético entre os dois progenitores envolvidos (*crossover*) e dará origem à nova descendência. Estes novos indivíduos poderão ainda ser alterados através dos efeitos do operador de mutação. A forma como se define a nova população pode variar. Se todos os indivíduos da população anterior forem substituídos por novas soluções, o processo designa-se por **geracional**[13] ou técnica de sobreposição[14]. Se apenas um grupo de indivíduos for

[13] do inglês, *generational*
[14] do inglês, *overlapping*

substituído (os piores), o processo designa-se por **estado estável**[15] ou técnica de não-sobreposição[16].

Estes passos repetem-se ao longo de um número de gerações ao longo das quais a população sofrerá um processo evolutivo que, em principio, conduzirá à solução óptima.

O funcionamento de Algoritmo Genético simples está representado na figura 2.1.

1. Gera população inicial
2. ENQUANTO *não terminar* FAZ
    2.1 Avalia população
    2.2 Selecciona progenitores para reprodução
    2.3 Gera descendência: aplicação do operador de *crossover*
    2.4 Aplica operador de mutação
    2.5 Substitui população antiga pela descendência gerada
FIM ENQUANTO

Figura 2.1 *- O Algoritmo Genético Clássico*

A figura 2.1 representa de forma genérica a estrutura e funcionamento de um AG clássico. Existem vários tipos de selecção que são normalmente utilizados, alguns dos quais serão descritos mais adiante. O mesmo se passa com o operador de *crossover*. Tradicionalmente utiliza-se este operador com um único ponto de corte, mas existem outras variantes que também serão abordadas a seguir. A escolha do operador de *crossover* é bastante importante, visto que ele permite ao AG explorar o espaço de procura e encaminhar-se para regiões mais promissoras, aproximando-se assim da "melhor" solução. Por outro lado, o operador de mutação altera aleatoriamente o valor de um ou mais genes do cromossoma permitindo explorar localmente o espaço de procura e impede a convergência prematura do AG para máximos locais.

Na próxima secção analisaremos em detalhe cada um dos elementos básicos de uma AG.

[15] do inglês, *steady-state*
[16] do inglês, *non-overlapping*

2.1.4. Representação

A escolha da representação das soluções candidatas é a primeira componente a ser decidida, uma vez que dela dependem o funcionamento de outros, como por exemplo, os operadores genéticos.

A estrutura de um vector-solução em qualquer problema de procura e optimização depende do problema subjacente. Em alguns problemas uma solução consiste numa vector de valores reais que especificam as dimensões dos parâmetros chave desse problema. Noutro tipo de problemas, uma solução consiste numa estratégia ou algoritmo para atingir a concretização de uma tarefa [Deb 1997]. Da mesma forma que a natureza das soluções varia de problema, uma solução para um problema particular pode também ser representada de diferentes formas. Portanto, a escolha da representação mais adequada à tarefa que se pretende resolver é um aspecto muito importante, da qual dependerá o desempenho do AG.

Se um AG tradicional utiliza uma representação binária para codificar as soluções candidatas para o problema, muitos outros tipos de representações existem, das quais destacamos além desta, a representação em números reais e em árvore. Outras representações menos utilizadas serão também abordadas com menos detalhe.

Representação binária

A implementação tradicional do AG utiliza a **representação binária** para os cromossomas. Nesta representação, cada cromossoma é constituído por uma sequência de bits e têm, normalmente, tamanho fixo.

Por exemplo, suponhamos que se pretendia um AG para optimizar a seguinte função [Michalewicz 1999]:

$$f(x) = x.sin(10\Pi.x) + 1.0$$

Esta função é unidimensional, sendo o domínio da variável ***x*** definido por [-1, 2]. Um cromossoma deve codificar esta variável através de uma cadeia binária. Suponhamos que a precisão dos valores a utilizar é de seis casas decimais. Como o domínio da variável tem comprimento igual a 3, a precisão exigida implica que o

intervalo [-1, 2] seja dividido em pelo menos 3*1.000.000 intervalos de igual dimensão. Assim, são necessários 22 bits para codificar a variável *x*:

$$1.097.152 = 2^{21} < 3.000.000 \leq 2^{22} = 4.194.304$$

O mapeamento de uma cadeia binária $(b_{21}b_{20}... b_0)$ no seu correspondente real ***x*** é definida em duas etapas:

1. converter a cadeia binária da base 2 para a base 10:

$$\left(\langle b_{21}b_{20}...b_0\rangle\right)_2 = \left(\sum_{i=0}^{21} b_i.2^i\right)_{10} = x'$$

2. encontrar o número real **x** correspondente:

$$x = -1.0 + x'.\frac{3}{2^{22}-1}$$

onde -1.0 é o limite inferior do domínio de **x** e 3 é o tamanho do domínio.

Por exemplo, o cromossoma (1000101110110101000111) representa o número 0.637197, uma vez que:

$x' = (1000101110110101000111)_2 = 2288967$

e

$$x = -1.0 + 2288967.\frac{3}{4194303} = 0.637197$$

Obviamente, os cromossomas (0000000000000000000000) e (1111111111111111111111) representam os limites do domínio -1.0 e 2.0, respectivamente.

<u>*Representação real*</u>

Apesar de alfabetos de menor dimensão garantirem maior paralelismo implícito, não existe qualquer evidência empírica que indique que a representação binária é mais eficiente na resolução de problemas de optimização envolvendo valores reais [Deb 1997]. De facto, a representação binária apresenta algumas desvantagens quando aplicada a problemas numéricos multidimensionais e de alta precisão [Michalewicz 1999]. Esta representação, apesar das vantagens que lhe estão subjacentes - o já referido

paralelismo implícito e uma análise teórica mais fácil - torna-se desadequada para certo tipo de problemas, sendo mais natural a utilização de vectores-solução com números reais.

Neste tipo de representação, cada solução candidata é representada por um vector de números reais que devem estar confinados a valores pertencentes ao domínio do problema em causa. Quando é usada uma representação real, os operadores de mutação e *crossover* têm que ser adaptados à nova codificação das soluções. O operador de *crossover* é análogo ao utilizado com cadeias binárias, isto é, consiste na troca de porções entre os números reais. O operador de mutação, afecta aleatoriamente um número real do vector. O seu resultado consiste na alteração deste valor por outro pertencente ao domínio. Além destes operadores tradicionais, outros foram implementados para utilizar em algoritmos com representação real, por exemplo, a mutação não-uniforme e o *crossover* aritmético [Michalewicz 1999].

Representação em árvore

A representação em árvore é tradicionalmente utilizada nas implementações com programação genética. Nos estudos desta área, uma solução é, geralmente, um programa em LISP especificando uma estratégia ou algoritmo para resolver um determinado problema. Para construir uma solução válida utilizam-se funções e terminais, sendo mantida a sintaxe e a estrutura de cada função. Portanto, se, por exemplo, na solução se utilizar a função OR , então pelo menos dois argumentos, retirados do conjunto de terminais, terão que ser atribuídos a esta função de forma a que ela execute uma operação válida [Koza 1992]. Assim, em programação genética, uma população consiste num conjunto de programas compostos por funções primitivas e terminais. Cada indivíduo da população é representado através de uma árvore, na qual as folhas são terminais que funcionam como argumento para nodos superiores dessa árvore, isto é, as funções. A figura 2.2 representa um programa contendo as funções OR e AND e os terminais *Arg1*, ... *Arg4*.

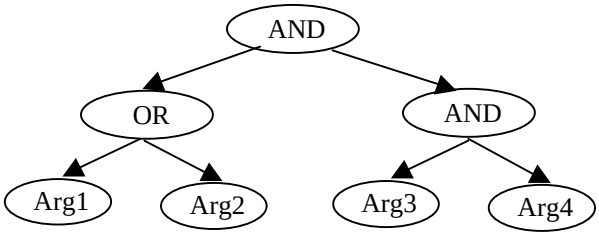


Figura 2. 2 - *Representação em Árvore*

A evolução da população inicial até à solução (programa) desejada é conseguida à custa de vários operadores genéticos. Além da mutação, que consiste na alteração de porções da árvore, e do *crossover* que se caracteriza pela troca de ramos entre duas árvores diferentes, outros operadores são vulgarmente utilizados nesta abordagem. Os operadores de duplicação e deleção podem ser aplicados a uma secção ou a um argumento da estrutura arborescente, duplicando ou eliminando, porções da árvore ou argumentos, respectivamente. De forma análoga, o operador de criação[17] pode ser aplicado na produção de ramos ou argumentos adicionais.

A representação em árvore pode trazer algumas vantagens, por exemplo, a possibilidade de formar novas estruturas que aumentem o espaço de procura pesquisado, uma vez que se podem formar árvores de qualquer tamanho através do operador de *crossover*. A maior desvantagem que este tipo de representação acarreta é o facto de as árvores poderem crescer descontroladamente evitando a formação de soluções hierarquicamente mais estruturadas [Mitchell 1996].

Outras Representações

Além das representações anteriormente focadas, as abordagens evolucionárias podem utilizar outro tipo de representação para as soluções candidatas ao problema. Por exemplo, representações de máquinas de estado finito [Fogel 1991], [Fogel 1993], [Fogel 1995], representações híbridas utilizando valores discretos e contínuos (Bäck et al. 1995] e intrões [Levenick 1991].

2.1.5. Métodos de Selecção

Quando se implementa um AG é necessário definir a forma como será realizada a selecção dos indivíduos que vão produzir a nova geração.

Tradicionalmente, o mecanismo de selecção deve possibilitar que os melhores indivíduos se reproduzam mais vezes para que, desta forma, a população vá evoluindo até à convergência.

O método de selecção deve ser conjugado com os operadores de *crossover* e mutação para que o AG atinja a solução para o problema. Assim, uma selecção muito exigente levará a que a população seja dominada muito rapidamente pelos melhores indivíduos o que pode levar à estagnação num máximo local. Por outro lado, uma selecção pouco exigente poderá conduzir a um processo de evolução muito lento.

Vários métodos de selecção têm sido propostos por diversos autores, não existindo no entanto uma resposta para qual o melhor método a utilizar numa dada situação. A referência a muitos destes (e outros) tipos de mecanismos de selecção podem ser analisados com mais detalhe em [Blickle et al. 1996], [Petrowsky 1997], [Syswerda 1991a] e [Thierens 1997].

Apresenta-se de seguida uma breve descrição dos métodos de selecção mais usados.

Selecção através de Roleta

Este método de selecção, também designada por **amostragem estocástica com substituição**[18] ou selecção **proporcional ao mérito**[19], foi utilizado no trabalho original de John Holland [Holland 1992]. Baseia-se no valor atribuído a cada indivíduo pela função de avaliação e na qualidade média da população. De acordo com a qualidade de cada indivíduo, atribui-se uma porção de uma circulo (roleta) a cada um deles.

A roleta é girada ***N*** vezes, tantas quantas o número de indivíduos da população, parando numa porção correspondente a um dos indivíduos. Ao fim de ***N*** experiências, estarão seleccionados os progenitores que irão gerar a próxima geração.

[17] do inglês, *creation*
[18] do inglês, *Stochastic Sampling with Replacement,*
[19] do inglês, *Fitness Proportional*

Em termos práticos, a roleta pode ser vista como um segmento. Os indivíduos da população são distribuídos em segmentos contíguos, sendo o tamanho de cada segmento calculado em função do mérito do indivíduo que lhe corresponde. De seguida é gerado um número aleatório e o primeiro indivíduo cujo segmento ultrapassa o valor desse número é escolhido para a reprodução. Este processo é repetido até se obter o número de progenitores desejado.

Vejamos o seguinte exemplo de uma população com 10 indivíduos:

NÚMERO do indivíduo	1	2	3	4	5	6	7	8	9	10
Função de Mérito =f(x)	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2
Probabilidade de Selecção $f(x_i)/\sum f(x_i)$	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02

Suponhamos que se pretendem obter 6 progenitores. Começamos por geral seis números aleatórios. Admitamos que se obtêm os seguintes:

0.81 0.32 0.96 0.01 0.65 0.42

Assim, os indivíduos seleccionados para gerar a descendência serão, respectivamente 6, 2, 9, 1, 5 e 3.

Método de Amostragem Universal Estocástica

[Baker 1987] propôs um método diferente, denominado por **amostragem universal estocástica**[20]. Tal como no método da roleta os indivíduos são mapeados em segmentos adjacentes cujo comprimento é igual ao valor dado pela função de avaliação a cada indivíduo.

Neste método dispõem-se ***N*** ponteiros igualmente espaçados entre si (***N*** = número de progenitores a seleccionar) e a roleta gira apenas uma vez. Os progenitores escolhidos são os indivíduos marcados pelos ***N*** ponteiros.

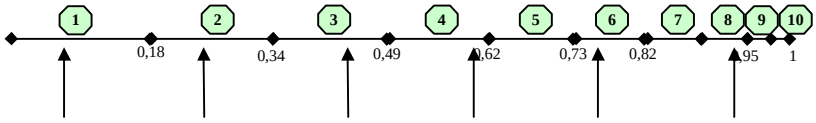
[20] do inglês, *Stochastic Universal Sampling*

A distância entre os ponteiros será **1/N** e a posição do primeiro ponteiro é dada por um número gerado aleatoriamente entre **0** e **1/N**.

Para o exemplo dado anteriormente temos 6 indivíduos, logo **1/N** = 0.167.

Gera-se um número aleatório, por exemplo, 0.1

Os 6 ponteiros vão ficar dispostos da seguinte forma:



Os indivíduos seleccionados para gerar a descendência serão os indivíduos 1, 2, 3, 4, 6 e 8.

Selecção Elitista[21]

Este método de selecção foi introduzido por [De Jong 1975] e serve de complemento a um dos outros métodos de selecção utilizados num AG. Procura evitar que os indivíduos de melhor qualidade se percam, retendo um determinado número dos melhores indivíduos para a geração seguinte.

Os restantes indivíduos são obtidos utilizando um dos outros métodos de selecção actuando sobre toda a população inicial.

Selecção Baseada na Posição[22]

Este método de selecção procura evitar a rápida convergência do AG para um máximo local [Mitchell 1996]. A população é ordenada de acordo com o seu mérito e o valor esperado de cada indivíduo é calculado de acordo com a posição que ocupa e não de acordo com o valor absoluto dado pela função de avaliação.

Na versão proposta por [Baker 1987] a população é ordenada por ordem crescente do seu mérito, de 1 até **N** (tamanho da população). O valor esperado de cada indivíduo é dado por:

[21] do ingles, *Elitist*
[22] do inglês, *Rank-based*

$$ValEsp(i,t) = Min + (Max - Min) * \frac{rank(i,t) - 1}{N - 1}$$

Onde ***Max*** é o valor esperado do indivíduo da posição **N** e ***Min***, o valor esperado do indivíduo da posição **1**. Os valores de ***Max*** e ***Min*** são escolhidos pelo utilizador, de acordo com algumas restrições.

Selecção por Torneio

Este método de selecção é computacionalmente mais eficiente que o anterior e funciona do seguinte modo:

Escolhem-se aleatoriamente dois de indivíduos da população. Gera-se um número aleatório, **r**, entre 0 e 1. Se **r < k,** o melhor dos dois indivíduos é seleccionado para reprodução. Caso contrário, selecciona-se o pior indivíduo. Os dois cromossomas são devolvidos à população e podem voltar a ser escolhidos, visto que este processo é repetido um número de vezes igual ao tamanho da população.

Selecção Estado Estável

De uma forma geral, os AG mais utilizados e descritos na literatura são **geracionais**. Isto é, toda a população é substituída de geração para geração. Em alguns métodos de selecção, por exemplo, os elitistas, alguns indivíduos (normalmente os melhores) são preservados para a geração seguinte. A fracção de novos indivíduos produzidos em cada geração designa-se por lacuna de gerações[23] [De Jong 1975].

No tipo de selecção estado estável, apenas um pequeno número de indivíduos é substituído de geração em geração. Os indivíduos a alterar serão os piores indivíduos da população, substituídos por novos indivíduos gerados por *crossover* dos melhores indivíduos e posterior mutação.

Este método foi já analisado por [Syswerda 1989], [Syswerda 1991a], [Whitley 1989] e [De Jong et al. 1993b].

[23] do inglês, *generation gap*

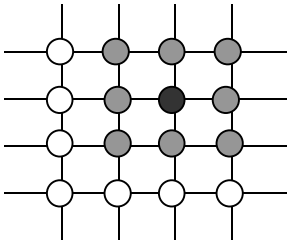
Selecção por Truncatura

O método de selecção por truncatura é normalmente utilizado para populações de grandes dimensões. Os indivíduos da população são ordenados de acordo com a sua função de mérito e apenas os melhores indivíduos poderão ser seleccionados para a reprodução. A percentagem de indivíduos que pode ser seleccionado depende de um parâmetro, que varia geralmente entre 10% e 50%, e que indica a percentagem dos melhores indivíduos que se irá reproduzir.

Selecção local

Ao contrário dos outros métodos de selecção que permitem que um indivíduo acasale com qualquer outro indivíduo da população, neste método de selecção todos os indivíduos se encontram limitados a um ambiente restrito, designado por vizinhança local. Cada indivíduo apenas pode interagir e acasalar com indivíduos da mesma vizinhança.

Por exemplo, se a vizinhança for igual a 1, o indivíduo assinalado a preto apenas poderá gerar descendência e interagir com os indivíduos marcados a cinzento.



2.1.6. Operadores de Crossover

Este operador genético permite manter a diversidade da população. Os progenitores seleccionados para reprodução trocam o seu material genético com uma probabilidade que tipicamente varia entre 0.5 e 0.8, produzindo indivíduos com novas características.

A forma como é trocado o material genético depende do tipo de *crossover* utilizado. Existem três tipos de *crossover* que se tornaram os mais vulgares num AG:

- *Crossover* com um ponto de corte;
- *Crossover* com **N** (N>1) pontos de corte;
- *Crossover* Uniforme.

Vários estudos foram feitos sobre o papel que o operador de *crossover* desempenha num AG. Por exemplo, [Spears 1992a] e [Spears 1995] apresentam um trabalho empírico em que o AG, à medida que é executado, escolhe qual o tipo de *crossover* (2 pontos de corte ou uniforme) que maximiza o desempenho do AG.

[Spears et al. 1990] e [De Jong et al. 1992] apresentam uma análise cuidada para o operador de *crossover* com ***N*** pontos de corte. Neste trabalho, os autores procuram responder à questão de quantos pontos de corte se devem usar no operador de forma a maximizar os resultados.

Apesar de, tradicionalmente, o AG utilizar o operador genético de *crossover* com uma probabilidade fixa, [Davis 1989] mostra que adaptando as probabilidades ao longo da simulação do AG, o desempenho do algoritmo para tarefas mais difíceis pode ser melhorado.

Outros trabalhos interessantes sobre o operador de *crossover*, que incluem alterações ao operador dependentes do domínio de aplicação são [Arabas et al. 1995], [Davidor 1989], [D'Haeseleer 1994], [Eshelman et al. 1993], [Harries et al. 1997], [Jones 1995], [Kahng et al. 1995], [Mathias et al. 1992], [McDonnell et al. 1995], [Schaffer et al. 1991], [Spears et al. 1991a], [Spears et al. 1991b], [Starkweather et al. 1991], [Syswerda 1993] e [Wu et al. 1997].

Vejamos agora, com mais detalhe, o funcionamento dos operadores de *crossover* mais importantes.

Crossover com um ponto de corte

Gera-se aleatoriamente qual o *locus* que irá funcionar como ponto de corte. De seguida troca-se o material genético dos dois progenitores tendo em conta o ponto de corte seleccionado.

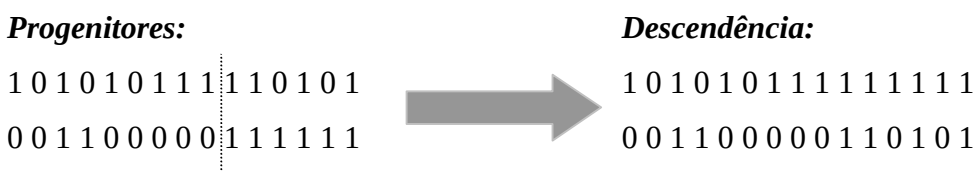


Figura 2. 3 - ***Funcionamento do Operador de Crossover com 1 ponto de corte***

Crossover com N (N > 1) pontos de corte

Funciona de forma análoga ao caso anterior. A troca de material genético será feita dependendo do número de pontos de corte utilizados.

Vejamos um exemplo para 2 pontos de corte (gerados aleatoriamente):

Progenitores:		***Descendência:***
1 0 1 0 1 0 1 1 1 1 1 0 1 0 1	→	1 0 1 0 0 0 0 0 0 1 1 0 1 0 1
0 0 1 1 0 0 0 0 0 1 1 1 1 1 1		0 0 1 1 1 0 1 1 1 1 1 1 1 1 1

Figura 2. 4 - ***Funcionamento do Operador de Crossover com 2 pontos de corte***

Crossover Uniforme

Esta forma de *crossover*, proposta por [Syswerda 1989], pretende generalizar o método de N pontos de corte, fazendo com que cada *locus* do cromossoma seja um potencial ponto de corte. Neste trabalho de 1989 o autor demonstra que, na maioria dos casos, este tipo de *crossover* consegue melhores resultados que os anteriores.

O funcionamento deste mecanismo baseia-se numa máscara, gerada aleatoriamente e cujo tamanho coincide com o tamanho do cromossoma. A paridade dos bits da máscara indicará qual dos progenitores irá fornecer o material genético que fará parte dos novos indivíduos.

Para o primeiro filho, um bit a **0** na máscara indica que o gene será do progenitor 1, um bit a **1** indica que o gene pertence ao outro progenitor. Para o segundo filho o processo é inverso: um bit a **0** indica que o gene provém do progenitor 2 e um bit a **1** indica que o gene é fornecido pelo pai 1.

Eis um exemplo deste processo:

Progenitores		***Descendência***	
Pai 1	1 0 1 0 1 0 1 1 1 1 1 0 1 0 1	Filho 1	0 0 1 0 1 0 0 1 1 1 1 1 1 1 1
Pai 2	0 0 1 1 0 0 0 0 0 1 1 1 1 1 1	Filho 2	1 0 1 1 0 0 1 0 0 0 1 0 1 0
Máscara	**1 1 1 0 0 1 1 0 0 0 1 1 0 1 0**		

Figura 2. 5 - ***Funcionamento do Operador de Crossover Uniforme***

2.1.7. Operador de Mutação

A grande maioria da comunidade dos AG vê como operador principal o operador de *crossover*. No entanto, depois de gerada a descendência, esta pode ser sujeita aos efeitos de outro operador genético: a mutação.

A mutação consiste na alteração do valor de um ou mais genes de um cromossoma. Este operador é, normalmente, utilizado com uma probabilidade muito baixa, tipicamente 0.001.

Vejamos um exemplo:

Figura 2. 6 - ***Funcionamento do Operador de Mutação***

Apesar de visto como operador secundário em relação ao operador de *crossover*, a mutação desempenha um papel importante no funcionamento do AG. Este operador evita que o algoritmo estagne num máximo local, "agitando-o" e fazendo com que este explore outras regiões eventualmente mais promissoras.

Um estudo comparativo entre os dois operadores genéticos foi realizado por [Spears 1992b]. O autor demonstra teoricamente que existem algumas características importantes de um operador que não são capturadas pelo outro. Por exemplo, tal como Holland, Spears conclui que o papel do operador de *crossover* consiste em construir blocos construtores de ordem mais elevada, a partir de blocos de ordem inferior. O operador de mutação não desempenha este papel com tanta eficácia. O operador de mutação permite a destruição de blocos construtores existentes, permitindo a exploração de espaços menos prováveis de serem visitados pelo AG.

Apesar de, normalmente, o AG ser utilizado com uma probabilidade de mutação fixa ao longo de todas as gerações, existem estudos realizados com taxas variáveis. Por exemplo, [Bäck 1993] apresenta um trabalho em que utiliza uma taxa de mutação variável ao longo do AG, com vista a maximizar o seu desempenho. O autor realizou um estudo empírico em que o AG foi testado com três funções de avaliação concluindo

que a variação da taxa de mutação, à medida que o AG é executado, é vantajosa em determinadas situações.

Também [Fogarty 1989] utiliza a mesma ideia numa aplicação industrial. Tal como Bäck, este autor conclui que a variação da taxa de mutação ao longo do tempo em que o AG é simulado melhora substancialmente o desempenho do algoritmo na resolução do problema. O autor salienta, no entanto, que variar a probabilidade de mutação ao longo do AG pode não ser vantajoso em todas as aplicações, implicando um custo computacional desnecessário.

Outros trabalhos envolvendo o estudo do operador de mutação podem ser encontrados em [Hinterding et al. 1995], [Jones 1995] e [Wu et al. 1997].

2.1.8. Função de avaliação

O papel da função de avaliação consiste em fornecer uma medida de qualidade das soluções codificadas pelo AG. A escolha desta função de avaliação depende do problema a resolver. Se nalguns casos a sua escolha é bastante fácil, noutros, que envolvam restrições de algum tipo, a escolha da função de avaliação tem que ser feita com algum cuidado. Por exemplo, a introdução de um factor de penalização ou a combinação de diferentes objectivos na mesma função, tornam a sua escolha um processo mais difícil.

2.1.9. População

Como já foi referido, o AG opera sobre uma população de soluções candidatas para um determinado problema e selecciona as de melhor qualidade para se reproduzirem. Existem algumas escolhas que têm que ser estabelecidas no que diz respeito à população.

Em primeiro lugar é necessário definir o tamanho da população. Populações maiores permitem uma maior diversidade de soluções, mas consomem mais tempo de avaliação. Assim, o tamanho da população deve ser escolhido com algum cuidado de forma a garantir por um lado, que a diversidade da população seja mantida e, por outro, que o tempo de computação não comprometa o funcionamento do AG na resolução do problema.

Em [Reeves 1993] é apresentado um estudo teórico sobre a utilização de AG com populações de pequena dimensão. A definição do tamanho "pequeno" é dada por um conjunto de condições, particularmente o facto de qualquer ponto do espaço de procura poder ser atingido apenas com o operador de *crossover*. Outra das constatações deste trabalho, é que, para populações de pequena dimensão, o AG deve gerar a população inicial não de uma forma aleatória, mas sim de um forma sistemática para que se obtenham pontos que permitam uma cobertura mais ampla do espaço procura.

[Balázs 1999] propõe uma abordagem na qual o AG utiliza uma população de tamanho variável, controlada por um modelo matemático inspirado na biologia. O autor propõe ainda uma forma de controlar a pressão selectiva do AG através da alteração dinâmica do tamanho da população.

Outro aspecto importante refere-se à forma como, em cada geração, se define a nova população. Se a população for substituída na totalidade por novos indivíduos corre-se o risco de os melhores indivíduos se perderem e não voltarem a surgir durante algumas gerações. Para evitar este problema utilizam-se técnicas de selecção elitistas, nas quais as melhores soluções (a elite) é mantida de geração para geração. Neste caso, apenas se substituirá o conjunto de indivíduos que não pertence à elite.

2.2. Aspectos teóricos

John Holland [Holland 1992] utilizou no seu trabalho original um AG com representação binária, propondo o **teorema do esquema** como justificação teórica para o seu bom funcionamento. A motivação inicial de Holland ao propor este teorema foi a de que problemas complexos poderiam ser resolvidos mais facilmente se fossem divididos em subproblemas mais simples.

O teorema do esquema parte do pressuposto que o AG utiliza uma população de cadeias binárias, selecção proporcional à qualidade, *crossover* com um único ponto de corte e mutação por mudança aleatória dos alelos.

Um **esquema** consiste numa palavra **H** sobre o alfabeto alargado A’ = {1,0,*}. Por exemplo $\mathbf{H_1}$**=1**01** é um esquema. Na prática um esquema representa um conjunto de palavras sobre o alfabeto **A**. No caso anterior teremos 4 palavras associadas:

10001, 10101, 11001, 11101

A **ordem**[24] de um esquema **H**, denotado por **o(H)**, é o número de posições fixas no esquema, ou seja o número de 1's e 0's presentes no esquema.

No exemplo anterior o(H_1)**=3.**

O **comprimento definidor**[25] de um esquema **H**, denotado por **δ(H)**, é a distância entre a primeira e a última posições definidas no esquema.

Por exemplo, de novo no caso H_1=1**01 o comprimento **δ(H1) = 4.**

Um **bloco construtor**[26] é um esquema de alta adaptabilidade, baixa ordem e comprimento curto.

De acordo com estas definições, podemos agora ver como o AG manipula os esquemas de geração em geração.

Selecção

O mecanismo de selecção escolhe um indivíduo de acordo com a sua qualidade. Assim um indivíduo x_i de qualidade $f(x_i)$ será escolhido com uma probabilidade:

$$\frac{f(x_i)}{\sum f(x_j)}$$

Numa tiragem com reposição de **n** elementos teremos um número esperado de indivíduos x_i dado por:

$$n * \frac{f(x_i)}{\sum f(x_j)}$$

Se **n** for a dimensão da população então:

$$\frac{f(x_i)}{\sum f(x_j)/n} = \frac{f(x_i)}{\overline{f}}$$

com $\overline{f}$ a qualidade média da população.

Pensando agora em termos de esquemas, se **m(H,t)** designar o número de instâncias do esquema **H** numa população, no tempo **t**, então:

[24] do inglês, *order*
[25] do inglês, *defining length*
[26] do inglês, *building block*

$$E(m(H,t+1)) = \frac{\sum_{x \in H} f(x)}{\bar{f}(t)}$$

isto é, na geração seguinte são esperadas um número de instâncias proporcional ao quociente entre a soma das qualidades das instâncias do esquema presentes na população no instante **t**, e a qualidade média da população. Esta fórmula pode ainda ser transformada em:

$$E(m(H,t+1)) = m(H,t) * \frac{\bar{f}(H,t)}{\bar{f}(t)}$$

isto é, o número de elementos do esquema **H** esperados na geração **t+1** é igual ao produto do número desses elementos na geração **t** pelo quociente das qualidades média do esquema pela qualidade média da população.

Desta equação decorre um princípio bastante simples: esquemas de qualidade acima da média tendem a aumentar o número de instâncias ao longo do tempo. Esta variação ainda pode ser tornada mais precisa. Admitamos que a qualidade média do esquema varia de uma constante **c** relativamente à qualidade média da população. Então:

$$E(m(H,t+1)) = m(H,t) * \frac{(\bar{f}(t) + c\bar{f}(t))}{\bar{f}(t)}$$

ou seja:

$$= m(H,t) * \frac{(1+c) * \bar{f}(t)}{\bar{f}(t)} = m(H,t)(1+c)$$

Esta recorrência pode resolver-se por iteração vindo:

$$E(m(H,t+1)) = m(H,0) * (1+c)^{l}$$

Daqui resulta claro, mais uma vez, que a selecção atribui um número exponencialmente crescente (ou decrescente) de instâncias de esquemas acima (abaixo) da média.

Crossover

Pensemos agora nos efeitos destrutivos do *crossover* com um ponto de corte. Suponhamos que o esquema tem comprimento ***l***. Existem então **(*l*-1)** pontos de corte possíveis. Dados dois esquemas qual tem mais hipóteses de ser corrompido pelo *crossover*? Vejamos com um exemplo:

A = 011|1000

$\mathbf{H_1}$=*1*|***0

$\mathbf{H_2}$=***|10**

Temos um indivíduo, **A**, e dois dos seus esquemas possíveis $\mathbf{H_1}$ e $\mathbf{H_2}$. Admitamos que foi definido aleatoriamente o ponto de corte entre o bit 3 e o 4. Ao ser cruzado o indivíduo **A** o esquema $\mathbf{H_1}$ tem muitas hipóteses de ser destruído enquanto $\mathbf{H_2}$ se mantém sem problemas. Esta observação é geral para os diferentes pontos de corte: $\mathbf{H_2}$ sobrevive melhor que $\mathbf{H_1}$. A possibilidade de destruição será assim proporcional ao número de possibilidades de pontos de corte que separam valores definidos. Esse valor é dado pelo comprimento definidor do esquema. Se $\boldsymbol{p_c}$ for a probabilidade de *crossover*, estão podemos definir um limite inferior para a sobrevivência de um esquema:

$$1 - p_c * \frac{d(H)}{l-1}$$

Mutação

A mutação só destrói um esquema se afectar os seus bits definidos. Sendo $\boldsymbol{p_m}$ a probabilidade de mutação e **o(H)** a ordem do esquema termos a probabilidade de sobrevivência devido à mutação dada por:

$$(1 - p_m)^{o(H)}$$

Como em geral $\boldsymbol{p_m}$ é muito inferior a 1 podemos simplificar a expressão (decomposição em série de MacLaurin) que fica:

$$1 - o(H) * p_m$$

Juntando agora estes três efeitos obtemos a fórmula:

$$E(m(H,t+1)) \geq m(H,t) * \frac{\bar{f}(H,t)}{\bar{f}(t)}(1 - p_c * \frac{d(H)}{l-1})[(1 - o(H) * p_m]$$

Que traduz o **Teorema do Esquema**, que de uma forma simples se pode resumir da seguinte forma: o número de instâncias de um determinado esquema cresce exponencialmente ao longo do tempo se estes possuirem três caracteristicas - qualidade acima da média, baixa ordem e baixo comprimento definidor.

Outro aspecto importante é conhecido por **paralelismo implícito**. Admitamos que trabalhamos com uma população de **n** indivíduos, cada um deles de comprimento ***l***. Cada indivíduo da população é instância de 2^l esquemas. Na realidade é como se trabalhássemos num alfabeto binário (o bit definido ou '*'). Deste modo em toda a população haverá no máximo $n*2^l$ esquemas. Mas quantos serão efectivamente processados? De acordo com John Holland [Holland 1992], esse número é da ordem $\mathbf{O(n^3)}$. Este resultado significa que embora sejam apenas processados **n** indivíduos numa geração o algoritmo genético processa algo como $\mathbf{n^3}$ esquemas. Este resultado é conhecido pelo nome de <u>*paralelismo implícito*</u>.

2.3. Aplicações dos Algoritmos Genéticos

Os AG podem ser aplicados em diversas áreas, tais como em problemas de optimização, a simulação de interacções ecológicas, o estudo da genética das populações ou ainda o sistema imunitário.

O trabalho original de John Holland consistiu não só no desenvolvimento de sistemas computacionais adaptativos para a resolução de problemas, mas também na clarificação de alguns aspectos da própria biologia, nomeadamente o mecanismo da evolução, através da utilização de modelos computacionais. A utilização destes modelos computacionais no estudo do mecanismo da evolução é relativamente recente e, não tem

ainda grande receptividade pela comunidade de investigadores da biologia evolucionária [Mitchell et al. 1994]. Os métodos tradicionais utilizados por estes cientistas para o estudo do processo evolutivo nos seres vivos passam pela análise de fósseis para determinar como se processou a evolução ao longo das eras geológicas ou pelo estudo dos sistemas biológicos nos seus *habitats* naturais com vista a perceber os mecanismos implícitos na evolução dos seres vivos. Este estudo inclui por um lado o papel dos mecanismos genéticos e, por outro lado, a função de várias características físicas e comportamentais dos organismos que influenciam a evolução das adaptações operadas. Podem ainda realizar-se experiências laboratoriais nas quais a evolução ao longo de muitas gerações, em populações de organismos relativamente simples, é estudada e controlada. Pode-se também efectuar o estudo da evolução ao nível molecular através da análise do ADN e ARN, nomeadamente as alterações sofridas por estas moléculas ao longo do tempo quando sujeitas a determinados factores. Este estudo inclui também a determinação dos processos evolutivos de espécies relacionadas comparando o seu ADN de forma a reconstruir filogenias, isto é, a história evolutiva de famílias de espécies relacionadas. Outra alternativa consiste no desenvolvimento de modelos matemáticos da evolução na forma de equações que possam ser resolvidas ou aproximadas [Mitchell 1996].

Como é facilmente perceptível estes métodos tradicionalmente utilizados pelos biólogos apresentam algumas limitações inerentes. Por exemplo, os fósseis analisados são normalmente muito incompletos e muitas vezes difíceis de interpretar. Torna-se difícil, se não impossível, realizar experiências controladas em sistemas biológicos naturais, sendo as escalas de tempo evolucionário muito longas para que os cientistas possam observar directamente as mudanças operadas nos referidos sistemas. O nível molecular é muitas vezes ambíguo. Por exemplo, existem ainda muitas incertezas sobre a informação codificada em certas porções da molécula de ADN. Finalmente, para que os modelos matemáticos de evolução possam ser resolvidos têm que ser bastante simplificados, fazendo com que muitas vezes não traduzam dados coerentes sobre a evolução real.

2.3.1. Modelos Científicos

Os computadores permitem uma nova abordagem para o estudo do processo evolutivo dos seres vivos: a simulação. Um programa de computador pode simular a evolução de uma população de indivíduos ao longo de milhões de gerações num tempo muito curto. Os resultados obtidos nas simulações podem ser usados para testar certas teorias sobre questões ainda em aberto relativamente ao processo da evolução. Uma das grandes vantagens dos programas de computadores é que permitem realizar experiências que podem ser controladas e repetidas, permitindo ainda a alteração de parâmetros para verificação das alterações produzidas.

Apesar do grande número de vantagens das simulações através do uso do computador, estas apresentam também algumas limitações como modelos dos fenómenos da vida real. Frequentemente elas simplificam drasticamente a realidade de forma a poderem ser computacionalmente tratáveis e produzir resultados perceptíveis. Mesmo com modelos muito simples, não é claro se os resultados podem ser aplicados ao modelo real correspondente. Por outro lado, modelos mais realistas implicam longos tempos de simulação e produzem enormes quantidades de informação que muitas vezes é difícil de interpretar.

Os AG são um dos métodos mais usados nas simulações microanalíticas da evolução. A sua utilização nesta área está em vasto crescimento como resultado do crescente interesse de investigadores em construir modelos computacionais dos processos biológicos. Assim, os AG são usados de várias formas em muitos problemas científicos dos quais se destacam os seguintes:

- **Modelos do sistema imunitário:** modelação de vários aspectos do sistema imunitário natural, incluindo mutações somáticas durante a vida de um indivíduo e na descoberta de famílias multigene durante o tempo evolucionário.

  Diversos autores têm realizado trabalhos nesta área. Destacamos os seguintes: [Dasgupta et al.1996], [Forrest et al. 1991], [Forrest et al. 1992], [Forrest et al. 1997], [Hightower et al. 1995], [Hightower et al. 1996], [Perelson et al. 1996], [Smith et al. 1992] e [Somayaji et al. 1997].

- **Modelos de sistemas ecológicos:** modelação de fenómenos ecológicos como por exemplo, a co-evolução, hospedeiro-parasita e o fluxo de recursos num ecossistema. Diversos autores enriqueceram esta área de estudo. Por exemplo, [Bedau et al. 1992], [Collins et al. 1991b], [Collins et al. 1992], [D'Haeseleer et al. 1996], [Forrest et al. 1994], [Hraber et al. 1996], [Hillis 1991], [Holland 1993], [Jefferson et al. 1991], [Jones et al. 1993], [Lindgren 1991], [Lindgren et al. 1994], [Lindgren et al. 1995], [Packard 1988], [Ray 1991a], [Ray 1991b], [Taylor et al. 1995] e [Wheeler et al. 1995].

- **Modelos de genética das populações:** questões relacionadas com a genética de populações, tais como "em que condições um gene será evolutivamente viável?" e a formação de espécies[27]. Destacamos os seguintes trabalhos: [Beasley et al. 1993c], [Bergman et al. 1992], [Deb et al. 1989], [Eshelman et al. 1993], [Fogel et al. 1990], [Menczer et al. 1992] , [Petrowski 1997], [Schaffer et al. 1991] e [Spears 1994], .

- **Modelos de sistemas sociais:** estudo de aspectos evolucionários de sistemas sociais tais como a evolução da cooperação, a evolução da comunicação ou o comportamento das formigas no seguimento de uma trilha. Alguns trabalhos desenvolvidos no estudo de modelos sociais são [Axelrod 1987], [Collins et al. 1991a], [Koza 1993], [MacLennan 1991], [MacLennan et al. 1994], [Miller 1989] e [Werner et al. 1991].

- **Modelos de selecção sexual:** análise de aspectos relacionados com a evolução sexual, nomeadamente a forma como a fêmea escolhe o seu parceiro. Para mais detalhe citamos alguns trabalhos desenvolvidos nesta área: [Collins et al. 1992], [Menczer et al. 1992], [Miller 1994], [Miller et al. 1995], [Miller 1996], [Todd et al. 1993], [Todd 1996], [Todd et al. 1997], [Webb et al. 1996].

[27] do inglês, *speciation*

- **Interacções entre evolução e aprendizagem:** estudo da forma como a aprendizagem individual e a evolução das espécies podem estar inter-relacionadas. Este aspecto, diferente da ideia de Lamarck que defendia que os caracteres adquiridos pelos indivíduos de uma população podiam ser transmitidos geneticamente, é conhecido por ***efeito de Baldwin*** [Simpson 1953], [Maynard-Smith 1987]. Este efeito traduz-se na evolução de uma espécie pela capacidade que os seus indivíduos têm em aprender certos comportamentos vantajosos. Assim, estes indivíduos têm mais possibilidades de sobreviver e de se reproduzir e, indirectamente, a sua descendência possuirá as mesmas características que os seus antepassados "aprenderam".

  Esta característica presente na natureza foi adaptada aos sistemas computacionais. O primeiro trabalho tentando explicar que consequências poderiam surgir quando se utilizasse a capacidade de aprendizagem é de [Hinton et al. 1987].

  Depois deste trabalho muitos outros têm surgido na literatura científica, por exemplo, [Ackley et al. 1991], [Ackley et al. 1993], [Hightower et al. 1995], [Littman 1996], [Mayley 1996], [Mayley 1997], [Nolfi et al. 1994], [Parisi et al. 1992], [Parisi et al. 1996], [Pereira et al. 1997], [Perelson et al. 1996] e [Todd et al. 1993], [Menczer et al. 1998] e [Pereira et al. 1999].

Este conjunto de tópicos reflecte uma boa parte das questões que podem ser abordados através de programas de computadores e nas quais se podem utilizar AG para a sua modelização e resolução.

Optámos por apresentar referências de trabalhos desenvolvidos nas diferentes áreas de aplicação específica visto que, a descrição pormenorizada de todos eles tornaria, este texto demasiado longo.

De seguida descreveremos alguns trabalhos desenvolvidos noutra área de aplicação dos AG: a resolução de problemas.

2.3.2. Resolução de Problemas

As aplicações dos AG na resolução de problemas incluem-se em muitas áreas, das quais destacamos as seguintes: planeamento, *design*, simulação e classificação, controlo, classificação e optimização de funções. Uma síntese mais completa de muitos trabalhos desenvolvidos nestas áreas de aplicação pode ser encontrado em [Beasley 1997].

Planeamento

Um dos problemas de optimização combinatória mais conhecido é, provavelmente, o problema do caixeiro viajante[28]. Neste problema, um caixeiro viajante tem que visitar um número de cidades uma e uma só vez e, depois disso, regressar a casa. Pretende-se descobrir qual a ordem pela qual o caixeiro deve visitar as diferentes cidades, de forma a minimizar a distância total percorrida. Um dos objectivos na resolução deste problema consiste em optimizar o compromisso entre a velocidade do algoritmo e a precisão da solução. Na resolução deste problema destacam-se os trabalhos de [Goldberg et al. 1985], [Fogel 1988], [Mühlenbein 1989], [Whitley et al. 1989] e [Homaifar et al. 1993].

Outro problema de planeamento semelhante ao anterior, mas generalizado a mais do que um viajante é o problema da rota de veículos[29]. Neste problema existe uma frota de veículos responsáveis pela entrega de uma encomenda a um determinado número de clientes. Pretende-se determinar qual a rota a seguir por cada veículo de forma a minimizar o custo. Na resolução deste problema através de AG destacam-se os trabalhos de [Blanton et al. 1993] e [Thangia et al. 1993].

Outras aplicações na área de planeamento enquadram-se em problemas de transporte[30], de planeamento de trajectórias para robots, escalonamento[31], e de empacotamento[32] .

No problema de transporte, um artigo tem que ser distribuído por um conjunto de clientes através de vários armazéns. Cada cliente pode receber artigos de um ou mais

[28] do inglês, *Traveling Salesman Problem*
[29] do inglês, *Vehicle Routing Problem*
[30] do inglês, *Transportation problem*
[31] do inglês, *Scheduling*
[32] do inglês, *Packing*

armazéns. O objectivo do problema consiste em encontrar a solução de custo mínimo. Este problema foi abordado com estratégias evolucionárias por [Michalewicz 1993] e [Michalewicz 1999].

No planeamento de trajectórias para robots pretende-se descobrir qual a trajectória que pode conduzir um robot de uma posição inicial a uma posição final de forma segura, sem colisões. Neste tipo de problema, um robot pode ser visto ou como um braço no qual se pretendem determinar os movimentos das suas junções ou como um robot que se movimenta num ambiente rodeado de obstáculos. Como trabalho nesta área destacamos os de [Davidor 1991], [McDonnell et al. 1992], [Page et al. 1992] e [Neves et al. 1999].

Os problemas de escalonamento envolvem o planeamento de um conjunto de actividades num determinado período de tempo, em que as actividades a realizar requerem recursos limitados. Além disso, as actividades têm que obedecer a um conjunto de restrições, pretendendo-se maximizar um ou mais objectivos. Dentro desta categoria de problemas existem algumas variantes, como por exemplo, escalonamento de tarefas[33] ou planeamento de horários. Destacam-se os seguintes trabalhos: [Davis 1985], [Syswerda 1991b], [Fang et al. 1993], [Easton et al. 1993], [Kidwell 1993], [Altman et al. 1993] e [Fogel et al. 1996].

Um dos problemas típicos de empacotamento é o problema da mochila[34]. Consideremos uma mochila de determinada capacidade e um conjunto de artigos, cada um deles com um determinado peso e valor. Pretende-se saber que artigos escolher, de tal modo que estes caibam na mochila e maximizem o valor total. Quando os valores dos pesos e do custo são inteiros o problema é designado do tipo **0/1**, quando são reais é designado **fraccionário**. Existem outros tipos de problemas de empacotamento, como por exemplo, a alocação de canais de comunicação a clientes com diferentes taxas de facturação, a forma de cortar porções de uma material de forma a minimizar os desperdícios ou ainda a maneira de posicionar circuitos integrados de forma a minimizar a área total que estes ocupam. Têm sido utilizadas algumas abordagens evolucionárias na resolução deste tipo de problemas, das quais se destacam: [Smith 1985], [Cohoon et al. 1987], [Chan et al. 1991], [Fujita et al. 1993] e [Juliff 1993].

[33] do inglês, *Job-shop scheduling*

Design

As abordagens genéticas têm sido utilizadas no *design* de circuitos eléctricos e digitais [Fogel 1991], [Fonseca et al. 1993], [Schaffer et al. 1993], de sistemas de processamento de sinal [San Martin et al. 1993], de circuitos integrados [Louis et al. 1991], [Rahmani et al. 1993], [Smith et al. 1993], da topologia e dos pesos de redes neuronais [Miller et al. 1989], [Harp et al. 1991], [Feldman 1993], [Gruau 1993], [Romaniuk 1993], [Zhang et al. 1993] e da topologia dos mapas de características[35] de Kohonen [Bramlette et al. 1991] [Watabe et al. 1993], [Furuya et al. 1993], [Powell et al. 1993].

