Carlos Manuel Chorro Simões Barrico

Optimização Evolucionária Multi-Objectivo em Ambientes Incertos — Pesquisa de Soluções Robustas

Departamento de Engenharia Electrotécnica e de Computadores
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

2007

Carlos Manuel Chorro Simões Barrico

Optimização Evolucionária Multi-Objectivo em Ambientes Incertos — Pesquisa de Soluções Robustas

Tese submetida para obtenção do grau de Doutor em
Engenharia Electrotécnica e de Computadores,
Especialidade em Informática

Orientação de Prof. Doutor Carlos Henggeler Antunes

Departamento de Engenharia Electrotécnica e de Computadores
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

Coimbra
2007

Agradecimentos

Ao Prof. Carlos Henggeler Antunes, pela orientação científica do trabalho que serviu de base a este documento, pela importante ajuda na escrita deste documento, assim como pelo constante apoio dado ao longo do tempo que durou todo o trabalho.

Ao Prof. Mário Freire pelo apoio dado na elaboração deste trabalho.

À minha família pela paciência que tiveram durante este tempo todo.

Aos meus colegas do INESC de Coimbra e do Departamento de Informática da Universidade da Beira Interior, pelo apoio manifestado em todos os momentos, em especial, nos mais difíceis.

Os meus agradecimentos são extensíveis a todos aqueles que, directa ou indirectamente, contribuíram para a realização deste trabalho.

Índice Geral

Resumo

Capítulo 1

Introdução Geral

1. Optimização multi-objectivo.....1

2. Optimização evolucionária multi-objectivo.....2

3. Análise de robustez em optimização.....5

4. Análise de robustez em optimização evolucionária.....6

5. Objectivos do trabalho.....7

6. Organização da tese.....8

7. Contribuições científicas da tese.....9

Capítulo 2

Optimização Evolucionária Multi-Objectivo

1. Optimização mono-objectivo.....11

2. Optimização multi-objectivo.....12

2.1. Formulação de um problema de optimização multi-objectivo.....12

2.2. Princípios de optimização multi-objectivo.....13

2.3. Diferenças com a optimização mono-objectivo.....14

2.4. Dominância e optimalidade de Pareto.....15

2.4.1. Pontos especiais.....16

2.4.2. Conceitos de dominância e de optimalidade de Pareto.....16
2.4.3. Determinação de uma frente não dominada.....18
2.4.4. Determinação de todas as frentes não dominadas.....18
3. Métodos para resolver problemas de optimização MO.....20
3.1. Métodos clássicos.....20
3.2. Meta-heurísticas.....22
4. Algoritmos Evolucionários.....25
4.1. Estratégias Evolucionárias.....26
4.2. Programação Evolucionária.....26
4.3. Algoritmos Genéticos.....27
4.3.1. Representação (codificação) dos indivíduos.....29
4.3.2. Início do processo.....31
4.3.3. Função de avaliação (aptidão).....32
4.3.4. População.....33
4.3.5. Mecanismos de selecção.....34
4.3.6. Operador genético cruzamento.....36
4.3.7. Operador genético mutação.....37
4.3.8. Mecanismos de substituição.....38
4.3.9. Critérios de paragem.....39
5. Algoritmos genéticos multi-objectivo.....40
5.1. Função de aptidão e mecanismo de selecção.....40
5.2. Diversidade da população.....41
5.2.1. Pré-selecção.....42
5.2.2. Técnica de multidões.....42
5.2.3. Partilha do valor de aptidão.....42
5.2.4. Técnica de agrupamentos.....44
5.3. Principais abordagens com Algoritmos Genéticos.....45
5.3.1. Abordagens Não Elitistas.....46
5.3.1.1. Vector Evaluated Genetic Algorithm (VEGA).....46
5.3.1.2. Weight-Based Genetic Algorithm (WBGA).....47
5.3.1.3. Multiple Objective Genetic Algorithm (MOGA).....48
5.3.1.4. Non-dominated Sorting Genetic Algorithm (NSGA).....49
5.3.1.5. Niched Pareto Genetic Algorithm (NPGA).....50

5.3.2. Abordagens Elitistas.....51
5.3.2.1. Strength Pareto Evolutionary Algorithm.....51
5.3.2.2. Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II).....53
5.3.2.3. Strength Pareto Evolutionary Algorithm 2.....54

Capítulo 3

Análise de Robustez em Optimização

1. Caracterização da incerteza.....57
1.1. Introdução.....57
1.2. Tipos e causas da incerteza.....58
1.3. Modelos que incorporam a incerteza.....60
2. Métodos associadas a modelos com incerteza.....61
3. Conceitos de robustez.....64
3.1. Decisão robusta.....64
3.2. Solução robusta.....65
3.3. Conclusão robusta.....66
3.4. Método robusto.....66
4. Optimização robusta.....68
4.1. Abordagem de Mulvey, Verderbel e Zenios.....69
4.2. Optimização discreta (combinatória) robusta.....71
5. Optimização robusta usando meta-heurísticas.....74