Simulação e Identificação

Os problemas de simulação envolvem o projecto de um modelo para um sistema, de forma a determinar o seu comportamento em determinadas circunstâncias. As abordagens evolucionárias, nesta área de aplicação, foram testadas em problemas de química e biologia. Por exemplo, para determinar o equilíbrio químico de um sistema reactivo [Roosen et al. 1992], na determinação da estrutura tri-dimensional das proteínas, dadas uma sequência de ácidos nucleicos [Lucasius et al. 1991] ou na procura de configurações energéticas estáveis para as proteínas [Unger et al. 1993].

O problema de identificação é inverso ao da simulação: consiste em encontrar o *design* de um sistema dado o seu comportamento. Dos trabalhos com abordagens genéticas na área da identificação destacamos os seguintes: [Flockton et al. 1993], [Manela et al. 1993] e [Tanaka et al. 1993].

Controlo

Existem duas abordagens genéticas distintas na área de controlo: em tempo real[36] e em diferido[37]. A abordagem em diferido utiliza os AG (ou outras abordagens evolucionárias) para projectar o controlador que posteriormente controlará o sistema. A abordagem em tempo real utiliza os AG como uma parte activa no processo de controlo.

[34] do inglês, *Knapsack problem*
[35] do inglês, *Feature Maps*
[36] do inglês, *on-line*
[37] do inglês, *off-line*

Nesta área de aplicação salientam-se os trabalhos de [De Jong 1980], [Fonseca et al. 1993], [Abu Zitar et al. 1993], [Spencer 1993] e [Kim et al. 1995].

Classificação

Na área de classificação foram utilizadas abordagens evolucionárias na implementação de sistemas classificadores[38] [Holland 1985], [Booker 1985], [Holland 1987], [Fogarty 1994], em tácticas para jogos [Axelrod 1987], [Fogel 1993], em sistemas de classificação de plantas [Güvenir et al. 1993], na tomada de decisão financeira em problemas de determinação de valores crédito ou ainda o estudo de risco de mercados económicos [Oliver 1993].

Na área da biologia foram implementados algoritmos para a classificação das localizações de segmentos particulares de proteínas [Handley 1993] e na classificação de amostras de óleos [Punch et al. 1993].

Outro tipo de problemas de classificação inclui o processamento de imagem [Bala et al. 1993] e [Tackett 1993] e a recolha de informação [Jones et al. 1993] e [Yang et al. 1993].

Optimização de funções

No domínio de optimização de funções muito trabalho tem sido feito desde o aparecimento dos AG.

O trabalho de De Jong, ***Analysis of the Behavior of a Class of Genetic Adaptive Systems*** [De Jong 1975] constitui uma obra de referência para qualquer investigador em AG. Apesar do seu interesse em vastas áreas de aplicação como por exemplo, desenvolvimento de algoritmos e estruturas de dados, sistemas operativos, controlo adaptativo, De Jong utiliza o domínio da optimização de funções como objecto do seu estudo. Uma das razões que condicionaram esta escolha foi a importância de utilizar um ambiente de experimentação totalmente controlado e relativamente simples. Neste sentido, De Jong utiliza o AG, o ambiente e as medidas de desempenho na sua forma mais elementar. E foi devido a estas simplificações que De Jong conseguiu um trabalho notável que serve de base a qualquer estudo sobre AG.

[38] do inglês, *Classifier Systems*

O trabalho de De Jong utiliza um ambiente de teste constituído por cinco funções cujo valor mínimo se pretende encontrar. Este conjunto foi seleccionado de forma a abranger funções possuindo determinadas características, tais como:

- Continua/descontinua
- Convexa/não-convexa
- Unimodal/multimodal
- Quadrática/não-quadrática
- De baixa dimensão/de alta dimensão

Estas cinco funções (também utilizadas no nosso trabalho) são descritas com detalhe no capítulo 5.

Para medir a eficiência do AG De Jong utilizou duas medidas de desempenho: uma para medir o desempenho do algoritmo e outra para medir a convergência. Designou estas medidas por medida **em tempo real** e medida **em diferido** (convergência), respectivamente.

A medida **em tempo real** $x_e(s)$ é definida por:

$$x_e(s) = \frac{1}{T}\sum_{1}^{T} f_e\ (t)$$

Onde f_e é a função objectivo no instante **t**, sendo **T** o número de gerações executadas. Esta medida consiste na média de todas as avaliações da função objectivo até à iteração corrente.

A medida **em diferido** $x^*_e(s)$, traduz-se por:

$$x^*_e(s) = \frac{1}{T}\sum_{1}^{T} f_e^*(t)$$

Com $f_e^* = \max\{f_e(1), f_e(2), ..., f_e(t)\}$

Assim, a medida de desempenho em diferido traduz a média dos melhores indivíduos obtidos até um determinado instante **T**.

Com um conjunto de cinco funções de teste e duas medidas de eficiência, De Jong planeou o seu trabalho como um conjunto de variações ao que chamou AG simples. Começou com uma primeira variação, designada **R1** (plano reprodutivo 1) que utiliza o AG com:

- Selecção através da roleta
- *Crossover* simples (1 ponto de corte)
- Mutação simples

O plano **R1** inclui uma família de planos dependentes de quatro parâmetros:

n : tamanho da população

$\mathbf{p_c}$: probabilidade de *crossover*

$\mathbf{p_m}$: probabilidade de mutação

G: lacuna de gerações

O estudo de De Jong consistiu em observar os efeitos da variação destes quatro parâmetros e permitiu chegar a algumas conclusões:

- Ao variar a população entre 50, 100 e 200 indivíduos, De Jong concluiu que populações maiores permitem ao AG melhor desempenho em diferido devido à maior diversidade de esquemas presentes na população. No entanto, para a medida em tempo real, o aumento do tamanho da população conduz a uma maior inércia no início das simulações. Por outro lado, populações menores permitem obter mudanças mais rápidas e, portanto, um melhor desempenho em tempo real.
- A variação da probabilidade de mutação foi outro dos factores analisados. Aumentando a $\mathbf{p_m}$ de 0.001 até 0.1, a eficiência em diferido aproxima-se cada vez mais de uma procura aleatória. Por outro lado, o aumento da $\mathbf{p_m}$ degrada também o desempenho em tempo real.
- O estudo da variação da probabilidade de *crossover* $\mathbf{p_c}$ permitiu concluir que um valor igual a 0.6 é um compromisso razoável entre os desempenhos em diferido e em tempo real.

- Relativamente à lacuna de gerações, de Jong concluiu que, para o plano reprodutivo 1, utilizando G = 1 (não permitindo indivíduos da geração anterior na nova população), obtêm-se melhores resultados na maioria das aplicações de optimização.

O estudo de De Jong abrangeu um factor mais alargado de variações. Assim, o plano **R1** foi alterado em cinco vertentes diferentes:

R2: modelo elitista

R3: modelo com valor esperado;

R4: modelo elitista com valor esperado

R5: modelo com o factor multidão[39]

R6: modelo com *crossover* generalizado.

Com a análise destas cinco variações De Jong conseguiu um estudo exaustivo e minucioso que é ainda hoje uma obra de referência na área dos AG.

No entanto, a área da optimização de funções é também objecto de estudo de outros autores. Por um lado, há os que se preocupam em construir funções de teste de boa qualidade e outros que utilizam essas funções como medidas de desempenho de diversos estudos.

O nosso trabalho está na linha do de De Jong e procurou, nesta área de aplicação, por um lado, encontrar um conjunto de funções de teste já bastante estudadas, como é o caso das cinco funções de De Jong. Por outro lado, pretendeu abranger um vasto leque de características. A pesquisa na literatura ofereceu-nos um grande número de possibilidades das quais seleccionámos um conjunto de dezoito funções que, quer pelas suas características, quer por serem objecto de estudo de outros autores, constituem medidas de desempenho de referência.

[39] do inglês, *Crowding factor*

2.4. Alterações ao AG clássico de inspiração biológica

Desde o seu aparecimento que os AG têm a sua inspiração nos processos biológicos da evolução e da genética. A utilização de uma população de indivíduos que evoluem de geração em geração através da actuação dos mecanismos de selecção e dos operadores genéticos, não é mais do que uma inspiração dos processos que encontramos nos seres vivos. Apesar da grande simplificação que um AG tradicional faz desses processos, existem alguns trabalhos que, procurando inspiração na biologia, propõem algumas alterações ao seu funcionamento. Nesta secção apresentaremos alguns destes trabalhos.

2.4.1. Operador de inversão

O trabalho original de Holland [Holland 1992] utiliza além dos operadores de *crossover* e mutação o operador de inversão. Este operador, também de inspiração biológica (ver capítulo 3) é um operador de reordenamento utilizado em situações em que os alelos têm uma função específica, independente da posição em que se encontram no cromossoma.

Nos sistemas biológicos, a função de um gene é muitas vezes independente da posição que ocupa no cromossoma. Assim, invertendo parte do cromossoma a semântica da porção original é preservada.

Para usar o operador de inversão num AG, é necessário encontrar uma forma de interpretar qual a função de um determinado alelo, independentemente da posição que ocupa na cadeia binária. Holland propôs que a cada alelo se indexasse um valor, correspondendo à sua posição "real". Por exemplo, a cadeia **00010101** seria codificada da seguinte forma:

{(1,0) (2,0), (3,0) (4,1) (5,0) (6,1) (7,0) (8,1)}

Nesta representação o primeiro elemento de cada par corresponde à posição que o alelo (2º elemento) ocupa.

Assim, a mesma cadeia pode ser codificada por:

{(1,0) (2,0), (6,1) (5,0) (4,1) (3,0) (7,0) (8,1)}

O operador de inversão actua através da escolha de dois pontos do cromossoma e da inversão da ordem dos bits contidos entre esses dois pontos. No exemplo dado

anteriormente os pontos escolhidos foram os pontos 3 e 6. Esta reordenação não altera o valor do mérito do cromossoma, uma vez que esta é calculada a partir da ordem original. No entanto, a inversão altera as **ligações**: a ideia subjacente a este operador consiste em gerar cadeias ordenadas em que esquemas de melhor qualidade tenham mais possibilidades de sobreviver. Suponhamos que, no exemplo dado, a ordem do esquema 00**01** é muito importante. Após a utilização do operador de inversão, este esquema passa a ser 0010****. Este esquema é de alta qualidade e tem mais hipóteses de sobreviver ao efeito do operador de *crossover* do que a cadeia na sua ordem original.

Surge no entanto, um problema quando se combinam os operadores de *crossover* e inversão. Vejamos o seguinte exemplo:

{(1,0) (2,0) (6,1) | (5,0) (4,1) (3,0) (7,0), (8,1)}

cruza com

{(5,1) (2,0) (3,1) | (4,1) (1,1) (8,1) (6,0), (7,0)}

através de um ponto de corte situado no *locus* 3.

Os descendentes obtidos são:

{(1,0) (2,0) (6,1) (4,1) (1,1) (8,1) (6,0), (7,0)}

{(5,1) (2,0) (3,1) (5,0) (4,1) (3,0) (7,0), (8,1)}

Como se pode constatar o primeiro filho possui duas cópias para o alelo das posições 1 e 6 e nenhuma para os alelos das posições 3 e 5. Por outro lado, o segundo filho possui duas cópias dos bits 3 e 5 e nenhuma para os bits 1 e 6. Uma questão se coloca: como garantir que o operador de *crossover* produzirá sempre descendentes com um conjunto completo de bits para todas as posições?

Holland propôs duas soluções possíveis: a primeira consiste em permitir que o operador de *crossover* actue apenas entre cromossomas com a mesma permutação de posições. Esta solução, apesar de funcionar limita bastante a actuação deste operador [Mitchell 1996]. A segunda solução consiste em utilizar uma abordagem

"Dono/Escravo"[40]: escolhe-se um dos progenitores para ser o dono e reordena-se temporariamente o outro progenitor (escravo) pela mesma ordem do seu dono. Após a actuação do *crossover*, o segundo progenitor é reordenado para a configuração original.

A inversão foi utilizada nos trabalhos iniciais dos AG, mas nunca provou ser um operador capaz de introduzir vantagens no desempenho do algoritmo [Goldberg 1989]. Mais recentemente, outras formas deste operador foram usadas com sucesso para resolução de problemas de ordenação, como por exemplo a assimilação de fragmentos de ADN [Parsons et al. 1995].

Não existe, portanto, nenhuma conclusão definitiva sobre as vantagens da utilização deste operador, sendo necessários estudos empíricos e teóricos mais aprofundados.

2.4.2. Operadores de duplicação e deleção

Biologicamente, a duplicação e a deleção são aberrações cromossómicas em que há duplicação ou perda, respectivamente, de parte de um cromossoma.

Nas abordagens evolucionárias que utilizam representações com cromossomas de tamanho variável, os operadores de duplicação e deleção desempenham um papel primordial na obtenção de novos indivíduos. As representações em árvore, usadas principalmente em programação genética [Koza 1992], [Banzhaf et al. 1998] são aquelas onde estes operadores são mais utilizados.

[Haynes 1998] utiliza o operador de duplicação (em programação genética) para obter segmentos de código que irão substituir segmentos sem qualquer utilidade, removidos dos indivíduos da população.

2.4.3. Operador de Transdução

A transdução é um mecanismo biológico envolvendo interacções entre vírus e bactérias e que permite que um vírus, transportando DNA de uma bactéria consiga infectar uma população inteira. Através deste processo é possível propagar uma característica particular de uma célula para a totalidade da população. No capítulo 3 explicam-se com mais detalhes os vários tipos de transdução.

[40] do inglês, *Master/Slave*

Este operador foi utilizado numa abordagem genética por [Furuhashi et al. 1994]. Os autores apresentam uma abordagem para encontrar regras vagas[41] para que um *robot* móvel consiga evitar um conjunto de obstáculos. O operador biológico é implementado para que os melhores genes de um cromossoma sejam transferidos para outros, melhorando assim a qualidade dos indivíduos da população. O operador genético permite, portanto, seleccionar as melhores porções de um conjunto de cromossomas e transferi-las para outras zonas do cromossoma, eliminado os genes de qualidade inferior. Após este operador, os cromossomas são sujeitos ao operador tradicional de *crossover*. Com este novo mecanismo [Furuhashi et al. 1994] conseguiram superar resultados conseguidos por outras abordagens, provando ainda que a transdução favorece a criação de esquemas de elevada qualidade.
O mesmo operador foi utilizado em abordagens semelhantes por [Yoshikawa et al. 1997], [Nawa et al. 1997], [Nawa et al. 1998] e [Nawa et al. 1999].

2.4.4. Operador de Conjugação

A conjugação é um mecanismo biológico presente nas bactérias e que permite que estas, reproduzindo-se assexuadamente, possuam grande diversidade genética entre os indivíduos da população. O seu funcionamento caracteriza-se pela transferência unidireccional de material genético entre duas células. Este operador será descrito com mais detalhe no capítulo 3.

Existem alguns trabalhos em que este operador foi incorporado no AG com o objectivo de garantir a diversidade dos indivíduos da população.

Os primeiros trabalhos utilizando este operador genético são de [Smith 1996a, 1996b]. O autor apresenta três variantes do operador de conjugação e incorpora-os num AG para resolução de problemas **NP-completos**.

Na primeira variante, os indivíduos da população são dispostos numa matriz e apenas os cromossomas ocupando posições adjacentes podem trocar informação genética.

Outra versão proposta por [Smith 1996a] é o operador de conjugação simples que funciona do seguinte modo: após a selecção de dois indivíduos através de um dos

[41] do inglês, *Fuzzy Rules*

mecanismos de selecção tradicionais, escolhe-se aleatoriamente qual deles irá funcionar como dador e receptor. De seguida, definem-se os pontos de origem e de fim, de forma análoga aos pontos de corte do *crossover*. Para uma cadeia de comprimento ***l***, existem ***l-1*** pontos de corte. Num indivíduo, o ponto de origem mais à esquerda não pode ser ponto de fim e o ponto de fim mais à direita não pode ser ponto de origem.

Vejamos um exemplo:

Dois progenitores:

X_1: xxxxxx

Y_1: yyyyyy

Definição dos pontos de origem (o_i) e de fim (f_i):

o_1 o_2 o_3 o_4 o_5 o_6
X X X X X X
f_1 f_2 f_3 f_4 f_5 f_6

O mecanismo de conjugação simples transfere do dador para o receptor uma quantidade de material genético definida por um ponto de origem e um ponto de fim $\{o_i, f_i\}$, $i \leq j$. O material genético compreendido entre estes dois pontos é então injectado na célula receptora.

No exemplo anterior, consideremos que $\mathbf{X_1}$ é o dador e $\mathbf{Y_1}$ o receptor e que os pontos seleccionados aleatoriamente foram $\{o_2,f_4\}$:

X_1: x|xxx|xx (dador)

Y_1: y|yyy|yy (receptor)

O novo indivíduo será $\mathbf{Y_1}$': y**xxx**yy

A terceira variante estudada por [Smith 1996a, 1996b] foi proposta por Inman Harvey.

Em [Harvey 1996] é proposto um algoritmo simplificado, designado por **AG Microbial**[42], em que se mantêm os mecanismos de selecção e mutação e se introduz a

[42] do inglês, *Microbial GA*

conjugação como substituto do operador de *crossover*. Para a actuação do operador de conjugação, após selecção aleatória de dois progenitores, estes são sujeitos a um torneio. O vencedor do torneio funcionará como dador de material genético e o perdedor como receptor do mesmo. O autor salienta que este operador não gera novos indivíduos de raiz, apenas altera os já existentes na população, através de injecção de nova informação genética proveniente de melhores indivíduos. Relativamente aos resultados obtidos com este AG Microbial , o autor não é muito claro, referindo apenas que "foi testado com sucesso em problemas de teste standard".

[Odutayo 1996] realizou um estudo empírico comparando os mecanismos de conjugação simples e baseado em torneio com os operadores de *crossover* com um ponto de corte, dois pontos de corte e uniforme. Como medida de desempenho [Odutayo 1996] utilizou as cinco funções de teste de De Jong [De Jong 1975]. Foram realizadas experiências com uma população de 100 indivíduos, variando a probabilidade de mutação e o tipo de selecção. A autora utilizou taxas de mutação iguais a 0.0, 0.01 e 0.1 e mecanismos de selecção por roleta e por torneio. O AG foi simulado ao longo de 50 gerações em séries de dez experiências. Os resultados analisados foram as médias dos valores máximos, mínimos e médios obtidos nas dez simulações.

O resultado deste estudo permitiu concluir que, na maioria dos testes, não houve superioridade clara de nenhum dos mecanismos utilizados (*crossover* ou conjugação). No entanto, com taxas de mutação mais elevadas o operador de conjugação permitiu ao AG obter melhores resultados que os operadores de *crossover*. Também o mecanismo de selecção influenciou o desempenho do AG. Por exemplo, testes executados com selecção do tipo torneio e o operador de conjugação ultrapassaram os resultados obtidos com *crossover*.

[Odutayo 1996] conclui o seu trabalho salientando que o operador de conjugação é um potencial substituto dos operadores tradicionais de *crossover*, mas realça a necessidade de estudos mais aprofundados que permitam estabelecer em que condições este novo operador pode trazer vantagens relativamente aos mecanismos tradicionalmente utilizados.

2.4.5. Representações Poliplóides

Outros trabalhos de inspiração biológica, com resultados interessantes, são os que utilizam cromossomas poliplóides.

A maioria dos trabalhos realizados na área dos AG utilizam o tipo mais simples de genótipo presente na natureza : genótipo haplóide.

No entanto, nos sistemas naturais, os indivíduos mais desenvolvidos (ex. o Homem) possuem genótipos diplóides. Neste tipo de genótipo os cromossomas dos indivíduos estão agrupados em pares, designados por cromossomas homólogos. A informação referente a uma determinada característica pode ser encontrada nos dois genes correspondentes dos dois cromossomas homólogos. No entanto, aquela que se manifesta no fenótipo depende do tipo de dominância existente entre os dois genes homólogos. Se um indivíduo possui o mesmo alelo nos dois genes homólogos diz-se *homozigótico*. Se possui alelos diferentes diz-se *heterozigótico*. Nos indivíduos heterozigóticos existe um dos genes que é dominante relativamente ao outro. Neste caso a característica que se manifesta no fenótipo corresponde à do gene dominante. Uma característica recessiva só se manifesta num indivíduo heterozigótico.

Vejamos um exemplo simples para a cor do cabelo de um indivíduo (o exemplo é simplificado relativamente àquilo que pode acontecer na natureza). O gene responsável por esta característica pode ter dois valores possíveis: **castanho** e **louro**.

A característica **castanho** é dominante relativamente à característica louro. Os alelos designam-se por:

C (castanho; maiúscula indicando que e dominante) ;

l (louro, minúscula indicando que é recessivo).

Imaginemos agora uma situação envolvendo dois progenitores possuindo os seguintes pares de cromossomas homólogos (apenas se indicam os alelos para a característica em estudo):

Descendência possível:

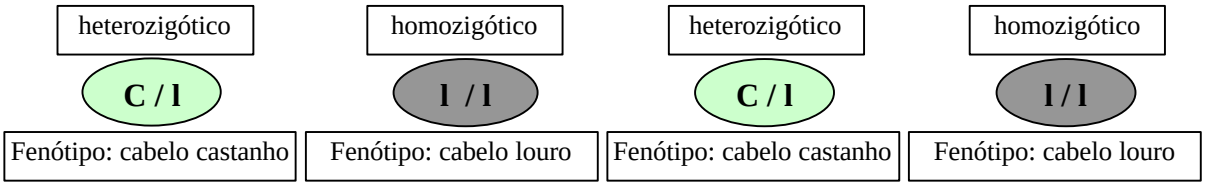


Figura 2. 7 - ***Efeito dos genes recessivos e dominantes na descendência de dois indivíduos***

A introdução da diploidia em sistemas computacionais conduz à existência de informação redundante, funcionando como uma memória que permite aos indivíduos poderem voltar a ter características que se perderam do seu fenótipo, mas preservadas de forma recessiva no genótipo.

Em situações em que as características do meio ambiente se possam alterar, este mecanismo permite aos indivíduos evoluírem rapidamente para fenótipos de alta adaptabilidade.

Em [Goldberg 1989] pode encontrar-se uma introdução a diversos métodos de diploidia. Salientamos três trabalhos nesta área:

[Ng et al. 1995] descrevem um novo esquema diplóide envolvendo um mapa de dominância. Os autores testaram esta nova representação com um conjunto de diferentes funções, apresentando apenas os resultados referentes ao problema da mochila [Cormen et al. 1990]. Para este problema, os resultados atingidos com a representação diplóide foram superiores aos conseguidos com um AG de indivíduos haplóides.

Em [Collingwood et al. 1996] é proposto um AG poliplóide em que cada indivíduo da população é representado por um conjunto de 3 cromossomas. Neste caso, existe um conjunto de vários candidatos para o mesmo gene. A forma como se define qual dos genes se encontra activo é dada por um operador de dominância. Este tipo de

representação garante algumas vantagens ao AG: a diversidade da população é incrementada e o facto de permitir que alguns genes se mantenham inactivos evita que eles desapareçam da população, voltando a surgir em situações em que se apresentem vantajosos. Além disso, garantindo uma maior diversidade da população evita a convergência do AG para máximos locais, garantindo que este encontre sempre a solução desejada.

Os autores testaram uma representação poliplóide (***3*** cromossomas) na resolução de dois problemas: o problema **indeciso(k)**[43]e o problema de **maximização de uns**[44]. O primeiro problema caracteriza-se por um espaço de procura com múltiplos subóptimos e apenas um valor óptimo, estando todos estes pontos à mesma distância de Hamming uns dos outros. O único máximo difere dos valores subóptimos de um valor muito pequeno, sendo esta diferença muito difícil de determinar para um AG. Esta dificuldade reside no facto de o AG ser facilmente "atraído " para os máximos locais, afastando a procura do verdadeiro máximo.

O problema *max* caracteriza-se pela função de avaliação calcular o número de uns da cadeia binária. Para que este encontre a solução óptima tem que gerar um indivíduo formado apenas com uns. A "dificuldade" que um AG pode encontrar na resolução deste problema reside na capacidade de "converter" os poucos zeros em uns de forma a concluir o processo iterativo.

Os resultados deste estudo permitem concluir que a multiploidia torna-se bastante útil em casos onde material genético importante (em situações futuras) pode ser perdido irreversivelmente. Nos dois problemas em estudo, verificou-se, no entanto, que uma representação multiplóide não trás nenhuma vantagem no problema de maximização de uns. Os autores justificam este resultado pelo facto do problema a resolver ser relativamente simples, facilmente resolvido por um AG tradicional com representação haplóide. Por outro lado, no problema **indeciso(k)** um AG com representação diplóide permitiu atingir uma rápida convergência com resultados muito superiores ao AG haplóide.

[Hadad et al. 1997] utilizam um AG com indivíduos diplóides e tetraplóides e realizam uma comparação deste AG com o tradicional utilizando cromossomas

[43] do inglês, *Indecisive(k)*

haplóides. A medida de desempenho utilizada foi o problema da mochila do tipo **0/1** [Cormen et al. 1990]. As conclusões deste trabalho vão de encontro às do trabalho anterior: maior diversidade na população e a capacidade de memorizar alelos. Estes alelos podem ter sido vantajosos num determinado momento e que, apesar de o deixarem de ser noutro estágio do processo, podem voltar a ser recuperados a qualquer momento. No problema utilizado, o ambiente está sujeito a alterações. Os autores compararam o desempenho do AG com indivíduos diplóides, tetraplóides e haplóides e concluíram que as abordagens poliplóides ultrapassaram grandemente a representação haplóide. Além disso, constataram que, em algumas situações, o AG com indivíduos tetraplóides atinge melhores resultados que o AG com representação diplóide.

[44] do inglês, *Max problem*

CAPÍTULO 3.
INSPIRAÇÕES DA BIOLOGIA

Sumário

A inspiração dos algoritmos genéticos provém das ideias da biologia, nomeadamente da genética e do princípio da selecção natural. No entanto, na natureza, os processos biológicos, desde os que envolvem o material genético até ao processo de evolução das espécies, são de uma enorme complexidade quando comparados com o que faz um AG. Os AG, inspirando-se nas ideias da genética e da evolução, provaram mesmo assim ser uma forte ferramenta em muitos domínios.

Mas porquê restringir o AG a apenas a inspirações de alguns mecanismos dos processos biológicos e não procurar na natureza novas ideias que se possam incorporar no AG e verificar se, tal como nos processos biológicos, os resultados são positivos?

Neste capítulo apresentaremos um vasto conjunto de mecanismos, presentes nos seres vivos, responsáveis por alterações estruturais do material genético. Estes mecanismos têm uma grande importância na evolução das espécies de determinados organismos, uma vez que a variação genética que permitem constitui um factor essencial para a actuação da selecção natural. Como iremos verificar, a variação genética não é mantida apenas pelos mecanismos de *crossover* e mutação. De facto, desde o ser mais simples ao mais complexo podemos encontrar um conjunto de processos responsáveis por essa variação. De todos estes mecanismos, destacaremos e detalharemos apenas aqueles que nos pareceram ter uma possível adaptação computacional.

Todos os mecanismos biológicos, que a seguir serão abordados, são de enorme complexidade. Neste capítulo não entraremos em detalhes particulares que se podem encontrar nos mais variados organismos vivos. Pelo contrário, tentaremos fazer uma descrição genérica do seu funcionamento geral. Descrições mais pormenorizadas podem ser encontradas em manuais de Biologia, por exemplo, [Bresch et al. 1972], [Gould et al. 1996] e [Russell 1998].

3.1. Biologia: algumas noções

A vida depende de um equilíbrio dinâmico entre a estabilidade e a mudança. Por um lado, a estabilidade é essencial à replicação do ácido desoxirribonucleico (ADN), por outro, a mudança é um dos factores imprescindíveis ao processo evolutivo das espécies. Um genoma estático conduziria à extinção de uma espécie que não se conseguiria adaptar e evoluir em caso de alterações climáticas, de recursos ou de competição.

É devido à diversidade dos seres vivos que uns se tornam mais aptos às condições actuais do ambiente e, portanto, têm mais probabilidade de viver e de se reproduzir e, consequentemente, garantir a evolução.

Normalmente, a diversidade dos seres vivos é vista como consequência de factores como o *crossover* do material genético no momento da formação dos gâmetas, a reprodução sexuada e o efeito de mutações. No entanto, na natureza podemos encontrar outros mecanismos responsáveis por profundas alterações cromossómicas e, consequentemente, pela diversidade dos seres vivos.

Um vasto leque de mecanismos responsáveis por estas alterações estruturais no genoma dos seres vivos será descrito na próxima secção.

No entanto, para que melhor se possa perceber o funcionamento desses processos, torna-se conveniente explicitar algumas noções e terminologia usadas na literatura da biologia.

3.1.1. Terminologia

Todos os organismos vivos são constituídos por células que, dentro da mesma espécie, contêm o mesmo número de cromossomas (cariótipo). Como já foi dito no capítulo anterior, os cromossomas são formados por cadeias de **ADN** (Ácido Desoxirribonucleico) constituídas por genes que codificam determinadas características dos indivíduos. O modelo do ADN proposto por Watson e Crick foi baseado nos dados radiográficos de Franklin e Wilkins e nos dados químicos de Charfagg [Bresch et al. 1972], [Watson 1968]. Estes dados permitiram concluir que uma molécula de ADN consiste em dois cordões, com polaridade inversa, que, enroscados, formam uma hélice. Existem quatro bases azotadas na constituição química dos ácidos nucleicos: **adenina**

(A), **timina (T)**, **guanina (G)** e **citosina (C)**. Estas bases emparelham duas a duas: a adenina com a timina e a citosina com a guanina [Russell 1998].

Nos sistemas biológicos podemos encontrar células **eucarióticas** e **procarióticas**, sendo a principal diferença entre as duas a existência de uma membrana nuclear bem definida nas primeiras e a não existência dessa membrana nas segundas. De uma forma geral as células eucarióticas são **diplóides** e as procarióticas **haplóides**. Uma célula diz-se diplóide se os seus cromossomas se encontram aos pares, haplóide caso contrário.

Nos organismos diplóides os cromossomas encontram-se agrupados em pares de **cromossomas homólogos**, cada um deles proveniente das células reprodutoras dos seus progenitores e contendo a mesma sequência de genes.

Nas primeiras etapas da divisão celular, quando os cromossomas se condensam e se tornam visíveis, cada um deles já se duplicou para produzir um par de gémeos unidos por um ponto designado **centrómero**. A cada um destes elementos chama-se **cromatídeo** (ver figura 3.1).

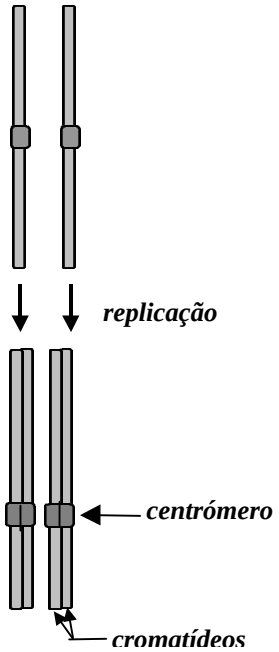


Figura 3. 1 – ***Par de cromossomas homólogos***

Em cada lugar (**locus**) de cada par de cromossomas homólogos encontra-se um par de genes responsáveis pela mesma característica que se designam por **genes alelos**. Cada alternativa que um gene pode ter é designada por **alelo**.

Uma vez que uma célula diplóide contém duas cópias de cada gene (uma em cada cromossoma homólogo), pode possuir duas cópias do mesmo alelo ou uma cópia de um alelo e outra de outro alelo (ver figura 3.2). Células com duas cópias do mesmo alelo dizem-se **homozigóticas**. As que possuem alelos de valores diferentes dizem-se **heterozigóticas**.

Os genes podem ser dominantes ou recessivos. São **dominantes** quando se manifestam no fenótipo independentemente da cópia do segundo cromossoma. São **recessivos** quando, na presença de um dominante não se expressam, apenas o conseguindo quando se encontram na presença de outro recessivo (numa célula homozigótica).

Por exemplo, imaginemos o gene responsável pela cor dos olhos. Os alelos possíveis para esse gene são **a** (azul; a minúscula indica que é um gene recessivo) e **C** (castanho; a maiúscula indica que é dominante). Podem ocorrer as seguintes combinações:

- **aa**: ser homozigótico com dois genes recessivos. Terá olhos azuis.
- **Ca** ou **aC**: ser heterozigótico com uma gene recessivo e outro dominante. Terá olhos castanhos, visto que o gene dominante se sobrepõe-se ao recessivo, manifestando-se no fenótipo.
- **CC**: ser homozigótico com dois gene dominantes. Terá olhos castanhos.

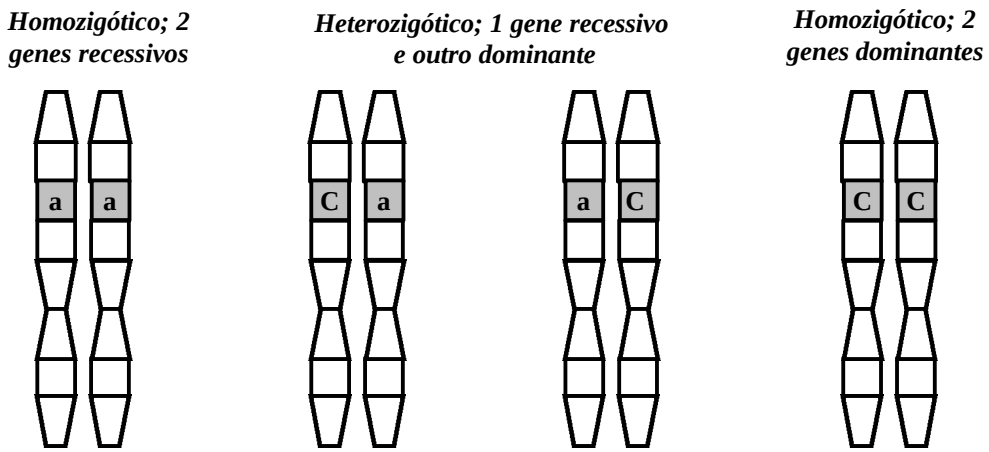


Figura 3. 2 – ***Células homozigóticas, heterozigóticas, com genes dominantes e recessivos***

A tabela 3.1 sintetiza as possibilidades anteriormente esquematizadas.

C – gene dominante para a cor castanha dos olhos		*a – gene recessivo para a cor azul dos olhos*
Genótipo	***Fenótipo***	***Caracterização***
CC	Olhos castanhos	Homozigótico dominante
Ca ou **aC**	Olhos castanhos	Heterozigótico
aa	Olhos azuis	Homozigótico recessivo

Tabela 3. 1 – ***Genes dominantes e recessivos para a cor dos olhos***

Os organismos diplóides utilizam a **reprodução sexuada** e os haplóides a **reprodução assexuada**.

Na **reprodução sexuada,** os organismos dividem as suas células diplóides (e os pares de cromossomas homólogos) formando células haplóides (**gâmetas** ou células reprodutoras). O processo de divisão celular para formação das células reprodutoras designa-se por **meiose**. A meiose é um processo bastante complexo, constituída por várias etapas, ocorrendo numa delas o fenómeno de ***crossover*** entre dois cromossomas homólogos. Este processo caracteriza-se pela troca de material genético entre dois cromossomas homólogos. Ver figura 3. 3

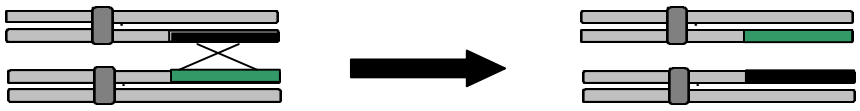


Figura 3. 3 - ***O Processo de Crossover***

Neste tipo de reprodução, estão envolvidos dois progenitores que contribuem cada um deles com uma célula reprodutora (gâmetas). Quando ocorre a fecundação de dois gâmetas, volta a formar-se uma célula diplóide (o **ovo** ou **zigoto**) que dará origem a um novo ser vivo (também diplóide).

As restantes células (com excepção das células reprodutoras) são designadas **células somáticas** e reproduzem-se através de um processo designado por **mitose**.

A **reprodução assexuada**, utilizada pelos organismos haplóides caracteriza-se fissão binária de uma célula-mãe que dará origem a duas células-filhas contendo o mesmo material genético da sua progenitora.

Em qualquer dos casos, a nova descendência gerada pode ser sujeita a mutações que provocam alterações no material genético e, consequentemente, no seu fenótipo.

Dentro das células procarióticas encontram-se as **bactérias** que, quer pelas suas características genéticas, quer pelos mecanismos de rearranjo do material genético que utilizam, serão abordadas com mais detalhe na próxima secção.

3.1.2. O Caso das Bactérias

As bactérias são células procarióticas constituídas por dois tipos de ADN:

- **ADN cromossómico** que regula a actividade da bactéria e essencial para o seu funcionamento;
- **plasmídeos**[45]: cadeia circular de ADN extracromossomático capaz de se replicar e de ser transferida para outra célula. Os plasmídeos possuem genes úteis para algumas actividades, mas são considerados não essenciais para a vida da célula a que pertencem.

As bactérias não têm sexo, reproduzem-se assexuadamente por divisão celular: uma célula dá origem a várias células iguais a si. No entanto, verifica-se que o material genético de diferentes bactérias apresenta grande variedade. Essa variedade é assegurada por vários mecanismos de recombinação que possibilitam obter novas combinações do material genético das células envolvidas.

Existem quatro formas de recombinação presentes nas bactérias:

- Transformação;
- Conjugação;
- Transdução;
- Transposição[46]

[45] do inglês, *plasmids*

Estes mecanismos permitem às bactérias, apesar de se reproduzirem assexuadamente, ter uma grande variação genética nas suas populações. Esta variação genética garante-lhes certas vantagens, como seja a resistência a alguns antibióticos.

Alguns destes mecanismos, por exemplo, a transformação foram adoptados pela engenharia genética para manipulação de células fornecendo-lhes características vantajosas em determinadas circunstâncias: manipular geneticamente vertas espécies, para que possuam resistência a pesticidas [Russell 1998].

3.1.3. Para além das Bactérias

Nos seres vivos em geral, além dos mecanismos focados para o caso particular das bactérias, existem outros que garantem a diversidade do seu material genético. Estes mecanismos podem ser agrupados em quatro categorias que não são disjuntas mas, pelo contrário, se sobrepõem e complementam. Essas categorias implicam fenómenos de:

- Inserção de genes;
- Duplicação de genes;
- Eliminação de genes;
- Movimentação de genes.

A cada um destes fenómenos estão associados vários mecanismos que produzem, de uma forma ou de outra, alterações no material genético, nos quais se incluem os mecanismos enumerados anteriormente. Além desses, existem outros, como por exemplo, o *crossover desigual*[47], a *quebra e fusão*[48], a *inversão*, a *deleção*, a *duplicação* e a *translocação*[49] [Bresch et al. 1972], [Russell 1998]. Estes mecanismos serão abordados com mais detalhe na próxima secção.

Na tabela 3.2 encontra-se o resumo dos fenómenos/mecanismos que podem ser encontrados nos sistemas biológicos em geral, incluindo o caso particular das bactérias [Gould et al. 1996].

[46] a transposição não é exclusiva das células procarióticas; em células eucarióticas também se manifesta, inclusive no ser humano
[47] do inglês, *unequal crossover*
[48] do inglês, *break and fusion*
[49] do inglês, *translocation*

Mecanismo	Fenómenos associados	Descrição
Transdução	Movimentação, inserção	Genes de uma bactéria, integrados acidentalmente em cápsulas virais, são introduzidos numa nova bactéria no momento em que o vírus a infecta.
Conjugação	Movimentação, inserção	Material genético de uma bactéria dadora é introduzido numa célula receptora produzindo bactérias com diferentes características.
Transformação	Inserção, duplicação	Genes de uma célula morta são importados do meio ambiente para o cromossoma de uma bactéria.
Transposição	Duplicação, eliminação, movimentação	Porções de ADN movem-se no genoma, abandonado a sua posição original e integrando-se noutra parte do ADN. Quando abandonam um ponto do ADN podem ou não deixar duplicações.
Deleção	Eliminação	Segmentos do cromossoma são eliminados
Duplicação	Duplicação	Segmentos do cromossoma são duplicados, podendo ocorrer diversas configurações.
Inversão	Movimentação, inversão	Parte do cromossoma abandona-o, roda 180º e reintegra-se na mesma posição.
Translocação	Duplicação, eliminação, movimentação	Consiste numa alteração na posição de certos segmentos do cromossoma e na sequência de genes que esses segmentos contêm.
Crossover	Movimentação	Troca de material genético entre dois cromossomas homólogos. A troca pode fazer-se num único ponto ou em vários.
Crossover desigual	Duplicação, eliminação, movimentação	No momento do *crossover* os cromossomas homólogos podem estar desalinhados. O *crossover* faz-se em pontos de corte desiguais. Certos gâmetas obterão cópias duplicadas de alguns genes.
Quebra e Fusão	Duplicação, eliminação, movimentação	Uma porção de um cromossoma parte-se acidentalmente fundindo-se com a extremidade de outro cromossoma.

Tabela 3.2 – ***Síntese dos mecanismos que provocam a alteração da estrutura do material genético presentes nos seres vivos***

Na próxima secção descreveremos com detalhe cada um destes mecanismos.

3.2. Diversidade Genética

A secção anterior deu-nos já uma ideia do grande número de mecanismos e dos fenómenos associados que podem provocar alterações na estrutura do material genético. Mesmo seres que se reproduzem assexuadamente (como é o caso das bactérias) e que, portanto, não vêm o seu material genético sujeito ao mecanismo de *crossover*, apresentam também uma grande diversidade genética.

Nesta secção explicaremos com detalhe o funcionamento de um grande conjunto de mecanismos que, além dos processos bem conhecidos de mutação e *crossover*, produzem variação genética nas populações.

Não será feita qualquer diferenciação no que se refere aos seres vivos ou ao tipo de reprodução que originam esses mecanismos. Essa referência já foi feita anteriormente. Nesta secção enumeraremos e descreveremos sequencialmente os mecanismos já sintetizados na tabela 3.2.

3.2.1. Transdução

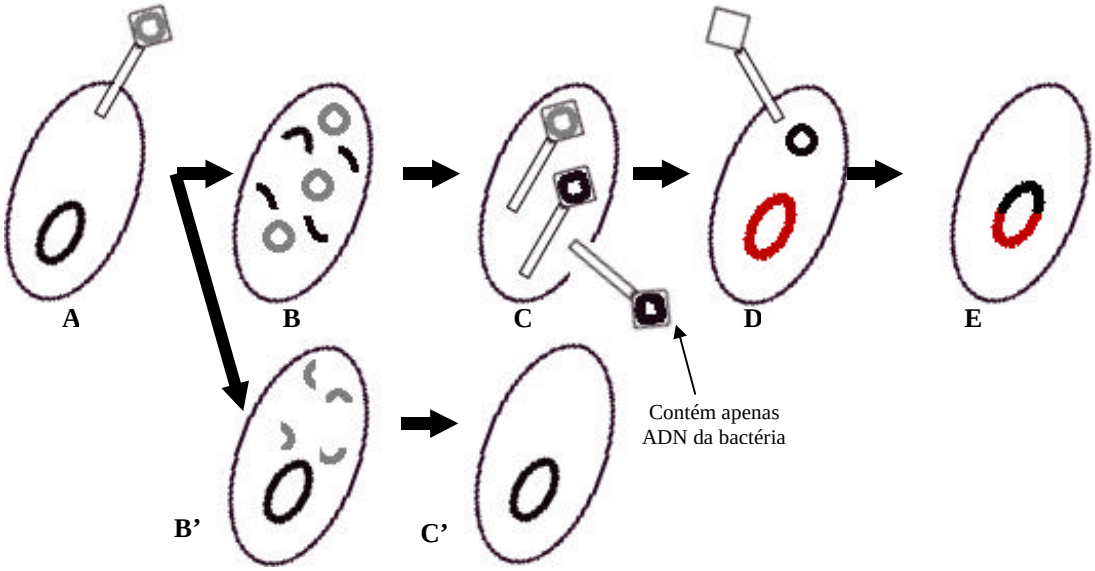
O mecanismo de transdução foi descoberto em 1952 por Zider e Lederberg [Bresch et al. 1972]. Este processo envolve vírus (ou fagos) que possuem capacidades de induzir as bactérias a produzir novos vírus. Os vírus microrganismos infecciosos que consistem num núcleo de ácido nucleico envolvido num revestimento proteico (cápside). Multiplicam-se parasitariamente em células de outros organismos. Fora das células hospedeiras não têm metabolismo próprio. Os vírus podem ser considerados como pequenos genomas móveis que exercem o seu efeito em células estranhas. Este efeito consiste, sobretudo, em produzirem novas partículas de vírus, o que prejudica e destrói a célula hospedeira. Os vírus são, portanto, "parasitas ao nível genético" [Bresch et al. 1972].

No mecanismo de transdução, quando um vírus "infecta" uma bactéria os novos vírus produzidos poderão possuir ADN da bactéria e quando infectam uma nova célula o ADN estranho pode incorporar-se no ADN da célula hospedeira.

Este fenómeno não ocorre sempre. Quando um vírus infecta uma bactéria, o seu ADN pode degradar-se não havendo formação de novas cápsulas virais, pelo que o mecanismo de transdução não se concretizará.

Existem dois mecanismos de transdução [Russell 1998]:

- *Transdução generalizada*: o vírus utiliza a bactéria para produzir novos vírus. O ADN da bactéria fragmenta-se ao ser invadida por um vírus. Estes fragmentos de ADN bacterial podem ser rodeados acidentalmente por cápsulas virais. Se este vírus infectarem outras bactérias, não lhe causarão qualquer dano visto que o ADN não possui capacidades para desfragmentar o material genético e produzir novos vírus. Em vez disso, o ADN estranho pode recombinar-se com zonas homólogas do ADN hospedeiro. Ver figura 3.4.



A - Um vírus infecta bactéria;

B - O ADN da bactéria parte-se. O ADN do vírus reproduz-se;
B' - O ADN viral degrada-se, não afectando o ADN da bactéria.

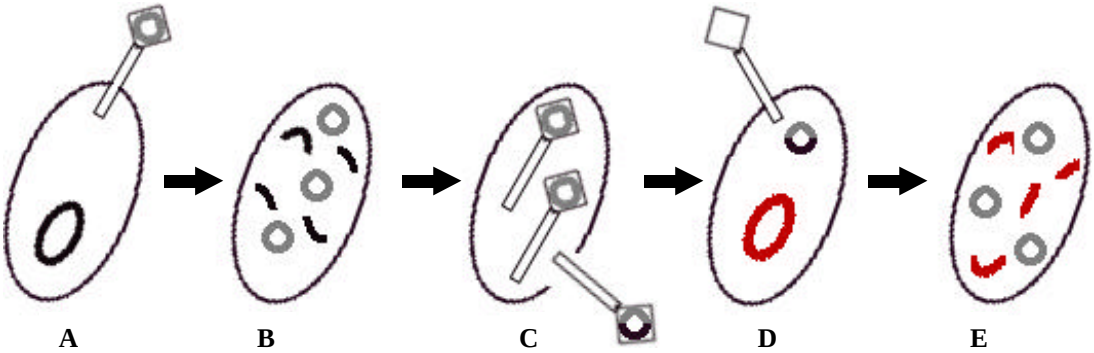
C - O ADN é rodeado por cápsulas virais. Novos vírus são formados, alguns deles contendo ADN da bactéria;
C' - A célula permanece inalterada;

D - ADN estranho entra numa nova célula.

E - ADN estranho integrou-se com o ADN da célula hospedeira. A bactéria não é lesada.

*Figura 3. 4 – **Mecanismo de transdução generalizada***

- *Transdução especializada*: o vírus infecta a bactéria e liberta o seu material genético. Este material genético pode misturar-se com o ADN bacterial formando-se uma nova cadeia com ADN do vírus e ADN da bactéria. Este material genético é encapsulado por vírus e posteriormente pode infectar uma nova bactéria que, ao receber o material genético híbrido será induzida a produzir novos vírus, acabando por morrer. Ver figura 3.5.



A - Um vírus infecta bactéria;

B - O ADN da bactéria fragmenta-se. O ADN do vírus reproduz-se;

C - O ADN é rodeado por cápsulas virais. Novos vírus são formados, alguns deles contendo ADN misto: ADN bactéria + ADN vírus

D - ADN híbrido entra numa nova célula.

E - ADN viral induz à produção de novos vírus. A célula acabará por morrer.

*Figura 3. 5 – **Mecanismo de transdução especializada***

3.2.2. Conjugação

O mecanismo de conjugação foi descoberto por Lederberg e Tatum e lavado ao seu estado actual de conhecimentos através de uma série de importantes trabalhos de Lederberg, Hayes, Jacob e Wollman [Bresch et al. 1972].

Neste mecanismo estão envolvidas duas células, uma que funciona como dadora e outra como receptora. Quando as duas células se aproximam, estabelece-se uma ponte que liga o citoplasma das duas células. Será através desta ponte e após replicação do plasmídeo da célula dadora, que se fará a sua transferência para a célula receptora.

Nas bactérias, apesar de não existir reprodução sexuada, as células possuem um elemento sexual, o plasmídeo **F** que determina o seu 'sexo' e que atribui características de fertilidade às células que o possuem. Assim, células que possuam o elemento **F** são designadas células $\mathbf{F^+}$ e funcionam como "machos". As células que não possuem o factor **F** são células $\mathbf{F^-}$ e funcionam como "fêmeas" [Russell 1998].

A conjugação estabelece-se entre células $\mathbf{F^+}$ (dadora) e $\mathbf{F^-}$ (receptora) podendo o resultado ser uma célula $\mathbf{F^+}$. O plasmídeo da célula macho replica-se e quando se estabelece a ponte entre duas células diferentes, é transferido para o citoplasma da célula $\mathbf{F^-}$. Como resultado esta célula ao receber o elemento **F** passará ser uma célula $\mathbf{F^+}$.

A partir do momento em que uma célula recebe o elemento **F** pode tornar-se dadora e transferir o seu material genético para outra célula.

A conjugação pode manifestar-se também de outro modo: quando o elemento **F** não fica isolado no citoplasma, mas se combina com o ADN cromossomático da célula, o resultado não é uma bactéria $\mathbf{F^+}$ mas sim uma célula **Hfr**[50]. Ver figura 3.6.

Esta célula continua a possuir características de dadora, mas em vez do elemento **F** ela transferirá o ADN cromossomático. Esta manifestação **Hfr** garante grande variedade genética numa população de bactérias [Gould et al. 1996].

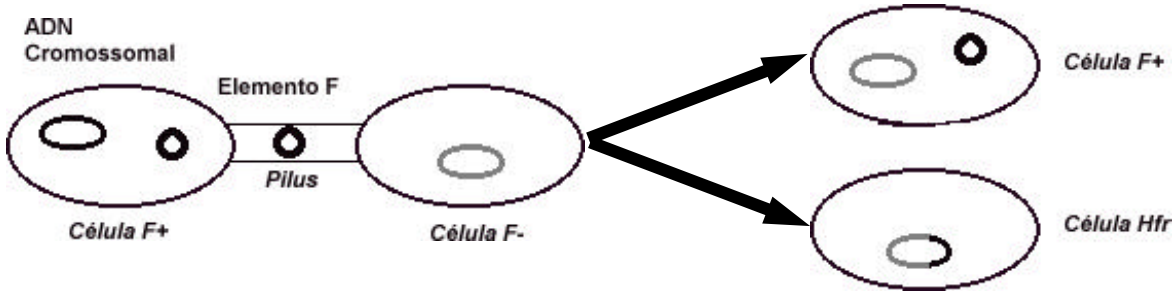


Figura 3. 6 - ***Mecanismo de conjugação entre duas células***

Uma célula ***Hfr*** poderá estar sujeita a duas ocorrências (ver figura 3.7):

- O factor **F** liberta-se do ADN cromossomático e a célula torna-se numa célula do tipo $\mathbf{F^+}$;
- O factor **F** liberta-se do ADN cromossomático mas arrasta consigo parte desse ADN. Neste caso estamos perante o factor **F'**. Quando o

[50] abreviatura de *High Frequency of Recombination*

factor **F'** é transferido por conjugação o processo designa-se por *sexdução*[51].

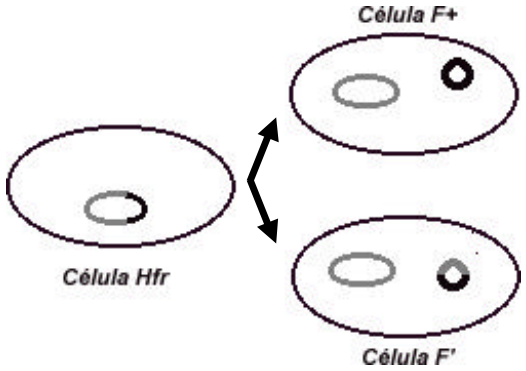


Figura 3. 7 – ***Mecanismo de conjugação: formação de células F+ e F' a partir de células Hfr***

Assim, se tudo parecia indicar que o processo de conjugação conduziria a uma mudança para o "sexo masculino" de uma população de bactérias, tal não se verifica. Este aspecto deve-se ao facto de o processo de conjugação não conduzir sempre à formação de células $\mathbf{F^+}$, existindo, pelo contrário, uma grande taxa de células **Hfr**. Experiências realizadas com a junção de células **Hfr** com células $\mathbf{F^-}$ permitiram concluir que estas permanecem recombinantes $\mathbf{F^-}$, não havendo nenhuma célula que tenha tomado o carácter **Hfr**. Isto acontece porque, como as células **Hfr** possuem o elemento **F** integrado no seu ADN cromossomático, quando ocorre a conjugação de uma célula **Hfr** com uma célula $\mathbf{F^-}$ a transferência do material genético nunca é levada até ao fim, sendo sempre interrompido antes que o elemento **F** seja transferido. Logo a célula receptora permanece $\mathbf{F^-}$ [Russell 1998], [Bresch et al. 1972].

Apenas algumas bactérias possuem resistência a vírus e/ou antibióticos. O elemento **F** é um factor de resistência. Através do processo de conjugação, quando o elemento **F** é transmitido isoladamente ou não a uma célula receptora, pode transferir para essa célula os genes responsáveis pela resistência aos fagos e aos antibióticos. Assim, através do processo de conjugação uma colónia de bactérias pode evoluir de tal forma que se torna imune aos seus mais directos inimigos - os vírus - e/ou aos antibióticos desenvolvidos para as destruir.

[51] do inglês, *sexduction*

Estes factores de resistência são de grande interesse médico, visto que eles transportam genes - como o factor F - que por formação do pilus possibilitam a transmissão dos factores a outras células. Além destes, possuem genes que dão à bactéria portadora uma resistência contra alguns fármacos, por exemplo, sulfamidas, penincilina, tetraciclina, neomicina, ampicilina, entre outros [Bresch et al. 1972].

3.2.3. Transformação

Algumas bactérias possuem a capacidade de 'apanhar' pequenas porções de ADN disperso no meio que as rodeia. Este ADN, por exemplo, pertencente a bactéria mortas, é importado para uma célula e pode recombinar-se com uma zona semelhante do genoma hospedeiro. Ver figura 3.8.

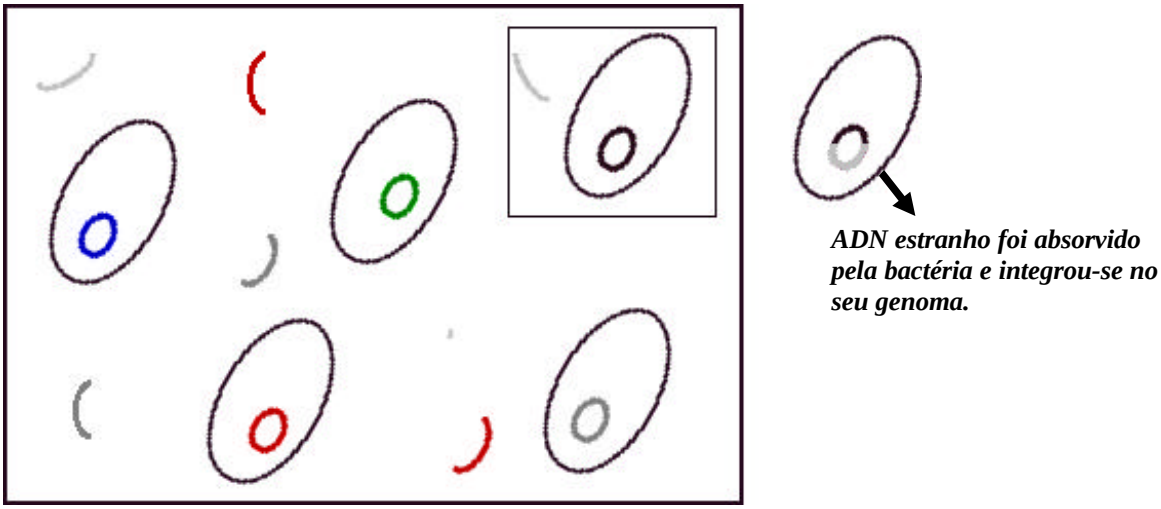


Figura 3. 8 – ***Mecanismo de Transformação***

Nem todas as bactérias possuem esta capacidade de absorver ADN estranho e integrá-lo no seu material genético. As células capazes de transformação designam-se por **competentes** [Russell 1998].

Algumas técnicas de engenharia genética consistem em tornar certas células normais em células competentes para que possam ter capacidades de transformação de ADN estranho no seu próprio material genético.

3.2.4. Deleção

Uma **deleção genética** consiste numa mutação cromossómica envolvendo a perda de um segmento de um cromossoma (figura 3.9.). O segmento eliminado pode estar localizado em qualquer ponto do cromossoma. Esta eliminação começa nos pontos onde ocorrem quebras no cromossoma devidas a certos factores. Por exemplo, a quebra de um cromossoma pode ser induzida por agentes, como o calor, radiação, vírus, produtos químicos, transposões, ou ainda erros de *crossover* [Russell 1998].

Figura 3. 9– ***Mecanismo de deleção de um segmento de um cromossoma***

As consequências de uma deleção dependem do gene ou do conjunto de genes que foram removidos. Nos organismos diplóides os efeitos podem ser minimizados pela presença, no cromossoma homólogo, de uma cópia dos genes eliminados. No entanto, se o homólogo contiver uma cópia recessiva dos genes e estes estiverem danificados, as consequências poderão ser muito graves.

Um grande número de doenças no ser humano são causadas pela deleção de segmentos cromossómicos. Em muitos casos surgem anormalidades que podem ser letais, uma vez que o número de cópias genes é um factor fundamental para o desenvolvimento normal e o bom funcionamento do organismo. Exemplos de doenças causadas pelas eliminações de segmentos são o síndroma *cri-du-chat* (resultante da perda de um pequeno segmento do cromossoma 5) e o síndroma *Prader-Willi* (resultante da perda de uma grande porção do cromossoma 15) [Russell 1998].

3.2.5. Duplicação

O fenómeno da **duplicação genética** consiste numa mutação cromossómica envolvendo a replicação de um segmento de um cromossoma (figura 3.10). O tamanho

do segmento duplicado pode variar consideravelmente e podem ocorrer em diferentes localizações do genoma ou numa configuração *tandem* (isto é, adjacentes uns aos outros).

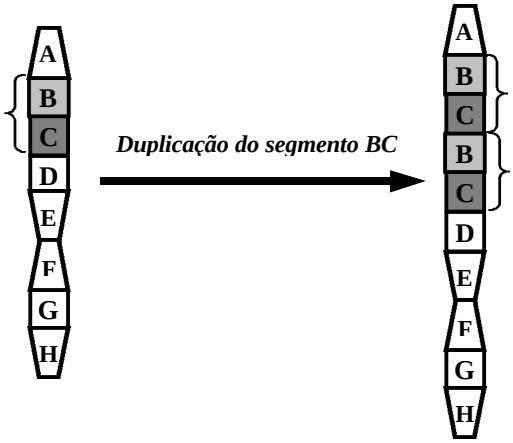


Figura 3. 10 - ***Mecanismo de duplicação de um segmento de um cromossoma***

Quando a ordem dos genes num segmento duplicado é contrária à ordem original estamos presentes uma **duplicação de *tandem* inversa**; quando os segmentos duplicados estão dispostos numa configuração tandem na extremidade do cromossoma a duplicação designa-se **duplicação *tandem* terminal** [Russell 1998]. A figura 3.11. ilustra estes diferentes fenómenos.

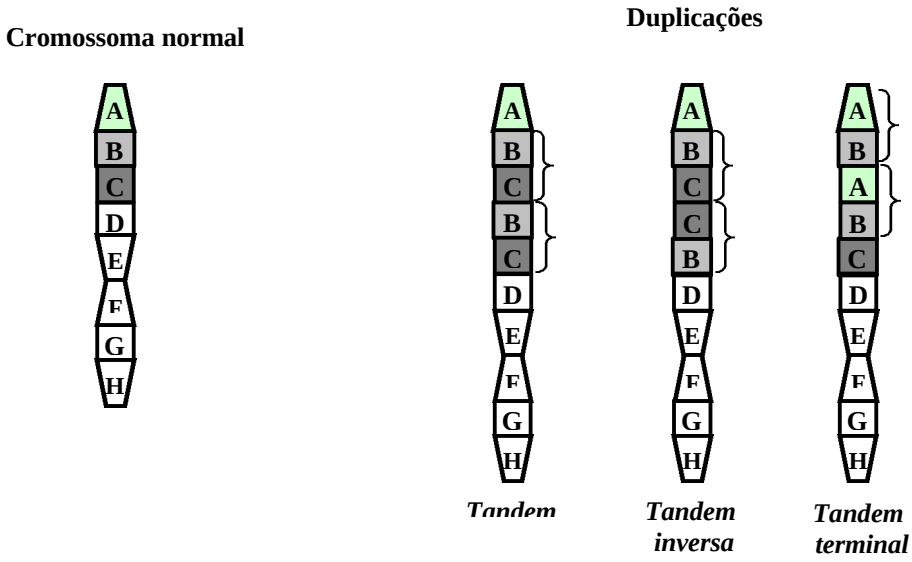


Figura 3. 11 – ***Formas de duplicação de segmentos: tandem, tandem inversa e tandem terminal***

A duplicação de segmentos desempenhou um papel importante na evolução de famílias multigénicas. Por exemplo, cada molécula de hemoglobina contem duas cópias de duas diferentes sub-unidades, os polipétideos α-globina e β-globina. Em diferentes etapas do desenvolvimento embrionário, um indivíduo possui moléculas diferentes de hemoglobina assembladas a partir dos diferentes polipetpideos α-globina e β-globina. Os genes para cada um dos tipos diferentes de α-globina são agrupados num mesmo cromossoma, enquanto que os genes para o outro polipeptideo são agrupados noutro cromossoma. A sequência de genes na família de polipeptideo α-globina é bastante semelhante (tal como acontece com o poplipeptideo β-globina). Pensa-se que este aspecto se deve à evolução de cada uma das famílias de polipeptideos, a partir de um gene ancestral que sofreu duplicações e subsequente divergência [Russell 1998], [Gould et al. 1996].

Um fenómeno de duplicação com características particulares designa-se por ***crossover*–desigual** e é descrito na próxima secção.

3.2.6. Crossover desigual

Certas duplicações ou eliminações num cromossoma surgem devido a um processo conhecido por *crossover* desigual. Este fenómeno acontece durante o processo de *crossover* do material genético que ocorre durante a meiose e deve-se ao facto de os cromossomas homólogos no momento do *crossover* se encontrarem desalinhados.

O desemparelhamento dos pares de genes em cada cromossoma homólogo leva a que, após o *crossover*, os cromossomas resultantes possuam duplicação e deleção de um ou mais genes. A figura 3.12. ilustra este mecanismo.

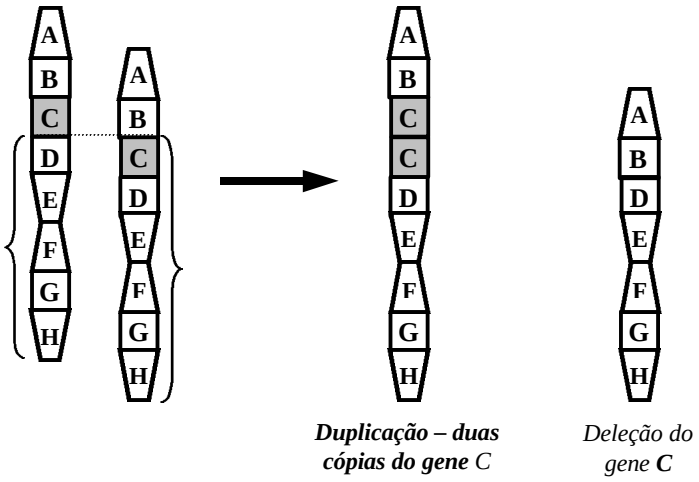


Figura 3. 12 – ***Mecanismo de crossover desigual***

Quando os cromossomas mutantes são novamente sujeitos ao processo de *crossover* desigual, surgem cópias triplicadas do mesmo gene num dos cromossomas produzidos, enquanto que o outro cromossoma regressa à sua configuração normal, visto que uma cópia do cromossoma duplicado é eliminada [Russell 1998]. A figura 3.13 mostra este processo.

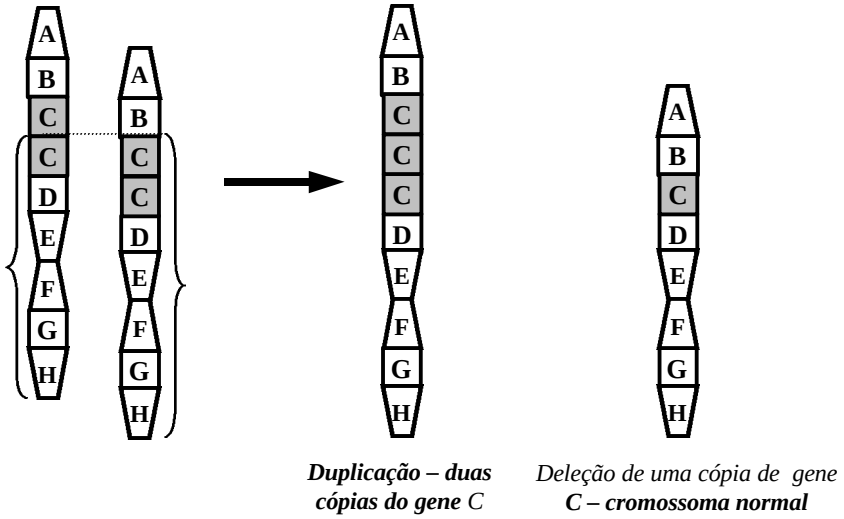


Figura 3. 13 – ***Mecanismo de crossover desigual a partir de cromossomas mutantes***

3.2.7. Quebra e Fusão

Muitas vezes uma porção de um cromossoma parte-se e funde-se na extremidade do cromossoma homólogo. Este processo origina um tipo especial de duplicação num cromossoma e eliminação no seu homólogo. A figura 3.14 mostra este mecanismo.

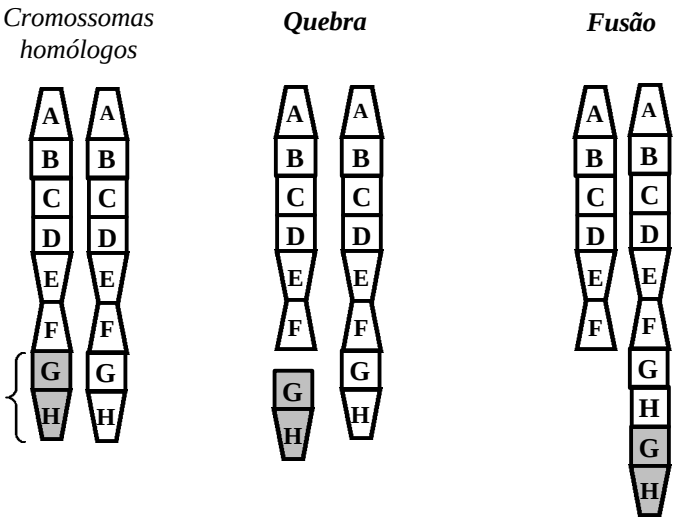


Figura 3. 14 – ***Mecanismo de quebra e fusão***

As consequências deste tipo de mutação nas células somáticas são de efeito mínimo. De facto, não ocorrendo perda de informação genética no par de cromossomas homólogos (apenas existe uma troca de posição), quando os cromossomas permanecem emparelhados (como é o caso das células somáticas) os efeitos não se fazem sentir.

No entanto, estas alterações dos cromossomas homólogos têm sérias consequências no processo da meiose que conduz à formação das células reprodutoras. Neste processo, os cromossomas homólogos separam-se para a formação dos gâmetas. Assim, metade das células reprodutoras produzidas terá uma lacuna de informação genética correspondente à parte eliminada. Um gâmeta com uma lacuna de informação é normalmente inviável. Qualquer ovo produzido pela fusão de uma destas células terá a mesma lacuna nos genes de um dos cromossomas homólogos. Assim, se a única cópia dos genes estiver danificada, sendo recessiva, numa situação normal não se manifestaria. Neste caso, não existindo a segunda cópia no cromossoma homólogo, os seus efeitos aparecerão no fenótipo [Russell 1998].

3.2.8. Inversão

O fenómeno de inversão caracteriza-se por uma alteração cromossómica de um segmento do cromossoma que se separa deste, roda 180º e se reintegra na mesma posição (figura 3.15).

Quando o segmento invertido inclui o centrómero, a inversão diz-se **pericêntrica**. Quando a inversão ocorre numa das extremidades do cromossoma não envolvendo o centrómero, designa-se por **paricêntrica**. Quer seja pericêntrica ou paricêntrica, a inversão tem consequências importantes na estrutura do cromossoma e no comportamento das inversões durante o período de *crossover* [Russell 1998].

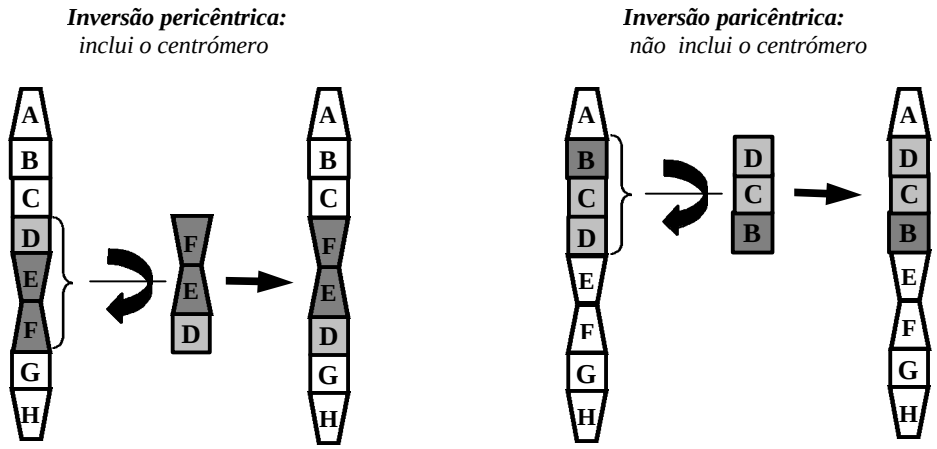


Figura 3. 15 – ***Mecanismo de inversão – inversão pericêntrica e inversão paricêntrica***

De uma forma geral, não se perde material genético durante o fenómeno de inversão, apesar de haver consequências ao nível do fenótipo quando os pontos de inversão definem regiões que controlam a expressão dos genes. Por outro lado, como a ordem dos genes pode afectar a regulação genética, as inversões podem corromper essa regulação.

As consequências da inversão podem ir mais longe, por exemplo, quando ocorre o processo de *crossover* (durante a meiose) entre dois cromossomas heterozigóticos (e.g. A**BCD**EFGH / A**DCB**EFGH) ou quando o processo de inversão for pericêntrica. Este último provoca emparelhamentos entre os cromossomas que levam à formação de laços.

3.2.9. Translocação

O fenómeno de **translocação** consiste numa mutação na qual ocorre uma alteração na posição de certos segmentos do cromossoma e na sequência de genes que esses segmentos contêm. Não existe perda ou ganho de material genético durante o fenómeno de translocação. Podem ocorrer dois tipos de translocação: Um tipo envolve a alteração na posição de um segmento dentro do mesmo cromossoma: este tipo designa-se por **translocação intracromossómica**. O outro tipo envolve a troca de material genético entre dois cromossomas não-homólogos: designa-se por **translocação intercromossómica** [Russell 1998)]. Neste último caso, se a translocação corresponder à transferência do segmento de um cromossoma para outro, estamos perante o fenómeno de **translocação não-recíproca**. Se a translocação se caracterizar pela troca de dois segmentos entre os dois cromossomas, trata-se do mecanismo de **translocação recíproca**. A figura 3.16 ilustra estes mecanismos.

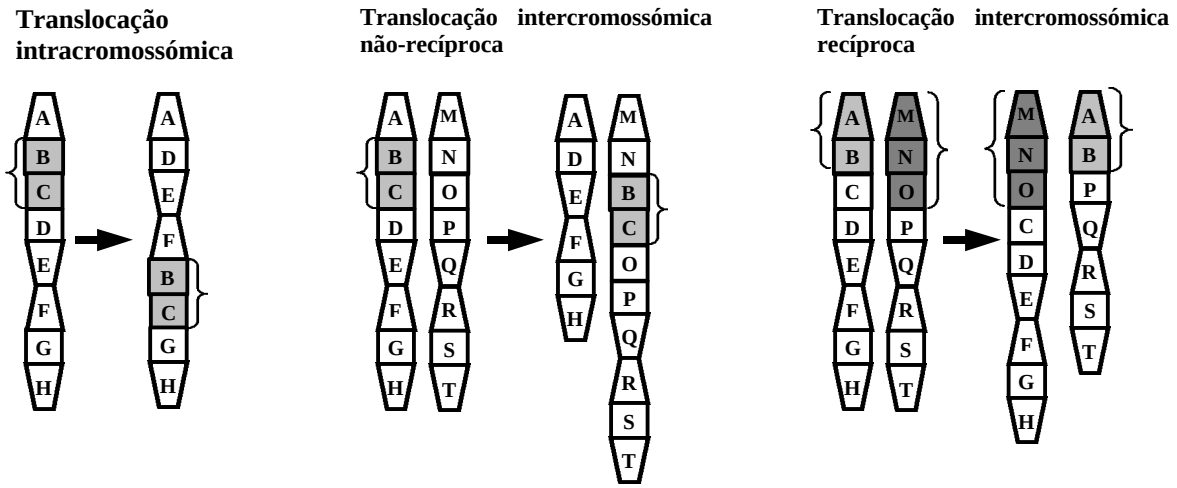


Figura 3. 16 – ***Mecanismo de translocação: intracromossómica, intercromossómica não-recíproca e recíproca.***

A principal consequência do mecanismo de translocação ocorre no processo da meiose, durante a formação dos gâmetas. Muitas vezes, estas células são desequilibradas, com duplicações e/ou deleções que, normalmente, provocam a sua inviabilidade. No caso de surgirem gâmetas viáveis podem levar ao aparecimento de síndromas, como o síndroma de *Down* (ou mongolismo) que surge como consequência de uma duplicação originada pelo mecanismo de translocação [Russell 1998].

3.2.10. Transposição

Este mecanismo será descrito com detalhe no próximo capítulo, visto ser o objecto principal do nosso estudo. No entanto, para que a descrição dos mecanismos incluídos na tabela 3.2 fique completa, ele será aqui abordado superficialmente.

O mecanismo de transposição caracteriza-se pela existência de unidades genéticas com capacidades de movimentação dentro do genoma. Estas entidades designam-se por transposões[52] ou "genes saltitões".

Um transposão pode consistir num conjunto de um ou mais genes ou numa unidade de controlo e possui a capacidade de se movimentar de diversas formas no genoma de um indivíduo. Para que um elemento transponível seja reconhecido como tal é necessário que sejam detectadas sequências flanqueadoras com determinadas características.

A movimentação do transposão resume-se em duas etapas distintas: a excisão, quando abandona o cromossoma e a integração, quando encontra uma região do genoma apropriada para se inserir.

3.3. Incorporação dos mecanismos descritos no Algoritmo Genético

Como se pode constatar pela leitura dos secções anteriores, a biologia molecular é uma fonte de inspiração bastante rica no que se refere a operadores biológicos que alteram a estrutura dos cromossomas.

Os AG, tendo a sua inspiração nas ideias da biologia, utilizam, na sua forma clássica, o operador de *crossover* (inspirado no processo de *crossover* que ocorre na meiose) e o operador de mutação (processo simples que altera ao valor de um gene). No entanto, como já foi referido no capítulo anterior, várias alterações têm sido propostas, no que diz respeito à utilização de novos mecanismos baseados nos sistemas biológicos. Por exemplo, o operador de inversão [Holland 1992], o operador de conjugação [Harvey 1996], [Smith 1996a], [Smith 1996b], o operador de transdução [Furuhashi et al. 1994], [Yoshikawa et al. 1997], [Nawa et al. 1997], [Nawa et al. 1998] e [Nawa et al. 1999] os operadores de deleção e duplicação [Haynes 1998], [Koza 1992] foram já objecto de

estudo. Também o operador de translocação foi já utilizado por [Oates et al. 1999]. Os autores denominam o seu operador por "*skewed crossover*" não o identificando com o seu correspondente biológico. No entanto, o seu funcionamento é similar ao que ocorre no mecanismo de transposição biológico.

Neste trabalho iremos aprofundar mais um desses mecanismos - a transposição - para verificar se a sua incorporação num AG conduz a algum tipo de vantagens. Estudos preliminares apontam resultados bastante promissores.

Para além dos operadores focados, restam os mecanismos de transformação, quebra e fusão e *crossover* desigual. Mesmo os operadores de conjugação, translocação e transposição merecem um estudo mais aprofundado focando outras abordagens. De facto, o operador de conjugação apenas explorou a transferência entre células do tipo $\mathbf{F}^+$ - $\mathbf{F}^-$ (desprezando as células **Hfr**). Por outro lado, operador de transposição (que estudaremos de seguida) apenas se aplica em situações com cromossomas de tamanho fixo.

Pensamos, portanto, que muito trabalho se pode fazer nesta área, quer a nível empírico, quer a nível teórico, de forma a que se possa concluir da viabilidade da utilização de todos estes operadores num AG e se alguns deles, mascarados sob a forma de outro nome, já não terão sido propostos em abordagens genéticas mais específicas.

[52] do inglês, *transposons*

CAPÍTULO 4.
O MECANISMO DE TRANSPOSIÇÃO

Sumário

Neste capítulo faremos uma descrição detalhada do novo mecanismo de recombinação que propomos neste trabalho como alternativa aos operadores tradicionais de *crossover*.

Inicialmente, será abordado o funcionamento do mecanismo de transposição nos seres vivos. Este mecanismo apresenta muitas variantes, dependendo do tipo de seres vivos onde actua. No entanto o seu funcionamento genérico é semelhante em todos eles. Será esta descrição genérica que será feita no âmbito deste capítulo.
Todos os aspectos abordados nesta secção basearam-se fundamentalmente nos textos de [Gould et al. 1996] e [Russell 1998].

De seguida, centraremos a descrição no mecanismo de transposição computacional. Descreveremos quais as simplificações assumidas para a sua implementação e proporemos duas variantes deste novo operador: transposição simples e transposição baseada em torneio.

Para que o funcionamento do novo operador possa ser compreendido de forma mais clara, demonstraremos o seu funcionamento através de um exemplo simples.

4.1. A transposição nos seres vivos

Como já foi descrito anteriormente, o objectivo deste trabalho consiste na proposta de um novo operador genético que possa ser incorporado no AG como alternativa aos operadores tradicionais de *crossover*. Assim, tal com estes, o mecanismo a propor terá como finalidade a manutenção da diversidade de população de geração em geração.

O mecanismo que a seguir se descreve baseia-se em fenómenos biológicos presentes nos seres vivos e caracteriza-se por alterações cromossómicas de grande escala. Estas alterações manifestam-se não ao nível de um gene isoladamente, mas num bloco de genes [Russell 1998], [Gould et al. 1996]. O mecanismo designa-se por **transposição**.

As alterações genéticas provocadas pelo fenómeno de transposição são responsáveis por variações que vão desde o aparecimento de células cancerígenas até à potencial alteração evolutiva das espécies.

Muito do trabalho feito actualmente em engenharia genética faz uso deste e de outros fenómenos deste tipo. Muitos destes trabalhos caracterizam-se pela utilização de agentes de mobilidade genética para o desenvolvimento de projectos de interesse científico, médico ou comercial [Gould et al. 1996]. Por exemplo, estas manipulações genéticas permitem obter espécies vegetais resistentes a determinadas doenças ou animais capazes de produzir grandes quantidades de hormonas como a insulina.

O mecanismo de transposição caracteriza-se pela existência de unidades genéticas com capacidades de movimentação dentro do genoma. Estas entidades designam-se por **transposões**. Estas unidades podem movimentar-se de várias formas: ou abandonando um cromossoma e introduzindo-se noutro, ou deixando uma cópia atrás de si e só depois a introdução na nova posição, ou ainda gerando várias cópias suas e disparando-as para diferentes cromossomas. A movimentação pode ainda ser efectuada dentro do mesmo cromossoma.

O fenómeno da transposição foi descoberto por Barbara McClintock na década de 50 (altura em que ainda não era completamente conhecida a estrutura do ADN). O seu estudo provou que certos fenómenos ocorridos em seres vivos expostos a radiações UV

não podiam ser resultado da recombinação normal ou das mutações conhecidas. Os transposões foram vistos durante muito tempo com alguma *anormalidade* mas, em 1983, quando B. McClintock foi galardoada com o prémio Nobel muitos transposões já eram do conhecimento cientifico e o seu papel na evolução começava a ser reconhecido.

Ao contrário do que possa parecer o mecanismo de transposição não é um fenómeno raro. Por exemplo, 5% do genoma humano possui cópias de um transposão (o L1-retrotransposão) responsável pela forma mais comum de hemofilia.

Um transposão pode consistir num conjunto de um ou mais genes ou numa unidade de controlo. Esta unidade genética é constituída por um conjunto de genes delimitado por sequências flanqueadoras com determinadas características. Para que um elemento transponível seja reconhecido como tal é necessário que essas sequências sejam detectadas. As sequências flanqueadoras, dependendo do tipo de transposão, podem ser iguais ou inversas. O exemplo que a seguir se apresenta, mostra as sequências flanqueadoras como conjuntos de bases azotadas (A – adenina; G – guanina; T – timina – C- citosina), elementos construtores do material genético. O símbolo ***N*** representa o restante material genético, que não apresenta as características das sequências que ladeiam o transposão.

Assim, conforme dissemos, poderemos ter transposões formados a partir da existência de sequências flanqueadoras idênticas ou inversas, tal como se mostra na figura 4.1.

NNNNN**ATTGA***(Transposão)***AGTTA**NNNNNN *(sequências flanqueadoras inversas)*

ou

NNNNN**ATTGA***(Transposão)***ATTGA**NNNNNN *(sequências flanqueadoras iguais)*

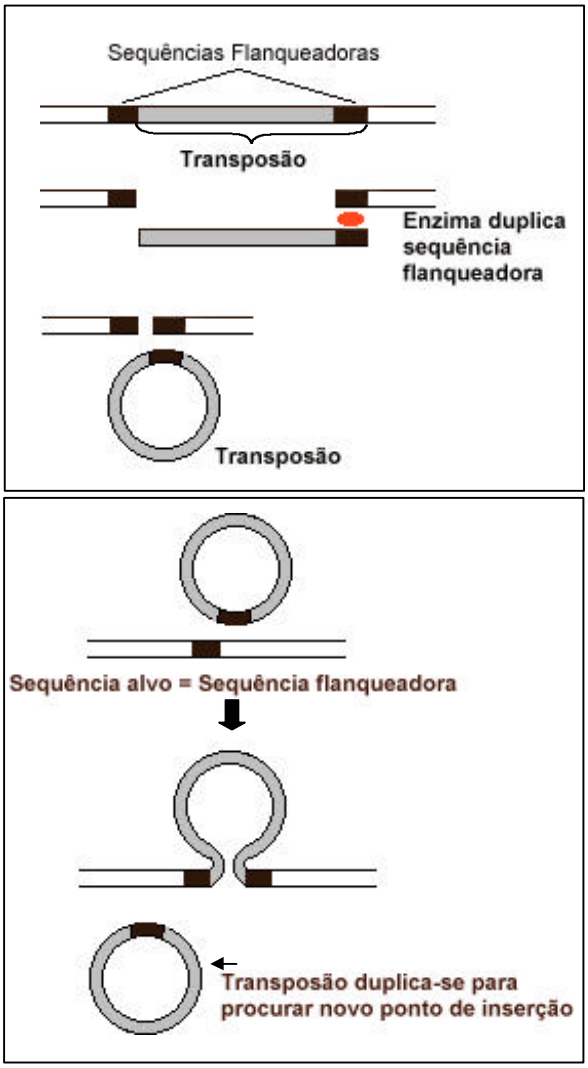
Figura 4. 1*– Transposão formado por sequências flanqueadoras iguais ou inversas*

O movimento do transposão caracteriza-se por duas fases distintas:

- Excisão – quando o transposão abandona a posição original
- Integração - quando o transposão se insere numa nova posição do genoma.

Quando se movimenta no genoma, pelo menos uma das sequências acompanhará o transposão. O ponto de inserção, requerido para o processo de integração, não requer qualquer homologia com o elemento transponível, o que confere a este mecanismo uma maior liberdade no processo de rearranjo de material genético, quando comparado com o *crossover*. Assim, o ponto de inserção é escolhido dependendo do tipo de transposão. Existem alguns elementos que não têm qualquer especificidade em relação ao ponto de inserção, outros inserem-se numa região que tenha alguma correspondência com a sequência flanqueadora que o acompanha, havendo outros ainda que se caracterizam pelo facto de possuírem um grau de preferência para se inserirem numa determinada região do cromossoma. No limite, existem elementos que apenas aceitam introduzir-se no *loci* correspondente a um gene específico da cadeia de ADN.

Ao abandonar o cromossoma o transposão (excisão) pode (ou não) deixar uma cópia sua e, depois disso dispõe-se em forma de circulo movimentando-se no genoma até ser reintroduzido (integração) numa nova posição. Quando se introduz na nova posição o transposão pode duplicar-se, continuando assim o processo de integração noutras zonas do genoma. Ver figura 4.2.



EXCISÃO

INTEGRAÇÃO

Figura 4. 2 *– processo de transposição: EXCISÃO + INTEGRAÇÃO*

Ao contrário do mecanismo de recombinação, os transposões não requerem qualquer correspondência entre os pontos de inserção do elemento transponível e da cadeia de ADN. Durante o processo de *crossover* os cromossomas trocam material genético, através de pontos de corte que têm que ser homólogos nos dois cromossomas. Na transposição, esta exigência não se coloca, uma vez que o ponto de inserção é determinado quer pelas características das sequências flanqueadoras, quer pelo grau de preferência que o transposão possa ter por uma região específica do genoma. Por esta

razão, a transposição é normalmente referida como a ***recombinação ilegítima*** [Russell 1998], [Gould et al. 1996].

É da responsabilidade de uma enzima (codificada pelo próprio transposão) o reconhecimento de uma sequência de genes que corresponda à sequência flanqueadora e, portanto a detecção de uma zona possível para a inserção da unidade genética.

4.1.1. Diferentes tipos de transposões

O tipo mais simples de elementos transponíveis são as **sequências de inserção**[53] [Russell 1998], [Gould et al. 1996]. Estes elementos transportam apenas os genes necessários para a movimentação do elemento e respectiva inserção numa nova posição. Além do transposão simples não transportam qualquer informação genética adicional. Ironicamente, a simplicidade da sua estrutura, torna-os bastante difíceis de estudar visto que não transportam material genético que possa ser facilmente marcado. São detectados pela sua capacidade de provocar mutações e são estudados indirectamente a partir dos efeitos que provocam. Caracterizam-se pelo seu pequeno tamanho, normalmente 1 Kb (Kilobase = 1000 *bp*[54]) de comprimento.

Outro tipo de elementos transponíveis são os **transposões compostos**[55]. Estas unidades transponíveis são mais complexas e maiores que as anteriormente referidas. Neste caso, o transposão é ladeado por sequências de inserção e transporta informação genética que permite, por exemplo, a resistência a determinados antibióticos. Tal como foi referido estas sequências possuem também elas próprias sequências iguais ou inversas a flanquear o transposão que permitem que o transposão composto possua também a capacidade de movimentação.

A terceira categoria de elementos transponíveis são os **transposões complexos**[56] ou transposões não-compostos[57]. Diferem dos transposões compostos pelo facto de não possuírem sequências de inserção a flanquearem os restantes genes. No entanto, as sequências flanqueadoras deste tipo de unidades transponíveis possuem as mesmas características das anteriores, isto é, podem ser sequências iguais ou inversas.

[53] do inglês, *Insertion Sequences*
[54] do inglês, *Base pairs*, que significa pares de bases azotadas
[55] do inglês, *Compound Transposons* ou *Composite Transposons*
[56] do inglês, *Complex Transposons*
[57] do inglês, *Noncomposite Transposons*

Como se verifica, apesar da diferente complexidade e do tipo de informação genética que cada tipo de unidade transponível transporta, todas as variantes de transposões possuem características semelhantes. Destas características salienta-se a sua capacidade de se movimentarem no genoma, provocando alterações cromossómicas quer no ponto de excisão quer na sua nova localização. Esta capacidade de movimentação é devida, em todos os elementos, à existência de sequências flanqueadoras, que apesar de diferentes, possuem a mesma característica genérica: podem ser sequências iguais ou inversas.

Resta-nos saber como se movimentam no genoma dos organismos cada um destes tipos de transposões. Este aspecto será abordado de seguida.

4.1.2. Como se movimentam os diferentes tipos de transposões

Apesar da variedade de elementos transponíveis existente, o modo como eles se movimentam no genoma é semelhante para todos eles. Assim, para o mesmo transposão, o tamanho das sequências flanqueadoras é constante. Mas, transposões diferentes podem ser flanqueados por sequências de diferentes tamanhos. A tabela 4.1 exemplifica o tamanho das sequências para alguns tipos de transposões que se podem encontrar nos sistemas biológicos [Russell 1998].

Transposão	Tamanho da sequência (bp)
Tn3	5
Tn10	9
IS4	11
IS5	4

Tabela 4. 1 – *Correspondência entre diferentes transposões e o tamanho das sequências flanqueadoras.*

As sequências flanqueadoras permitem que uma determinada unidade seja reconhecida como tendo capacidade de se movimentar no genoma. Os transposões podem abandonar a posição original e, neste caso, provocam grandes alterações no cromossoma de excisão, visto que a funcionalidade dos genes que se movimentam fica destruída. Este tipo de transposição designa-se por **transposição conservativa** ou **não**

replicativa[58] ou ainda **inserção simples**[59]. Noutros casos, o transposão duplica-se (uma ou mais vezes) e são estas cópias que se irão integrar em novas posições. Este tipo de transposição designa-se por **transposição replicativa**[60]. Quando se inserem numa nova posição do genoma, provocam também alterações genéticas, visto que o cromossoma passa a ter um conjunto de genes que anteriormente não possuía.

O processo de transposição é iniciado através da actividade de uma enzima designada transposase (ou outras enzimas nos casos mais complexos). A acção das enzimas consiste, por exemplo, em reconhecer o ponto de inserção para o transposão em causa, quer pelo reconhecimento de uma sequência de determinadas características, quer pela localização de uma região preferencial para o elemento a integrar.

No momento de transposição, pelo menos uma das sequências acompanhará a unidade transponível e poderá servir para que a enzima reconheça noutro cromossoma o ponto de inserção requerido para o transposão.

Resumindo, verifica-se que todos os tipos de transposões possuem a capacidade de se movimentar no genoma. Podem fazê-lo de várias formas, como por exemplo, através de duplicação e posterior inserção ou sem duplicação, com perda de funcionalidades dos genes do cromossoma que abandona. A integração do transposão não é feita de forma aleatória, existindo um conjunto de regras que se podem aplicar, dependendo dos casos em concreto. Assim, alguns transposões possuem preferências regionais por uma determinada zona ou gene[61]. Existem outros que apenas se inserem em posições que possuam uma correspondência com as sequências flanqueadoras do transposão. Todo o mecanismo de transposição é regulado por uma ou mais enzimas codificadas pelo próprio transposão.

4.2. Transposição Computacional

De forma a poder ser implementado no AG, o mecanismo biológico de transposição foi alvo de algumas simplificações.

[58] do inglês, *conservative or replicative transposition*
[59] do inglês, *simple insertion*
[60] do inglês, *nonreplicative transposition*
[61] do inglês, *integration hotspots*

O objectivo principal deste operador será a manutenção da diversidade da população através do rearranjo do material genético, que será feito de forma diferente dos operadores tradicionais de *crossover*. Tal como acontece nos sistemas biológicos, nos sistemas computacionais este mecanismo irá ser usado como um operador de recombinação ilegítima, isto é, sem correspondência entre os pontos de troca do material genético. Estes serão escolhidos seguindo um processo semelhante ao que acontece nos organismos vivos.

O mecanismo de transposição será implementado de forma a manter as duas fases que o caracterizam: excisão e integração. Assim, quando o transposão estiver reconhecido, a partir da existência de sequências flanqueadoras de determinadas características, abandonará o cromossoma onde se posiciona. A nova localização será definida através da pesquisa de um ponto de inserção noutro cromossoma. Este ponto de inserção não será necessariamente homólogo com o ponto de onde partiu o transposão.

Para definir claramente o mecanismo de transposição interessa definir alguns aspectos:

- Como se forma o transposão;
- Como se movimenta o transposão;
- Como se define o ponto de inserção;
- Qual o tamanho das sequências flanqueadoras;
- Se há ou não duplicação do transposão quando abandona o *loci* inicial e quando se insere num novo *loci*;

Vejamos cada um destes aspectos com mais detalhe.

4.2.1. Como se forma o transposão

A transposição ocorrerá entre dois cromossomas escolhidos pelo método de selecção utilizado no AG (no nosso caso, roleta com elitismo). O gene que marcará o início do transposão será escolhido aleatoriamente entre todos os genes que constituem o cromossoma

Estando previamente definido um tamanho para as sequências flanqueadoras (**N**) , a sequência inicial consistirá nos **N** bits que antecedem o gene marcado para o início do transposão. A segunda etapa para que se consiga formar o transposão consiste em

encontrar a outra sequência flanqueadora. Esta será formada tendo em conta a primeira: após o gene marcado para início do transposão procura-se um conjunto de bits que formem uma sequência igual ou inversa à primeira sequência encontrada.