Capítulo 4

Análise de Robustez em Optimização Evolucionária Multi-objectivo

1. Introdução.....77
2. Optimização evolucionária em ambientes incertos.....77
2.1. Função de aptidão perturbada.....78
2.1.1. Cálculo explícito da média.....79
2.1.2. Cálculo implícito da média.....80
2.1.3. Selecção modificada.....80

2.1.4. Casos relacionados....81
2.2. Função de aptidão aproximada....82
2.2.1. Métodos de aproximação....83
2.2.2. Mecanismos para incorporar meta-modelos....84
2.2.3. Controlo da evolução ou gestão de meta-modelos....84
2.2.4. Convergência....87
2.3. Ambientes dinâmicos....87
2.3.1. Considerações gerais....88
2.3.2. Tratamento de problemas de optimização dinâmica com AE....89
2.3.3. Algumas abordagens....90
2.4. Robustez....92
2.4.1. Optimização da função de aptidão esperada....92
2.4.1.1. Cálculo explícito da média....94
2.4.1.2. Cálculo implícito da média....96
2.4.2. Abordagens multi-objectivo....97
3. Abordagem de Deb e Gupta....99
3.1. Robustez em optimização mono-objectivo....100
3.2. Robustez em optimização multi-objectivo....101
4. Uma nova abordagem à optimização robusta multi-objectivo....102
4.1. Perturbações nos valores das variáveis de decisão....103
4.2. Perturbações nos dados associados às funções objectivo....105
4.3. Perturbações nos dados associados às restrições....106
4.4. Parâmetros de robustez....107
4.5. Métricas....110
4.6. Abordagem evolucionária....110
4.6.1. Construção das vizinhanças....110
4.6.2. Algoritmo genético....112
4.6.3. Função de aptidão....112
4.6.4. Mecanismo de manutenção da diversidade....114
4.6.5. População inicial....117
4.6.6. Descrição do algoritmo....117

Capítulo 5
Testes Computacionais e Análise de Resultados

1. Introdução.....119
2. Formalização matemática do problema de teste.....120
3. Frente de Pareto obtida com e sem análise de robustez.....121
4. Análise do parâmetro p1.....122
5. Análise do parâmetro p2.....124
6. Análise dos parâmetros p1 e p2 em conjunto.....126
7. Análise do parâmetro η.....128
8. População secundária só com soluções robustas.....132
9. Utilização de métricas.....134

Capítulo 6
Estudo de Caso - Compensação de energia reactiva em redes radiais de distribuição

1. Introdução.....137
2. O problema da compensação de energia reactiva em redes de distribuição.....138
   2.1. Introdução.....138
   2.2. Colocação de condensadores em sistemas eléctricos.....139
3. Abordagem sem análise de robustez.....140
   3.1. Algoritmo genético.....140
   3.2. Rede testada.....142
   3.3. Frente óptima de Pareto.....142
   3.4. População inicial.....143
4. Abordagem com análise de robustez.....145
   4.1. Algoritmo genético.....146
   4.2. Frente não dominada.....147
   4.3. Análise do parâmetro p1.....148
   4.4. Análise do parâmetro p2.....150

4.5. Análise dos parâmetros p1 e p2 em conjunto.....152
4.6. Análise do parâmetro η.....154
4.7. Conclusão.....155

Capítulo 7
Conclusões Finais e Pistas para Futuros Desenvolvimentos

1. Conclusões Finais.....157
2. Pistas para Futuros Desenvolvimentos.....159

Referências.....161

ANEXO

Problema da compensação de energia reactiva em redes de distribuição

1. Modelo matemático.....187
2. Redes radiais.....190
3. Definição do cenário de carga.....191
4. Características das redes usadas.....192
   4.1. Rede portuguesa.....192
   4.2. Rede indiana.....197

Índice de Figuras

Figura 2.1: Classificação das soluções por níveis de dominância.....19

Figura 4.1: Definições das vizinhanças nos espaços das variáveis de decisão e das funções objectivo (para espaços bidimensionais e todas as funções a minimizar)....104

Figura 4.2: Definições das vizinhanças nos espaços dos cenários e das funções objectivo (para espaços bidimensionais e todas as funções a minimizar).....106

Figura 4.3: Forma de gerar h=8 pontos vizinhos em redor de uma solução x=(a, b)....111

Figura 4.4: Mecanismo de preservação da diversidade populacional.....116

Figura 5.1: Frente de Pareto (análise sem robustez).....121

Figura 5.2: Frente de Pareto (com análise de robustez, p1 = p2 = 100% e η = 1.5)....122

Figura 5.3: Frente de Pareto (p1 = 95%; p2 = 100%; η = 1.5).....123

Figura 5.4: Frente de Pareto (p1 = 85%; p2 = 100%; η = 1.5).....123

Figura 5.5: Frente de Pareto (p2 = 95%).....124

Figura 5.6: Frente de Pareto (p2 = 85%).....125

Figura 5.7: Frente de Pareto (p2 = 75%).....126

Figura 5.8: Frente de Pareto (p1$\oplus$p2 = 95%).....127

Figura 5.9: Frente de Pareto (p1$\oplus$p2 = 85%).....128

Figura 5.10: Frente de Pareto (η = 1.0).....129

Figura 5.11: Frente de Pareto (η = 0.5).....130

Figura 5.12: Frente de Pareto (η = 2.0).....131

Figura 5.13: Frente de Pareto (η = 2.5).....132

Figura 5.14: Frente de Pareto só com soluções robustas (p1 = p2 = 100%).....133

Figura 5.15: Frente de Pareto só com soluções robustas (p1 = p2 = 95%).....133

Figura 5.16: Frente de Pareto só com soluções robustas (p1 = 95%; p2 = 85%).....134

Figura 5.17: Frente de Pareto (usando a distância normalizada relativa).....135

Figura 5.18: Frente de Pareto (usando a distância não normalizada absoluta).....136

Figura 5.19: Frente de Pareto (usando a distância não normalizada relativa).....136

Figura 6.1: População secundária final (frente óptima de Pareto).....143

Figura 6.2: População secundária final versus população inicial.....143

Figura 6.3: Frentes de Pareto (mesmos valores para os parâmetros genéticos).....144

Figura 6.4: Populações secundárias finais (os melhores resultados).....145

Figura 6.5: Frente óptima de Pareto (sem considerar análise de robustez).....147

Figura 6.6: Frente de Pareto (com análise de robustez: p1 = p2 = 100%
e η = 0.004).....148

Figura 6.7: Frente de Pareto (com análise de robustez: p1 = 95%).....149

Figura 6.8: Frente de Pareto (com análise de robustez: p1 = 90%).....150

Figura 6.9: Frente de Pareto (com análise de robustez: p2 = 95%).....151

Figura 6.10: Frente de Pareto (com análise de robustez: p2 = 90%).....151

Figura 6.11: Frente de Pareto (com análise de robustez: p1 $\oplus$ p2 = 95%).....153

Figura 6.12: Frente de Pareto (com análise de robustez: p1 $\oplus$ p2 = 90%).....153

Figura 6.13: Frente de Pareto (com análise de robustez: η = (0.003, 150)).....154

Figura 6.14: Frente de Pareto (com análise de robustez: η = (0.005, 150)).....155

Figura 6.15: Frente de Pareto (com análise de robustez: (η = 0.006, 150)).....155

Figura 1: Exemplo de um sistema de distribuição radial.....191

Índice de Tabelas

Tabela 1: Comparação entre as codificações binárias tradicional e gray.....30

Tabela 2: Resultados da aplicação do mecanismo de preservação da diversidade.....115

Tabela 3: Características dos troços da rede portuguesa.....193

Tabela 4: Potências tomadas, activa e reactiva, nos nós da rede portuguesa.....194

Tabela 5: Caracterização dos alimentadores da rede portuguesa.....195

Tabela 6: Caracterização dos condensadores a instalar na rede portuguesa.....195

Tabela 7: Perdas activa e resistiva nos troços da rede portuguesa sem compensação.. 196

Tabela 8: Perdas totais na rede portuguesa sem compensação.....197

Tabela 9: Tensão nos nós da rede portuguesa sem compensação.....197

Tabela 10: Características dos troços da rede indiana.....198

Tabela 11: Potências tomadas, activa e resistiva, nos nós da rede indiana.....198

Tabela 12: Caracterização dos alimentadores da rede indiana.....198

Tabela 13: Caracterização dos condensadores a instalar na rede indiana.....199

Tabela 14: Perdas activa e resistiva nos troços da rede indiana sem compensação.....199

Tabela 15: Perdas totais na rede indiana sem compensação.....199

Tabela 16: Tensão nos nós da rede (indiana) sem compensação.....199

Resumo

A modelação de muitos problemas de optimização, quer multi-objectivo quer mono-objectivo, depende da informação disponível, a qual, contudo, muitas vezes não se conhece com exactidão, dado que, normalmente, envolve previsões ou estimativas de fenómenos naturais, económicos ou sociais. Em alguns casos esta incerteza é simplesmente ignorada (trabalha-se com a melhor estimativa dos dados), mas é comum a construção de modelos que procuram incorporar explicitamente a incerteza (probabilísticos, difusos, com intervalos, com cenários, etc.).