O transposão encontra-se delimitado por estas duas sequências. A figura 4.3 exemplifica o processo descrito, com o tamanho das sequências igual a 4.

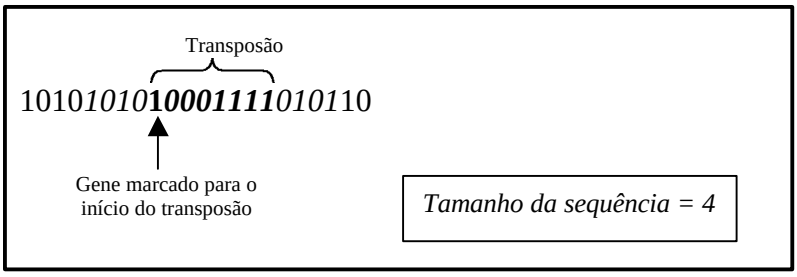


Figura 4. 3 – *Processo de formação do transposão*

A sequência que acompanhará o transposão quando este se reintegrar noutra posição será sempre a segunda sequência flanqueadora.

Em situações em que o gene escolhido para início da unidade móvel seja o primeiro ou o último gene do cromossoma e em todas as situações em que o tamanho escolhido para a sequência flanqueadora não consiga abranger os bits que antecedem o gene escolhido, o cromossoma será visto como um círculo fechado. Assim, torna-se sempre possível a escolha dos **N** bits que antecedem o gene seleccionado aleatoriamente. A figura 4.4 mostra esquematicamente o cromossoma do exemplo anterior visto como um círculo. O seu início está assinalado.

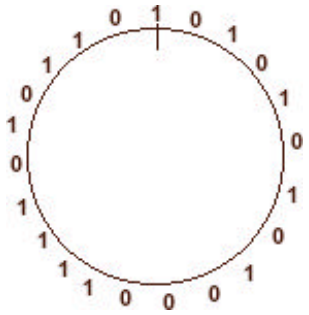


Figura 4. 4 – *Representação do cromossoma em circulo*

4.2.2. Como se define o ponto de inserção

A escolha do segundo indivíduo (que servirá como cromossoma receptor para concretizar a fase de integração) é feita de acordo com o valor dado pela função de avaliação. Depois de seleccionado, este indivíduo será analisado desde o primeiro gene em busca de uma sequência de bits igual ou inversa à sequência que acompanha o transposão. Quando se encontrar uma sequência de bits que satisfaça a condição descrita, o ponto de inserção será marcado a partir do gene que se segue ao último bit da sequência encontrada.

A figura 4.5 exemplifica este processo.

Transposão formado em:
Cromossoma 1 – 111*01***1111*01***011001

Procurar ponto de inserção em:
Cromossoma 2 – 0*01*01010101111111

Ponto de inserção do transposão

Figura 4. 5 – *Processo de pesquisa do ponto de inserção no cromossoma receptor*

4.2.3. Como se movimenta o transposão

Depois de formado o transposão torna-se necessário definir a forma como ele se movimentará no genoma. A primeira abordagem para o problema consistiu em escolher aleatoriamente um dos progenitores, previamente seleccionados, para ser o cromossoma do qual sairia o transposão, sendo o outro, o cromossoma para a integração. Após a formação do transposão no cromossoma 1 e respectiva identificação do ponto de inserção no cromossoma 2, troca-se uma quantidade de material genético igual ao tamanho do transposão entre os dois indivíduos seleccionados. Assim, o transposão desloca-se para o cromossoma 2 mas, como nos casos em estudo a representação exige cromossomas de tamanho fixo, o espaço deixado em vazio neste cromossoma é preenchido pela mesma quantidade de material genético proveniente do cromossoma 2. A esta primeira abordagem designamos mecanismo de **transposição simples**.

As primeiras experiências feitas com este mecanismo demonstraram que os novos indivíduos gerados por este processo de troca provocavam uma grande flutuação na qualidade média da população. Conseguiram atingir-se valores máximos de grande qualidade, mas surgiram também indivíduos de qualidade muito baixa. Para tentar evitar esta oscilação exagerada, implementámos uma segunda variante do mecanismo de transposição inspirada no trabalho de [Harvey 1996]. Neste artigo o autor propõe um AG que utiliza o operador de conjugação já descrito no capítulo 2.

Esta variante do mecanismo de transposição, designada **por transposição baseada em torneio** (abreviada para transposição/torneio), coloca em competição os dois indivíduos previamente seleccionados. O que ganhar o torneio será o escolhido para a constituição do transposão (cromossoma dador) e o segundo para a escolha do ponto de inserção (cromossoma receptor). A escolha deste ponto será feita da mesma forma que no caso anterior.

Após ser encontrado o transposão e o respectivo ponto de integração, a unidade móvel será duplicada e inserida no cromossoma receptor ocupando a posição dos **N** bits (tamanho do transposão) que se seguem ao ponto de inserção encontrado. Ao contrário do caso anterior, em que os dois cromossomas eram alterados pela troca bidireccional de material genético, neste caso apenas o pior indivíduo é modificado através da injecção do transposão no seu código genético (troca unidireccional de genes). A figura 4.6 esquematiza as duas variantes propostas. No exemplo apresentado, o cromossoma dador é o mesmo nos dois casos.

Transposição Simples	**Transposição baseada em torneio**
Cromossoma 1 – 111*01***1111*01***011001	Cromossoma 1 – 111*01***1111*01***011001
Cromossoma 2 – 0*01***010101**01111111	Cromossoma 2 – 0*01***010101**01111111
Filho 1 – 111*01***010101**011001	Filho 1 – 111*01***1111*01***011001
Filho 2 – 0*01***1111*01***01111111	Filho 2 – 0*01***1111*01***01111111

Figura 4. 6 - *Representação do funcionamento das duas variantes propostas para o mecanismo de transposição*

Quando se utiliza este processo de transposição, a flutuação registada na caso anterior diminuiu ligeiramente. Relativamente aos valores máximos, estes continuam a ser atingidos quando o AG é implementado com esta variante.

No estudo empírico realizado neste trabalho o AG foi executado com as duas variantes, para podermos comparar com mais detalhe os resultados alcançados com cada uma delas.

Como este estudo não tem qualquer precedente que nos permita à partida definir um valor para as sequências flanqueadoras, faremos uma variação desde tamanho de 1 até um valor máximo que dependerá do tamanho do cromossoma que codifica o problema em causa.

4.2.5. Casos particulares

Nos processos de formação do transposão através da procura da segunda sequência flanqueadora e da pesquisa do ponto de inserção, dois casos particulares podem acontecer:

1. não ser encontrada a segunda sequência flanqueadora para a formação do transposão;
2. não ser encontrada sequência alvo coincidente com a desejada para definição do ponto de integração.

Ao considerar o cromossoma como um circulo a primeira situação fica resolvida, visto que no limite, o transposão coincidirá com o próprio cromossoma e, neste caso não se fará reintegração.

Para a segunda situação, se a sequência alvo não for encontrada, o mecanismo de transposição não ocorre, sendo a descendência igual aos dois progenitores.

4.3. O Algoritmo

Depois de definidos os aspectos básicos para o funcionamento do mecanismo de transposição é já possível fazer uma descrição do processo total, nas suas duas variantes (transposição simples e transposição/torneio), sob a forma do algoritmo implementado.

O algoritmo descreve-se na figura 4.7, em pseudo-código.

```
PROCEDIMENTO TRANSPOSIÇÃO (cromossoma 1, cromossoma 2, tipo)
 BEGIN
        If (tipo == "simples")        // transposição simples
          dador = aleat (cromossoma1, cromossoma 2);
        else      // transposição baseada em torneio
          dador = torneio (cromossoma1, cromossoma 2);
        receptor = outro_cromossoma (dador, cromossoma1, cromosoma2);
        inicioT = aleat (dador);       // escolhe aleatoriamente gene para início do transposão
        sequência = forma_sequência (dador, inicioT, tamanho_seq);
        transposão = forma_transposão (dador, inicioT, sequência);
        inserção = encontra_ponto (receptor, sequência);

//Troca de material genético
        If (tipo == "simples")        // transposição simples
            troca_material( dador, receptor, inserção, inicioT,  transposão);

        else      // transposição baseada em torneio
            insere_transposão (dador, inserção, transposão);
END.
```

Figura 4. 7 – *Algoritmo do mecanismo de transposição*

Como é facilmente perceptível, o procedimento recebe como entrada os dois cromossomas envolvidos na transposição, bem como o tipo de transposição seleccionado. Depois de definidos os cromossomas dador e receptor, o transposão é localizado no indivíduo dador e o ponto de inserção define-se no cromossoma receptor.

A troca do material genético entre os dois indivíduos seleccionados depende do tipo de transposição. No mecanismo de transposição simples, a troca é bidireccional. Quando se utiliza a variante com torneio, a troca é unidireccional, isto é, o transposão é injectado no cromossoma receptor.

4.4. Um exemplo

Apresenta-se de seguida um exemplo que ilustra detalhadamente o funcionamento do mecanismo implementado. O exemplo escolhido é bastante simples para que melhor se perceba o seu funcionamento. Este exemplo mostra a

implementação de cada uma das variantes do mecanismo, isto é, transposição simples e transposição baseada em torneio.

As características do exemplo escolhido são as seguintes:

- codificação binária para o problema;
- tamanho do cromossoma = 50
- tamanho das sequências flanqueadoras = 4
- Gene de início do transposão = **inicioT** = bit 10
- função de avaliação ⇒ f = maximizar (número de 1's)

4.4.1. Mecanismo de transposição simples

Nesta variante do mecanismo de transposição, os cromossomas dador e receptor são escolhidos aleatoriamente entre os dois progenitores anteriormente seleccionados para reprodução. A troca do material genético é bidireccional, o que implica que os dois indivíduos da descendência são gerados pela troca da mesma quantidade de material genético.

P1 - Progenitor 1 – 11101010100011000011011001111001000100110101010111 (f = 26)
P2 - Progenitor 2 – 11101001110101101111111100000000100110111111111011 (f = 32)

Dador = aleat(P1, P2) = P1

definir sequência flanqueadora e formar transposão

P1 – dador - 11101*0101***0**0011000*1010*1101111001000100110101010111 (f = 26)

Transposão

Procurar ponto de inserção

P2 – receptor - 11*1010***0**11101011011111111000000001001101111111111011 (f = 32)

Ponto de inserção

TROCA DE MATERIAL GENÉTICO

P1 – dador - 11101*0101***000110000*1010***1101111001000100110101010111 (f = 26)

P2 – receptor - 11*1010***0111010110111**1111100000000100110111111111011 (f = 32)

Filho 1 - 11101*0101***0111010110111**1101111001000100110101010111 (f = 31)
Filho 2 - 11*1010***000110000*1010***1111100000000100110111111111011 (f = 27)

4.4.2. Mecanismo de transposição baseada em torneio

Nesta variante do mecanismo de transposição, os dois progenitores anteriormente seleccionados para reprodução são sujeitos a um torneio. O vencedor do torneio será o cromossoma dador onde se formará o transposão e o perdedor será o cromossoma receptor, no qual se injectará a unidade formada. Neste caso, a troca de material genético é unidireccional, o que implica que a descendência será gerada do seguinte modo: o pior indivíduo (perdedor do torneio) é modificado com a assimilação da unidade móvel, enquanto o outro indivíduo permanece inalterado.

P1 - Progenitor 1 – 1110101010001100001101100111100100010011010101010111 (f = 26)
P2 - Progenitor 2 – 1110100111010110111111110000000010011011111111111011 (f = 32)

Dador = melhor(P1, P2) = P2

definir sequência flanqueadora e formar transposão

P2 – dador – 11101*0011*1010110111111*1100* 0000001001101111111111011 (f = 32)

Transposão

Procurar ponto de inserção

P1 – receptor - 1110101010*0011***0**0001101100111100100010011010101010111 (f = 26)

Ponto de inserção

TROCA DE MATERIAL GENÉTICO

P2 – dador - 11101*0011***1010110111111*1100*** 0000001001101111111111011 (f = 32)

P1 – receptor - 1110101010*0011**0**0001101100111100*1000100110101010111 (f = 26)

Filho 1 = P2 - 11101*0011***1010110111111*1100*** 0000001001101111111111011 (f = 32)
Filho 2 - 1110101010*0011***1010110111111*1100***1000100110101010111 (f = 30)

CAPÍTULO 5

CARACTERÍSTICAS DO AMBIENTE DE EXPERIMENTAÇÃO

Sumário

Para comparar o desempenho dos operadores genéticos tradicionais com o do novo mecanismo proposto, tornou-se necessário definir uma medida de desempenho. Para tal, usou-se o problema clássico de optimização de funções abrangendo um problema de maximização e outro de minimização. Para cada um deles seleccionou-se um conjunto de funções de teste em parte já bastante exploradas por diversos autores. Estas funções foram seleccionadas de modo a abranger um vasto conjunto de características, como por exemplo a continuidade/descontinuidade, baixa e alta dimensionalidade, número de máximos (ou mínimos locais), convexidade/não convexidade, entre outras. Para o problema de maximização escolheram-se as funções de teste de De Jong e um conjunto de funções suplementar que foi escolhido de forma a complementar as características das funções de De Jong. Para o problema de minimização utilizou-se um conjunto de funções sugeridas por Koon e Shebald. No total, em ambas as categorias (maximização e minimização) reunimos 18 funções de teste que serviram de base ao trabalho experimental realizado e permitiram comparar os resultados obtidos pelos operadores genéticos em estudo.

Neste capítulo serão abordadas estas 18 funções, salientado como se codificou o problema para cada uma delas e ainda as suas características mais relevantes. No final do capítulo apresentaremos os parâmetros usados no AG tal como este foi implementado para estudar o problema. Destacam-se os seguintes aspectos: a forma como o mecanismo de transposição substituiu o operador de *crossover*, as probabilidades dos operadores de *crossover* (ou transposição) e mutação, o mecanismo de selecção utilizado, o tamanho da população e das sequências flanqueadoras.

No **ANEXO 1** encontram-se as representações gráficas das funções. Estas representações foram executadas com a versão de estudante do Matlab 5, que apresenta

algumas limitações. Assim, todas as funções com mais de duas variáveis foram desenhadas com apenas duas. Além disso a precisão utilizada no Matlab não corresponde àquela que foi implementada. No entanto, apesar de algumas representações não corresponderem com total exactidão à utilizada na realidade, permitem dar uma ideia visual das características mais importantes de cada um das funções.

5.1. Problema de maximização

Para medir o desempenho do AG quando utilizado com os diferentes operadores genéticos, tornou-se necessário definir um método de avaliação dos resultados obtidos. As funções de teste constituem medidas de desempenho que permitem avaliar o desempenho do AG em determinadas circunstâncias [Foster 1995], [Whitley et al. 1995].

No nosso estudo experimental pretendemos avaliar o AG em situações de optimização de funções. Para o primeiro caso, seleccionámos as funções de De Jong já bastante utilizadas neste tipo de problemas. No entanto, um bom conjunto de teste deve incluir funções cujas características abranjam um vasto leque de opções [Whitley et al. 1995], tais como:

- Discretas e contínuas;
- Unimodais e multimodais;
- Baixa e alta dimensão;
- Quadráticas e não-quadráticas
- Convexas e não convexas.

Assim, para complementar o conjunto de funções de De Jong, seleccionámos um conjunto de teste suplementar constituído pelas seguintes funções:

- Função de Michalewicz [Michalewicz 1999];
- 2 Funções de Bohachevsky [Fogel 1995];
- Função de Griewangk [Whitley et al. 1995];
- Função de Rastrigin [Whitley et al. 1995];
- Função de Schwefel [Whitley et al. 1995].

De seguida apresentaremos detalhadamente as características de cada uma das funções, incluindo a forma como o cromossoma codificou o problema em cada caso.

5.1.1. Funções de teste de K. De Jong

Este conjunto de teste consiste em 5 funções, e serviram de base ao estudo realizado por De Jong em [De Jong 1975]. De seguida descrevem-se as características

destas funções que a seguir passaremos a designar por **Funções de Teste de De Jong** (F1 a F5)

F1 – Função de teste de De Jong

A função F1 de De Jong é definida por:

$$F_1(x_i) = \sum_{i=1}^{3} x_i^{\,2} * i \qquad (5.1.)$$

Características da função

A função $\mathbf{F_1}$ é uma parábola simples de dimensão 3. É contínua, convexa, unimodal, de baixa dimensão (3 variáveis - x_i) e quadrática. A representação gráfica da versão de duas variáveis desta função encontra-se na Figura 1 no ANEXO 1.

O máximo da função coincide com o valor **78,6**.

Nos testes realizados $\mathbf{F_1}$ foi restringida ao espaço definido por $-5{,}12 \leq x_i \leq 5{,}12$, onde i = 1, 2, 3, com uma precisão de duas casas decimais.

Partindo destas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Precisão 2 casas decimais.
- Tamanho do domínio para x_i: 10,24
- Número de bits para x_i: 10,24* 100 = 1024 ➪ 2^{10} = 1024 ➪ 10 bits por variável
- Tamanho do cromossoma = Dimensão * 10 = 3 * 10 = **30 bits**

F2 – Função de teste de De Jong Test

A função F2 de De Jong é definida por:

$$F_2(x_1, x_2) = 100 * \left(x_1^{\,2} - x_2^{\,2}\right)^2 + \left(1 - x_1\right)^2 \qquad (5.2)$$

Características da função

A função F2 é uma função standard na literatura de optimização. É uma função contínua, não convexa, unimodal, quadrática e de baixa dimensão (2 variáveis – x_i). A representação gráfica desta função encontra-se na Figura 2 no ANEXO 1.

O máximo da função coincide com o valor **3905,926**.

Nos testes realizados **F_2** foi restringida ao espaço definido por $-2{,}048 \leq x_i \leq 2{,}048$, onde i = 1, 2, com uma precisão de três casas decimais.

Com estas características o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 4,096
- Número de bits para x_i: 4,096 * 1.000 = 4096 ➪ 2^{12} = 4096 ➪ 12 bits por variável
- Tamanho do cromossoma = Dimensão * 12 = 2* 12 = **24 bits**

F3 – Função de teste de De Jong

A função F3 de De Jong é definida por:

$$F_3(x_i) = \sum_{i=1}^{5} integer(x_i) \qquad (5.3.)$$

Características da função

Esta função é uma função em degrau de 5 variáveis. É descontínua, não convexa, unimodal, não quadrática e de média dimensão (5 variáveis – x_i). Do conjunto de funções de De Jong é a única que se caracteriza pela descontinuidade. *Integer* representa o maior inteiro menor ou igual a x_i. A sua representação gráfica encontra-se no ANEXO 1, Figura 3.

O máximo da função coincide com o valor **25**.

Nos testes realizados **F_3** foi restringida ao espaço definido por $-5{,}12 \leq x_i \leq 5{,}12$, onde i = 1, 2, 3, com uma precisão de duas casas decimais.

Considerando estas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 10,24
- Número de bits para x_i: 10,24 * 100 = 1024 ➪ 2^{10} = 1024 ➪ 10 bits por variável
- Tamanho do cromossoma = Dimensão * 10 = 5 * 10 = **50 bits**

F4 - De Jong Test Bed Function

A função F4 de De Jong é definida por:

$$F_4(x_i) = \sum_{i=1}^{30} i x_i^4 + Gauss(0,1) \qquad (5.4)$$

Características da função

A função F4 é contínua, convexa, unimodal, quadrática, de alta dimensão (30 variáveis – x_i), com ruído Gaussiano. A função GAUSS devolve um valor aleatório compreendido entre 0 e 1 que segue uma distribuição normal de Gauss. A sua representação gráfica encontra-se no ANEXO 1, Figura 4.

O máximo da função (sem ruído gaussiano) coincide com o valor **1248.20**.

Nos testes realizados **F4** foi restringida ao espaço definido por $-1{,}28 \leq x_i \leq 1{,}28$, onde i = 1, 2, ..., 30, com uma precisão de duas casas decimais.

Em função destas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 2,56
- Número de bits para x_i: 2,56 * 100 = 256 ➪ 2^8 = 256 ➪ 8 bits por variável
- Tamanho do cromossoma = Dimensão * 8 = 30 * 8 = **240 bits**

F5 - De Jong Test Bed Function

A função F5 de De Jong é definida por:

$$1/F_5(x_i) = \frac{1}{K} + \sum_{i=1}^{2} \frac{1}{f_j(x)} \qquad (5.5.)$$

onde

$$f_j(x_i) = c_j + \sum_{i=1}^{2} \left(x_i - a_{ij}\right)^6$$

e

$$a_{ij} = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & \dots & 32 & 32 & 32 \end{bmatrix}$$

Características da função

A função F5 é contínua, não convexa, multimodal, não quadrática e de baixa dimensão (2 variáveis – x_i). Estas características podem ser observadas na Figura 5 do ANEXO 1.

O máximo da função coincide com o valor **500**.

Nos testes realizados, utilizámos, tal como sugerido por De Jong, K = 500 e c_j = j. **F_5** foi restringida ao espaço definido por $-65{,}536 \leq x_i \leq 65{,}536$, onde i = 1, 2, , com uma precisão de três casas decimais.

Com estas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 131,072
- Número de bits para x_i: 131,072 * 1.000 = 131.072 ➪ 2^{17} = 131.072 ➪ 17 bits por variável
- Tamanho do cromossoma = Dimensão * 17 = 2 * 17 = **34 bits**

5.1.2. Conjunto de teste suplementar

Para complementar o conjunto de teste de De Jong, com funções que abrangessem outras características formou-se um conjunto de teste suplementar.

F6 - Função de Michalewicz

A função de Michalewicz (F6) é definida por:

$$F_6(x_1, x_2) = 21.5 + x_1.\sin(4Px_1) + x_2.\sin(20Px_2) \qquad (5.6)$$

Características da função

A função de Michalewicz é contínua, não convexa, multimodal, não quadrática e de baixa dimensão (2 variáveis – x_i). Esta função encontra-se representada graficamente na Figura 6 do ANEXO 1.

O máximo da função coincide com o valor **38,85**.

Nos testes realizados, a função **F_6** foi restringida ao espaço **A_1 e A_2** definidos por

$A_1 = -3{,}0 \leq x_1 \leq 12{,}1$

$A_2 = 4{,}1 \leq x_2 \leq 5{,}8$

com uma precisão de quatro casas decimais.

Partindo destas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_1: 15,1
- Tamanho do domínio para x_2: 1,7
- Número de bits para x_1: 15,1* 10.000 = 151.000 ➪ 2^{18} = 262.144 ➪ 18 bits
- Número de bits para x_2: 1,7* 10.000 = 17.000 ➪ 2^{15} = 32.768 ➪ 15 bits
- Tamanho do cromossoma = 18 + 15 = **33 bits**

F7 – Função de Bohachevsky I

A função de Bohachevsky I é definida por:

$$F_7(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3Px_1) - 0.4\cos(4Px_2) + 0.7 \qquad (5.7)$$

Características da função

A função F7 é contínua, não convexa, multimodal, quadrática e de baixa dimensão (2 variáveis – x_i). A representação gráfica desta função encontra-se no ANEXO 1, Figura 7.

O máximo da função coincide com o valor **7500**.

Nos testes realizados, a função $\mathbf{F_7}$ foi restringida ao espaço **A** definido por -50,0 ≤ x_i £ 50,0, onde i = 1, 2 e com uma precisão de três casas decimais.

Em função destas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 100
- Número de bits para x_i: 100* 1.000 = 100.000 ➪ 2^{17} = 131.072 ➪ 17 bits
- Tamanho do cromossoma = Dimensão * 17 = 2*17 = **34 bits**

F8 – Função de Bohachevsky II

A função de Bohachevsky 2 é definida por:

$$F_8(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3[\cos(3Px_1)0.4\cos(4Px_2)] + 0.3 \qquad (5.8)$$

Características da função

A função F8 é contínua, não convexa, multimodal, quadrática e de baixa dimensão (2 variáveis – x_i). Estas características podem ser observadas na sua representação gráfica - ANEXO 1, Figura 8.

O máximo da função coincide com o valor **7500**.

Nos testes realizados, a função $\mathbf{F_8}$ foi restringida ao espaço **A** definido por -50,0 ≤ x_i £ 50,0, onde i = 1,2 e com uma precisão de três casas decimais.

Com estas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 100
- Número de bits para x_i: 100* 1.000 = 100.000 ➪ 2^{17} = 131.072 ➪ 17 bits
- Tamanho do cromossoma = Dimensão * 17 = 2*17 = **34 bits**

F9 – Função de Griewangk

A função de Giewangk é definida por:

$$F_9(x_i) = 1 + \sum_{i=1}^{n}\left[\frac{x_i^{\,2}}{4000}\right] - \prod_{i=1}^{n}\left[\cos\left(\frac{x_i}{\sqrt{i}}\right)\right] \qquad (5.9)$$

Características da função

A função F9 é contínua, unimodal, de dimensão escalonável, convexa e quadrática. A versão de duas varáveis (**n** = 2) desta função encontra-se na Figura 9 do ANEXO 1.

Nos testes realizados, a função $\mathbf{F_9}$ foi restringida ao espaço definido por -600,0 ≤ x_i ≤ 600,0, onde i = 1, ... **n** e com uma precisão 3 casas decimais.

Em função destas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 1.200
- Número de bits para x_i: 1.200 * 1.000 = 1.200.000 ➪ 2^{21} = 2.097.152 ➪ 21 bits por variável
- Tamanho do cromossoma = **n** * 21 bits.

- Como utilizámos **n** = 10, Tamanho do cromossoma = 10 * 21 = **210 bits**

F10 – Função de Rastrigin

A função de Rastrigin é definida por:

$$F_{10}(x_i) = n * A + \sum_{i=1}^{n} \left[x_i^{\,2} - A * cos(2Px_i) \right] \qquad (5.10)$$

Com A = 10,0.

Características da função

A função F10 é contínua, de dimensão escalonável, multimodal, não convexa e quadrática. A representação desta função, com **n** = 2, encontra-se na Figura 10 do ANEXO 1.

Nos testes realizados, a função $\mathbf{F_{10}}$ foi restringida ao espaço definido por $5{,}12 \leq x_i \leq 5{,}12$, onde i = 1, ..., **n** e com uma precisão de três casas decimais.

Com estas características o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 10,24
- Número de bits para x_i: 10,24 * 1.000 = 10.240 ➪ 2^{14} = 16.384 ➪ 14 bits por variável
- Tamanho do cromossoma = **n** * 14 bits
- Como utilizámos **n** = 20, Tamanho do cromossoma = 20 * 14 = **280 bits**

F11 - Função de Schwefel (Sine Root)

A função de Schwefel é definida por:

$$F_{11}(x_i) = V * n + \sum_{i=1}^{n} \left[-x_i * sin\left(\sqrt{|xi|}\right) \right] \qquad (5.11)$$

com V = 418,9829

Características da função

A função F11 é contínua, unimodal, de dimensão escalonável, não convexa e não quadrática (Figura 11 do ANEXO 1).

Nos testes realizados, a função $\mathbf{F_{11}}$ foi restringida ao domínio definido por $-500{,}0 \leq x_i \leq 500{,}0$, onde i = 1, ... **n** e com uma precisão três casas decimais.

Partindo destas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 1.000;
- Número de bits para x_i: 1.000 * 1.000 = 1.000.000 ➪ 2^{20} = 1.048.576 ➪ 20 bits por variável;
- Tamanho do cromossoma = **n** * 20 bits;
- Como utilizámos **n** = 10, Tamanho do cromossoma = 10 * 20 = **200 bits**

5.2. Problema de Minimização

Para o problema de minimização, seleccionámos um conjunto de funções propostas por [Koon et al. 1995]. Os autores sugerem um vasto número de funções, apropriadas para testar estratégias evolucionárias. Das funções utilizadas pelos autores no trabalho seleccionámos um total de sete funções:

- F12 - Função 6-Hump CamelBack
- F13 - Função de Shubert
- F14 - Função de teste N-Dimensional (N = 1, 2, 3, 4)
- F15 - Função de teste Bi-Dimensional I
- F16 - Função de teste Bi-Dimensional II (N = 1, 2, 3, 4)
- F17 - Função de teste Bi-Dimensional de Rastrigin
- F18 - Função de teste Uni-Dimensional

5.2.1. Conjunto de teste de Koon

F12 - Função 6 Hump CamelBack

A função 6-Hump Camelback é definida por:

$$F_{12}(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + \left(-4 + 4x_2^2\right)x_2^2 \qquad (5.12)$$

Características da função

A função **F12** é contínua, não convexa, multimodal (6 mínimos locais), quadrática e de baixa dimensão (2 variáveis – x_i). Esta função encontra-se representada graficamente na Figura 12 do ANEXO 1.

O mínimo da função coincide com o valor **–1,0320**.

Nos testes realizados, a função $\mathbf{F_{12}}$ foi restringida aos seguintes domínios: $-3{,}0 \leq x_1 \leq 3{,}0$ e $-2{,}0 \leq x_2 \leq 2{,}0$ e com uma precisão quatro casas decimais.

Partindo destas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 10,0;
- Número de bits para x_i: 10,0* 10.000 = 100.000 ➪ 2^{17} = 131.072 ➪ 17 bits por variável;
- Tamanho do cromossoma = Dimensão*17 = 2* 17 = **34 bits**

F13 - Função de Shubert

A função de Shubert é definida por:

$$F_{13}(x_1, x_2) = \left[\sum_{i=1}^{5} i\cos[(i+1)x_1 + i]\right] * \left[\sum_{i=1}^{5} i\cos[(i+1)x_2 + i]\right] \qquad (5.13)$$

Características da função

A função de Shubert é contínua, não convexa, multimodal (760 mínimos locais), não quadrática e de baixa dimensão (2 variáveis – x_i). A sua representação gráfica corresponde à Figura 13 do ANEXO 1.

O mínimo da função coincide com o valor **-186,731**.

Nos testes realizados, a função $\mathbf{F_{13}}$ foi restringida ao espaço definido por $-5{,}0 \leq x_i \leq 5{,}0$, onde i = 1, 2 e com uma precisão quatro casas decimais.

Em função destas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 10,0;
- Número de bits para x_i: 10,0* 10.000 = 100.000 ➪ 2^{17} = 131.072 ➪ 17 bits por variável;

- Tamanho do cromossoma = Dimensão * 17 = 2 * 17 = **34 bits**.

F14 - Função de teste N-dimensional

A função de teste N–dimensional é definida por:

$$F_{14}(x_i) = \frac{1}{2}\sum_{j=1}^{N}(x_i^4 - 16x_i^2 + 5x_i) \qquad (5.14)$$

Características da função

A função F14 é contínua, não convexa, multimodal (2^N mínimos locais), quadrática e de dimensão escalonável (**N** variáveis - x_i). A função com N = 2 encontra-se representada na Figura 14 do ANEXO 1.

Para esta função variámos o parâmetro **N** de 1 até 4.

O mínimo da função depende do valor de **N** e coincide com os seguintes valores:

N = 1 ➪ Mínimo: **-39,1660**;

N = 2 ➪ Mínimo: **-78,3320**;

N = 3 ➪ Mínimo: **-117,4980**;

N = 4 ➪ Mínimo: **-156,6650**;

Nos testes realizados, a função $\mathbf{F_{14}}$ foi restringida ao espaço definido por $-20{,}0 \leq x_i \leq 20$, onde i = 1, 2, ..., N (no nosso caso, ***N = 4***) e com uma precisão quatro casas decimais.

Partindo destas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 40,0;
- Número de bits para x_i: 40,0* 10.000 = 400.000 ➪ 2^{19} = 524.288 ➪ 19 bits por variável;
- Tamanho do cromossoma = N * 19 bits;
  - ✓ **N = 1** ➪ Tamanho do cromossoma = **19 bits**;
  - ✓ **N = 2** ➪ Tamanho do cromossoma = **38 bits**;
  - ✓ **N = 3** ➪ Tamanho do cromossoma = **57 bits**;
  - ✓ **N = 4** ➪ Tamanho do cromossoma = **76 bits**.

F15 – Função de teste bidimensional I

A função de teste bidimensional I é definida por:

$$F_{15}(x_1, x_2) = 0.5x_1^2 + 0.5[1 - cos(2x_1)] + x_2^2 \qquad (5.15)$$

Características da função

A função de teste bidimensional I é contínua, convexa, multimodal, quadrática e de baixa dimensão (2 variáveis – x_i). A sua representação gráfica encontra-se na Figura 15 do ANEXO 1.

O mínimo da função coincide com o valor **0,0**.

Nos testes realizados, a função $\mathbf{F_{15}}$ foi restringida ao espaço definido por $-5{,}0 \leq x_i \leq 5{,}0$, onde i = 1, 2 e com uma precisão quatro casas decimais.

Em função destas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 10,0;
- Número de bits para x_i: 10,0* 10.000 = 100.000 ➪ 2^{17} = 131.072 ➪ 17 bits por variável;
- Tamanho do cromossoma = Dimensão * 17 = 2* 17 = **34 bits**.

F16 – Função de teste bidimensional II

A função de teste di-dimensional II (F16) é definida por:

$$F_{16}(x_1, x_2) = 10^n x_1^2 + x_2^2 - \left(x_1^2 + x_2^2\right)^2 + 10^m \left(x_1^2 + x_2^2\right)^4 \qquad (5.16)$$

com **n = -m.**

Características da função

A função de teste bidimensional II é contínua, convexa, multimodal (> 3 mínimos locais), quadrática e de baixa dimensão (2 variáveis – x_i). Ver Figura 16 do ANEXO 1.

Nesta função variámos o valor de **n** de 1 até 4. O mínimo da função depende do valor de **n** e coincide com os seguintes valores:

n = 1 ➪ Mínimo: **-0,407**;

n = 2 ➪ Mínimo: **-18,059**;

n = 3 ➪ Mínimo: **-227,766**;

n = 4 ➪ Mínimo: **-2429,415**;

Nos testes realizados, a função $\mathbf{F_{16}}$ foi restringida aos seguintes domínios, dependendo de **n**:

n = 1, 2 ➪ $-5{,}0 \leq x_i \leq 5{,}0$, onde i = 1, 2 e com uma precisão de quatro casas decimais;

n = 3, 4 ➪ $-20{,}0 \leq x_i \leq 20{,}0$, onde i = 1, 2 e com uma precisão de quatro casas decimais.

Considerando estas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

n = 1, 2:

Tamanho do domínio para x_i: 10,0;

Número de bits para x_i: 10,0* 10.000 = 100.000 ➪ 2^{17} = 131.072 ➪ 17 bits por variável;

Tamanho do cromossoma = Dimensão * 17 = 2 * 17 = **34 bits**.

n = 3, 4:

Tamanho do domínio para x_i: 40,0;

NÚMERO de bits para x_i: 40,0* 10.000 = 400.000 ➪ 2^{19} = 524.288 ➪ 19 bits por variável;

Tamanho do cromossoma = Dimensão * 19 = 2 * 19 = **38 bits**.

F17 – Função bidimensional de Rastrigin

A função bidimensional de Rastrigin é definida por:

$$F_{17}(x_1, x_2) = x_1^2 + x_2^2 - cos(18x_1) - cos(18x_2) \qquad (5.17)$$

Características da função

A função bidimensional de Rastrigin é contínua, não convexa, multimodal (> 50 mínimos locais), quadrática e de baixa dimensão (2 variáveis – x_i). A sua representação gráfica encontra-se no ANEXO 1, Figura 17.

O mínimo da função coincide com o valor **–2,0**.

Nos testes realizados, a função $\mathbf{F_{17}}$ foi restringida ao domínio definido por $-5{,}0 \leq x_i \leq 5{,}0$, onde i = 1, 2 e com uma precisão quatro casas decimais.

Em função destas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x_i: 10,0;
- Número de bits para x_i: 10,0* 10.000 = 100.000 ➪ 2^{17} = 131.072 ➪ 17 bits por variável;
- Tamanho do cromossoma = Dimensão * 17 = 2 * 17 = **34 bits**.

F18 – Função de teste unidimensional

A função de teste unidimensional é definida por:

$$F_{18}(x) = \sum_{i=1}^{5} sin[(i+1)x + i] \qquad (5.18)$$

Características da função

A função de teste unidimensional é contínua, não convexa, multimodal (20 mínimos locais), não quadrática e de baixa dimensão (1 variável). O seu gráfico corresponde à Figura 18 do ANEXO 1.

O mínimo da função coincide com o valor **–3,3730**.

Nos testes realizados, a função $\mathbf{F_{18}}$ foi restringida ao domínio definido por $-20{,}0 \leq x \leq 20{,}0$, e com uma precisão quatro casas decimais.

Considerando estas características, o tamanho do cromossoma utilizado para codificar esta função foi encontrado a partir dos seguintes cálculos:

- Tamanho do domínio para x: 40,0;

- Número de bits para *x*: 40,0* 10.000 = 400.000 ➪ 2^{19} = = 524.288 ➪ 19 bits por variável;
- Tamanho do cromossoma = Dimensão * 19 = 1* 19 = **19 bits**.

5.3. Resumo das características das funções

A tabela seguinte resume as características das dezoito funções de teste. Uma cruz na coluna correspondente a uma determinada característica indica que a função em causa a possui. As características focadas estão na tabela ordenadas da seguinte forma: contínua e descontínua, convexa e não convexa, quadrática e não-quadrática, baixa, alta e dimensão escalonável, unimodal e multimodal.

	Características										
Função	*Cont.*	*Descont.*	*Conv.*	*Não Conv.*	*Quad*	*Não Quad*	*Baixa dim*	*Alta dim*	*Esc.*	*Unim*	*Multim*
F1	**X**		**X**		**X**		**X**			**X**	
F2	**X**			**X**	**X**		**X**			**X**	
F3		**X**		**X**		**X**	**X**			**X**	
F4	**X**		**X**		**X**			**X**		**X**	
F5	**X**			**X**		**X**	**X**				**X**
F6	**X**			**X**		**X**	**X**				**X**
F7	**X**			**X**	**X**		**X**				**X**
F8	**X**			**X**	**X**		**X**				**X**
F9	**X**		**X**		**X**				**X**	**X**	
F10	**X**			**X**	**X**				**X**		**X**
F11	**X**			**X**		**X**			**X**	**X**	
F12	**X**			**X**	**X**		**X**				**X**
F13	**X**			**X**		**X**	**X**				**X**
F14	**X**			**X**	**X**				**X**		**X**
F15	**X**		**X**		**X**		**X**				**X**
F16	**X**		**X**		**X**		**X**				**X**
F17	**X**			**X**	**X**		**X**				**X**
F18	**X**			**X**		**X**	**X**			**X**	

Tabela 5. 1 - ***Resumo das características das funções de teste***

Relativamente às restantes características, como sejam a precisão do eixo utilizada e o número de variáveis para as funções escalonáveis, a sua escolha deveu-se a alguns critérios. Por um lado, a escolha de um diferente número de casas decimais para

as diversas funções foi propositada. Essa escolha foi feita de forma a que os cromossomas codificadores das variáveis em causa possuíssem tamanhos diferentes. Pensámos, tal como se veio a verificar, que a escolha do tamanho das sequências flanqueadoras poderia estar relacionada com o tamanho do cromossoma. Assim, existindo cromossomas de diferentes tamanhos, o estudo dessa relação foi mais objectivo.

Por outro lado, a escolha da dimensão das funções escalonáveis foi feita tentando estabelecer um compromisso entre a complexidade da função a testar e o tamanho do cromossoma resultante. Por exemplo, na função de *Rastrigin* (F10) escolhemos **n=20**, visto que um valor superior tornaria a cadeia binária muito longa para os nossos recursos computacionais.

5.4. O Algoritmo Genético utilizado

De forma a comparar o desempenho dos operadores genéticos tradicionais com o do novo mecanismo proposto utilizou-se um algoritmo genético tradicional, tal como foi descrito no capítulo 2. A única alteração feita no AG, consistiu na introdução do mecanismo de transposição como alternativa ao operador de *crossover*.

O AG foi utilizado como optimizador de funções, pelo que, surgiram algumas restrições na escolha do mecanismo de selecção. Relativamente às probabilidades de *crossover* e mutação, utilizaram-se valores de referência utilizados por diversos autores, nomeadamente [De Jong 1975].

Todos estes aspectos serão abordados de seguida.

5.4.1. O AG e o mecanismo de transposição

O AG tradicional utiliza uma população de indivíduos que evoluem ao longo de gerações através do princípio de selecção natural. Para que este princípio possa actuar é necessário que a diversidade da população seja mantida, e que a descendência seja gerada a partir dos melhores indivíduos. Os operadores genéticos de *crossover* são responsáveis por manter essa diversidade, através do rearranjo do material genético de 2 indivíduos previamente seleccionados de acordo com a sua qualidade. Este processo resume-se com o esquema da figura 5.1, que se retoma da secção 2.1.

1. Gera população inicial
2. ENQUANTO *não terminar* FAZ
    2.1 Avalia população
    2.2 Selecciona progenitores para reprodução
    2.3 Gera descendência: aplicação do operador de *crossover*
    2.4 Aplica operador de mutação
    2.5 Substitui população antiga pela descendência gerada
FIM ENQUANTO

Figura 5. 1 – ***O Algoritmo Genético Clássico***

A utilização do mecanismo de transposição tem como objectivo o cruzamento do material genético, tal com o operador de *crossover*.

Assim, na experimentação realizada, o operador de *crossover* será substituído pelo mecanismo de transposição, exactamente na mesma posição do AG (2.3).

5.4.2. Mecanismo de selecção

A maneira clássica de utilizar o AG como optimizador de funções consiste em manter o melhor indivíduo encontrado globalmente, independentemente do melhor indivíduo da geração actual [De Jong 1993]. Portanto, a população é vista como uma base de dados de amostras a partir das quais se podem obter melhores indivíduos. Os operadores genéticos em conjugação com o mecanismo de selecção permitem preservar a diversidade da população, gerando indivíduos de melhor qualidade.

Para que o melhor indivíduo, encontrado num determinado momento do processo evolutivo, não se perca ao longo das gerações, é necessário utilizar um mecanismo de selecção que preserve as melhores soluções atingidas. Uma forma de manter estes indivíduos consiste num mecanismo de selecção baseado na ordenação dos indivíduos segundo a sua qualidade [De Jong 1993]. Ao manter a população ordenada de acordo com o seu mérito e ao seleccionar os progenitores de acordo com esta ordenação, consegue-se manter um diferencial constante entre os melhores e o piores indivíduos na população [Whitley 1989].

Frequentemente, a melhoria do desempenho de um AG ao longo das gerações é conseguida através de métodos de selecção elitistas, onde o(s) melhor(es) indivíduo(s)

encontrado(s) até um determinado momento são preservado e passados para a próxima geração. Este método, apesar de garantir que a melhor solução encontrada não se perde devido aos efeitos dos operadores genéticos, não havendo conhecimento *a priori* das funções a optimizar, pode conduzir o AG a máximos locais não permitindo que este atinja o valor óptimo da função [De Jong 1993].

Entre estas duas alternativas, nas nossas experiências utilizámos, em todos os casos, o método de **selecção da roleta com elitismo**. A tamanho da elite escolhida correspondeu a 20% do tamanho da população utilizada em cada experimentação.

5.4.3. Probabilidades dos operadores genéticos

De Jong no seu trabalho de 1975 [De Jong 1975], realizou um estudo aprofundado sobre o efeito da variação de determinados parâmetros nomeadamente a probabilidade de mutação e de *crossover*. De Jong verificou que a melhor opção, na maioria dos casos, consiste em escolher uma probabilidade de mutação igual a ao inverso do tamanho da população.

Ao variar a probabilidade de mutação no plano reprodutivo **R1** verificou que aumentando a probabilidade de mutação (a partir de 0.001) o desempenho do AG deteriorava-se devido à perda de alelos. No entanto, uma probabilidade muito alta também degradava os resultados alcançados pelo algoritmo. Para uma população de 50 indivíduos, uma probabilidade de 0.02 e 0.01 permitiu obter os melhores resultados no desempenho em diferido.

No estudo da variação do efeito da probabilidade de *crossover*, [De Jong 1975] efectuou um estudo envolvendo ***Pc*** = 0.8, 0.6 e 0.4, para uma população de 50 indivíduos.

Verificou que ao diminuir a probabilidade de mutação de 1 até 0,4 inicialmente o desempenho do AG melhorou, mas com ***Pc*** = 0.4 o seu desempenho degradou-se. Os melhores resultados foram obtidos com ***Pc*** = 0.6 e 0.8.

No nosso estudo, pretendemos utilizar a mesma probabilidade de mutação e *crossover* em todos os casos, independentemente do tamanho da população, para que apenas pudéssemos estudar as diferenças produzidas pelos operadores genéticos.

Baseando-nos no trabalho de De Jong optámos por um compromisso e escolhemos uma **probabilidade de mutação** igual a **0.01** e uma **probabilidade de *crossover* (ou transposição)** igual a **0.7**.

5.4.4. Tamanho da população

Para verificar o efeito que o tamanho da população teria quando se substituiu o operador de *crossover* pelo mecanismo de transposição, utilizámos a mesma variação que [De Jong 1975] usou, isto é , populações de 50, 100 e 200 indivíduos.

5.4.5. Método de inserção

O processo de inserção consiste na colocação da descendência criada na população. Existem dois tipos básicos de técnicas inserção: técnica de **não-sobreposição** e técnica de **sobreposição** [De Jong 1975]. Na primeira opção, nenhum indivíduo da população anterior pode ser introduzido na próxima geração. Por outro lado, a técnica de substituição permite que indivíduos da população anterior façam parte da nova geração. Esta técnica pode ser implementada de diversas formas, sendo a mais comum a substituição de um número específico dos indivíduos da população (os piores) por novos indivíduos gerados.

No nosso estudo utilizámos o **método de inserção por sobreposição**. Isto é, em cada geração foram mantidos os melhores indivíduos da geração anterior e os restantes criados a partir da reprodução dos progenitores seleccionados.

5.4.6. Avaliação do desempenho do AG

Para comparar o desempenho do AG quando utilizado com os operadores de *crossover* e transposição utilizou-se a medida de desempenho *em diferido*. A medida de *em diferido* consiste na média dos melhores indivíduos em cada geração. Executaram-se 10 iterações para cada geração e a média das 10 simulações foi calculada através do somatório dos valores máximos (ou mínimos) obtidos no total das simulações e dividindo esta soma por 10.

A formula 5.19. traduz o cálculo do desempenho do AG para a geração **g** quando se executam **T** simulações [De Jong 1975]. No nosso caso, T = 10.

$$X^{*}{}_{e}(g) = \frac{1}{T} * \sum_{t=1}^{T} f_{e}{}^{*}(t) \qquad (5.19)$$

T é igual ao número de simulações executadas e $\mathbf{f^{*}_{e}(t)}$ o melhor indivíduo encontrado em cada simulação.

Porquê a escolha desta medida de desempenho em vez da medida **em tempo real**? Esta escolha deveu-se a duas razões essenciais. Por um lado, a carga computacional que o trabalho experimental realizado já em si representou seria profundamente agravado se as mesmas experiências tivessem que ser repetidas com a medida em tempo real. Isto limitou a nossa escolha a apenas uma das medidas. Escolhemos a medida em diferido, uma vez que, no estudo comparativo realizado foi mais importante avaliar a qualidade das soluções encontradas do que a velocidade de convergência. Por outro lado, [De Jong 1975] afirma que *o desempenho em diferido está mais próximo da medida de avaliação utilizada nos optimizadores de funções*. Uma vez que utilizámos o domínio da optimização de funções para estudar o nosso problema, a escolha entre as duas medidas de avaliação estava feita.

5.4.7. Tamanho das sequências flanqueadoras

O intervalo de variação das sequências flanqueadoras dependeu do tamanho do cromossoma que codificou cada função. O intervalo variou sempre entre 1 e um valor máximo que foi escolhido de acordo com a dimensão do cromossoma. Assim, para cromossomas de "pequena" dimensão o valor máximo do intervalo de variação foi 8 ou 10. Para cromossomas de "média" dimensão o tamanho máximo escolhido foi 15 e para cromossomas de "grande" dimensão o tamanho máximo escolhido foi 20. A tabela 5.1. resume, para cada função, o tamanho do cromossoma utilizado e o limite superior do intervalo de variação das sequências flanqueadoras.

		Função	*Tamanho do cromossoma*	*Tamanho máximo das sequências flanqueadoras*
MAXIMIZAÇÃO	F1	De Jong F1	30	10
	F2	De Jong F2	25	10
	F3	De Jong F3	50	15
	F4	De Jong F4	240	20
	F5	De Jong F5	34	10
	F6	Michalewicz	33	10
	F7	Bohachevsky I	34	10
	F8	Bohachevsky II	34	10
	F9	Griewangk (N = 10)	210	20
	F10	Rastrigin (N = 20)	280	20
	F11	Schwefel (N = 10)	200	20
MINIMIZAÇÃO	F12	6-Hump CamelBack	34	10
	F13	Shubert	34	10
	F14	N-Dimensional (N = 1)	19	8
	F14	N-Dimensional (N = 2)	38	10
	F14	N-Dimensional (N = 3)	57	15
	F14	N-Dimensional (N = 4)	74	20
	F15	Bi-Dimensional I	34	10
	F16	Bi-Dimensional II (N = 1)	34	10
	F16	Bi-Dimensional II (N = 2)	34	10
	F16	Bi-Dimensional II (N = 3)	38	10
	F16	Bi-Dimensional II (N = 4)	38	10
	F17	Bi-Dimensional de Rastrigin	34	10
	F18	Uni Dimensional	19	8

Tabela 5. 2 – ***Relação entre o tamanho do cromossoma e o tamanho máximo das sequências flanqueadoras***

5.5. Plano da experimentação realizada

Como já foi referido, utilizámos um total de dezoito funções de teste que serviram de termo de comparação para o desempenho do AG quando usado com os diferentes operadores genéticos. Efectuámos um estudo experimental vasto que permitisse a comparação dos resultados, quer operador a operador, quer ao nível da variação dos tamanhos da população e das sequências flanqueadoras.

Assim, **para cada função** realizámos cinco séries experimentais e, dentro de cada série alterámos alguns parâmetros. As série foram as seguintes:

Série 1: operador de *crossover* com 1 ponto de corte;

Série 2: operador de *crossover* com 2 pontos de corte;

Série 3: operador de *crossover* uniforme;

Série 4: operador de transposição simples;

Série 5: operador de transposição baseada em torneio.

De seguida apresentam-se quais as variações efectuadas em cada uma das séries realizadas.

5.5.1. Série 1 - Operador de crossover com 1 ponto de corte

Experiência 1.1. - População de 50 indivíduos
Experiência 1.2. - População de 100 indivíduos
Experiência 1.3. - População de 200 indivíduos

5.5.2. Série 2 - Operador de crossover com 2 pontos de corte

Experiência 2.1. - População de 50 indivíduos
Experiência 2.2. - População de 100 indivíduos
Experiência 2.3. - População de 200 indivíduos

5.5.3. Série 3 - Operador de crossover uniforme

Experiência 3.1. - População de 50 indivíduos
Experiência 3.2. - População de 100 indivíduos
Experiência 3.3. - População de 200 indivíduos

5.5.4. Série 4 – Mecanismo de transposição simples

Em cada uma das experiências, o tamanho das sequências flanqueadoras foi sempre variado de 1 até ao máximo.

Experiência 4.1. - População de 50 indivíduos
Experiência 4.2. - População de 100 indivíduos
Experiência 4.3. - População de 200 indivíduos

5.5.5. Série 5 – Mecanismo de transposição baseado em torneio

Em cada uma das experiências, o tamanho das sequências flanqueadoras foi sempre variado de 1 até ao máximo.

Experiência 5.1. - População de 50 indivíduos
Experiência 5.2. - População de 100 indivíduos
Experiência 5.3. - População de 200 indivíduos

Devido ao grande número de experiências executadas com cada função de teste, a quantidade de resultados produzidos foi muito elevada. Após uma análise cuidada de toda essa informação, seleccionámos os resultados que melhor permitem comparar os mecanismos genéticos em estudo. A apresentação e análise desses resultados encontra-se no próximo capítulo.

CAPÍTULO 6.
ANÁLISE DOS RESULTADOS

Sumário

Neste capítulo faremos uma análise detalhada dos resultados obtidos em todos os casos, para cada uma das funções de teste, atrás descritas.

Inicialmente, estudaremos a transposição isoladamente, de forma a compreender os factores que influenciam o seu desempenho. De seguida, far-se-á a comparação dos mecanismos propostos com os diferentes operadores genéticos tradicionais.

Devido às enormes quantidades de informação produzidas com o trabalho experimental realizado (cerca de 3 Gigabytes de informação), procedeu-se a uma selecção dos dados que melhor traduzissem os resultados em análise. Assim, para cada função de teste, proceder-se-á a uma comparação entre os melhores e os piores resultados alcançados com a transposição. Serão analisados por um lado, os valores máximos (ou mínimos) alcançados para cada uma das funções e o valor médio calculado a partir dos melhores resultados em todas as gerações simuladas. Será sobre esta informação que se mostrarão os respectivos gráficos e tabelas.

A análise dos resultados remeterá frequentemente o leitor para a observação dos gráficos que se encontram nos ANEXOS 2, 3, 4 e 5.

6.1. O Mecanismo de Transposição

Todas as experiências realizadas fizeram variar o tamanho das sequências flanqueadoras de 1 até um tamanho máximo que dependeu do tamanho do cromossoma utilizado para codificar a função em causa. Tal como se previa, os resultados obtidos são dependentes do tamanho dessas sequências.

Além disso, utilizámos populações de 50, 100 e 200 indivíduos e, mais uma vez, este factor influenciou os resultados obtidos.

Para flanqueadores de determinado tamanho, o mecanismo de transposição permitiu obter resultados superiores aos operadores de *crossover*. Para que o uso deste novo mecanismo se torne viável, é necessário, no entanto, que tenhamos alguma informação sobre a forma como escolher o tamanho das sequências para cada caso especifico. Se assim for, não será necessário executar a variação do tamanho das sequências de 1 até um tamanho máximo, mas será possível restringir este intervalo. Desta forma, será poupado muito tempo de CPU, de tempo despendido na análise comparativa de dados, ao mesmo tempo que se economiza o espaço em disco utilizado.

Para conseguir este objectivo, faremos uma análise das sequências flanqueadoras que permitiram obter os melhores resultados em cada uma das funções de teste. Pretendemos assim chegar a uma regra que nos diga, para cada caso, qual(ais) o(s) tamanho(s) da sequência que dará(ão) melhores resultados.

6.1.1. Influência do Tamanho da População

Tal como já foi referido, no trabalho experimental realizado para cada uma das funções de teste, variámos o número de indivíduos da população: de 50, 100 e 200 indivíduos. De uma forma geral, populações com mais indivíduos originaram soluções de melhor qualidade. Utilizando populações de tamanho 200, na maioria dos casos, obtivemos soluções que superaram aquelas obtidas com qualquer um dos tipos de *crossover*. Nas secções 6.2, 6.3 e 6.4 estes resultados serão referidos com mais detalhe.

Os gráficos 1, 2, 3, 4 do ANEXO 2, referem-se aos melhores resultados obtidos nas funções de De Jong e ilustram o efeito da variação do tamanho da população.

Para as restantes funções verificou-se uma generalização deste comportamento.

6.1.2. Influência do Tamanho das Sequências Flanqueadoras

Como já referido no capítulo anterior, a variação do tamanho das sequências flanqueadoras dependeu do tamanho do cromossoma utilizado para cada função.

Não é possível indicar uma regra genérica que antecipe o comportamento do AG como consequência da variação destes tamanhos. De uma forma geral, para o mecanismo de transposição, sequências menores (3, 4, 5) produzem melhores resultados do que sequências de maior tamanho. No mecanismo de transposição/torneio, as melhores soluções são obtidas com sequências de tamanho médio entre 6 e 7.

Os gráficos 5, 6, 7, 8 do ANEXO 2, ilustram, para as funções de De Jong, os efeitos produzidos com a variação do tamanho das sequências.

Ao estudar a forma como varia o tamanho do transposão, constatámos que à medida que se aumenta o tamanho das sequências flanqueadoras, o tamanho médio do transposão aumenta. O número de vezes em que o transposão encontrado coincide com o cromossoma inicial e, portanto, não ocorre mecanismo de transposição também aumenta.

A tabela 6.1.1 refere-se à função de teste ***F1*** de De Jong e mostra numericamente a variação destes dois aspectos: o tamanho médio do transposão e o número de vezes em que este coincide com o tamanho do cromossoma. Neste caso o cromossoma tem tamanho 30 e verifica-se que, ao aumentar-se o tamanho da sequência flanqueadora, este tamanho vai também aumentado. Por outro lado, o número de vezes (mostrado em %) em que não ocorre transposição, também aumenta.

Tamanho das Sequências	**Tamanho Médio do transposão**	**% média tamanho = 30 (não ocorre transposição)**
1	2,65	0,40%
2	5,47	0,53%
3	7,09	3,22%
4	9,67	7,53%
5	14,73	17,50%
6	18,75	24,20%
7	21,26	31,99%
8	19,55	36,54%
9	20,46	39,12%
10	24,59	57,12%

*Tabela 6.1.1- **Efeitos da variação do tamanho dos flanqueadores na função de teste F1 de De Jong***

No ANEXO 2, os gráficos 13.1 a 17.2 ilustram as variações efectuadas neste dois aspectos nas funções de teste de De Jong. Para as restantes funções, este mesmo comportamento foi um geral.

Este aspecto permite tirar conclusões acerca da relação entre o tamanho das sequências, a melhor solução e o tamanho médio do transposão que conduz a essa solução. Para o mecanismo de transposição, quando se utilizam sequências de grande dimensão obtêm-se, de uma forma geral, piores soluções. Tamanhos de transposão de dimensão pequena/média (3 a 6), normalmente permite obter melhores soluções.

O facto de quanto maior for o tamanho das sequências flanqueadoras implicar um aumento do número de vezes que não ocorre transposição, pode conduzir a uma perda de diversidade da população. Este factor poderá estar na origem das piores soluções obtidas com sequências de maior tamanho.

Portanto, a escolha do tamanho das sequências flanqueadoras é um factor muito importante, condicionador da qualidade das soluções encontradas. Na secção seguinte, analisaremos com detalhe quais as sequências que originaram melhores soluções. Serão propostas heurísticas para reduzir o intervalo de variação do tamanho das sequências de forma a que as melhores soluções consigam ser encontradas.

6.1.3. Escolha do Tamanho das Sequências Flanqueadoras

A análise dos dados alcançados não permitiu chegar a uma regra que seja aplicável em todos os casos sem qualquer margem de erro. No entanto, através da observação directa dos resultados obtidos, é possível chegar a algumas conclusões que permitem restringir o intervalo de variação do tamanho das sequências flanqueadoras [Simões et al. 1999b].

Os mecanismos de transposição simples e de transposição baseada em torneio não se comportam da mesma maneira. O tamanho das sequências, em cada um dos casos, tem características distintas. Vejamos, então, cada um dos casos separadamente.

Mecanismo de Transposição

O tamanho do cromossoma das diferentes funções de teste varia entre 19 e 280 bits. Dividimos estes tamanhos em três categorias: pequena, média e grande dimensão.

Verificámos que, para os casos com tamanho do cromossoma de pequena dimensão, os melhores resultados obtidos com o mecanismo de transposição variavam em torno de valores pequenos, com uma relação directa com o tamanho do cromossoma. Como se pode observar na tabela 6.1.2, os cromossomas de 19 a 38 incluem-se na categoria de pequena dimensão. Se restringirmos o intervalo para o tamanho das sequências flanqueadoras segundo a seguinte fórmula:

Tamanho da Sequência = 10% * Tamanho do Cromossoma ± 1 (6.1.)

temos a garantia que os melhores resultados serão alcançados. Surgiram apenas duas excepções, na função número 16 (para N = 3 e N = 4). No entanto, se analisarmos qual o tamanho da sequência que produziu os segundos melhores resultados, verificamos que, para o caso de N = 3 estes tamanhos são 3, 4 e 5 e, para N = 4 são 2 e 4. Como se pode observar, pelo menos um destes tamanhos encaixa com os fornecidos pela fórmula 6.1.

Para os cromossomas de média dimensão (de 50 a 74 bits de comprimento), a fórmula teve que ser ajustada para:

Tamanho da Sequência = 5% * Tamanho do Cromossoma ± 1 (6.2.)

Em todos os casos, o tamanho da sequência flanqueadora que originou os melhores resultados foi encontrado com esta fórmula.

Quando o tamanho do cromossoma atinge maiores dimensões, torna-se mais difícil encontrar uma regra que funcione para a maioria dos casos.

Neste caso, utilizando a fórmula

Tamanho da Sequência = 0.3% * Tamanho do Cromossoma ± 1 (6.3.)

apenas obtemos 2 casos positivos e 2 casos negativos (ver tabela 6.1.2). No entanto, podemos observar que, para o mecanismo de transposição, flanqueadores de tamanho menor, de uma forma geral, permitem obter melhores resultados. A média dos tamanhos das sequências que produziram soluções de melhor qualidade para todas as funções é de **3,9**.

Assim, 0.3% do tamanho do cromossoma de grandes dimensões (com uma margem de +-1), permite obter valores que variam entre 5 e 9. Como o tamanho médio das melhores soluções é 3,9, é de esperar que para cromossomas maiores, este tamanho seja um pouco superior. Assim, baseando-nos na estatística dos resultados obtidos, a fórmula 6.3. poderá indicar flanqueadores que conduzam a resultados de boa qualidade. Realmente, observando os flanqueadores que produziram resultados muito próximos dos melhores, verifica-se que os tamanhos estão dentro da margem esperada.

A tabela 6.1.2 sintetiza os resultados obtidos. A informação encontra-se ordenada por ordem crescente do tamanho do cromossoma. Aplicando as respectivas fórmulas, obtemos um intervalo de 3 valores para o tamanho das sequências. Na coluna seguinte, é indicado qual o tamanho que produziu os melhores resultados (quando este valor aparece a negrito/itálico, indica que aplicação da respectiva fórmula não encontrou o valor desejado). Finalmente, na última coluna, são indicados, quais as sequências que produziram resultados muito próximos do melhor.

Função de teste	Tamanho do Cromossoma (TC)	Transposição				
		(10% TC) ± 1			*Melhor*	***» Melhor***
N-Dim. (N = 1)	19	1	2	3	2	1, 3
Uni Dim.	19	1	2	3	3	4, 1, 2
De Jong F2	25	2	3	4	3	5, 2
De Jong F1	30	2	3	4	4	6, 3
Michalewicz	33	2	3	4	4	2, 5
De Jong F5	34	2	3	4	todas	todas
Bohachevsky I	34	2	3	4	3	4, 9
Bohachevsky II	34	2	3	4	3	4, 2
6-Hump CamelBack	34	2	3	4	4	3, 5
Shubert	34	2	3	4	2	4, 3
Bi-Dim. I	34	2	3	4	2	3, 4
Bi-Dim. II (N = 1)	34	2	3	4	3	4, 7
Bi-Dim. II (N = 2)	34	2	3	4	2	7, 3
Bi-Dim. de Rastrigin	34	2	3	4	3	5, 4, 2
N-Dim. (N = 2)	38	3	4	5	3	4, 5
Bi-Dim. II (N = 3)	38	3	4	5	***10***	**3, 4, 5**
Bi-Dim. II (N = 4)	38	3	4	5	***6***	**2, 4**
		(5% TC) ± 1			***Melhor***	***» Melhor***
De Jong F3	50	2	3	4	2	3, 4, 5
N- Dim. (N = 3)	57	2	3	4	4	3, 6
N - Dim. (N = 4)	74	3	4	5	4	6, 3, 5
		(0.3% TC) ± 1			***Melhor***	***» Melhor***
Schwefel (N = 10)	200	5	6	7	***4***	5, 3, 6
Griewangk (N= 10)	210	5	6	7	***8***	7, 6, 4
De Jong F 4	240	6	7	8	***2***	7, 8, 9
Rastrigin (N = 20)	280	7	8	9	9	11, 3
Tamanho Médio das Sequências						***3.9***

Tabela 6.1.2 – ***Síntese dos sequências flanqueadoras que permitiram obter os melhores resultados para o mecanismo de transposição***

Mecanismo de Transposição/Torneio

Para o caso do mecanismo de transposição/torneio, o comportamento da variação do tamanho das sequências flanqueadoras produziu resultados diferentes do caso anterior.

Para este mecanismo, uma divisão do tamanho dos cromossomas em 2 categorias abrange todos os casos analisados. A observação directa dos resultados experimentais,

permitiu distinguir duas categorias com comportamentos distintos. Assim, os cromossomas com tamanhos compreendidos entre 19 e 38 foram agrupados numa categoria. Os restantes (74 a 280) foram agrupados noutra categoria.

Com a utilização deste mecanismo, observa-se que os melhores resultados são obtidos com flanqueadores de tamanho superior, relativamente ao mecanismo anteriormente analisado. De facto, o tamanho médio dos flanqueadores que produziram melhores resultados é **7,13**.

Para a primeira categoria, a regra que maximizou o número de casos positivos foi:

Tamanho da Sequência = 18% * Tamanho do Cromossoma ± 1 (6.4.)

No entanto, surgem alguns casos em que esta fórmula não indicou as sequências que maximizaram a solução. No entanto, se observarmos os resultados mais próximos dos melhores verifica-se que a fórmula, apesar de em alguns casos não indicar a melhor possibilidade, apresenta tamanhos que originaram resultados muito próximos da melhor solução.

Para os cromossomas da segunda categoria, a fórmula foi ajustada para:

Tamanho da Sequência = 5% * Tamanho do Cromossoma ± 1 (6.5.)

Para esta situação, na maioria dos casos, a fórmula apresentou intervalos que abrangem a melhor solução. Na única excepção (função número 14, n=3), a análise dos resultados indica que os tamanhos indicados pela fórmula originam soluções muito próximas das melhores.

A tabela 6.1.3 sintetiza os resultados obtidos e está organizada da mesma forma que a tabela 6.1.2.

Função de Teste	Tamanho do Cromossoma (TC)	Transposição/Torneio				
		(18% TC) ± 1			*Melhor*	***» Melhor***
N-Dim. (N = 1)	19	2	3	4	***6***	**4, 3**
Uni Dim.	19	2	3	4	***5***	**3, 4**
De Jong F2	25	4	5	6	***8***	**6, 4, 5**
De Jong F1	30	4	5	6	6	4, 5
Michalewicz	33	5	6	7	***8***	**3, 4, 5**
De Jong F5	34	5	6	7	todas	todas
Bohachevsky I	34	5	6	7	6	9, 7
Bohachevsky II	34	5	6	7	***9***	**6, 7, 5**
6-Hump CamelBack	34	5	6	7	***8***	**6, 2, 7**
Shubert	34	5	6	7	***4***	**6, 7**
Bi-Dim. I	34	5	6	7	***10***	**7, 5, 6**
Bi-Dim. II (N = 1)	34	5	6	7	6	9, 7
Bi-Dim. II (N = 2)	34	5	6	7	5	3, 8
Bi-Dim. de Rastrigin	34	5	6	7	5	2, 6, 4
N-Dim. (N = 2)	38	6	7	8	6	4, 7
Bi-Dim. II (N = 3)	38	6	7	8	6	9, 8
Bi-Dim. II (N = 4)	38	6	7	8	6	***8, 5***
		(5% TC) ± 1			*Melhor*	***» Melhor***
De Jong F3	50	2	3	4	3	4, 8
N- Dim. (N = 3)	57	2	3	4	***6***	4, 5
N - Dim. (N = 4)	74	3	4	5	3	4, 5, 6
Schwefel (N = 10)	200	9	10	11	10	11, 9
Griewangk (N= 10)	210	10	11	12	10	12
De Jong F 4	240	11	12	13	***20***	**13, 12**
Rastrigin (N = 20)	280	13	14	15	***11***	12, 13, 14
Tamanho Médio das sequências						***7.13***

Tabela 6.1.3 – ***Síntese das sequências flanqueadoras que permitiram obter os melhores resultados para o mecanismo de transposição/torneio***

Podemos concluir que a escolha do tamanho das sequências flanqueadoras assume uma grande relevância para a optimização das funções de teste. Efectuar uma variação exaustiva deste tamanho, tal como foi realizado no nosso trabalho experimental, é um trabalho demorado e que produz muita informação que só após uma análise detalhada permitirá tirar conclusões acerca da qualidade das soluções encontradas.

A restrição do intervalo de variação das sequências flanqueadoras surge como um dos métodos para simplificar esta tarefa. O estudo minucioso dos resultados obtidos não nos permitiu chegar a uma regra universal aplicável em todas as situações. No entanto,

escalonámos o problema em diferentes categorias e, para cada caso, encontrámos uma heurística que apresenta um intervalo de variação para o tamanho dos flanqueadores com algum grau de segurança.

6.2. Comparação da Transposição com os operadores de *Crossover* (1 ponto de corte, 2 pontos de corte e uniforme)

Nesta secção será apresentado o estudo comparativo detalhado dos resultados obtidos com os mecanismos de transposição e com os operadores tradicionais de *crossover* (com 1 ponto de corte, 2 pontos de corte e uniforme). A análise destes resultados focará o melhor e o pior resultados obtidos com transposição e a sua comparação com os operadores genéticos de *crossover*. Além do valor máximo (ou mínimo) alcançado em cada função, serão analisados os valores médios calculados a partir dos melhores valores obtidos na totalidade de gerações executadas. Os gráficos relativos a este estudo comparativo encontram-se em três anexos distintos:

ANEXO 3. – transposição *vs crossover* 1 ponto de corte

ANEXO 4. – transposição *vs crossover* 2 pontos de corte

ANEXO 5. – transposição *vs crossover* uniforme.

6.2.1. Funções de teste de De Jong (maximização)

O conjunto de teste de De Jong é constituído pelas 5 funções já descritas no capítulo 5. Os resultados obtidos com estas funções serão abordados de seguida.

As tabelas de apresentação dos resultados, além do máximo obtido por ambos os operadores em análise, apresentam também a média dos melhores resultados obtidos ao longo de todas as gerações.

F1 - De Jong (F1)

Relativamente ao máximo obtido, exceptuando o operador de *crossover* com 2 pontos de corte e população com 50 indivíduos, todos os operadores atingiram o máximo da função (78,6432). No caso de mecanismo de transposição, este valor foi alcançado quer pela melhor solução (transposição com sequências de tamanho 4; transposição/torneio com sequências de tamanho 6), quer pela pior solução (transposição com sequências de tamanho 8; transposição/torneio com sequências de tamanho 3). Os gráficos 1, 2 e 3 dos ANEXOS 3, 4 e 5, referem-se à melhor solução alcançada com os mecanismos de transposição e respectiva comparação com o operador de *crossover* com 1 ponto de corte, 2 pontos de corte e uniforme, respectivamente. Pela observação destes gráficos, verifica-se que, apesar de todos os operadores atingirem o máximo, os operadores de transposição e transposição/torneio convergem para esse valor mais depressa que os operadores de *crossover*. Além disso, os mecanismos de transposição, com uma população de 100 indivíduos, permitem obter convergência mais rápida do que os operadores de *crossover* com uma população de 200 indivíduos. Mesmo a pior solução alcançada com transposição/torneio com uma população de 200 indivíduos, consegue superar os operadores de *crossover* (ver gráficos 6 dos ANEXOS 2, 3 e 4 para comparação com os operadores de *crossover* com 1, 2 pontos de corte e uniforme, respectivamente).

A média obtida com os melhores valores de todas as gerações (500 neste caso) em ambos os mecanismos está bastante próxima. No entanto, a melhor solução alcançada com os mecanismos de transposição e transposição/torneio apresenta valores superiores àqueles obtidos com os operadores de *crossover*. No caso do mecanismo de transposição simples, observa-se que com populações de 100 indivíduos, obtiveram-se melhores resultados do que com uma população de 200 utilizando os operadores tradicionais de *crossover*.

A análise dos valores médios permite verificar uma ligeira superioridade do operador de *crossover* uniforme relativamente aos operadores com 1 e 2 pontos de corte.

Analisando a pior solução obtida com os mecanismos de transposição, verifica-se que os valores médios estão muito próximos dos obtidos com *crossover*. No entanto, os

mecanismos de transposição, com uma população de 100 e 200 indivíduos distanciam esta proximidade de valores.

A tabela 6.2.1 apresenta os valores em análise para todos os operadores genéticos.

		Valores Máximos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	78,6432	78,6432	78,6432	76,9453	77,3634	77,7218
	Crossover 2 pontos de corte	78,6330	78,6432	78,6432	76,3286	77,1538	77,7550
	Crossover Uniforme	78,6432	78,6432	78,6432	77,1807	77,6053	77,7233
Melhor	Transposição - 4	78,6432	78,6432	78,6432	76,8052	77,8551	77,9555
Solução	Transp. Torneio - 6	78,6432	78,6432	78,6432	77,0292	77,5059	78,0449
Pior	Transposição – 8	78,6432	78,6432	78,6432	77,0017	76,7858	77,6479
Solução	Transp. Torneio - 3	78,6432	78,6432	78,6432	75,8758	76,8129	77,9434

Tabela 6.2.1 ***– Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função De Jong – F1***

F2 - De Jong (F2)

Se analisarmos apenas os três operadores de *crossover* verifica-se, mais uma vez, uma ligeira supremacia do operador de *crossover* uniforme. No entanto, os mecanismos de transposição atingiram valores máximos superiores aos alcançados por este e, consequentemente, pelos restantes operadores. Mesmo a pior solução (transposição e transposição/torneio com sequências de tamanho 1), conseguiu, em algumas situações, resultados de melhor qualidade. Outra constatação, já encontrada na função anterior, é que populações menores com transposição têm um melhor desempenho do que o *crossover* com populações maiores. Os gráficos 7, 8 e 9 dos ANEXOS 3, 4 e 5 são ilustrativos desta conclusão.

Relativamente à média dos melhores valores obtidos nas 500 gerações simuladas, também a melhor solução para transposição apresenta valores mais elevados. A pior solução, para o mecanismo de transposição simples, atinge valores muito próximos, em alguns casos superiores, do que os operadores de *crossover*. De facto, os gráficos 10, 11 e 12 dos ANEXO 2, 3 e 4 evidenciam este aspecto.

Para a melhor solução, observa-se, mais uma vez, o factor população. Os mecanismos de transposição com menos indivíduos alcançam valores superiores aos

conseguidos com os operadores de *crossover*. A tabela 6.2.2 sintetiza os valores mais relevantes que suportam estas observações.

		Valores Máximos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	3904,2878	3902,6494	3902,6494	3807,2916	3865,9856	3881,7641
	Crossover 2 pontos de corte	3903,4686	3900,1918	3903,4686	3837,5086	3865,1199	3877,6395
	Crossover Uniforme	3905,1070	3902,6494	3905,1070	3799,6018	3862,9594	3884,8236
Melhor	Transposição - 3	3905,9262	3905,8014	3905,107	3844,5176	3889,7987	3886,5396
Solução	Transp. Torneio - 8	3905,9262	3904,2878	3904,2878	3861,8017	3873,6192	3889,6751
Pior	Transposição - 1	3905,107	3903,4686	3905,107	3800,6559	3880,8665	3887,1106
Solução	Transp. Torneio - 1	3904,2878	3905,107	3902,6494	3801,8488	3857,9466	3873,7381

Tabela 6.2.2 – ***Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função De Jong – F2***

F3 - De Jong (F3)

Exceptuando o operador de *crossover* com 2 pontos de corte, que nunca atingiu o máximo, o desempenho dos operadores genéticos, nesta função de teste, foi bastante semelhante. O mecanismo de transposição simples utilizando sequências de tamanho 2 atingiu sempre o máximo da função. Os operadores de transposição/torneio e *crossover* com 1 ponto de corte e uniforme, não o conseguiram em toda as situações. A convergência do AG para a máximo foi mais rápida quando se utilizou os mecanismos de transposição, com excepção para população com 50 indivíduos (ver gráficos 13, 14 e 15 dos ANEXOS 3, 4 e 5).

Mesmo a pior solução conseguida com os mecanismos de transposição, com 200 indivíduos permitiu alcançar o máximo e os valores médios estão muito próximos dos operadores de *crossover*, superando aqueles que foram alcançados com o operador de *crossover* com 2 pontos de corte.

Relativamente à média dos melhores valores obtidos nas 500 gerações simuladas, também a melhor solução para transposição apresenta valores mais elevados, com excepção para população com 50 indivíduos. O mecanismo de transposição/torneio obteve valore médios próximos dos operadores de *crossover*, apenas o superando com

uma população com 200 indivíduos. A tabela 6.2.3 apresenta detalhadamente os valores obtidos em cada um dos casos.

		Valores Máximos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	25	24	25	23,4360	23,1340	23,8980
	Crossover 2 pontos de corte	24	24	24	22,9520	22,5920	23,2440
	Crossover Uniforme	24	24	25	22,4040	22,7140	23,8320
Melhor Solução	Transposição – 2	25	25	25	23,0380	23,9420	24,0920
	Transp. Torneio – 3	24	25	25	22,9460	23,9820	24,2440
Pior Solução	Transposição – 15	24	24	25	21,9140	22,4040	23,7480
	Transp. Torneio – 6	24	25	25	21,4800	22,9460	23,5540

Tabela 6.2.3 – ***Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função De Jong – F3***

F4 - De Jong (F4)

A tabela 6.2.4 mostra claramente que os mecanismos de transposição, nomeadamente na sua versão com torneio, alcançaram soluções de qualidade consideravelmente superior. No caso do mecanismo de transposição/torneio, com uma população de 50 indivíduos, o AG superou todos os valores alcançados com os operadores de *crossover* utilizando populações de 50, 100 e 200 cromossomas.

Os gráficos 19, 20 e 21 dos ANEXO 3, 4 e 5, ilustram o melhor desempenho dos mecanismos de transposição e sua contraposição com os operadores de *crossover*. Os gráficos 22, 23 e 24, dos mesmos anexos, mostram os piores resultados obtidos com os operadores de transposição. Também aqui se verificou uma superioridade do mecanismo de transposição/torneio.

Quando analisamos os valores médios calculados a partir dos melhores indivíduos de todas as gerações, observa-se uma notória supremacia do mecanismo de transposição/torneio, apresentado o mecanismo de transposição uma ligeira superioridade relativamente aos três operadores tradicionais.

A tabela 6.2.4 apresenta detalhadamente os valores em causa.

		Valores máximos			Valores médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	1132,9995	1168,0291	1184,2729	982,0331	1040,7946	1087,937
	Crossover 2 pontos de corte	1124,7321	1167,8977	1182,9567	975,9754	1054,9602	1081,0621
	Crossover Uniforme	1134,3248	1168,2534	1183,5416	985,9043	1042,0513	1089,1933
Melhor Solução	Transposição - 2	1141,6718	1172,9825	1184,1625	1004,4236	1053,8135	1090,1143
	Transp. Torneio - 20	1239,748	1264,0081	1265,3879	1100,8754	1179,3685	1173,1016
Pior Solução	Transposição - 17	1127,8061	1168,111	1183,966	978,3947	1029,6914	1071,2332
	Transp. Torneio - 3	1194,2515	1254,2089	1264,6689	1015,7931	1130,6683	1162,6243

Tabela 6.2.4 – ***Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função De Jong – F4***

F5 - De Jong (F5)

Esta função de teste não permitiu chegar a qualquer conclusão sobre os mecanismos em estudo. Em todos os casos, o máximo foi alcançado à primeira geração, pelo que os valores médios são constantes. De qualquer forma, a tabela 6.2.5 apresenta os resultados obtidos em todos os casos.

		Valores máximos			Valores médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	500,0000	500,0000	500,0000	500,0000	500,0000	500,0000
	Crossover 2 pontos de corte	500,0000	500,0000	500,0000	500,0000	500,0000	500,0000
	Crossover Uniforme	500,0000	500,0000	500,0000	500,0000	500,0000	500,0000
Todas as Soluções	Transposição	500,0000	500,0000	500,0000	500,0000	500,0000	500,0000
	Transp. Torneio	500,0000	500,0000	500,0000	500,0000	500,0000	500,0000

Tabela 6.2.5 ***– Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função De Jong – F5***

6.2.2. Funções de teste suplementar (maximização)

Este conjunto de funções de teste, das quais se pretendeu obter o valor máximo, é constituído pelas seguintes funções: função de *Michalewicz*, funções de *Bohachevsky* 1 e 2, função de *Griewangk* (n = 10), função de *Rastrigin* (n = 20) e função de *Schwefel* (n = 10). As características destas funções foram já descritas no capítulo anterior.

F6 - Michalewicz

O desempenho dos mecanismos de transposição superou todos os valores alcançados com os operadores de *crossover*, aproximando-se mais dos valores obtidos com o operador de *crossover* com 2 pontos de corte. Mesmo a pior solução ultrapassou os valores máximos e médios obtidos com os operadores genéticos tradicionais.

Os gráficos 25, 26 e 27 dos ANEXOS 3, 4 e 5, ilustram os melhores resultados obtidos com transposição e os gráficos 28, 29 e 30, os piores. Tal como se pode observar, o desempenho dos mecanismos de transposição foi superior aos operadores de *crossover* em todos os casos. Na melhor situação, com uma população de 50 indivíduos, a transposição conseguiu melhores resultados que o *crossover* com 50, 100 e 200 indivíduos.

Também a análise dos valores médios conseguidos com as melhores soluções em todas as iterações, mostra que, os valores alcançados com os mecanismos de transposição são superiores em todos os casos, com maior proximidade aos obtidos com o operador de *crossover* com 2 pontos de corte.

A tabela 6.2.6 sumaria os resultados em análise.

		Valores Máximos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	37,2310	38,1090	38,4177	36,8982	37,5830	38,0210
	Crossover 2 pontos de corte	37,8113	38,6383	38,7679	37,3551	38,1954	38,5041
	Crossover Uniforme	37,6557	37,9786	38,6056	37,0656	37,4155	38,2867
Melhor	Transposição - 4	38,7791	38,7886	38,7868	38,3612	38,4484	38,5191
Solução	Transp. Torneio - 8	38,8503	38,8493	38,8439	38,6042	38,8043	38,8069
Pior	Transposição - 10	38,0557	38,6095	38,6774	37,5593	38,0216	38,4694
Solução	Transp. Torneio - 9	38,4499	38,4439	38,4503	38,4388	38,1603	38,4331

Tabela 6.2.6 – ***Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função de Michalewicz.***

F7 - Bohachevsky (I)

Relativamente ao valor máximo, ele foi atingido por todos os operadores genéticos em estudo, com excepção do operador com 2 pontos de corte e população com 50 indivíduos. Mas, tal como se pode ver nos gráficos 31, 32 e 33 dos ANEXOS 3, 4 e 5 os mecanismos de transposição, principalmente o mecanismo de transposição

simples, permitiram uma convergência mais rápida do AG para esse valor máximo, mesmo utilizando uma população de 50 indivíduos.

Analisando os valores médios alcançados ao fim de 500 gerações, os valores são, na maioria dos casos superiores nos mecanismos de transposição com sequências que permitiram alcançar os melhores resultados (3 e 6).

A tabela 6.2.7 apresenta os valores em análise para esta função.

		Valores Máximos			*Valores Médios*		
		50 Ind.	**100 Ind.**	**200 Ind.**	**50 Ind.**	**100 Ind.**	**200 Ind.**
	Crossover 1 ponto de corte	7500,0000	7500,0000	7500,0000	7452,3339	7477,0706	7472,0692
	Crossover 2 pontos de corte	7499,9847	7500,0000	7500,0000	7436,9660	7458,5999	7481,8225
	Crossover Uniforme	7500,0000	7500,0000	7500,0000	7459,0408	7468,9505	7462,1071
Melhor Solução	Transposição - 3	7500,0000	7500,0000	7500,0000	7474,9069	7477,7992	7478,9642
	Transp. Torneio - 6	7500,0000	7500,0000	7500,0000	7459,4808	7475,1411	7478,4017
Pior Solução	Transposição - 1	7493,8439	7499,4749	7500,0000	7399,9432	7433,1507	7462,2154
	Transp. Torneio - 1	7500,0000	7500,0000	7500,0000	7428,7533	7461,3225	7468,4008

Tabela 6.2.7 – ***Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bohachevsky (I)***

F8 - Bohachevsky (II)

O desempenho dos operadores genéticos nesta função assemelha-se ao alcançado com a função anterior. No entanto, a superioridade dos mecanismos de transposição é neste caso mais evidente. Os valores médios são superiores nas melhores soluções alcançadas o que indica, convergência mais rápida para o valor máximo. De facto, os gráficos 37, 38 e 39 dos ANEXOS 2, 3 e 4 evidenciam este aspecto. Mais uma vez, os mecanismos de transposição com populações menores (50 e 100 indivíduos) alcançaram melhores resultados (neste caso, convergência mais rápida) do que os operadores de *crossover* utilizando uma população com 200 indivíduos.

A tabela 6.2.8 apresenta os valores em análise.

		Valores Máximos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	7500,0000	7500,0000	7500,0000	7437,1680	7466,7100	7469,4010
	Crossover 2 pontos de corte	7499,9924	7500,0000	7500,0000	7439,2425	7472,1124	7467,9128
	Crossover Uniforme	7500,0000	7500,0000	7500,0000	7432,7595	7463,5809	7466,0201
Melhor Solução	Transposição - 3	7500,0000	7500,0000	7500,0000	7465,5424	7477,4800	7472,8667
	Transp. Torneio - 9	7500,0000	7500,0000	7500,0000	7482,5976	7478,8275	7478,0934
Pior Solução	Transposição - 9	7500,0000	7499,9924	7500,0000	7439,1719	7458,2389	7463,0492
	Transp. Torneio - 1	7499,3761	7500,0000	7500,0000	7394,5958	7462,1465	7472,5909

Tabela 6.2.8 – ***Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bohachevsky (II)***

F9 -Griewangk

Também neste caso, os mecanismos de transposição superaram os operadores de *crossover*. Relativamente aos valores máximos, podemos observar três situações:

- dos operadores de *crossover* tradicionais, a sua versão com 1 ponto de corte foi a que atingiu valores superiores;
- na melhor solução, os mecanismos de transposição com um população de 50 indivíduos obtiveram valores superiores aos do *crossover* com 50, 100 e 200 indivíduos. Esta constatação é clara na análise dos gráficos 43, 44 e 45 dos ANEXOS 3, 4 e 5.
- na pior solução, os mecanismos de transposição superaram, em alguns casos, os resultados do *crossover* com populações da mesma dimensão. Os gráficos 46, 47, 48 dos ANEXOS 3, 4 e 5 ilustram estes resultados.

Ao analisarmos os valores médios calculados a partir dos melhores indivíduos de cada geração, verificamos que a melhor solução dos mecanismos de transposição ultrapassa o valores obtidos pelos operadores genéticos tradicionais. Além disso, com populações menores, os mecanismos de transposição superam os resultados alcançados com os operadores de *crossover* com populações maiores.

A tabela 6.2.9 mostra a síntese destes resultados.

		Valores Máximos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	5278,0391	5718,9535	5982,7578	4885,2566	5205,8285	5531,2085
	Crossover 2 pontos de corte	5385,4920	5585,3969	5883,4415	5034,6361	5160,2579	5517,7952
	Crossover Uniforme	5400,3089	5786,2886	5834,3487	4923,4853	5375,5239	5399,4589
Melhor	Transposição – 8	6203,7959	6352,0879	6351,4147	5603,3318	5835,4364	5827,1700
Solução	Transp. Torneio – 10	5646,3728	6190,7717	6219,9724	5077,7399	5670,6735	5761,1459
Pior	Transposição – 18	5516,3503	5763,2261	5877,8594	5159,3031	5267,8639	5355,0293
Solução	Transp. Torneio – 20	5236,7780	5767,7271	5835,6181	4785,1578	5285,5579	5420,0062

Tabela 6.2.9 ***– Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função de Griewangk (N=10).***

F10 - Rastrigin

Para esta função, os mecanismos de transposição, utilizando uma população de 50 indivíduos, produziram resultados inferiores aos correspondentes obtidos com os operadores de *crossover*. Para populações com 100 e 200 indivíduos, os mecanismos alcançaram, para populações do mesmo tamanho, resultados superiores ao *crossover*. Os gráficos 49, 50 e 51 dos ANEXOS 3, 4 e 5, evidenciam estas observações. A observação destes gráficos permite concluir que o AG com os mecanismos de transposição convergiu mais depressa para os valores máximos. Também as piores soluções obtidas com os novos operadores genéticos, em algumas situações, superaram os valores obtidos com *crossover*.

A média dos melhores valores obtidos ao fim de 500 gerações, segue o mesmo raciocínio, isto é, apenas com populações com 50 indivíduos os mecanismos de transposição ficaram aquém dos resultados obtidos com *crossover*.

A tabela 6.2.10 mostra estes resultados.

		Valores Máximos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	780,5252	781,6576	797,8310	726,1677	733,0772	752,5922
	Crossover 2 pontos de corte	757,0067	778,6594	784,1078	712,5139	728,9449	736,2778
	Crossover Uniforme	749,5756	784,6014	782,5948	690,1870	735,5580	737,1849
Melhor	Transposição - 9	764,5783	796,0835	800,4473	704,9862	740,7537	741,0256
Solução	Transp. Torneio - 11	778,7807	796,4420	802,2541	724,6487	754,9991	759,8954
Pior	Transposição - 20	759,3568	770,1266	784,5841	698,8584	707,8575	736,0820
Solução	Transp. Torneio - 1	750,2637	776,6677	785,7193	701,4860	723,0063	742,9535

Tabela 6.2.10 – ***Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função de Rastrigin (N=20).***

F11 - Schwefel

Os mecanismos de transposição permitiram alcançar valores superiores aos operadores de *crossover*, quando utilizados com sequências de tamanho 4 e 10. Mais uma vez, populações com menos indivíduos conduziram a melhores resultados do que os operadores de *crossover* com populações de maiores dimensões. Os gráficos 55, 56 e 57, dos ANEXOS 3, 4 e 5, evidenciam de forma inequívoca os melhores resultados obtidos com transposição. A média dos melhores valores obtidos ao fim de 1000 gerações é, para o melhor caso da transposição, superior aos valores atingidos com *crossover*. Mesmo as piores soluções alcançadas com os mecanismos de transposição conseguem, em alguns casos, ultrapassar os valores conseguidos com o AG utilizando os operadores de recombinação tradicionais.

A tabela 6.2.11 mostra estes resultados.

		Valores Máximos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	7843,5418	8020,1151	8153,4297	7543,8486	7667,2040	7821,4787
	Crossover 2 pontos de corte	7862,4104	7969,3710	8116,8000	7546,8458	7553,4524	7749,8832
	Crossover Uniforme	7854,3587	7974,1173	8178,3854	7579,3010	7612,8715	7829,8623
Melhor	Transposição - 4	8061,2831	8202,7662	8276,6807	7724,9635	7893,0883	7948,4027
Solução	Transp. Torneio - 10	8130,3692	8201,1602	8346,3864	7768,0870	7810,2352	7965,7186
Pior	Transposição - 15	7841,5346	7935,0797	8114,7882	7477,6675	7576,9711	7720,4632
Solução	Transp. Torneio - 1	7885,8275	7783,6816	8086,4366	7574,4246	7517,6935	7772,1421

Tabela 6.2.11 ***– Síntese dos melhores e piores resultados (máximos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função de Schwefel (N=10).***

6.2.3. Funções de teste sugeridas por Koon (minimização)

O mecanismo de transposição e o operador de *crossover* foram também analisados num problema de minimização. Para realizar este estudo comparativo utilizámos um conjunto de 7 funções sugeridas por Koon em [Koon et al. 1995]. Essas funções são as seguintes: função *6-Hump Camelback*, função de *Shubert*, função *n-dimensional* (n=1, ..., 4), função *bi-dimensional 1*, função *bi-dimensional 2* (n=1, ..., 4), função *bi-dimensional de Rastrigin* e função *Uni-dimensional*. As características destas funções foram já descritas no capítulo anterior.

F12 - 6-Hump Camelback

Tal como se observa nos gráficos 61, 62 e 63 dos ANEXOS 3, 4 e 5, os mecanismos de transposição permitiram atingir melhores soluções. Mesmo com a pior solução (gráficos 64, 65, 66), em algumas situações, os resultados foram melhores do que aqueles obtidos com os mecanismos de *crossover*. Mais uma vez se verificou que o AG implementado com o mecanismo de transposição, com populações de 50 indivíduos, conseguiu atingir valores mínimos superiores ao alcançados com os operadores de *crossover* com populações de 50, 100 e 200 indivíduos.

A média dos melhores valores obtidos ao fim de 2000 gerações é, em todos os casos, superior para os mecanismos de transposição. Relativamente aos piores resultados obtidos com o mecanismo de transposição baseada em torneio, apenas com 50 indivíduos o AG obteve um desempenho ligeiramente inferior do que com os operadores de *crossover*. Em todos os restantes casos (transposição simples e transposição/torneio com populações maiores) os piores resultados obtidos com transposição superam os alcançados com os operadores tradicionais.

A tabela 6.2.12 mostra os resultados obtidos.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-1,0172	-1,0170	-1,0117	-0,9789	-1,0058	-0,9921
	Crossover 2 pontos de corte	-1,0093	-1,0084	-1,0164	-0,9953	-0,9877	-0,9995
	Crossover Uniforme	-1,0203	-0,9984	-1,0054	-0,9946	-0,9828	-0,9973
Melhor Solução	Transposição - 4	-1,0282	-1,0315	-1,0316	-1,0092	-1,0139	-1,0230
	Transp. Torneio - 8	-1,0258	-1,0271	-1,0317	-1,0075	-1,0111	-1,0154
Pior Solução	Transposição - 10	-1,0109	-1,0187	-1,0271	-0,9824	-1,0016	-1,0061
	Transp. Torneio - 10	-1,0008	-1,0109	-1,0084	-0,7590	-0,9082	-0,9426

Tabela 6.2.12 ***– Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função 6-HumpCamel Back.***

F13 - Shubert

As mesmas observações que nos casos anteriores se verificam para esta função. Apesar da proximidade dos resultados obtidos, os mecanismos de transposição alcançaram valores mínimos ligeiramente superiores, a convergência foi mais rápida e os valores médios são também superiores. Populações pequenas conjuntamente com os mecanismos de transposição permitiram obter valores mínimos de melhor qualidade do que populações maiores quando se usa AG com os operadores de *crossover*. Os gráficos 67, 68, 69, dos ANEXOS 3, 4 e 5, mostram estes resultados.

A tabela 6.2.13 sintetiza estes resultados.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-186,1501	-186,2806	-186,2813	-182,6958	-183,4441	-182,9962
	Crossover 2 pontos de corte	-186,4088	-186,4857	-186,6500	-178,3771	-184,4176	-185,2920
	Crossover Uniforme	-186,4893	-186,4891	-186,7302	-185,1405	-185,2669	-184,5317
Melhor	Transposição - 2	-186,6496	-186,7301	-186,7300	-183,7463	-184,9466	-185,5195
Solução	Transp. Torneio - 4	-186,4808	-186,3945	-186,4420	-183,1903	-184,9920	-184,3979
Pior	Transposição - 10	-179,5267	-186,4411	-186,5697	-176,5473	-184,3892	-183,5833
Solução	Transp. Torneio - 1	-182,2972	-186,4891	-186,4889	-177,4483	-183,6768	-183,0666

Tabela 6.2.13 ***– Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função de Shubert.***

F14 - N-dimensional

Esta função foi testada variando o valor da dimensão N de 1 até 4. Verificou-se que à medida que se aumentou a dimensionalidade da função, o AG utilizando os operadores de *crossover* não conseguiu alcançar melhores resultados enquanto que, com os mecanismos de transposição, o AG evoluiu de forma significativa. Assim, para esta função, utilizando dimensão superior, é mais clara a superioridade dos mecanismos de transposição. Vejamos então, com mais detalhe, cada um dos casos.

N = 1

Para **n** = 1, apesar dos mecanismos de transposição terem permitido obter valores mínimos superiores, a diferença destes valores com aqueles obtidos com os operadores de *crossover* é muito pequena. Tanto os gráficos 73, 74 e 75 dos ANEXOS 3, 4 e 5 como a tabela 6.2.14 evidenciam este aspecto.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-39,1646	-39,1653	-39,1655	-39,1321	-39,1339	-39,1566
	Crossover 2 pontos de corte	-39,1373	-39,1653	-39,1656	-39,1021	-39,1259	-39,1586
	Crossover Uniforme	-39,1505	-39,1639	-39,1661	-39,1355	-39,1044	-39,1569
Melhor Solução	Transposição - 2	-39,1659	-39,1662	-39,1662	-39,1528	-39,1513	-39,1600
	Transp. Torneio - 6	-39,1657	-39,1647	-39,1656	-39,1323	-39,1389	-39,1582
Pior Solução	Transposição - 7	-39,1377	-39,1650	-39,1662	-39,1093	-39,1298	-39,1583
	Transp. Torneio - 7	-39,1233	-39,1513	-39,1650	-39,1099	-39,1017	-39,1521

Tabela 6.2.14 – ***Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, transposição e transposição/torneio para a função N-dimensional (n=1).***

N = 2

Aumentado o valor de **n** para 2, torna-se mais evidente a superioridade do AG quando utiliza os mecanismos de transposição. Mesmo a pior solução da transposição é, muitas vezes, superior aos operadores tradicionais de *crossover*. A tabela 6.2.15 e os gráficos 79 a 84 dos ANEXOS 3, 4 e 5 mostram os resultados alcançados com os operadores genéticos em estudo.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-62,6285	-71,0200	-73,6592	-62,1344	-66,8135	-72,1389
	Crossover 2 pontos de corte	-65,4676	-67,2053	-71,9630	-65,1845	-65,2616	-69,3398
	Crossover Uniforme	-62,9881	-63,1683	-72,6211	-62,0899	-62,6410	-69,9426
Melhor Solução	Transposição - 3	-78,3306	-78,3306	-78,3315	-76,7094	-76,6853	-77,4684
	Transp. Torneio - 3	-77,6588	-78,3173	-78,3303	-73,2353	-76,0633	-76,4998
Pior Solução	Transposição - 10	-74,1360	-72,1526	-76,9178	-68,9929	-69,0958	-73,9892
	Transp. Torneio - 10	-64,7306	-67,0376	-75,1771	-62,7114	-65,4447	-72,8814

Tabela 6.2.15 – ***Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função N-dimensional (n=2).***

N = 3

Com a dimensão da função igual a 3, os resultados mantêm as mesmas características: superioridade dos mecanismos de transposição que, com população de

50 indivíduos alcança soluções muito melhores do que o operador de *crossover* com populações de 50, 100 e 200 indivíduos. Os gráficos 85, 86 e 87, dos mesmos anexos, mostram estes resultados.

Mais uma vez, mesmo a pior solução alcançada com os mecanismos de transposição, consegue em alguns casos superar os valores obtidos com os operadores tradicionais de recombinação genética. Os gráficos 88, 89 e 90 e a tabela 6.2.16 apresentam os valores obtidos.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-101,9069	-103,0480	-104,1435	-89,5382	-96,0090	-99,8163
	Crossover 2 pontos de corte	-106,6513	-102,8995	-105,7903	-102,5521	-93,4700	-100,5817
	Crossover Uniforme	-105,5717	-104,3740	-105,5198	-98,3490	-95,7928	-100,6486
Melhor Solução	Transposição - 4	-117,0075	-117,2489	-117,4985	-110,4575	-109,3206	-114,1573
	Transp. Torneio - 6	-112,5215	-114,5198	-117,0083	-103,6422	-106,1709	-110,6998
Pior Solução	Transposição - 13	-110,1189	-110,3453	-109,6940	-99,8084	-101,2853	-99,2870
	Transp. Torneio - 15	-107,7165	-104,1284	-101,9312	-99,3858	-97,6945	-94,4453

Tabela 6.2.16 ***– Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função N-dimensional (n=3).***

N = 4

As mesmas constatações se puderam verificar quando se aumentou **N** para 4. A tabela 6.2.17 e os gráficos 91 a 96 dos ANEXOS 3, 4 e 5 resumem os resultados obtidos e comprovam a superioridade dos mecanismos de transposição.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-127,9654	-115,0828	-137,6195	-108,3967	-113,6055	-133,2237
	Crossover 2 pontos de corte	-131,7322	-122,1456	-130,3566	-111,0872	-118,7131	-124,2315
	Crossover Uniforme	-121,2718	-128,0433	-130,3523	-101,9516	-122,9687	-120,6598
Melhor Solução	Transposição - 4	-153,9444	-155,3229	-156,6605	-131,3568	-143,2226	-151,4682
	Transp. Torneio - 3	-138,0874	-135,6424	-151,1868	-127,7007	-127,1585	-143,5059
Pior Solução	Transposição - 17	-127,2408	-132,1207	-141,8101	-81,0853	-118,8939	-128,4216
	Transp. Torneio - 1	-131,3819	-121,0924	-129,8045	-100,3704	-113,0753	-127,2940

Tabela 6.2.17 ***– Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função N-dimensional (n=4).***

F15 - Bi-dimensional (I)

O desempenho dos operadores genéticos foi, neste caso muito semelhante. O mínimo foi sempre alcançado e a média dos valores mínimos obtidos ao fim de 1000 gerações apresenta valores muito próximos. A tabela 6.2.18 e os gráficos 97 a 102 dos ANEXOS 3, 4 e 5 resumem os resultados obtidos.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	0,0000	0,0000	0,0000	0,0063	0,0025	0,0039
	Crossover 2 pontos de corte	0,0000	0,0000	0,0000	0,0073	0,0034	0,0036
	Crossover Uniforme	0,0000	0,0000	0,0000	0,0067	0,0032	0,0025
Melhor Solução	Transposição - 2	0,0000	0,0000	0,0000	0,0048	0,0035	0,0027
	Transp. Torneio - 10	0,0000	0,0000	0,0000	0,0045	0,0029	0,0032
Pior Solução	Transposição - 8	0,0000	0,0000	0,0000	0,0105	0,0038	0,0055
	Transp. Torneio - 3	0,0000	0,0000	0,0000	0,0075	0,0051	0,0025

Tabela 6.2.18 ***– Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional I.***

F16 - Bi-dimensional (II)

Esta função foi testada variando o valor de **N** de 1 até 4. Neste caso, o desempenho dos operadores genéticos foi muito próximo, mesmo quando se aumentou o valor de **N**.

Vejamos então, com mais detalhe, cada um dos casos.

N = 1

Os valores mínimos obtidos são muito próximos em todos os casos. Os gráficos 103, 104 e 105 dos ANEXOS 3, 4 e 5 e a tabela 6.2.19 comprovam esta proximidade no desempenho dos operadores de transposição e *crossover*. Analisando os dados com mais detalhe, conclui-se que, apesar da semelhança nos valores alcançados pelo AG em todos os casos, quando este foi executado com os mecanismos de transposição esses valores foram, na maioria dos casos, ligeiramente superiores.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-0,4060	-0,4060	-0,4071	-0,3582	-0,3719	-0,3839
	Crossover 2 pontos de corte	-0,4064	-0,3822	-0,4064	-0,3573	-0,3484	-0,3741
	Crossover Uniforme	-0,3930	-0,3964	-0,4064	-0,3261	-0,3607	-0,3581
Melhor Solução	Transposição - 3	-0,4075	-0,4071	-0,4060	-0,3666	-0,3815	-0,3825
	Transp. Torneio - 6	-0,4060	-0,4064	-0,4056	-0,3758	-0,3617	-0,3793
Pior Solução	Transposição - 9	-0,3799	-0,4070	-0,4071	-0,3152	-0,3670	-0,3856
	Transp. Torneio - 1	-0,3569	-0,4067	-0,4056	-0,2820	-0,3482	-0,3764

Tabela 6.2.19 ***– Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional II (n=1).***

N = 2

Também neste caso os valores se apresentam muito semelhantes em todos os casos. No entanto, para a melhor solução, existe uma ligeira superioridade dos mecanismos de transposição face aos operadores de *crossover*. Apesar dos gráficos 109, 110 e 111 dos ANEXOS 3, 4 e 5 não permitirem observar com muita facilidade esta ligeira superioridade, os valores apresentados na tabela 6.2.20 possibilitam essa diferenciação.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-18,0576	-18,0576	-18,0587	-17,4481	-18,0333	-18,0363
	Crossover 2 pontos de corte	-18,0536	-18,0580	-18,0587	-17,5959	-18,0302	-18,0369
	Crossover Uniforme	-18,0433	-18,0465	-18,0580	-17,4881	-18,0295	-18,0298
Melhor Solução	Transposição - 2	-18,0577	-18,0586	-18,0586	-17,5600	-18,0390	-18,0434
	Transp. Torneio - 5	-18,0536	-18,0468	-18,0576	-17,6824	-18,0239	-18,0398
Pior Solução	Transposição - 8	-18,0580	-18,0580	-18,0583	-17,1260	-18,0139	-18,0390
	Transp. Torneio - 4	-18,0535	-18,0532	-18,0583	-17,1430	-18,0127	-18,0432

Tabela 6.2.20 ***– Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional II (n=2).***

N = 3

Mais uma vez, existe uma grande proximidade nos resultados obtidos, existindo, no entanto uma ligeira supremacia dos valores obtidos com o AG implementado com os mecanismos de transposição (melhor solução).

Em alguns casos, evidentes a partir da observação da tabela, os novos mecanismos de recombinação permitiram obter melhores soluções com populações menores do que os operadores de *crossover* utilizando populações de maior dimensão.

Observando os valores da tabela 6.2.21 constata-se essa distinção.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-225,9019	-227,7416	-227,7485	-207,3518	-221,7613	-223,7404
	Crossover 2 pontos de corte	-220,3510	-226,2683	-227,7623	-204,6500	-217,4197	-221,3932
	Crossover Uniforme	-222,2328	-225,9152	-227,7520	-204,2497	-213,4075	-220,7577
Melhor Solução	Transposição - 10	-227,7623	-227,7549	-227,7520	-220,6613	-221,0019	-220,9827
	Transp. Torneio - 6	-227,7451	-227,7451	-227,7451	-220,7547	-219,5213	-221,9912
Pior Solução	Transposição - 7	-227,7510	-227,7520	-227,7451	-210,8425	-215,6807	-220,1985
	Transp. Torneio - 3	-225,9019	-227,7416	-227,7554	-209,5092	-214,6873	-222,4577

Tabela 6.2.21 ***– Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional II (n=3).***

N = 4

Para **n** = 4, verificam-se as mesmas conclusões dos casos anteriores. Assim, apesar da semelhança dos valores, existe uma ligeira superioridade nos resultados alcançados com os mecanismos de transposição. Esta ligeira vantagem regista-se quer nos valores mínimos obtidos, quer na média dos melhores valores ao fim de todas as gerações.

A tabela 6.2.22 apresenta os valores em causa.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-2429,1976	-2429,1988	-2429,0044	-2304,8886	-2366,0446	-2374,3515
	Crossover 2 pontos de corte	-2421,1700	-2429,1965	-2429,3073	-2267,7833	-2296,4224	-2388,2930
	Crossover Uniforme	-2429,3028	-2429,3017	-2428,9828	-2303,7627	-2336,8372	-2340,3034
Melhor Solução	Transposição - 6	-2429,0902	-2429,4125	-2429,4125	-2359,1542	-2320,4864	-2375,4060
	Transp. Torneio - 6	-2429,1954	-2421,0637	-2429,0913	-2360,9028	-2345,3019	-2367,1218
Pior Solução	Transposição - 7	-2397,0282	-2429,3028	-2429,3017	-2234,9148	-2275,4591	-2375,8736
	Transp. Torneio - 2	-2428,9806	-2428,9828	-2428,9817	-2282,1463	-2269,0139	-2353,5554

Tabela 6.2.22 ***– Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional II (n=4).***

F17 - Bi-dimensional de Rastrigin

O AG utilizando os operadores tradicionais de *crossover* apenas atingiu o valor mínimo da função com uma população de 200 indivíduos, na sua versão com 1 ponto de corte. Para a melhor solução alcançada com os operadores de transposição constata-se o seguinte:

- o mecanismo de transposição simples atingiu sempre (50, 100 e 200 indivíduos) o valor mínimo.
- o mecanismo de transposição/torneio não atingiu o valor mínimo, com 50 indivíduos na população.

Estes aspectos são claros na observação dos gráficos 127, 128 e 129 dos ANEXOS 3, 4 e 5.

Também em alguns casos da pior solução os mecanismos de transposição foram melhores que os operadores tradicionais de recombinação, visto que permitiram a obtenção do valor mínimo da função em algumas situações. A observação dos gráficos 130, 131 e 132 dos ANEXOS 2, 3 e 4 permitem a observação directa destas constatações.

Relativamente à média dos valores mínimos obtidos ao fim das 1000 gerações, também para os mecanismos de transposição estes valores são de melhor qualidade.

A tabela 6.2.23 apresenta os valores em causa.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-1,8651	-1,9572	-2,0000	-1,6653	-1,8542	-1,8879
	Crossover 2 pontos de corte	-1,8888	-1,9079	-1,8772	-1,7224	-1,7725	-1,8244
	Crossover Uniforme	-1,9444	-1,9693	-1,9693	-1,6814	-1,8980	-1,9223
Melhor Solução	Transposição - 3	-2,0000	-2,0000	-2,0000	-1,8349	-1,9118	-1,9359
	Transp. Torneio - 5	-1,9879	-2,0000	-2,0000	-1,8189	-1,9011	-1,9429
Pior Solução	Transposição - 9	-1,9243	-2,0000	-2,0000	-1,6805	-1,8808	-1,9097
	Transp. Torneio - 1	-1,9074	-1,9386	-2,0000	-1,6988	-1,8130	-1,9437

Tabela 6.2.23 ***– Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Bi-dimensional de Rastrigin.***

F18 - Uni-dimensional

Nesta função, o desempenho dos operadores genéticos foi muito semelhante em todos os casos. Quando utilizado com os mecanismos de transposição, o AG permitiu a obtenção de valores de melhor qualidade de uma forma mais consistente independentemente do tamanho da população. Quando utilizado com os operadores de *crossover*, o AG é menos eficaz com populações mais pequenas.

A tabela 6.2.24 apresenta os valores obtidos.

		Valores Mínimos			Valores Médios		
		50 Ind.	100 Ind.	200 Ind.	50 Ind.	100 Ind.	200 Ind.
	Crossover 1 ponto de corte	-3,3725	-3,3729	-3,3727	-3,3711	-3,3727	-3,3722
	Crossover 2 pontos de corte	-3,3728	-3,3723	-3,3729	-3,3717	-3,3720	-3,3727
	Crossover Uniforme	-3,3713	-3,3729	-3,3728	-3,3698	-3,3726	-3,3725
Melhor Solução	Transposição - 3	-3,3729	-3,3729	-3,3729	-3,3717	-3,3726	-3,3727
	Transp. Torneio - 5	-3,3729	-3,3726	-3,3729	-3,3723	-3,3720	-3,3723
Pior Solução	Transposição - 5	-3,3727	-3,3727	-3,3727	-3,3698	-3,3725	-3,3723
	Transp. Torneio - 8	-3,3717	-3,3720	-3,3725	-3,3700	-3,3713	-3,3723

Tabela 6.2.24 – ***Síntese dos melhores e piores resultados (mínimos e médios) obtidos com os operadores de crossover 1 ponto de corte, 2 pontos de corte, uniforme, transposição e transposição/torneio para a função Uni-dimensional.***

6.3. Síntese dos Resultados

Como se pode verificar pelos resultados apresentados nas secções anteriores, os mecanismos de transposição, em geral, permitem ao AG a obtenção de melhores soluções que os operadores de *crossover*. Assim, se utilizarmos os mecanismos de transposição com população de 200 indivíduos, temos a garantia que o AG encontrará sempre melhores soluções do que com qualquer operador de *crossover*. No entanto, uma população maior implica maiores custos computacionais que podem ser minimizados se utilizarmos populações com menos indivíduos. Neste caso, os mecanismos de transposição só permitirão ao AG alcançar melhores resultados, se se utilizarem sequências flanqueadoras de determinado tamanho. Os resultados experimentais permitiram definir um conjunto de heurísticas que, quando aplicadas, fornecem um intervalo de três valores para variação do tamanho da sequência. Escolhendo um destes três valores temos a garantia que o AG encontrará a melhor solução ou uma solução muito próxima do máximo, mesmo com populações de 50 indivíduos. Neste caso, a transposição torna-se mais vantajosa, visto que, mesmo com populações mais pequenas, fornece, na grande maioria dos casos, melhores resultados do que o *crossover*. A síntese dos resultados obtidos no conjunto de teste pode também ser encontrada em [Simões et al. 1999c].

As tabelas seguintes apresentam, para cada função, qual o operador genético que permitiu obter os melhores resultados. Na tabela 5.3 analisamos apenas as experiências envolvendo população de 50 indivíduos; na tabela 5.4 sintetizam-se os resultados obtidos com populações de 50 e 100 indivíduos; na tabela 5.5 todos os resultados são comparados.

Dentro da célula correspondendo a cada operador encontra-se a população que permitiu alcançar o melhor valor para a função em causa. Se, para a mesma função, existir mais do que um operador preenchido com o correspondente valor da população, significa que todos eles permitiram atingir o óptimo. Neste caso, o valor a sublinhado e a itálico indica qual deles permitiu ao AG uma convergência mais rápida para esse valor.

		Melhor Operador, população				
Função		*Cx1*	*Cx2*	*CxU*	*TS*	*TT*
F1		50		50	50	*50*
F2					50	*50*
F3		50			*50*	
F4						50
F5		50	50	50	50	50
F6						50
F7		50		50	*50*	50
F8		50		50	50	*50*
F9					50	
F10		50				
F11						50
F12					50	
F13					50	
F14	N = 1				50	
	N = 2				50	
	N = 3				50	
	N = 4				50	
F15		50	50	50	50	*50*
F16	N = 1				50	
	N = 2				50	
	N = 3				50	
	N = 4					50
F17					50	
F18					50	*50*

Tabela 5. 3 - ***Para cada função, qual o melhor operador com população até 50 indivíduos***

Com população de 50 indivíduos, o AG com os mecanismos de transposição atingiu na quase totalidade dos casos o melhor resultado. De facto, só em **F10**, isto não se verificou. Além disso, nos casos em que também os operadores de *crossover* atingiram o melhor valor, os mecanismos de transposição conferiram ao AG convergência mais rápida.

		Melhor Operador, população				
Função		*Cx1*	*Cx2*	*CxU*	*TS*	*TT*
F1		50	100	50	50	*50*
F2					50	*50*
F3		50			*50*	100
F4						50
F5		50	50	50	50	50
F6						50
F7		50	100	50	*50*	50
F8		50	100	50	50	*50*
F9					50	
F10						**100**
F11						50
F12					50	
F13					50	
F14	N = 1				50	
	N = 2				50	
	N = 3				50	
	N = 4				50	
F15		50	50	50	50	*50*
F16	N = 1				50	
	N = 2				**100**	
	N = 3				50	
	N = 4				**100**	
F17					50	
F18		100		100	50	*50*

Tabela 5. 4 - ***Para cada função, qual o melhor operador com população até 100 indivíduos***

Os resultados desta tabela englobam os obtidos com populações de 50 e 100 indivíduos. Pretende-se evidenciar que os mecanismos de transposição com população de tamanho 50 foram, na maioria dos casos, superiores aos resultados alcançados com *crossover* utilizando população de 100 indivíduos. Apenas em 3 casos (marcados a negrito), o AG com transposição precisou de uma população de 100 indivíduos para ultrapassar os valores obtidos com *crossover*. Mais uma vez, em caso de igualdade na obtenção do valor máximo, a convergência foi mais rápida com os mecanismos de transposição

		Melhor Operador, população				
Função		*Cx1*	*Cx2*	*CxU*	*TS*	*TT*
F1		50	100	50	50	*50*
F2					50	*50*
F3		50	200	200	50	*100*
F4						50
F5		50	50	50	50	50
F6						50
F7		50	100	50	*50*	50
F8		50	100	50	50	*50*
F9					50	
F10						**200**
F11					100	
F12					50	
F13					100	
F14	N = 1				100	
	N = 2				50	
	N = 3				50	
	N = 4				50	
F15		50	50	50	50	*50*
F16	N = 1				50	
	N = 2				100	
	N = 3				50	
	N = 4				100	
F17					50	
F18		100	200	100	50	*50*

Tabela 5. 5 - ***Para cada função, qual o melhor operador com população até 200 indivíduos***

Alargando a análise dos resultados para populações até 200 indivíduos, a síntese dos resultados salienta o que já foi referido com detalhe nas secções anteriores. Portanto, os mecanismos de transposição com populações de 50 e 100 indivíduos permitem ao AG obter melhores resultados e mais depressa do que os operadores de *crossover* com população de 200 indivíduos. Mais uma vez a função **F10** foi a única excepção a esta observação.

Como pensamos ter provado neste estudo, a transposição revelou-se um mecanismo bastante poderoso, superior aos operadores tradicionais de *crossover*. Resta saber se noutro domínio de aplicação o mesmo se verificará.

No domínio da optimização de funções, é interessante verificar que, apesar da grande diversidade de funções utilizadas (ver tabela 5.1), os mecanismos de transposição permitem quase sempre alcançar melhores resultados do que quando utilizado na sua forma clássica com um dos operadores de *crossover*. De facto, o comportamento da transposição aparenta ser genérico, sem qualquer relação com as características das funções.

CAPÍTULO 7.
CONCLUSÕES E TRABALHO FUTURO

Sumário

Concluído o trabalho experimental e analisados os resultados obtidos, importa agora referir os objectivos principais desta tese, o que foi alcançado e aquilo que ficou por clarificar e/ou fazer.

É isso que faremos de seguida.

7.1. Conclusões

Concluído o trabalho experimental, com o qual se pretendeu testar o desempenho do algoritmo genético clássico, quando modificado com um novo operador, alternativo aos operadores de recombinação tradicionais, podemos sintetizar algumas conclusões.

A inspiração na natureza na procura de novos mecanismos que possam enriquecer o AG constitui uma fonte inesgotável de ideias que poderão trazer vantagens, quer na qualidade das soluções, quer na convergência do AG, quer em domínios muito específicos. Muitos são os autores que alertam para o facto da falta de estudo de certos mecanismos presentes nos seres vivos e cuja adaptação computacional poderá trazer bons resultados para o desempenho dos Algoritmos Genéticos clássicos ou para a programação genética.

O objectivo principal deste trabalho foi procurar na biologia novas ideias e verificar quais as possibilidades da sua incorporação no AG tradicional.

Neste sentido, após uma análise cuidada dos diversos mecanismos de rearranjo do material genético presentes nos seres vivos, que foram apresentados no capítulo 3, seleccionámos um desses processos para verificar em que medida poderia ser utilizado no AG tradicional. O mecanismo estudado foi a **transposição**. Este mecanismo foi inspirado num processo similar existente nos seres vivos e caracteriza-se pela presença de unidades genéticas móveis (transposões) com capacidade de se movimentarem no genoma. Para que um transposão seja reconhecido é necessária a existência de sequências flanqueadoras que podem ser iguais ou inversas. Um transposão ao abandonar um cromossoma pode ou não duplicar-se e quando se integra numa nova posição pode apresentar alguma preferência especial ou ser conduzido pelas características das sequências flanqueadoras.

A transposição foi utilizada como alternativa aos operadores de *crossover* (1 ponto de corte, 2 pontos de corte, uniforme) normalmente implementados no AG.

Assim, após a selecção dos progenitores de acordo com o mecanismo de selecção utilizado, a nova descendência foi produzida misturando o material genético de acordo com o funcionamento do mecanismo de transposição

Para avaliar o desempenho do AG quando se incorporou este mecanismo, seleccionámos um conjunto de dezoito funções de teste que se pretenderam optimizar.

Realizámos um trabalho experimental extenso, que envolveu para cada função de teste o estudo dos operadores de *crossover* com 1 e 2 pontos de corte e uniforme, variando para cada um deles o tamanho da população de 50, 100 e 200 indivíduos. O mecanismo de transposição, implementado em duas variantes, substituiu o operador de *crossover* e foi simulado, para cada função, com populações de 50, 100 e 200 indivíduos, variando ainda o tamanho das sequências flanqueadoras de 1 até um valor máximo que dependeu do tamanho do cromossoma.

O mecanismo de transposição implementado e integrado no AG foi directamente inspirado no seu correspondente biológico. Após os primeiros resultados experimentais verificámos alguma instabilidade na média geral da população. Para minimizar este efeito, implementámos uma segunda versão da transposição, baseada no torneio entre dois cromossomas. O vencedor do torneio será o cromossoma onde o transposão será formado e o perdedor aquele onde o transposão será inserido. Ao contrário do primeiro mecanismo de transposição, em que havia uma troca bidireccional de material genético, neste caso apenas o indivíduo perdedor será alterado pela injecção do transposão. Estes mecanismos designam-se por *transposição simples* e *transposição baseada em torneio*, respectivamente.

Após este trabalho concluímos que o desempenho do AG foi, na grande maioria dos casos superior quando utilizado com os mecanismos de transposição. Esta superioridade manifestou-se principalmente na qualidade das soluções alcançadas: com o mecanismo de transposição o AG conseguiu, na maioria das funções, atingir valores óptimos superiores aos alcançados com os operadores de *crossover*. Em algumas situações, em que ambos os operadores atingiram o óptimo, o mecanismo de transposição conduziu o AG a uma convergência mais rápida para esse valor. Outra constatação verificada em muitos casos, foi o facto de o AG utilizado com o mecanismo de transposição, com populações de 50 ou 100 indivíduos ultrapassar o desempenho dos operadores de *crossover* com populações maiores (100 e 200 indivíduos). Muitas vezes, com uma população de 50 indivíduos o mecanismo de transposição superou os operadores de *crossover* com populações de tamanho igual a 200 indivíduos.

O bom desempenho do mecanismo de transposição foi alcançado com uma escolha adequada do tamanho das sequências flanqueadoras. Verificou-se que

sequências muito grandes ou muito pequenas produziram soluções de baixa qualidade. O trabalho empírico realizado permitiu chegar a algumas conclusões sobre a forma como escolher, para cada caso, o tamanho das sequências que produzam os melhores resultados. Apresentámos um conjunto de heurísticas que, apesar de não serem completamente seguras, permitem calcular um intervalo de variação para as sequências flanqueadoras, que permitirão alcançar os melhores resultados.

Apesar do mecanismo de transposição baseada em torneio permitir que a qualidade média da população se mantivesse mais estável, não se mostrou nem superior nem inferior ao mecanismo de transposição simples. Apesar de algumas diferenças de comportamento, ambos permitiram o bom desempenho do AG.

Consideramos, portanto, que este novo operador genético (nas suas duas variantes), inspirado nos processos biológicos, provou ser um bom substituto para os operadores tradicionais de *crossover*, conseguindo, na generalidade das situações, obter resultados de qualidade muito superior.

Temos consciência que este trabalho é apenas a ponta do *iceberg*. O capítulo 3, apresentou um conjunto de novas ideias que podemos procurar na biologia, limitadas ao assunto do **rearranjo do material genético**. Muitas mais existirão, abrangendo outros assuntos que permitirão com certeza enriquecer a investigação no domínio dos AG.

7.2. Trabalho Futuro

O trabalho realizado foi fortemente empírico e não tinha à partida como objectivo central encontrar uma justificação teórica para os resultados alcançados. A próxima etapa, no seguimento do trabalho realizado, será iniciar o estudo teórico que permita fundamentar esses resultados.

O estudo experimental centralizou-se apenas no domínio da optimização de funções. A procura de outros domínios de aplicação, nomeadamente a programação genética será mais uma etapa a concretizar.

Finalmente, a procura de inspiração na biologia é uma fonte inesgotável de ideias. Assim, muitos outros mecanismos biológicos (além da transposição) poderão ser

adaptados e implementados. Obviamente que a utilização de novos mecanismos estará sempre sujeita a fortes estudos quer empíricos, quer teóricos, que permitam concluir quais as possíveis vantagens/desvantagens que esses novos mecanismos podem introduzir no funcionamento do AG.

BIBLIOGRAFIA

[Abu Zitar et al. 1993] R. A. Abu Zitar, M. H. Hassoum, ***Regulator Control via Genetic Search and Assisted Reinforcement,*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 254-262, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Ackley et al. 1991] D. H. Ackley, M. L. Littman, ***Interactions Between Learning and Evolution.*** In C. G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, eds., Artificial Life II, pp. 487 - 507, 1991.

[Ackley et al. 1993] D. H. Ackley, M. L. Littman, ***A Case for Lamarkian Evolution .*** In C. G. Langton, ed., Artificial Life III, 1993.

[Altman et al. 1993] E. R. Altman, V. K. Agarwal, G. R. Gao, ***A Novel Methodology Using Genetic Algorithms for the Design of Caches and Cache Replacement.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 392-399, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Arabas el al. 1995] J. Arabas, J. J. Mulawka, J. Pokrasniewicz, ***A New Class of the Crossover Operators for the Numeric Optimization.*** In L. J. Eshelman editor, Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 42-48, Pittsburgh, Morgan Kaufmann Publishers, Inc, 1995.

[Axelrod 1987] R. Axelrod, ***The Evolution of Strategies in the Iterated Prisioner's Dilemma.*** In L. D. Davis, ed., Genetic Algorithms and Simulated Annealing, Research Notes in Artificial Intelligence, 1987.

[Bäck 1993] T. Bäck, ***Optimal Mutation Rates in Genetic Search.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 2-8, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Bäck et al. 1997] T. Bäck, D. B. Fogel, Z. Michalewicz (eds), ***Handbook of Evolutionary Computation.*** IOP Publishing, Oxford University Press, 1997.

[Baker 1987] J. E. Baker, ***Reducing Bias and Inefficiency in the Selection Algorithm,*** In J.J Grefenstette, ed., Proceedings of the Second International Conference on Genetic Algorithms and their Applications, Erlbaum, 1987.

[Bala et al. 1993] J. W. Bala, H. Wechsler, ***Learning to Detect Targets using Scale-Space and Genetic Search,*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 516-522, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Balázs 1999] M. E. Balázs, ***A Genetic Algorithm with Dynamic Population Size,*** In A. Dobnikar, N. Steele, D. Pearson, R. Albrecht, (eds.), Proceedings of the Fourth

International Conference on Neural Networks and Genetic Algorithms (ICANNGA 99), Springer-Verlag, 1999.

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

[Beasley et al. 1993a] D. Beasley, D. R. Bull, R. R. Martin***, An Overview of Genetic Algorithms: Part I, Fundamentals***, University Computing, 1993.

[Beasley et al. 1993b] D. Beasley, D. R. Bull, R. R. Martin***, An Overview of Genetic Algorithms: Part II Research Topics***, University Computing, 1993.

[Beasley et al. 1993c] D. Beasley, D. R. Bull, R. R. Martin***, A Sequential Niche Technique for Multimodal Function Optimization***, In Evolutionary Computation 1(2), pp. 101-125, 1993.

[Beasley 1997] D. Beasley, ***Why Evolutionary Computation? - Possible Applications of Evolutionary Computation***. In T. Bäck, D. Fogel, Z. Michalewicz, eds., Handbook of Evolutionary Computation, part A.1.2:1-10, IOP Publishing Ltd and Oxford University Press, 1997.

[Bedau et al. 1992] M. A. Bedau, N. H. Packard, ***Measurement of Evolutionary Activity.*** In C. G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, eds., Artificial Life II, pp. 431 - 61, 1992.

[Bergman et al. 1992] A. Bergman, M. W, Feldman, ***Recombination Dynamics and the Fitness Landscape.*** In Physica D, 56, pp. 57-67, 1992.

[Blanton et al. 1993] J. L. Blanton, L. N. Wainwright, ***Multiple Vehicle Routing With Time and Capacity Constraints.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 452-459, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Blickle et al. 1996] T. Blickle, L. Thiele, ***A Comparison of Selection Schemes Used in Evolutionary Algorithms.*** In Evolutionary Computation, 4(4), pp. 361 – 394, MIT Press, 1996.

[Booker 1985] L. Booker, ***Improving The Performance of Genetic Algorithms in Classifier Systems,*** In J. J. Grefenstette editor, Proceedings of the First International Conference on Genetic Algorithms, pp. 80-92, Pittsburgh, PA, Morgan Kaufmann Publishers, Inc, 1993.

[Bramlette et al. 1991] M. F. Bramlette, E. E. Bouchard, ***Genetic Algorithms in Parametric Design of Aircraft,*** In L. Davis editor, Handbook of Genetic Algorithms, ch. 10, pp. 109-123, Van Nostrand Reinhold, 1991.

[Bresch et al. 1972] C. Bresch, R. Hausmann, ***Genética Clássica e Molecular,*** 3ª edição, Fundação Calouste Gulbekian, 1972.

[Burke et al. 1998] D. S. Burke, K. De Jong, J. J. Grefenstette, C. L. Ramsey, A. S. Wu, ***Putting More Genetics into Genetic Algorithms,*** In Evolutionary Computation, 6(4) **,** pp. 387-410, MIT Press, 1998.

[Chambers 1995a] L. Chambers, ***Practical Handbook of Genetic Algorithms, Vol I***, CRC Press, Inc., 1995.

[Chambers 1995b] L. Chambers, ***Practical Handbook of Genetic Algorithms, Vol II***, CRC Press, Inc., 1995.

[Chan et al. 1991] H. Chan, P. Mazumder, K. Shahookar, ***Macro-cell and Module Placement by Genetic Adaptive Search with Bitmap-represented Chromosome,*** In Integration VLSI Journal, 1991.

[Cohoon et al. 1987] J. P. Cohoon, W. D. Paris, ***Genetic Placement,*** In IEEE Transactions Computer-Aided Design (6), pp. 956-964, 1987.

[Collingwood et al. 1996] E. Collingwood, D. Corne, P. Ross, ***Useful Diversity via Multiploidy***, Submitted to AISB Workshop on Evolutionary Computation, 1996.

[Collins et al. 1991a] R. J. Collins, D. R. Jefferson, ***Selection in Massively Parallel Genetic Algorithms.*** In R. K. Belew, L. B. Booker, eds., Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 249-256, Morgan Kaufmann Publishers, Inc, 1991.

[Collins et al. 1991b] R. J. Collins, D. R. Jefferson***, AntFarm: Towards Simulated Evolution***. In Christopher G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, eds., Artificial Life II, SFI Studies in the Science of Complexity, Addison-Wesley, 1991.

[Collins et al. 1992] R. J. Collins, D. R. Jefferson***, The Evolution of Sexual Selection and Female Choice***. In F. J. Varela, P. Bourgine, eds., Toward a Practice of Autonomous Systems: Proceedings of the First Conference on Artificial Life, pags 327-336, Cambridge, MA. MIT Press/Bradford Books, 1992.

[Cormen et al. 1990] T. H. Cormen, C. E. Leiserson, R. L. Rivest, ***Introduction to Algorithms,*** MIT Press, McGraw-Hill, 1990.

[Crutchfield et al. 1995] J. P. Crutchfield, M. Mitchell, ***The Evolution of Emergent Computation***, (SFI Technical Report 94-03-012). In the Proceedings of the National Academy of Sciences, USA , 92, 1995.

[Dasgupta et al. 1996] D. Dasgupta, S. Forrest, ***Novelty Detection in Time Series Data Using Ideas from Immunology***, 1996.

[Davidor 1989] Y. Davidor, ***Analogous Crossover***. In J. D. Shaffer editor, Proceedings of the Third International Conference on Genetic Algorithms, pp. 98-103, Morgan Kaufmann Publishers, Inc, 1989.

[Davidor 1991] Y. Davidor, ***A Genetic Algorithm Applied to Robot Trajectory Generation.*** In L. Davis editor, Handbook of Genetic Algorithms, ch. 11, pp. 124-143, Van Nostrand Reinhold, 1991.

[Davis 1985] L. Davis, ***Job Shop Scheduling With Genetic Algorithms*** . In J. J. Grefenstette editor, Proceedings of the First International Conference on Genetic Algorithms, pp. 136-140, Lawrence Elrbaum Associates, 1985.

[Davis 1989] L. Davis, ***Adapting Operator Probabilities in Genetic Algorithms***. In J. D. Shaffer editor, Proceedings of the Third International Conference on Genetic Algorithms, pp. 61-69, Morgan Kaufmann Publishers, Inc, 1989.

[Davis 1991] L. Davis, ***Handbook of Genetic Algorithms***. Van Nostrand Reinhold, 1991.

[Deb et al. 1989] K. Deb, D. E. Goldberg, ***An Investigation of Niche and Species Formation in Genetic Function Optimization***. In J. D. Shaffer editor, Proceedings of the Third International Conference on Genetic Algorithms, pp. 42-50, Morgan Kaufmann Publishers, Inc, 1989.

[Deb 1997] K. Deb, ***Representations***. In T. Bäck, D. Fogel, Z. Michalewicz, eds., Handbook of Evolutionary Computation, part C1.1:1-4, IOP Publishing Ltd and Oxford University Press, 1997.

[De Jong 1975] K. A. De Jong, ***Analysis of the Behavior of a Class of Genetic Adaptive Systems.*** Ph.D. Dissertation, Department of Computer and Communication Science, University of Michigan, 1975.

[De Jong 1980] K. A. De Jong, ***Adaptive Systems Design: a Genetic Approach,*** In IEEE Transactions System, Man, Cyberneticsm (10), pp. 566-574, 1980.

[De Jong et al. 1992] K. A. De Jong, W. M. Spears, ***A Formal Analysis of the Role of Multi-Point Crossover in Genetic Algorithms***. In Annals of Mathematics and Artificial Intelligence, Volume 5, #1, pp. 1-26, 1992.

[De Jong 1993] K. A. De Jong, ***Genetic Algorithms Are NOT Function Optimizers***. In L. Darrell Whitley, editor, Foundations of Genetic Algorithms 2, pp. 5-17, Morgan Kaufmann, 1993.

[De Jong et al. 1993a] K. A. De Jong, W. M. Spears***, On the State of Evolutionary Computation***. In Proceedings of the International Conference on Genetic Algorithms, pp. 618-623, 1993.

[De Jong et al. 1993b] K. A. De Jong, J. Sarma. ***Generation Gaps Revisited***. In L. D. Whitley, ed., Foundations of Genetic Algorithms 2, Morgan Kaufmann, 1993.

[D'Haeseleer 1994] P. D'Haeseleer, ***Context Preserving Crossover in Genetic Programming,*** In Proceedings of the 1994 IEEE World Congress on Computational Intelligence, Vol. 1, pp. 256-261, IEEE Press, pp. 27-29, June 1994.

[D'Haeseleer et al. 1996] P. D'Haeseleer, S. Forrest, P. Helman, ***An Immunological Approach to Change Detection: Algorithms, Analysis and Implications,*** In Proceedings of the IEEE Symposium on Security and Privacy, pp 110-119 - IEEE Computer Asociety Press, 1996.

[Easton et al. 1993] F.F. Easton, N. Mansour, ***A Distributed Genetic Algorithm For Employee Staffing and Scheduling Problems.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 360-367, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Eshelman et al. 1993] L. J. Eshelman, J. D. Schaffer, ***Crossover's Niche.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 9-14, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Fang et al. 1993] H-L. Fang, P. Ross, D. Corne, ***A Promising Genetic Algorithm Approach to Job-shop Scheduling, Resecheduling and Open-shop Scheduling Problems.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 375-382, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Feldman 1993] D. S. Feldman, ***Fuzzy Network Synthesis with Genetic Algorithms.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 312-317, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Flockton et al. 1993] S. J. Flockton, M. White, ***Pole-zero System Identification using Genetic Algorithms,*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 531-535, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Fogarty 1989] T. Fogarty, ***Varying the Probability of Mutation in the Genetic Algorithms***. In J. D. Shaffer editor, Proceedings of the Third International Conference on Genetic Algorithms, pp. 104-109, Morgan Kaufmann Publishers, Inc, 1989.

[Fogarty 1994] T. Fogarty, ***Co-evolving Co-operative Populations of Rules in Learning Control Systems,*** In Selected Papers of AISB Workshop, pp. 195-209, Springer-Verlag, 1994.

[Fogel 1988] D. Fogel, ***An Evolutionary Approach to the Traveling Salesman Problem,*** Biological Cybernetics, Vol. 60, pp. 139-144, 1989.

[Fogel et al. 1990] D. B. Fogel, L. J. Fogel, V. W. Porto, ***Evolving Neural Networks,*** In Biological Cybernetics, 63, pp. 487-493, 1990.

[Fogel 1991] D. B. Fogel, ***System Identification though Simulated Evolution,*** Needham, MA: Ginn, 1991.

[Fogel 1993] D. B. Fogel, ***Evolving Behaviors in the Iterated Prisoner's Dilemma,*** In Evolutionary Computation (1), pp. 77-97, 1993.

[Fogel 1995] D. B. Fogel, ***Evolutionary Computation: Toward a New Philosophy of Machine Intelligence.*** IEEE Press, 1995.

[Fogel et al. 1996] D. B. Fogel, L. J. Fogel, ***Using Evolutionary Programming to Schedule Tasks on a Suite of Heterogeneous Computers,*** In Computer Operation Res. (23), pp. 527-534, 1996.

[Fonseca et al. 1993] C. M. Fonseca, P. J. Fleming, ***Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416-423, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Forrest et al. 1991] S. Forrest, A. S. Perelson, ***Genetic Algorithms and the Immune System,*** In Lectures Notes in Computer Science, Vol. 496, 1991.

[Forrest et al. 1992] S. Forrest, B. Javornik, Robert E. Smith, A. S. Perelson, ***Using Genetic Algorithms to Explore Pattern Recognition in the Immune System,*** Computing Surveys Vol. 28:1, pp. 77-80, 1992.

[Forrest et al. 1994] S. Forrest, T. Jones, ***Modeling complex adaptive systems with Echo*** In R.J. Stonier and X.H. Yu (eds.), Complex Systems: Mechanism of Adaptation, IOS Press, pp. 3-21, 1994.

[Forrest 1996] S. Forrest, ***Genetic Algorithms.*** Computing Surveys Vol. 28:1, pp. 77-80, 1996.

[Forrest et al. 1997] S. Forrest, ***Computer Immunology,*** Communications of the ACM, 40(19), pp. 88-96., 1997.

[Foster 1995] J. A. Foster, ***On Test Suits***, Lecture Notes on Genetic Algorithms, 1995.

[Fujita et al. 1993] K. Fujita, S. Akagi, N. Hirokawa, ***Hybrid Approach for Optimal Nesting using Genetic Algorithm and a Local Minimization Algorithm,*** In Advances in Design Automation (1), pp. 477-484, 1993.

[Furuhashi et al. 1994] T. Furuhashi, K. Nakaoka, Y. Uchikawa, ***A New Approach to Genetic Based Machine Learning and an Efficient Finding of Fuzzy Rule***, In T. Furuhashi editor, Proceedings of the 1994 IEEE/Nagoya-University World Wisepersons

Workshop (WWW'94). Lecture Notes in Artificial Intelligence, vol. 1011, 173-189, Springer Verlag.

[Furuya et al. 1993] H. Furuya, R. T. Haftka, ***Genetic Algorithms for Placing Actuators on Space Structures,*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 536-542, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Goldberg et al. 1985] D. E. Goldberg, R. Lingle, ***Alleles, Loci and the Traveling Salesman Problem.*** In J. J. Grefenstette editor, Proceedings of the First International Conference on Genetic Algorithms, pp. 154-159, Pittsburgh, PA, Lawrence Erlbaum Associates, 1985.

[Goldberg 1989] David E. Goldberg, ***Genetic Algorithms in Search, Optimisation and Machine Learning.*** Addison-Wesley Publishing Company, Inc, 1989.

[Gould et al. 1996] J. L. Gould, W. T. Keeton, ***Biological Science***, W. W. Norton & Company, 1996.

[Gruau 1993] F. Gruau, ***Genetic Synthesis of Modular Neural Networks,*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 318-325, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Güvenir et al. 1993] H. A. Güvenir, I. Sirin, ***A Genetic Algorithm for Classification by Feature Recognition,*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 543-548, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Hadad et al. 1997] B. S. Hadad, C. F. Eick, ***Supporting Polyploidy in Genetic Algorithms Using Dominance Vectors***, In Proceedings of the Sixth Annual Conference on Evolutionary Programming, 1997.

[Handley 1993] S. Handley, ***Automated Learning of a Detector for α-helices in Protein Sequences via Genetic Programming,*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 271-278, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Harp et al. 1991] S. A. Harp, T. Samad, ***Genetic Synthesis of Neural Network Architecture.*** Handbook of Genetic Algorithms, ch. 15, pp. 202-221. 1991.

[Harries et al. 1997] K. Harries, P. Smith, ***Exploring Alternative Operators and Search Strategies in Genetic Programming.*** In Proceedings of the Second International Conference on Genetic Programming, pp. 147-155, Stanford University, California. Morgan Kaufmann, 1997.

[Harvey 1996] I. Harvey, ***The Microbial Genetic Algorithm.*** Submitted as a Letter to Evolutionary Computation, January 1996.

[Haupt et al. 1998] R. L. Haupt, S. E. Haupt, ***Practical Genetic Algorithms***, John Wiley & Sons, 1998.

[Haynes 1998] T. Haynes, ***Duplication of Coding Segments in Genetic Programming.*** In Proceedings of the AAAI, 1998.

[Hillis 1991] W. D. Hillis, ***Co-evolving parasites improve simulated evolution as an optimization procedure***. In C.G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, editors, Artificial Life II, SFI Studies in the Science of Complexity, Addison - Wesley, 1991.

[Hightower et al. 1995] R. Hightower, S. Forrest, A.S. Perelson, ***The Evolution of Emergent Organization in Immune System Gene Libraries***. In L.J. Eshelman, ed., Proceedings of the Sixth International. Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco, CA, pp. 344-350, 1995.

[Hightower et al. 1996] R. Hightower, S. Forrest, A.S. Perelson***. The Baldwin Effect in the Immune System: Learning by Somatic Hypermutation***. In R.K. Belew and M. Mitchell, eds., Adaptive Individuals in Evolving Populations, pp. 159-167, Addison-Wesley, Reading, MA, 1996.

[Hinterding et al. 1995] R. Hinterding, H. Gielewski. T. C. Peachey, ***The Nature of Mutation in Genetic Algorithms.*** In L. J. Eshelman editor, Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 65-72, Pittsburgh, Morgan Kaufmann Publishers, Inc, 1995.

[Hinton et al. 1987] G. E. Hinton, S. J. Nowlan, ***How Learning Can Guide Evolution.*** In Complex Systems, 1, pp. 495-502, 1987.

[Holland 1985] J. H. Holland, ***Properties of the Bucket-Brigade Algorithm,*** In J. J. Grefenstette editor, Proceedings of the First International Conference on Genetic Algorithms, pp. 1-7, Pittsburgh, PA, Lawrence Erlbaum Associates, 1985.

[Holland 1987] J. H. Holland, ***Genetic Algorithms and Classifier Systems: Foundations and Future Directions, ,*** In J. J. Grefenstette editor, Proceedings of the Second International Conference on Genetic Algorithms, pp. 82-89, Pittsburgh, PA, Lawrence Erlbaum Associates, 1987.

[Holland 1992] J. H. Holland, ***Adaptation in Natural and Artificial Systems: an introductory analysis with applications to biology, control and artificial intelligence***, MIT Press, 1992.

[Holland 1993] J. H. Holland, ***Echoing Emergence: objectives, rough definitions and speculations for Echo-class models.*** Technical Report 93-04-023, Santa Fe Institute, 1993.

[Homaifar et al. 1993] A. Homaifar, S. Guan, G. Liepins, ***A New Approach to the Traveling Salesman Problem by Genetic Algorithms***. In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 460-466, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Hraber et al. 1996] P. Hraber, T. Jones, S. Forrest, ***The Ecology of Echo,*** Artificial Life, 1996.

[Jefferson et al. 1991] D. Jefferson, R. Collins, C. Cooper, M. Dyer, M. Flowers, R. Kort, C. Taylor, A. Wang***, Evolution as a Theme in Artificial Life: The Genesis/Tracker System***. In C.G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, editors, Artificial Life II, SFI Studies in the Science of Complexity, Addison - Wesley, 1991.

[Jones et al. 1993] T. Jones, S. Forrest, ***An Introduction to SFI Echo***. Working Paper 93-12-074, Santa Fe Institute, 1993.

[Jones 1995] T. Jones, ***Crossover, Macromutation and Population-based Search.*** In L. J. Eshelman editor, Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 73-80, Pittsburgh, Morgan Kaufmann Publishers, Inc, 1995.

[Juliff 1993] K. Juliff, ***A Multi-Chromosome Genetic Algorithm for Pallet Loading.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 467-473, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Kahng et al. 1995] A. B. Kahng, B. R. Moon, ***Toward More Powerful Recombinations.*** In L. J. Eshelman editor, Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 96-103, Pittsburgh, Morgan Kaufmann Publishers, Inc, 1995.

[Kidwell 1993] M. D. Kidwell, ***Using Genetic Algorithms to Schedule Distributed Tasks on a Bus-based System.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 368-374, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Kim et al. 1995] J-H. Kim, H-S. Shim, ***Evolutionary Programming-based Optimal Robust Locomotion Control of Autonomous Mobile Robots,*** In J. R. McDonnell, R. G. Reynolds, D. B. Fogel, editors, Proceedings of the Fourth Annual Conference on Evolutionary Programming, pp. 631-644, MIT Press, 1995.

[Koon et al. 1995] G. H. Koon, A. V. Sebald, ***Some Interesting Test Functions for Evaluating Evolutionary Programming Strategies***, In Proceedings of the Fourth Annual Conference on Evolutionary Programming, San Diego, CA, 1995.

[Koza 1992] J. Koza, ***Genetic Programming: on the Programming of Computers by Means of Natural Selection,*** MIT Press, 1992.

[Koza 1993] J. Koza, ***Genetic Evolution and co-evolution of computer programs.*** In C.G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, editors, Artificial Life II, SFI Studies in the Science of Complexity, Addison - Wesley, 1991.

[Lindgren 1991] K. Lindgren, ***Evolutionary Phenomena in Simple Dynamics***. In C.G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, eds., Artificial Life II, SFI Studies in the Science of Complexity, Addison - Wesley, 1991.

[Lindgren et al. 1994] K. Lindgren, M. G. Nordahl, ***Artificial Food Webs***. In C.G. Langton, ed., Artificial Life III, SFI Studies in the Science of Complexity, Addison - Wesley, 1994.

[Lindgren et al. 1995] K. Lindgren, M. G. Nordahl***, Cooperation and Community Structure in Artificial Ecosystems***. In C.G. Langton, ed., Artificial Life: an Overview, pags. 15- 37, MIT Press/Bradford Books, 1995.

[Levenick 1991] J. R. Levenick, ***Inserting Introns Improves Genetic Algorithm Success Rate: Taking a Cue from Biology,*** In Belew. L. Booker editors, Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 123-127, San Diego, California, Morgan Kaufmann Publishers, Inc, 1991.

[Littman 1996] M. L. Littman, ***Simulations Combining Evolution and Learning***. In R. K. Belew, M. Mitchell, ed., Adaptive Individuals in Evolving Populations, SFI Studies in the Science of Complexity, Vol. XXVI, Addison-Wesley, 1996.

[Louis et al. 1991] S. J. Louis, G. J. E. Rawlins, ***Designer Genetic Algorithms: Genetic Algorithms in Structure Design***. In R. Belew. L. Booker editors, Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 53-60, San Diego, California, Morgan Kaufmann Publishers, Inc, 1991.

[Lucasius et al. 1991] C. B. Lucasius, M. J. Blommers, L. M. Buydens, G. Kateman, ***A Genetic Algorithm for Conformational Analysis of DNA,*** In L. Davis editor, Handbook of Genetic Algorithms, ch. 18, pp. 251-281, Van Nostrand Reinhold, 1991.

[MacLennan 1991] B. MacLennan***, Synthetic Ethology: An Approach to the Study of Communication***. In Christopher G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, eds., Artificial Life II, SFI Studies in the Science of Complexity, Addison-Wesley, 1991.

[MacLennan et al. 1994] B. J. MacLennan, G. M. Burghardt, ***Synthetic Ethology and the Evolution of Cooperative Communication***. In Adaptive Behavior, 2(2), pp. 161 – 188, MIT Press, 1994.

[Manela et al. 1993] M. Manela, N. Thornhill, J. A. Campbell, ***Fitting Spline Functions to Noisy Data using a Genetic Algorithms,*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 549-556, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Mathias et al. 1992] K. Mathias, D. Whitley, ***Genetic Operators, the Fitness Landscape and the Traveling Salesman Problem***, In R. Männer and B. Manderick, eds., Parallel Problem Solving from Nature-PPSN 2, pp. 219-228. North Holland-Elsevier, 1992.

[Mayley 1996] G. Mayley, ***The Evolutionary Cost of Learning***. In P. Maes, M. J. Mataric, J-A. Meyer, J. Pollack, S. W. Wilson eds, From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, pp. 458-467, Cambridge, MA, MIT Press/Bradford books, 1996.

[Mayley 1997] G. Mayley, ***Landscapes, Learning Costs and Genetic Assimilation***. In Evolutionary Computation, 4(3), pp. 213 – 234, MIT Press, 1997.

[Maynard-Smith 1987] J. M. Smith, ***When Learning Guides Evolution***. In Nature, 329, pp. 761-762, 1987.

[McDonnell et al. 1992] J. R. McDonnell, B. L. Andersen, W. C. Page, F. G. Pin, ***Mobile Manipulator Configuration Optimization Using Evolutionary Programming***. In D. B. Fogel, W. Atmar, editors, Proceedings of the First Annual Conference on Evolutionary Programming, pp. 52-62, 1992.

[McDonnell et al. 1995] J. R. McDonnell, D. E. Waagen, ***An Empirical Study of Recombination in Evolutionary Search***, In Proceedings of the Fourth Annual Conference on Evolutionary Programming, San Diego, CA, 1995.

[Menczer et al. 1992] F. Menczer, D. Parisi, ***A Model for Emergence of Sex in Evolving Networks: Adaptive Advantage or Random Drift?***. In F. J. Varela, P. Bourgine, eds., Toward a Practice of Autonomous Systems: Proceedings of the First Conference on Artificial Life, pp. 337-345, Cambridge, MA. MIT Press/Bradford Books, 1992.

[Menczer et al. 1998] F. Menczer, R. Belew, ***Adaptive Information Agents in Distributed Textual Environments***. In Proceedings of the Second International Conference on Autonomous Agents, 1998.

[Michalewicz 1993] Z. Michalewicz, ***A Hierarchy of Evolution Programs: an experimental study.*** In Evolutionary Computation, 1, pp. 51-76, 1993.

[Michalewicz 1999] Z. Michalewicz, ***Genetic Algorithms + Data Structures = Evolution Programs.*** Third, Revised and Extended Edition, Springer-Verlag, 1999.

[Miller et al. 1989] G. F. Miller, P. M. Todd, S. U. Hegde, ***Designing Neural Networks Using Genetic Algorithms.*** In J. D. Schaffer editor, Proceedings of the Third International Conference on Genetic Algorithms, pp. 379-384, Fairfax, VA, Morgan Kaufmann Publishers, Inc, 1989.

[Miller 1994] G. F. Miller, ***Exploiting Mate Choice in Evolutionary Computation: Sexual Selection as a Process of Search, Optimization and Diversification,*** In Lectures Notes on Computer Science, Vol. 865, 1994.

[Miller et al. 1995] G. F. Miller, P. M. Todd, ***The Role of Mate Choice in Biocomputation: Sexual Selection as a Process of Search, Optimization and Diversification.*** In Lectures Notes in Computer Science, Vol. 899, 1995.

[Miller 1996] G. F. Miller, ***A Review of Sexual Selection and Human Evolution: How Mate Choice shaped Human Nature,*** In Evolution and Human Behavior: Ideas, Issues and Applications, L. Erlbaum, 1996.

[Miller 1989] J. H. Miller, ***The Co-evolution of Automata in the Repeated Prisoner's Dilemma.*** Technical Report 89-003, Santa Fe Institute, 1989.

[Mitchell et al. 1994] M. Mitchell, S. Forrest, ***Genetic Algorithms and Artificial Life***, (SFI Working Paper, 93-11-072). In Artificial Life 1 (3), pp. 267-289, 1994.

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

[Mühlenbein 1989], H. Mühlenbein, ***Parallel Genetic Algorithms, population genetics and combinatorial optimization.*** In J. D. Schaffer editor, Proceedings of the Third International Conference on Genetic Algorithms, pp. 416-421, Morgan Kaufmann Publishers, Inc, 1989.

[Mühlenbein 1993] H. Mühlenbein, ***Evolutionary Algorithms: Theory and Applications*** , TR. 1993.

[Nawa et al. 1997] N. Nawa, T. Hashiyama, T. Furuhashi, Y. Uchikawa, ***A Study on Fuzzy Rules Using Pseudo-Bacterial Genetic Algorithm with Adaptive Operator***, In Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC'97), 589-593. IEEE Press, 1997.

[Nawa et al. 1998] N. Nawa, T. Furuhashi, ***Bacterial Evolutionary for Fuzzy System Design***, In Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics. IEEE Press, 1998.

[Nawa et al. 1999] N. Nawa, T. Furuhashi, T. Hashiyama, Y. Uchikawa, ***A Study of the Discovery of Relevant Fuzzy Rules Using Pseudo-Bacterial Genetic Algorithm***, In IEEE Transactions on Industrial Electronics, 1999.

[Neves et al. 1999] A. Neves, A. Silva, E. Costa, ***Evolutionary Path Planning for Nonholonomic Robots***. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, R. E. Smith, editors, GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference, Orlando, Florida, USA, 13-17 July, 1999.

[Ng et al. 1995] K. P. Ng, K. C. Wong, ***A New Diploid Scheme and Dominance Change Mechanism for Non-Stationary Function Optimization.*** In L. J. Eshelman editor, Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 159-166, Pittsburgh, Morgan Kaufmann Publishers, Inc, 1995.

[Nolfi et al. 1994] S. Nolfi, J. L. Elman, D. Parisi, ***Learning and Evolution in Neural Networks***. In Adaptive Behavior, 3(1), pp. 5–28, MIT Press, 1994.

[Oates et al. 1999] M. Oates, D. Corne, R. Loader, ***Skewed Crossover and the Dynamic Distributed Database Problem***, In A. Dobnikar, N. Steele, D. Pearson, R. Albrecht (eds.), Proceedings of the Fourth International Conference on Neural Networks and Genetic Algorithms (ICANNGA 99), Springer-Verlag, 1999.

[Odutayo 1996] M. A. Odutayo, ***An Empirical Study of Crossover and Conjugation using the De Jong's Test Bed of Optimization Functions.*** M. Eng. Degree in Software Engineering Dissertation. (tese não publicada), City University, 1996.

[Oliver 1993] J. R. Oliver, ***Discovering Individual Decision Rules: an Application of Genetic Algorithms,*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 216-222, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Packard 1988] N. H. Packard, ***Intrinsic Adaptation in a Simple Model for Evolution***. In C.G. Langton, ed., Artificial Life, SFI Studies in the Science of Complexity, Addison-Wesley, 1988.

[Page et al. 1992] W. C. Page, J. R. McDonnell, B. L. Andersen, ***An Evolutionary Programming Approach to Multi-dimensional Path Planning***. In D. B. Fogel, W. Atmar, editors, Proceedings of the First Annual Conference on Evolutionary Programming, pp. 63-70, 1992.

[Parisi et al. 1992] D. Parisi, S. Nolfi, F. Cecconi***, Learning, Behavior and Evolution.*** In F. J. Varela, P. Bourgine, eds., Toward a Practice of Autonomous Systems: Proceedings of the First Conference on Artificial Life, pp. 207-216, Cambridge, MA. MIT Press/Bradford Books, 1992.

[Parisi et al. 1996] D. Parisi, S. Nolfi, ***The Influence of Learning in Evolution***. In R. K. Belew, M. Mitchell, eds., Adaptive Individuals in Evolving Populations, SFI Studies in the Science of Complexity, Vol. XXVI, Addison-Wesley, 1996.

[Parsons et al. 1995] R. Parsons, S. Forrest, C. Burks, ***Genetic Algorithms, Operators, and DNA Fragment Assembly.*** In Machine Learning, pp. 1-24, 1995.

[Pereira et al. 1997] F. B. Pereira, E. Costa, ***The Influence of Learning in the Optimization of Royal Road Functions***. In Proceedings of MENDEL'97, 3rd International Mendel Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks, Rough Sets, pp. 244-249, Brno, Czech Republic,1997.

[Pereira et al. 1999], F. B. Pereira, E. Costa, ***The Influence of Learning in the Behavior of Information Retrieval Adaptive Agentes,*** Submitted to the Nineth Encontro Português em Inteligência Artificial (EPIA 99).

[Perelson et al. 1996] A. Perelson, R. Hightower, S. Forrest***, Evolution and Somatic Learning in V-Region Genes***. Research in Immunology Vol. 147, pp. 202-208, 1996.

[Petrowsky 1997] A. Petrowski, ***A New Selection Operator Dedicated to Speciation.*** In T. Bäck editor, Proceedings of the Seventh International Conference on Genetic Algorithms, pp. 144-151, East Lansing, MI, Morgan Kaufmann Publishers, Inc, 1997.

[Powell et al. 1993] D. Powell, M. M. Skolnick, ***Using Genetic Algorithms in Engineering Design Optimization with Non-linear Constraints,*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 424-431, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Punch et al. 1993] W. F. Punch, E. D. Goodman, M. Pei, L. Chia-Shun, P. Hovland, R. Enbody, ***Further Research on Feature Selection and Classification using Genetic Algorithms***, In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 557-564, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Rahmani et al. 1993] A. T. Rahmani, N. Ono, ***A Genetic Algorithm for Channel Routing Problem.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 494-498, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Ray 1991a] T. S. Ray, ***It is alive, or it is GA?***. In R. K. Belew, L. B. Booker, eds., Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 527-534, Morgan Kaufmann Publishers, Inc, 1991.

[Ray 1991b] T. S. Ray, ***An Approach to the Synthesis of Life***. In C.G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, eds., Artificial Life II, SFI Studies in the Science of Complexity, Addison - Wesley, 1991.

[Reeves 1993] C. R. Reeves, ***Using Genetic Algorithms with Small Populations.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 92-99, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Romaniuk 1993] S. G. Romaniuk, ***Evolutionary Growth Perceptrons.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 334-341, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Roosen et al. 1992] P. Roosen, F. Meyer, ***Determination of Chemical Equilibria by Means of an Evolution Strategy***, In R. Männer and B. Manderick, eds., Parallel Problem Solving from Nature-PPSN 2, pp. 219-228. North Holland-Elsevier, 1992.

[Russell 1998] P. J. Russell, ***Genetics***. Addison-Wesley, 5th edition, 1998.

[San-Martin et al. 1993] R. San Martin, J. P. Knight, ***Genetic Algorithms for Optimization of Integrated Circuit Synthesis.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 432-438, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Schaffer et al. 1991] J. D. Schaffer, L. J. Eshelman, ***On Crossover as an Evolutionarily Viable Strategy***. In R. K. Belew, L. B. Booker editors, Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 61-68, San Diego, CA, Morgan Kaufmann Publishers, Inc, 1991.

[Schaffer et al. 1993] J. D. Schaffer, L. J. Eshelman, ***Designing Multiplierless Digital Filters Using Genetic Algorithms.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 439-444, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Simões et al. 1999a] A. Simões, E. Costa, ***Transposition: A Biologically Inspired Mechanism to Use with Genetic Algorithms,*** In A. Dobnikar, N. Steele, D. Pearson, R. Albrecht (eds.), Proceedings of the Fourth International Conference on Neural Networks and Genetic Algorithms (ICANNGA 99), Springer-Verlag, 1999.

[Simões et al. 1999b] A. Simões, E. Costa, ***Enhancing Transposition Performance***, In Proceedings of the 1999 Congress on Evolutionary Computation (CEC 99), Piscataway, NJ: IEEE Press, 1999.

[Simões et al. 1999c] A. Simões, E. Costa, ***Transposition versus Crossover: An Empirical Study,*** In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, R. E. Smith (eds.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 99), Orlando, Florida, USA, San Francisco, CA: Morgan Kaufmann, 1999.

[Simpson 1953] G. G. Simpson, ***The Baldwin Effect***. In Evolution, 7, pp. 110-117, 1953.

[Smith et al. 1993] A. E. Smith, D. M. Tate, ***Genetic Optimization Using a Penalty Function.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 499-505, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Smith 1985] D. Smith, ***Bin-Packing with Adaptive Search.*** In J.J. Grefenstette editor, Proceedings of the First Conference on Genetic Algorithms, Lawrence Elrbaum Associates, 1985.

[Smith 1996a] P. Smith, ***Finding Hard Satisfiability Problems Using Bacterial Conjugation***. Presented at the AISB96 Workshop on Evolutionary Computing. University of Sussex April 1996.

[Smith 1996b] P. Smith, ***Conjugation - A Bacterially Inspired Form of Genetic Recombination***. In Late Breaking Papers of the First International Conference on Genetic Programming. Stanford University, California, July 1996.

[Smith et al. 1992] R. E. Smith, S. Forrest, A. S. Perelson. ***Population Diversity in an Immune System Model: Implications for Genetic Search,*** In FOGA'92, Foudations of Genetic Algorithms, pp. 24-29, 1992.

[Smith et al. 1993] R. E. Smith, S. Forrest, A. S. Perelson. ***Searching for Diverse, Cooperative Populations with Genetic Algorithms.*** Evolutionary Computation, 1(2), pp. 127-149, 1993.

[Somayaji et al. 1997] A. Somayaji, S. Hofmeyr, S. Forrest, ***Principles of a Computer Immune System.*** In New Security Paradigms Workshop, 1997.

[Spears et al. 1990] W. M. Spears, K. A. De Jong, ***An Analysis of Multi-Point Crossover.*** In Proceedings of the Foundations of Genetic Algorithms Workshop, pp. 301-315, 1990.

[Spears et al. 1991a] W. M. Spears, K. De Jong, ***On the Virtues of Parameterized Uniform Crossover***. In R. K. Belew, L. B. Booker editors, Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 230-236, San Diego, CA, Morgan Kaufmann Publishers, Inc, 1991.

[Spears et al. 1991b] W. M. Spears, V. Anand, ***A Study of Crossover Operators in Genetic Programming***. In Proceedings of the Sixth International Symposium on Methodologies for Intelligent Systems, pp. 409-418, 1991.

[Spears 1992a] W. M. Spears, ***Adapting Crossover in a Genetic Algorithm*** (Technical Report AIC-92-025). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research on Artificial Intelligence, 1992.

[Spears 1992b] W. M. Spears, ***Crossover or Mutation?***. In Proceedings of Foundations of Genetic Algorithms Workshop, pp. 221-237, 1992.

[Spears et al. 1993] W. M. Spears, K. A. De Jong, T. Bäck, D. Fogel, and H. de Garis. ***An Overview of Evolutionary Computation***. In Proceedings of the European Conference on Machine Learning, v667, pp. 442-459, 1993.

[Spears 1994] W. M. Spears, ***Simple Subpopulation Schemes***. In Proceedings of the Third Conference on Evolutionary Programming, World Scientific, 1994.

[Spears 1995] W. M. Spears, ***Adapting Crossover in Evolutionary Algorithms***. In Proceedings of the Fourth Conference on Evolutionary Programming, San Mateo, CA, pp. 367-384, 1995.

[Spencer 1993] G. F. Spencer, ***Automatic Generation of Programs for Crawling and Walking***, In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 654, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Starkweather et al. 1991] T. Starkweather, S. McDaniel, K. Mathias, D. Whitley, C. Whitley, ***A Comparison of Genetic Sequencing Operators***. In R. K. Belew,L. B. Booker editors, Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 69-76, San Diego, CA, Morgan Kaufmann Publishers, Inc, 1991.

[Syswerda 1989] G. Syswerda, ***Uniform Crossover in Genetic Algorithms***. In J. D. Shaffer editor, Proceedings of the Third International Conference on Genetic Algorithms, pp. 2-9, Morgan Kaufmann Publishers, Inc, 1989.

[Syswerda 1991a] G. Syswerda, ***A Study of Reproduction in Generational and Steady-state Genetic Algorithms***. In G. Rawlins, ed., Foundations of Genetic Algortitms, Morgan Kaufmann, 1991.

[Syswerda 1991b] G. Syswerda, ***Schedule Optimization Using Genetic Algorithms.*** In L. Davis editor, Handbook of Genetic Algorithms, ch. 21, pp. 332-349, Van Nostrand Reinhold, 1991.

[Syswerda 1993] G. Syswerda, ***Simulated Crossover in Genetic Algorithms***. In L. Darrell Whitley, editor, Foundations of Genetic Algortitms 2, pp. 239-255, Morgan Kaufmann, 1993.

[Tackett 1993] W. A. Tackett, ***Genetic Programming for Feature Discovery and Image Discrimination***, In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 303-309, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Tanaka et al. 1993] Y. Tanaka, A. Ishiguro, Y. Uchikawa, ***A Genetic Algorithms Application to Inverse Problems in Electromagnetics***, In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 656, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Taylor et al. 1988] C. E. Taylor, D. R. Jefferson, S. R. Turner, S. R. Goldman, ***RAM: Artificial Life for Exploration of Complex Biological Systems***. In C. G. Langton editor, Artificial Life, SFI Studies in the Science of Complexity, Addison - Wesley, 1988.

[Taylor et al. 1995] C. E. Taylor, D. R. Jefferson, ***Artificial Life as a Tool for Biological Inquiry***. In C. G. Langton editor, Artificial Life: an Overview, pp. 1-13, MIT Press/Bradford Books, 1995.

[Thangia et al. 1993] S. R. Thangia, R. Vinayagamoorthy, A. V. Gubbi, ***Vehicle Routing With Time Deadlines Using Genetic and Local Algorithms.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 506-513, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Thierens 1997] D. Thierens, ***Selection Schemes, Elitist Recombination and Selection Intensity.*** In T. Bäck editor, Proceedings of the Seventh International Conference on Genetic Algorithms, pp. 152-159, East Lansing, MI, Morgan Kaufmann Publishers, Inc, 1997.

[Todd et al. 1993] Peter M. Todd, G. F. Miller, ***Parental Guidance Suggested: How Parental Imprintings Evolves Through Sexual Selection as an Adaptive Learning Mechanism.*** In Adaptive Behavior, 2(1), pp. 5-47, MIT Press, 1993.

[Todd 1996] Peter M. Todd, ***Sexual Selection and the Evolution of Learning.*** In R. K. Belew, M. Mitchell, eds., Adaptive Individuals in Evolving Populations. SFI Studies in the Science of Complexity, Vol XXVI, Addison-Wesley, 1996.

[Todd et al. 1997] Peter M. Todd, G. F. Miller, ***Biodiversity through Sexual Selection,*** In Proceedings of the Fifth International Workshop on Artificial Life: Synthesis and Simulation of Living Systems (ALIFE-96), pp. 289-299, MIT Press, 1997.

[Unger et al. 1993] R. Unger, J. Moult, ***A Genetic Algorithm for 2D Protein Folding Simulations,*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 581-588, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Watabe et al. 1993] H. Watabe, N. Okino, ***A Study on Genetic Shape Design***, In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 445-450, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

[Watson 1968] J. D. Watson, ***The Double Helix***, New York:Atheneum, 1968.

[Webb et al. 1996] B. Webb, J. Hallam, ***How to attract females: further robotic experiments in cricket phonotaxis.*** In P. Maes, M. J. Mataric, J-A. Meyer, J. Pollack, S. W. Wilson editors, From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, pp. 75-83, Cambridge, MA, MIT Press/Bradford books, 1996.

[Werner et al. 1991] G. M. Werner, M. G. Dyer***, Evolution of Communication in Artificial Organisms***. In C.G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, eds., Artificial Life II, SFI Studies in the Science of Complexity, Addison - Wesley, 1991.

[Wheeler et al. 1995] M. Wheeler, P de Bourcier***, How Not to Murder Your Neighbor: Using Synthetic Behavioral Ecology to Study Aggressive Signaling***. In Adaptive Behavior, 3(3), pp. 273–309, MIT Press, 1995.

[Whitley 1989] D. Whitley, ***The Genitor and selection pressure: Why rank-based allocation of reproductive trials is best.*** In J. D. Schaffer, ed., Proceedings of the Third International Conference on Genetic Algorithms, pp. 116-121, Morgan Kaufmann Publishers, Inc, 1989.

[Whitley 1993] D. Whitley, ***A Genetic Algorithm Tutorial***. Technical Report CS-93-103, Department of Computer Science, Colorado State University, 1993.

[Whitley et al. 1989] D. Whitley, T. Starkweather , D. Fuquay, ***Scheduling problem and Traveling Salesmen: the genetic edge genetic recombination operator.*** In J. D. Shaffer editor, Proceedings of the Third International Conference on Genetic Algorithms, pp. 133-140, Morgan Kaufmann Publishers, Inc, 1989.

[Whitley et al. 1995] D. Whitley, K. Mathias, S. Rana, J. Dzubera, ***Building Better Test Functions.*** In L. J. Eshelman editor, Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 239-246, Pittsburgh, Morgan Kaufmann Publishers, Inc, 1995.

[Wu et al. 1997] A. S. Wu, R. K. Lindsay, R. Riolo, ***Empirical Observations on the Roles of Crossover and Mutation.*** In T. Bäck editor, Proceedings of the Seventh International Conference on Genetic Algorithms, pp. 362-369, East Lansing, MI, Morgan Kaufmann Publishers, Inc, 1997.

[Yang et al. 1997] J-M. Yang, J-T. Horgn, C-Y. Kao, ***A Continuous Genetic Algorithm for Global Optimization.*** In T. Bäck editor, Proceedings of the Seventh International Conference on Genetic Algorithms, pp. 230-237, East Lansing, MI, Morgan Kaufmann Publishers, Inc, 1997.

[Yoshikawa et al. 1997] T. Yoshikawa, T. Furuhashi and Y. Uchikawa, ***Knowledge Acquisition of Fuzzy Control Rules for Mobile Robots Using DNA Coding Method and Pseudo-Bacterial Genetic Algorithm***. In Proceedings of the First Asia-Pacific Conference on Simulated Evolution and Learning. Lecture Notes in Artificial Intelligence, vol. 1285, 126-135. Spinger-Verlag, 1997.

[Zhang et al. 1993] B-T. Zhang, H. Mühlenbein, ***Genetic Programming of Minimal Neural Nets Using Occam's Razor.*** In S. Forrest editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 342-349, Urbana, Champaigne, Morgan Kaufmann Publishers, Inc, 1993.

RECURSOS OBTIDOS NA INTERNET

Transposable elements
http://www.bio.cornell.edu/biochem/calvo/biobm232/lecture/lecture18/lecture18.html

Transposition in Bacteria - Notes for Two Lectures
http://www.leeds.ac.uk/mbiology/ug/ugteach/gene2020/transpos.htm

Recombination in bacteria - Brief explanation about conjugation, transduction and transformation.
http://www.leeds.ac.uk/mbiology/ug/ugteach/gene2020/transpos.htm

Notes on Microbial Genetics (1997)
http://www.life.uiuc.edu/micro/316/supplement.html

An introduction to Evolutionary Biology - by Chris Colby (1996)
http://www.ics.uci.edu/pub/origins/faq-intro-to-biology
http://wcl-l.bham.ac.uk/origins/faqs/faq-intro-to-biology.html

A Primer on Molecular Genetics - An introduction to molecular biology, genetics (1992)
http://www.bis.med.jhmi.edu/Dan/DOE/intro.html

Prokaryotic Genetics Tools - What biological phenomena do we use as tools to study prokaryotic genetics? (1997)
http://www.bis.med.jhmi.edu/Dan/DOE/intro.html

Bookmarks on Modern Biology - Web links related to Biology
http://info.bio.cmu.edu/Courses/03121/fall/ModBioBookmarks.html

Biology Genetics - Lectures given by Dr John Smith (1998)
http://bssv01.lancs.ac.uk/ADS/second.htm
http://bssv01.lancs.ac.uk/ADS/RJS/Smith.htm

GLOSSÁRIO

Este glossário sintetiza os termos utilizados em português e o seu correspondente em inglês.

Ácidos Nucleicos - Nucleic acids

Adenina - Adenine

ADN (Ácido Desoxirribonucleico) - DNA (Desoxirubonucleic Acid)

Alelo -Alele

Amostragem Estocástica com Substituição (método de selecção) - Stochastic Sampling with Replacing

Amostragem Estocástica Universal - Stochastic Universal Sampling

Bactéria - Bacterium

Baseado na Posição (método de selecção) - Rank based

Bloco Construtor - Building Block

Caixeiro Viajante (problema do) - Traveling Salesman Problem

Centrómero - Centromere

Citosina - Citosin

Comprimento Definidor - Defining Length

Conjugação - Conjugation

Criação -Creation

Cromatídeo - Chromatid

Cromossoma - Chromosome

Crossover - *Crossover*

***Crossover* Desigual** - Unequal *Crossover*

Deleção - Deletion

Diplóide - Diploid

Dominante (alelo) - Dominant

Duplicação - Duplication

Elitista (método de selecção) - Elitist

Em diferido - Off-Line

Em tempo real - On-line

Empacotamento (problema do) - Packing Problem

Escalonamento (problema de) - Scheduling Problem

Esquema - Schema

Estado Estável (método de selecção) - Steady State

Eucariótica (célula) - Eucariotic

Exão - Exon

Factor Multidão - Crowding Factor

Fenótipo - Phenotype

Função de Avaliação - Fitness Function

Gene - Gene

Genoma - Genome

Genótipo - Genotype

Geracional - Generational

Guanina - Guanine

Haplóide - Haploid

Heterozigótico - Heterozygous

Homólogos (cromossomas) - Homologous

Homozigótico - Homozygous

Indeciso (problema) - Indecisive Problem

Intrão - Intron

Inversão - Inversion

Lacuna de Gerações - Generation Gap

Locus - Locus

Mapa de Características - Feature Map

Maximização de uns (problema de)- MAX problem

Mérito - Fitness

Mochila (problema da) - Knapsack Problem

Mutação - Mutation

Não-sobreposição (método de substituição por) - Non-Overlapping

Ordem - Order

Paralelismo Implícito - Implicit Paralelism

Plasmídeo -Plasmid

Poliplóide - Poliploid

Procariótica (célula) - Procariot

Projecto - Design

Proporcional ao Mérito (método de selecção) -

Qualidade - Fitness

Quebra e Fusão - Break and Fusion

Recessivo (alelo) - Recessive

Recombinação - Recombination

Rota de veículos (problema da) - Route Vehicle Problem

Selecção - Selection

Sequência Flanqueadora - Flanking Sequence

Sequências de Inserção - Insertion Sequences

Sobreposição (método de substituição por) - Overlapping

Timina - Timine

Torneio (método de selecção) - Tournament

Transdução - Transduction

Transdução Especializada - Specialized Transduction

Transdução Generalizada - Generalized Transduction

Transformação - Transformation

Translocação - Translocation

Transporte (problema do) - Transportation Problem

Transposão - Transposon

Transposição - Transposition

Transposões Complexos - Complex Transposons

Transposões compostos - Composite Transposons

Truncatura (método de selecção) - Truncature

Vagas (regras) - Fuzzy Rules

Vírus - Virus

ANEXOS

ANEXO 1

REPRESENTAÇÃO GRÁFICA DAS FUNÇÕES DE TESTE

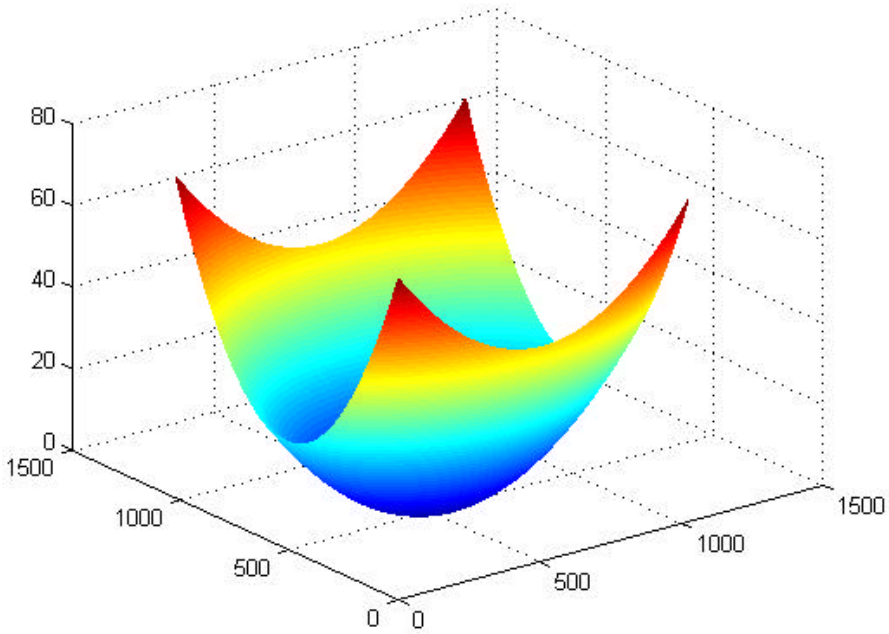


Figura 1 - Função de teste F1 de De Jong (F1)

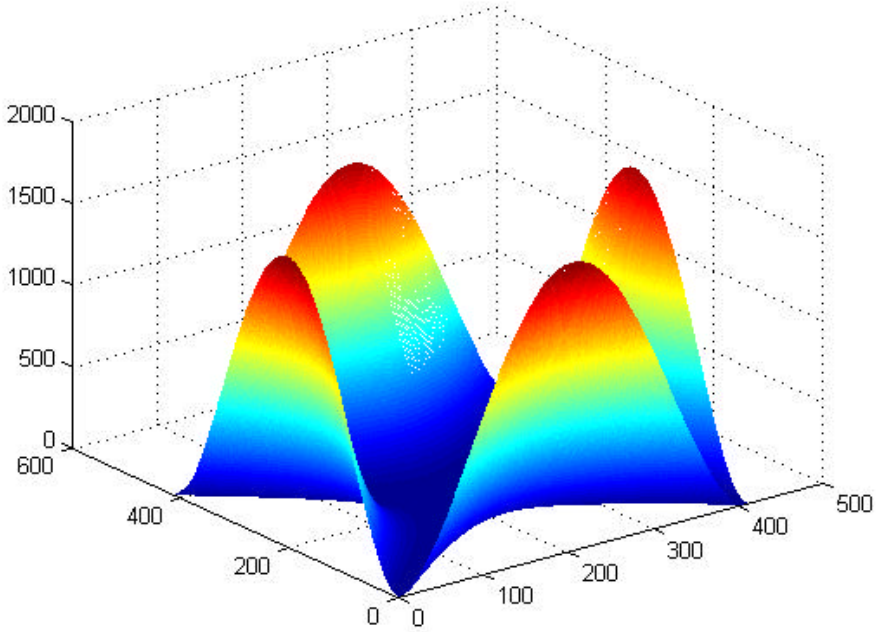


Figura 2 - Função de teste F2 de De Jong (F2)

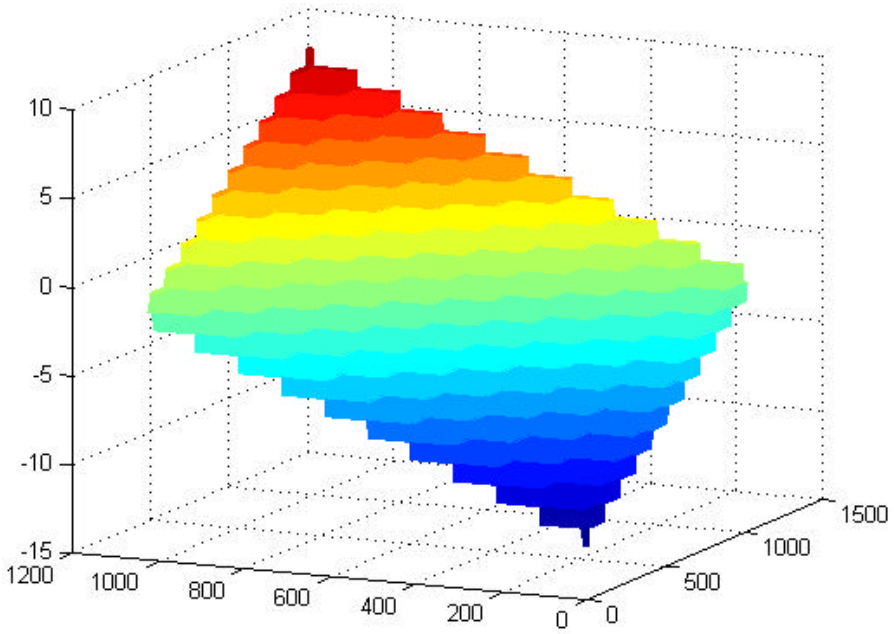


Figura 3 - Função de teste F3 de De Jong (F3)

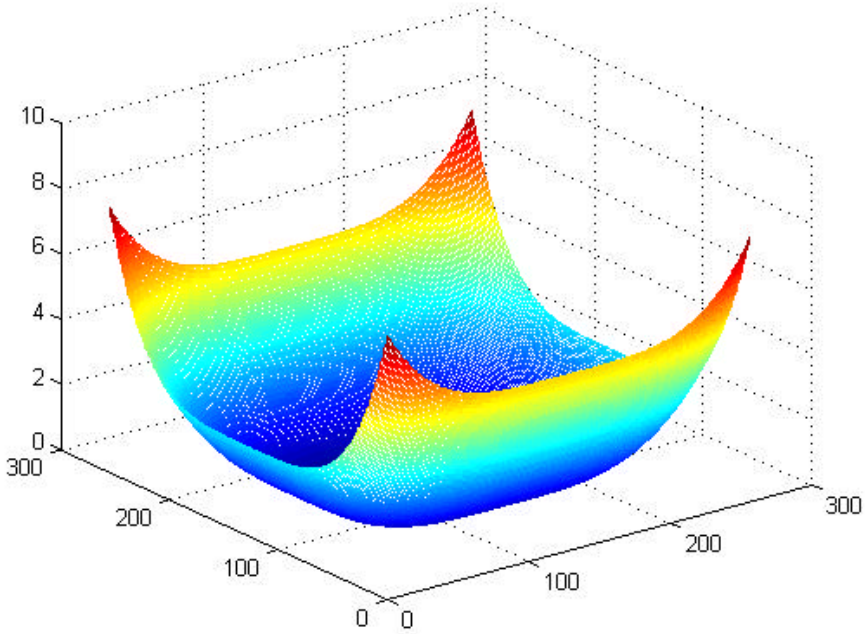


Figura 4 - Função de teste F4 de De Jong (F4)

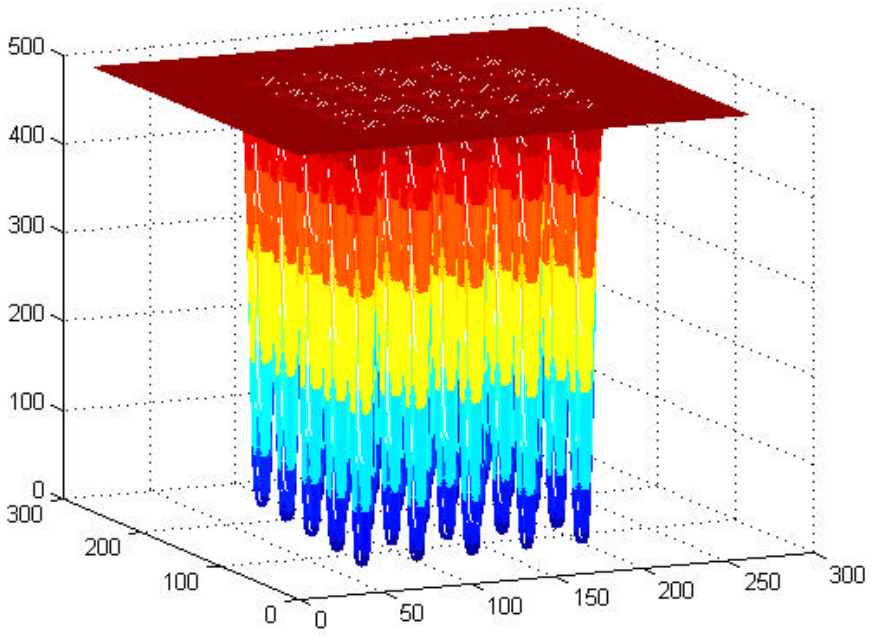


Figura 5 - Função de teste F5 de De Jong (F5)

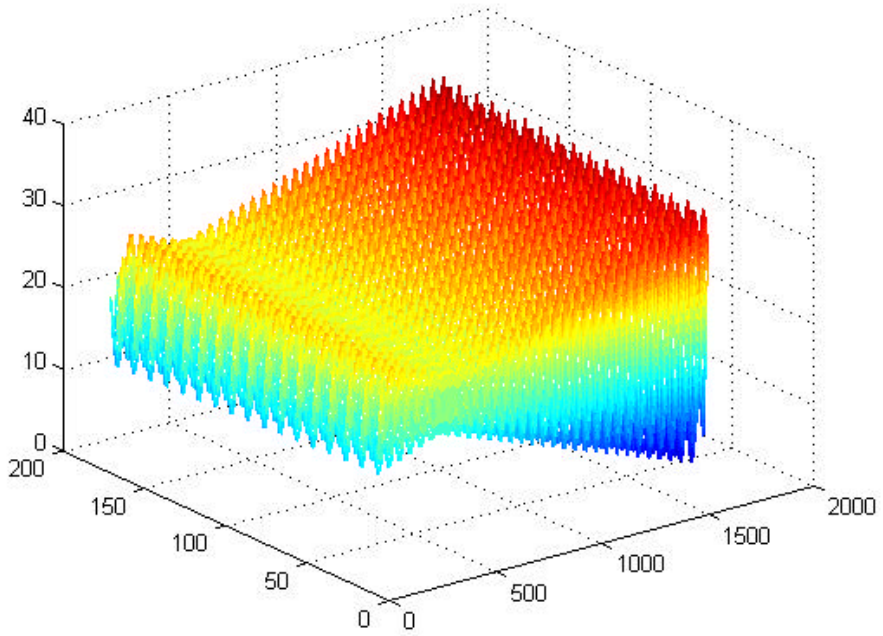


Figura 6 - Função de Michalewicz (F6)

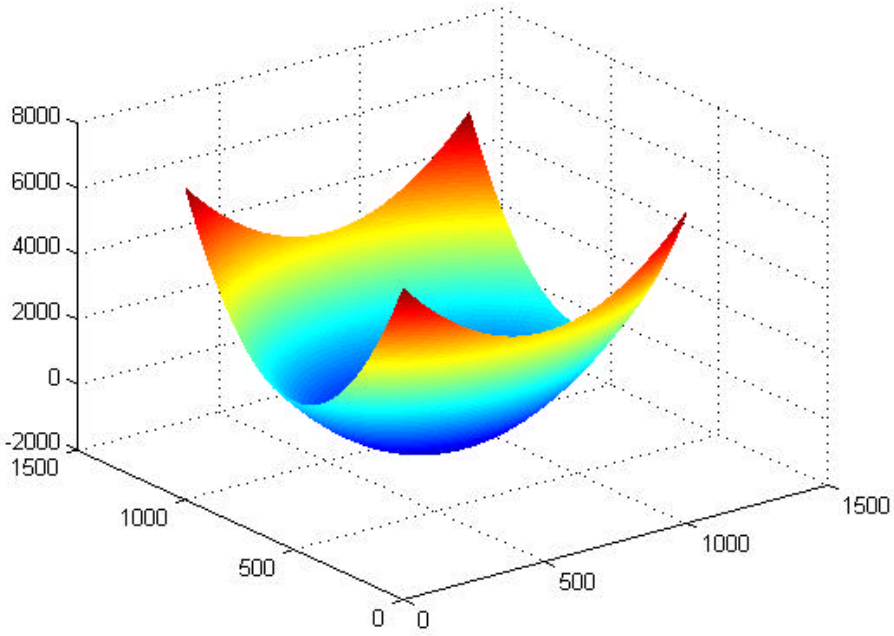


Figura 7 - Função de Bohachevsky I (F7)

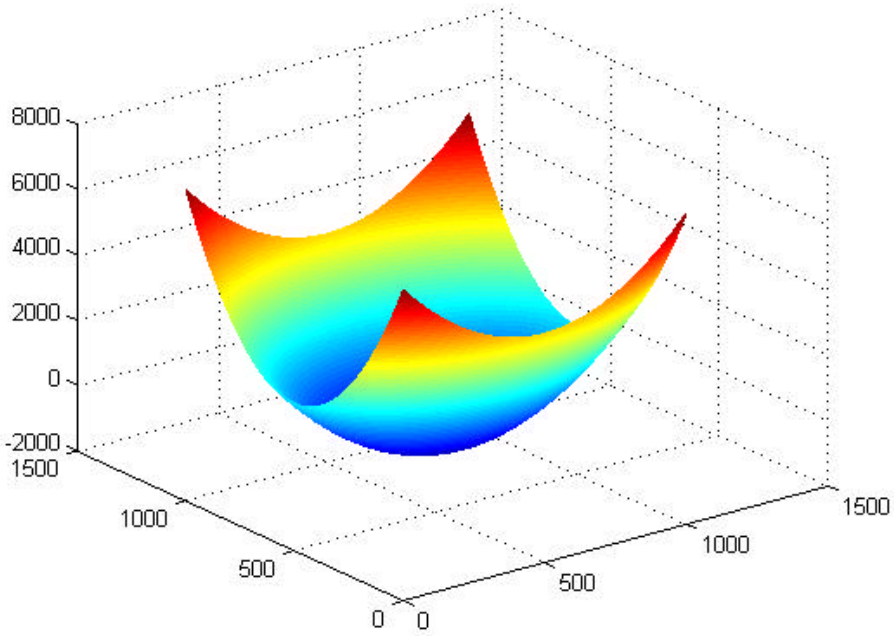


Figura 8 - Função de Bohachevsky II (F8)

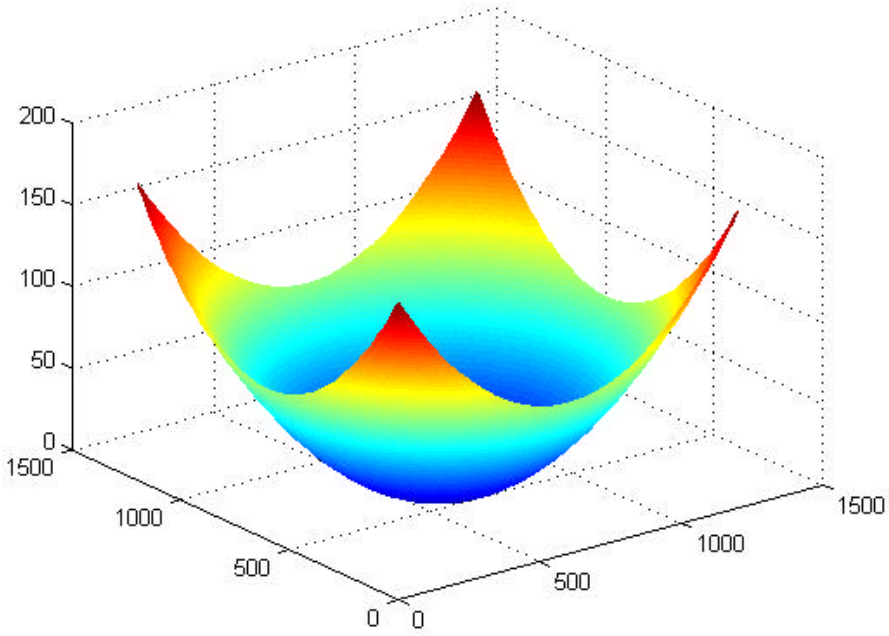


Figura 9 - Função de Griewangk (F9)

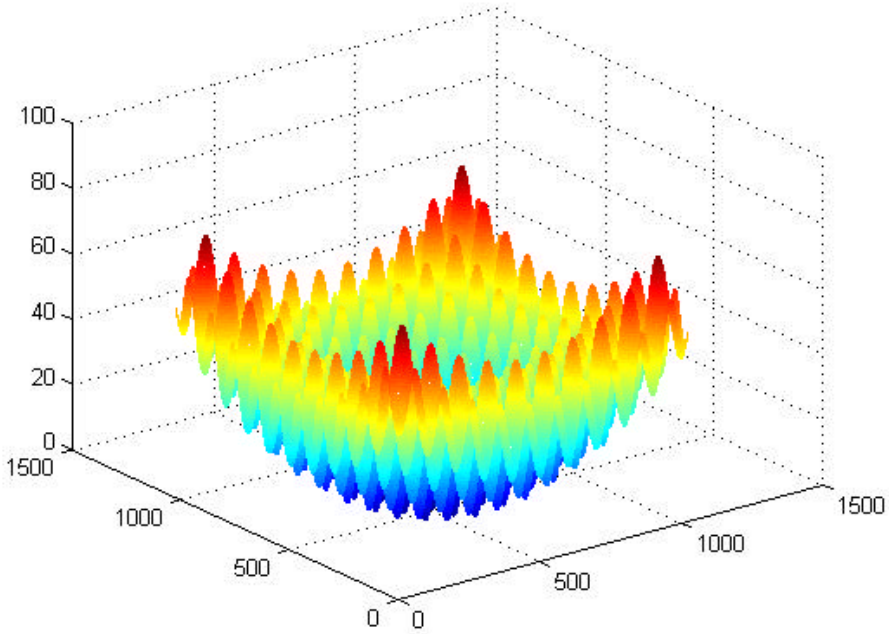


Figura 10 - Função de Rastrigin (F10)

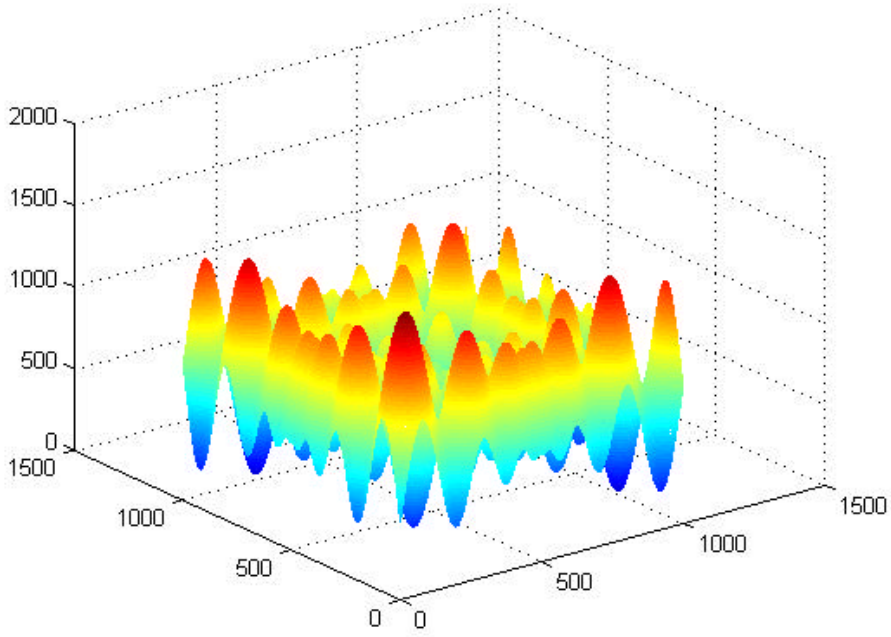


Figura 11 - Função de Schwefel (F11)

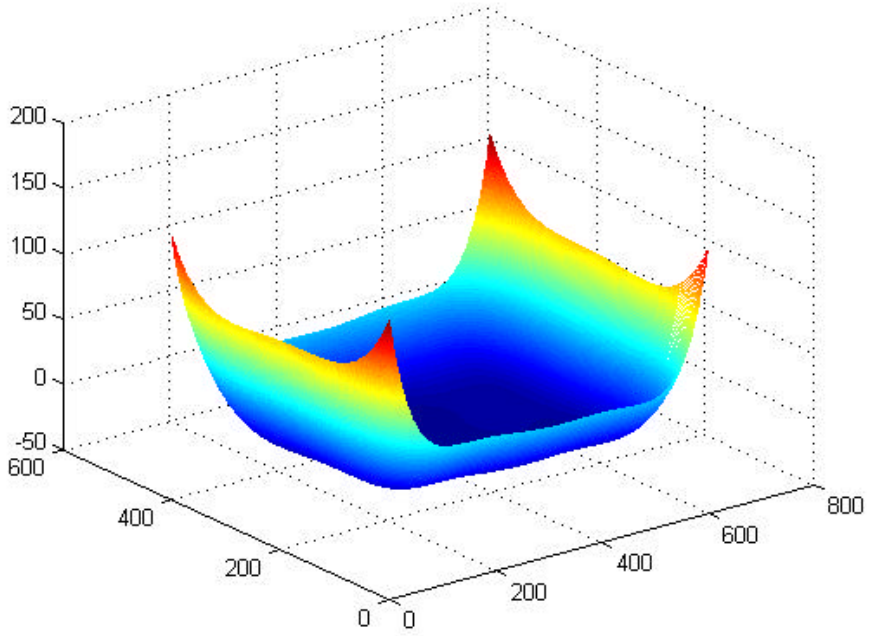


Figura 12 - Função 6-Hump Camelback (F12)

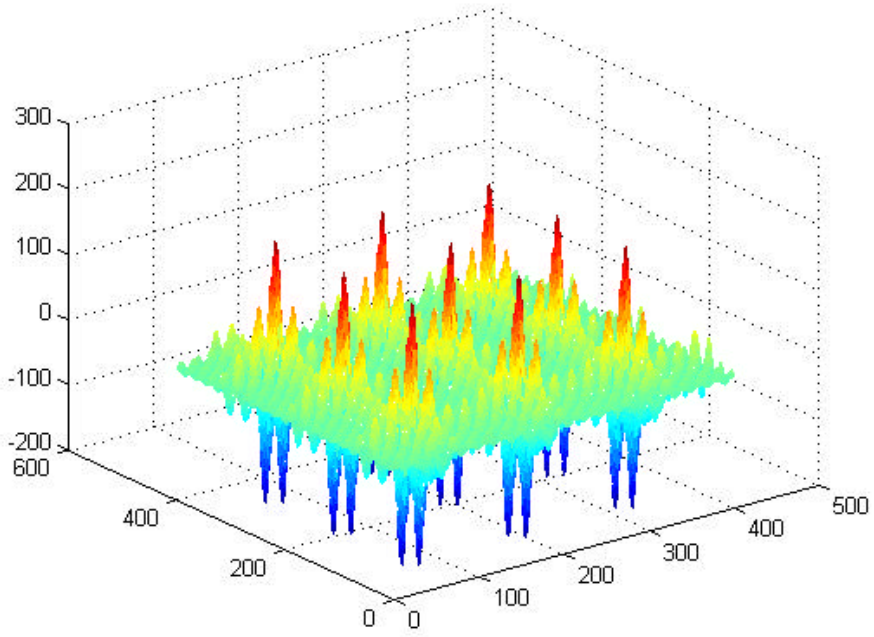


Figura 13 - Função de Shubert (F13)

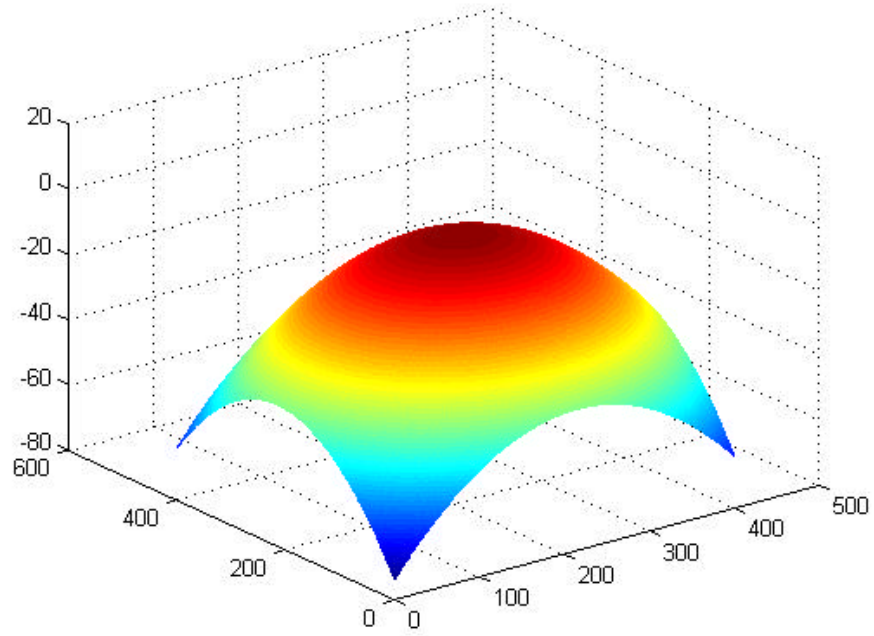


Figura 14 - Função N-Dimensional (F14)

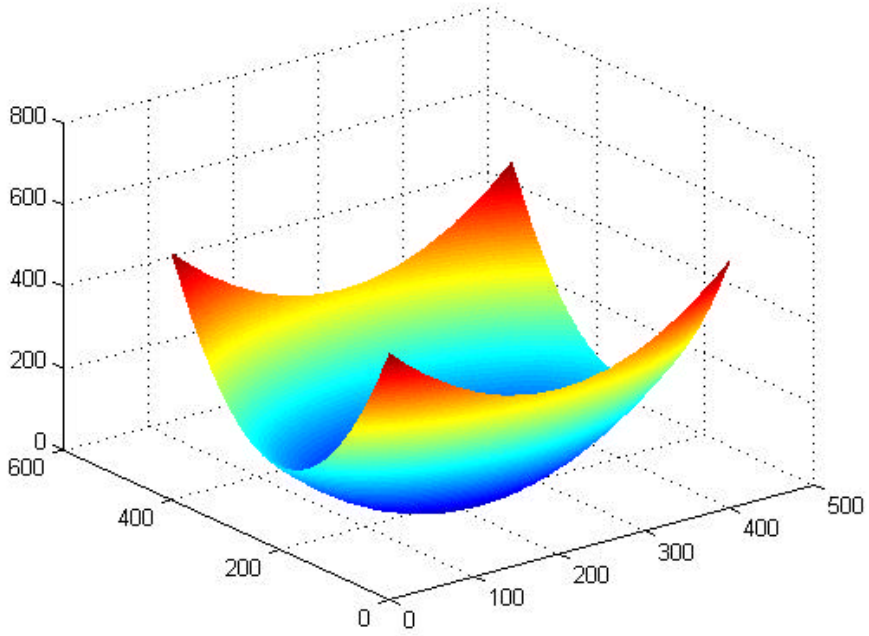


Figura 15 - Função Bi-dimensional I (F15)

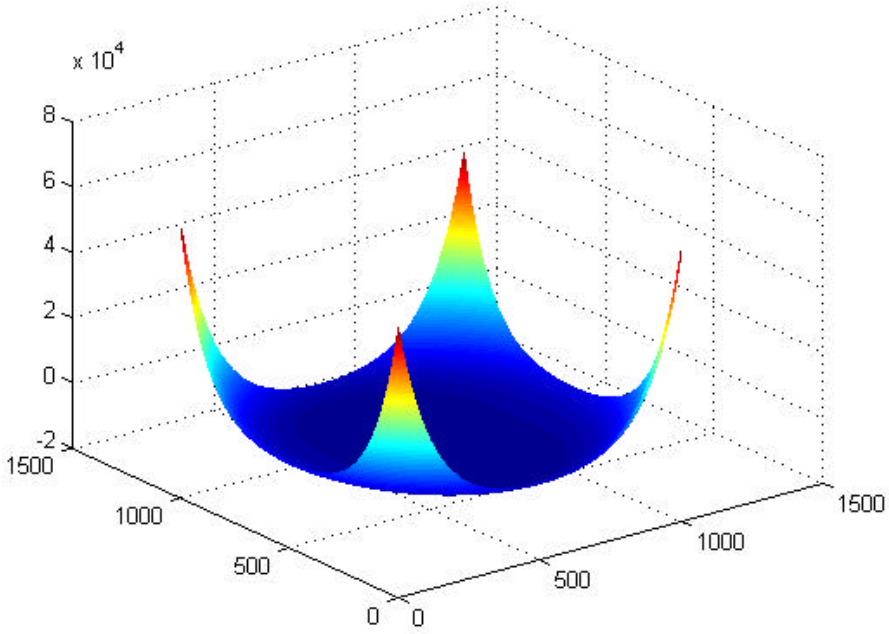


Figura 16 - Função Bidimensional II (F16)

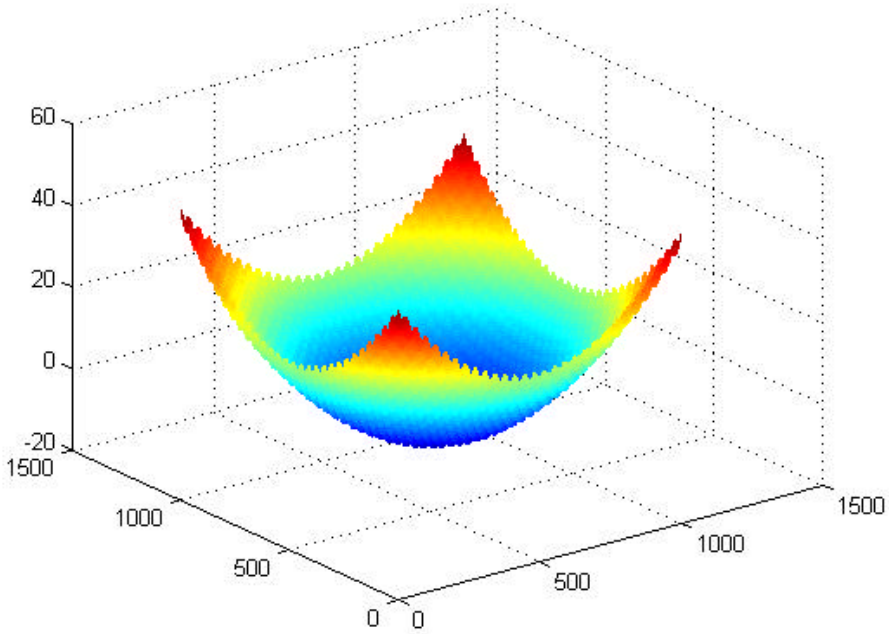


Figura 17 - Função Bidimensional de Rastrigin (F17)

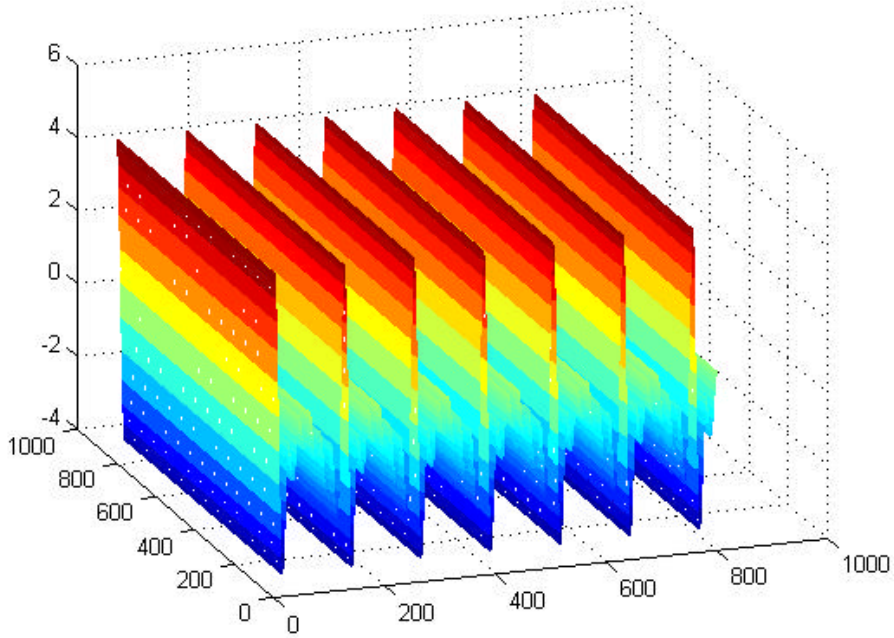
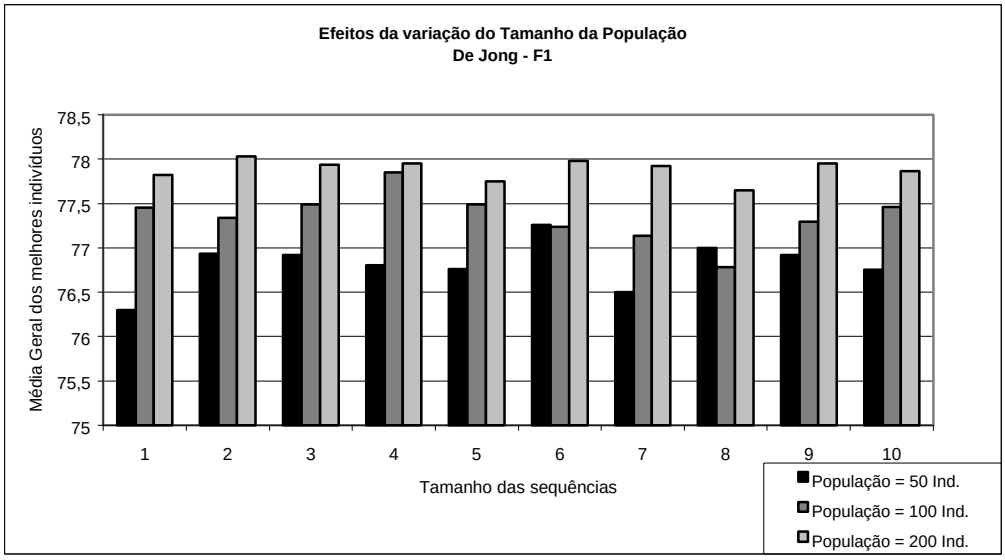


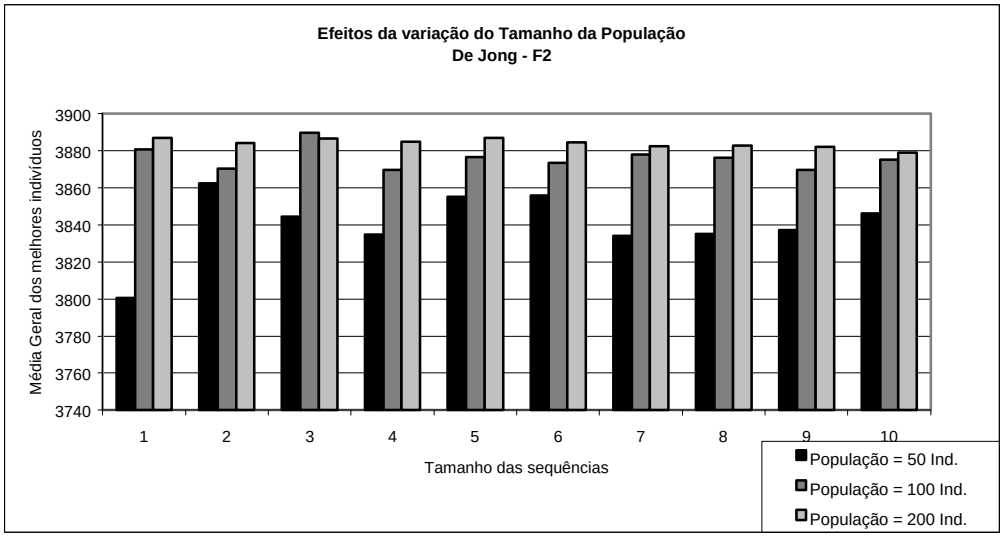
Figura 18 - Função Unidimensional (F18)

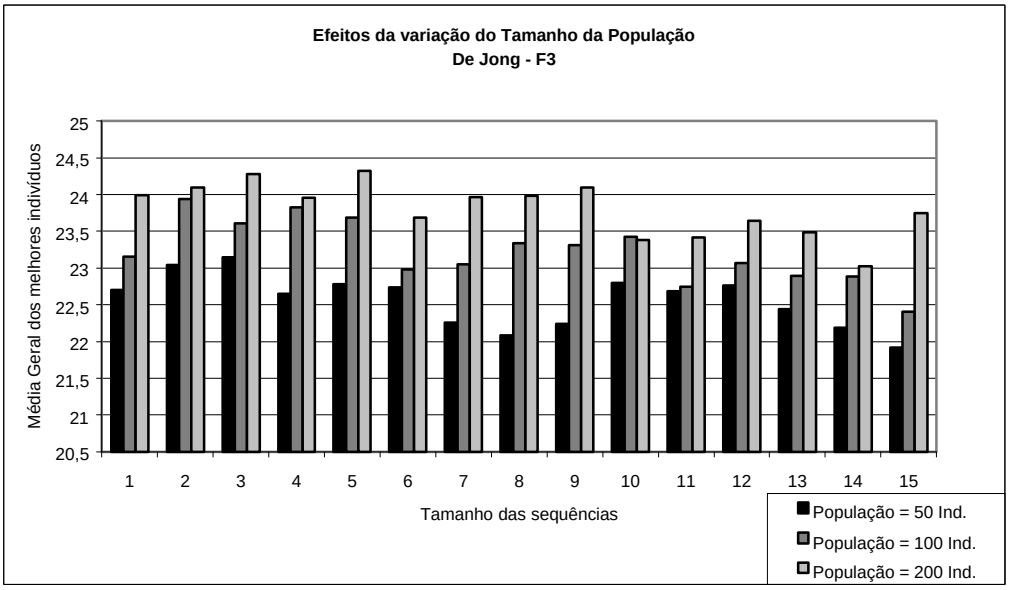
ANEXO 2

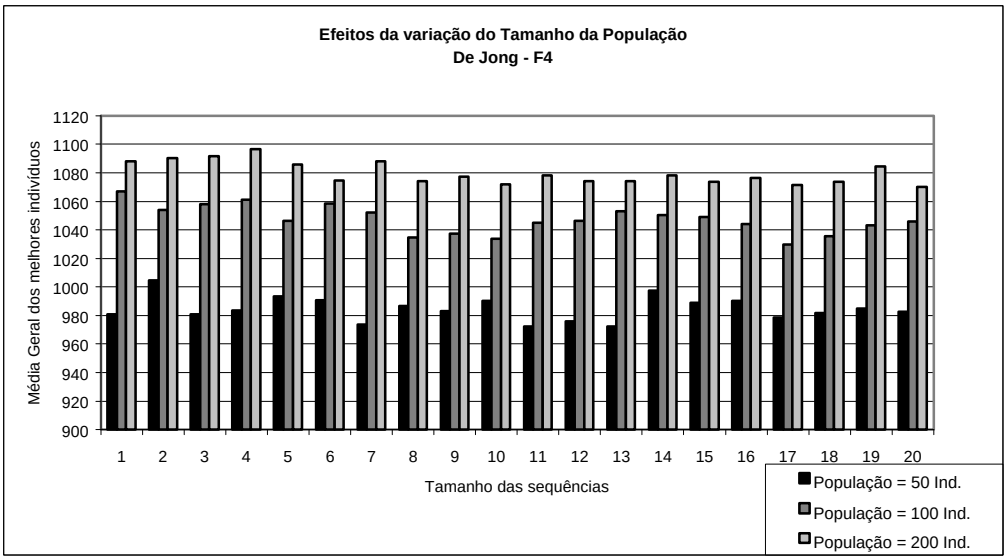
ESTUDO DO MECANISMO DE TRANSPOSIÇÃO

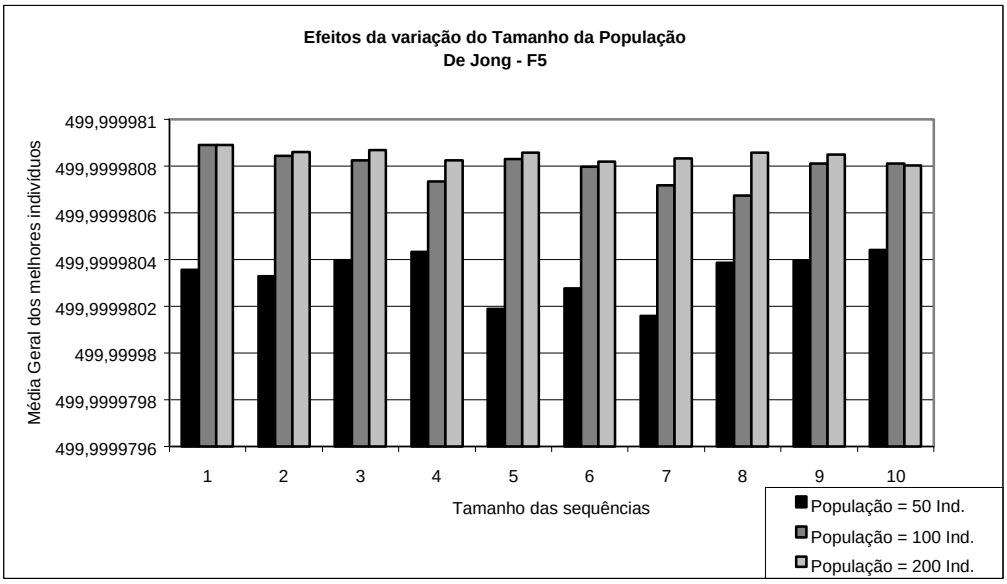
Estudo do Mecanismo de Transposição
Efeitos do Variação do Tamanho da População
Funções de Teste de K. De Jong



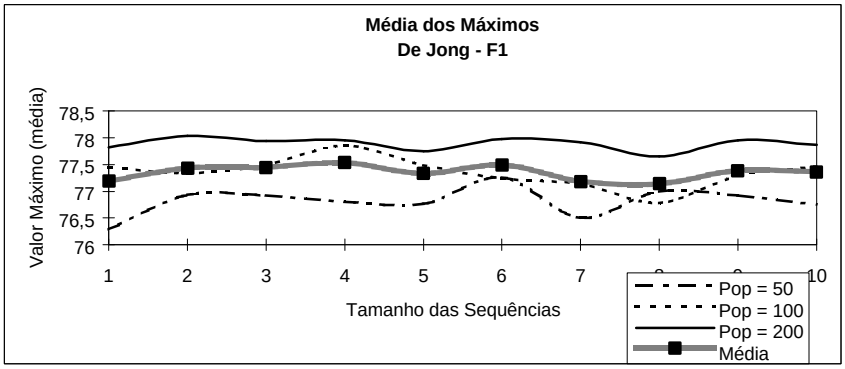


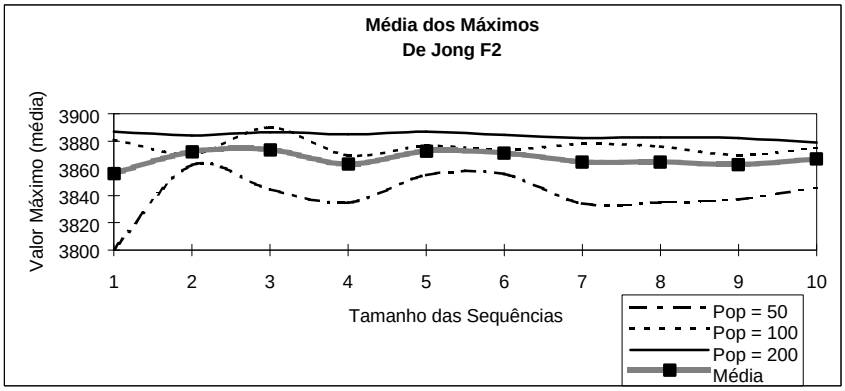


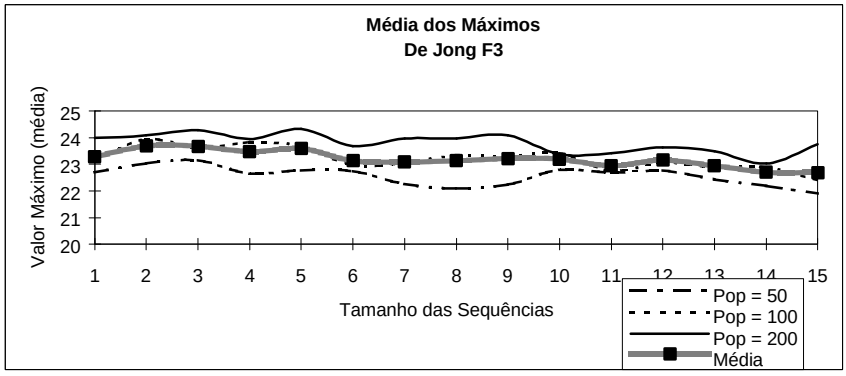

Efeitos da variação do Tamanho da População
De Jong - F4
Média Geral dos melhores indivíduos
1120
1100
1080
1060
1040
1020
1000
980
960
940
920
900
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Tamanho das sequências
População = 50 Ind.
População = 100 Ind.
População = 200 Ind.

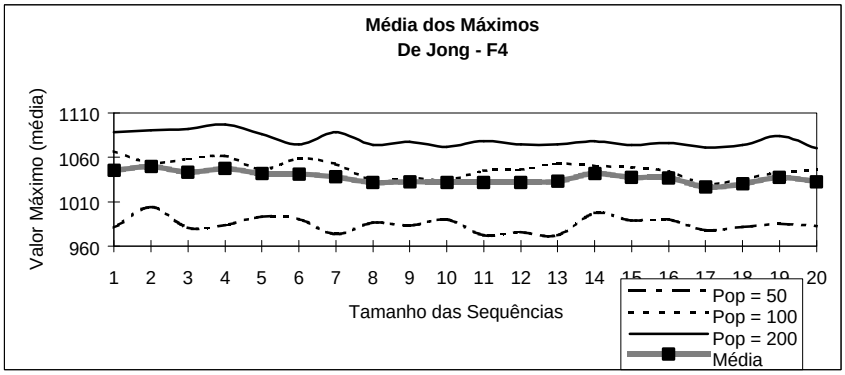

Efeitos da variação do Tamanho da População
De Jong - F5
Média Geral dos melhores indivíduos
499,999981
499,9999808
499,9999806
499,9999804
499,9999802
499,99998
499,9999798
499,9999796
1 2 3 4 5 6 7 8 9 10
Tamanho das sequências
População = 50 Ind.
População = 100 Ind.
População = 200 Ind.

Estudo do Mecanismo de Transposição Simples
Efeitos da variação do tamanho das sequências flanqueadoras
Funções de Teste de K. De Jong

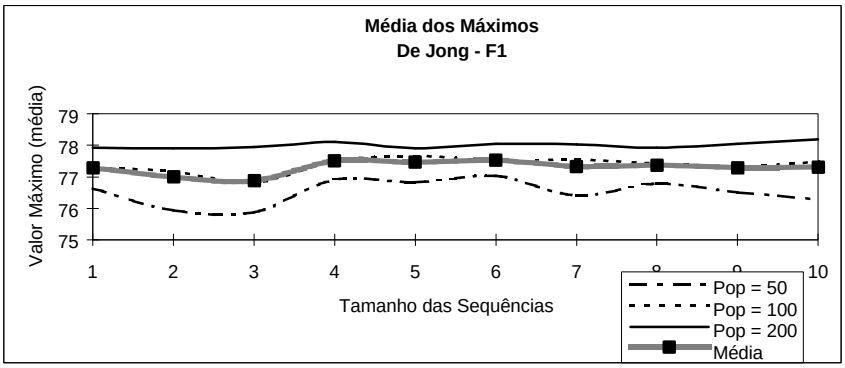


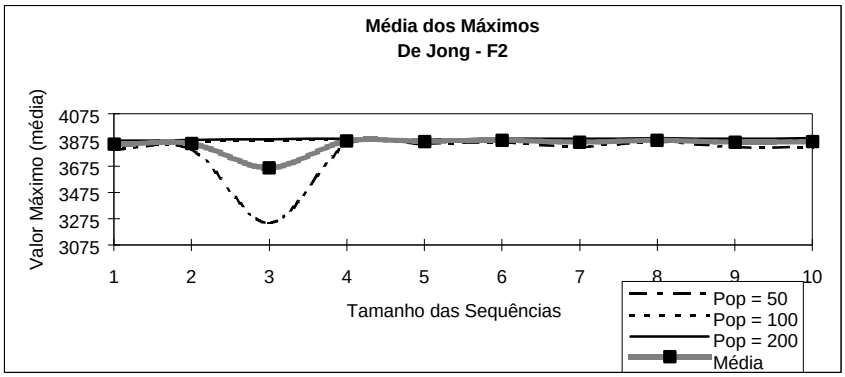


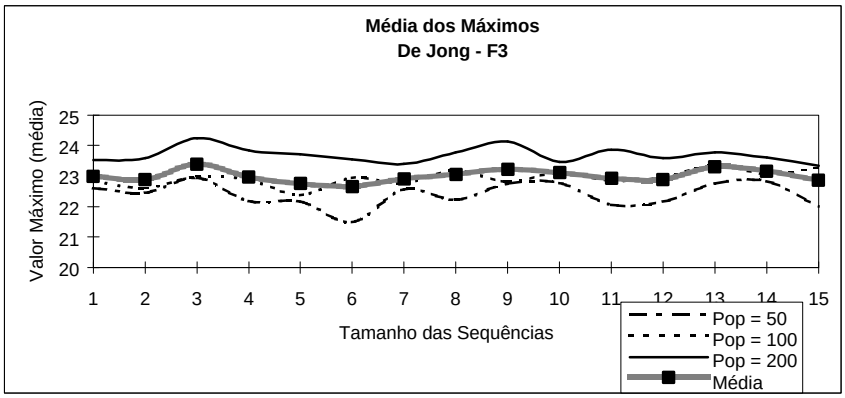


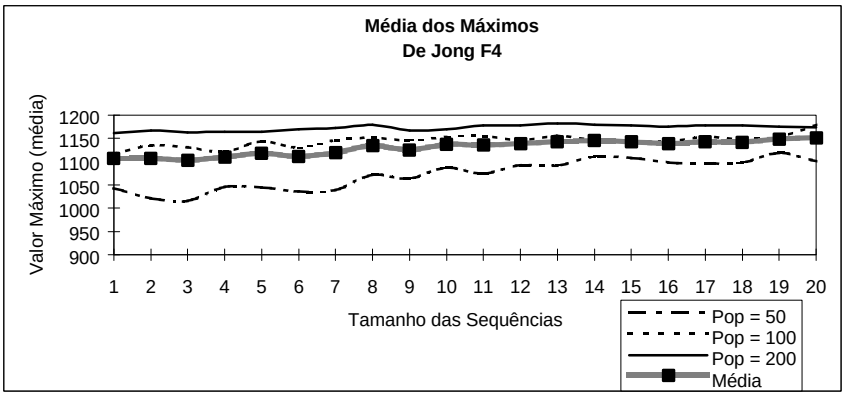


Estudo do Mecanismo de Transposição/Torneio
Efeitos da variação do tamanho das sequências flanqueadoras
Funções de Teste de K. De Jong









Estudo do Mecanismo de Transposição Simples
Efeitos da variação do tamanho das sequências flanqueadoras
Funções de Teste de K. De Jong

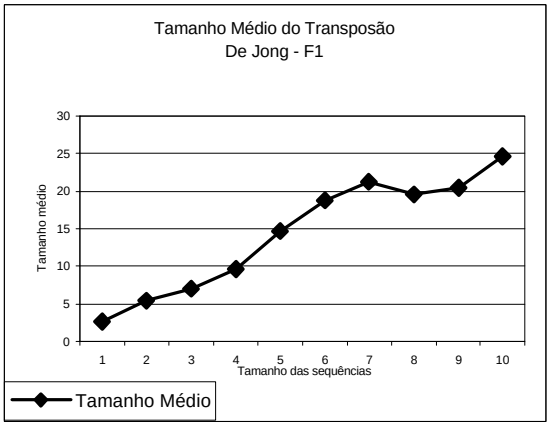


Gráfico 14.1

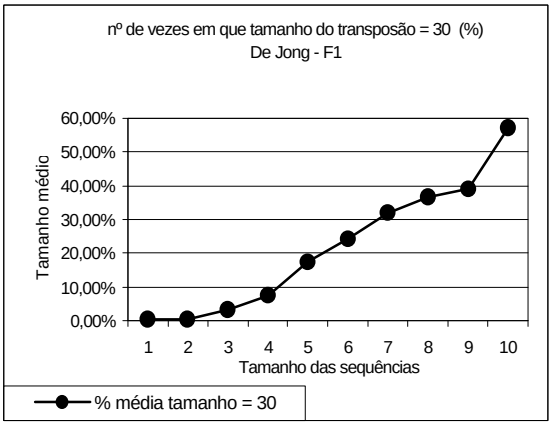


Gráfico 14.2

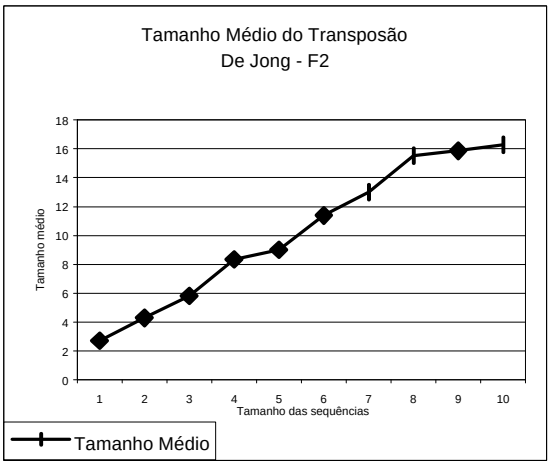


Gráfico 15.1

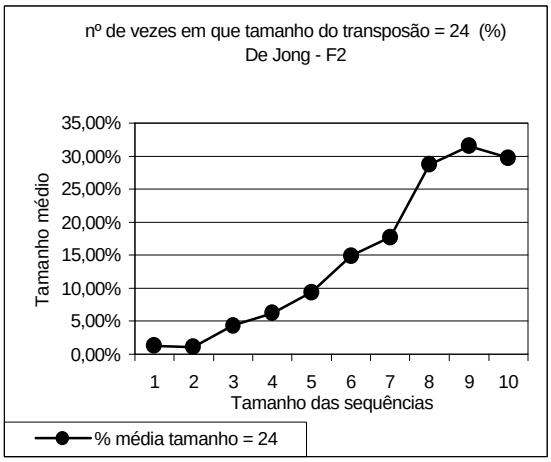


Gráfico 15.2

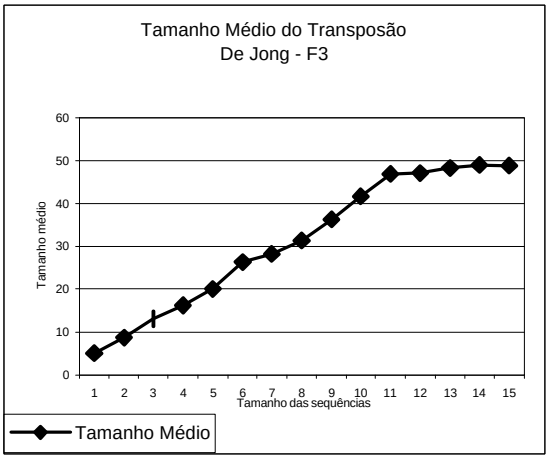


Gráfico 16.1

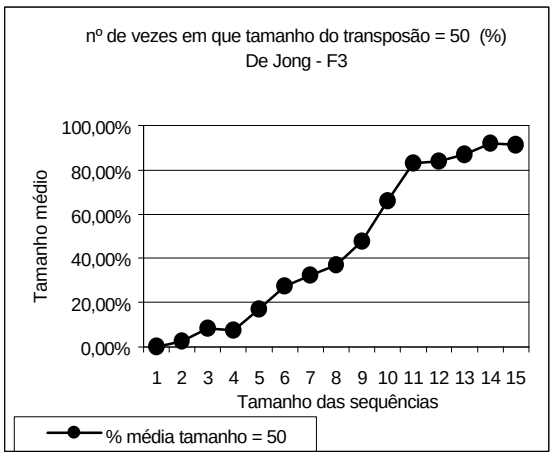


Gráfico 16.2

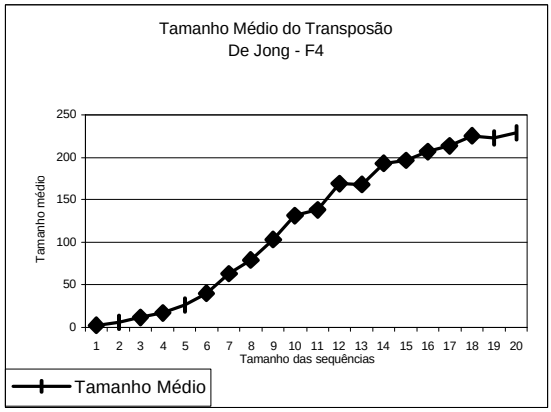


Gráfico 17.1

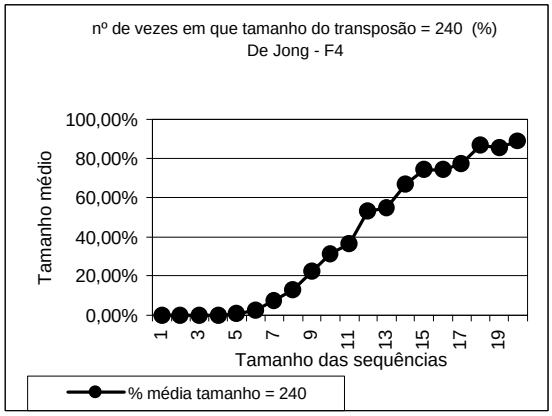


Gráfico 17.2

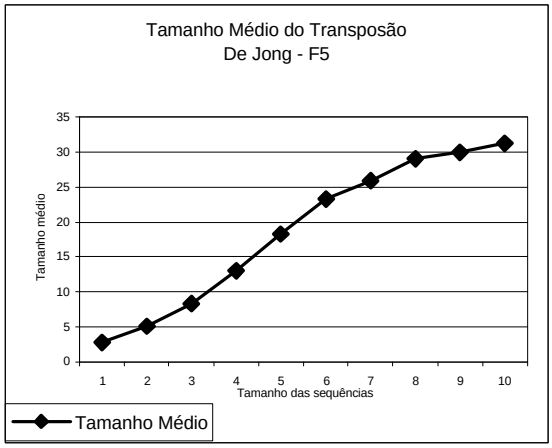


Gráfico 18.1

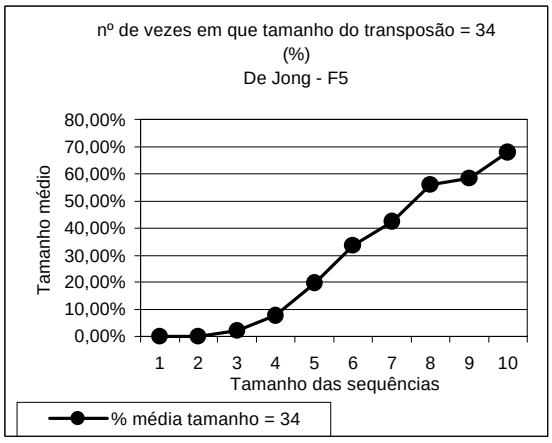


Gráfico 18.2

ANEXO 3

COMPARAÇÃO ENTRE O MECANISMO DE TRANSPOSIÇÃO E CROSSOVER 1 PONTO DE CORTE

(INDISPONIVEL)

ANEXO 4

COMPARAÇÃO ENTRE O MECANISMO DE TRANSPOSIÇÃO E CROSSOVER 2 PONTOS DE CORTE

(INDISPONIVEL)

ANEXO 5

COMPARAÇÃO ENTRE O MECANISMO DE TRANSPOSIÇÃO E CROSSOVER UNIFORME

(INDISPONIVEL)