A forma mais tradicional de modelar um problema de optimização consiste em formulá-lo como determinístico (isto é, não considerando a incerteza inerente ao problema). No entanto, tem-se verificado nos últimos anos um aumento do interesse na construção de modelos que incorporam explicitamente a incerteza. Esta incerteza pode não ser apenas estrutural, relacionada com o conhecimento global acerca do sistema a ser modelado, mas também ocorrer nos dados do problema, os quais podem sofrer de imprecisões, de imperfeições ou estarem sujeitos a alterações (perturbações).

A incerteza resulta, fundamentalmente, da imprecisão e das oscilações nos valores que alimentam os modelos, da subjectividade subjacente ao Ser Humano, da imprecisão presente na modelação, que é sempre uma aproximação da realidade, da ocorrência de imponderáveis relevantes e das simplificações necessariamente assumidas devido à crescente complexidade dos problemas actuais. A maioria das abordagens associadas a problemas de decisão sujeitos a incerteza está relacionada com os modelos de representação da incerteza, os quais podem ser sobretudo estocásticos, difusos, intervalares e baseados em cenários.

A análise de robustez está associada essencialmente ao modelo de representação da incerteza através de cenários, aos quais normalmente não estão associadas quaisquer probabilidades, mas pode também ser usada com os modelos difusos ou intervalares. O processo de optimização pode ser de qualquer tipo, mas os algoritmos evolucionários têm emergido e ganho importância na resolução deste tipo de problemas, em particular problemas complexos de natureza combinatória.

Os algoritmos evolucionários ajustam-se bem à resolução de problemas de optimização multi-objectivo, essencialmente devido à capacidade de trabalhar, em cada geração, com uma população de potenciais soluções. Existem vários tipos de algoritmos evolucionários, sendo os mais usados os algoritmos genéticos.

Os métodos para tratamento da incerteza em optimização evolucionária podem ser categorizados em quatro classes; três delas estão relacionadas com formas diferentes de tratar a função de aptidão e a outra com a análise de robustez das soluções. Por sua vez, os métodos pertencentes à classe da análise de robustez podem-se subdividir nos que optimizam o valor esperado (mono-objectivo) e nos multi-objectivo.

Apesar de existirem algumas abordagens evolucionárias para determinar soluções robustas em optimização, mono e multi-objectivo, a maioria destas abordagens está vocacionada para resolver problemas de optimização mono-objectivo, ou então converter o problema multi-objectivo original num mono-objectivo. Estas abordagens consistem em optimizar a média ou o desvio padrão da função de aptidão associada ao problema em estudo, as quais são obtidas através de uma amostra de pontos na vizinhança das soluções. Das poucas abordagens que tratam os problemas como multi-objectivo, a maioria considera como funções objectivo a optimizar o desempenho e a robustez; são raras as abordagens que modelam explicitamente os problemas como multi-objectivo.

Esta dissertação descreve uma nova metodologia para tratar problemas de optimização multi-objectivo sujeitos a incerteza, baseada em algoritmos evolucionários (genéticos). Nesta metodologia, a incerteza, traduzida por perturbações aleatórias em redor dos valores das variáveis de decisão e dos valores "nominais" dos dados do problema, é modelada através de cenários (aos quais não estão atribuídos quaisquer probabilidades) e usada a análise de robustez como método de resolução. Para tal, introduziu-se um novo conceito de robustez associado às soluções do problema, que consiste em categorizar estas soluções através de um grau de robustez, o qual é incorporado no algoritmo evolucionário, contribuindo de forma directa para o cálculo do valor da função de aptidão associado às soluções.

Foram desenvolvidas duas abordagens para determinar as soluções robustas em problemas de optimização multi-objectivo, usando este novo conceito de robustez, nas quais as soluções são classificadas de acordo com os respectivos graus de robustez, o que permite contribuir para uma melhor escolha, por parte do agente de decisão, de uma solução de compromisso robusta.

Na primeira abordagem as perturbações ocorrem nos valores das variáveis de decisão das soluções do problema (isto é, no espaço das variáveis de decisão). O grau de robustez de uma solução x depende do comportamento das soluções da vizinhança de x no espaço dos objectivos (na vizinhança de f(x)). O valor do grau de robustez da solução

x é obtido tendo em conta o número de soluções vizinhas de x admissíveis e o número de soluções vizinhas de x cujos valores das funções objectivo são melhores do que f(x) ou pertencem a uma certa vizinhança de f(x).

Na segunda abordagem as perturbações ocorrem nos dados associados às funções objectivo e às restrições, relativamente aos valores "nominais" destes dados (cenário de referência). O grau de robustez de uma solução x depende do seu comportamento na vizinhança do cenário de referência s e na vizinhança dos valores das funções objectivo de x para o cenário de referência s, $f^s(x)$. O valor do grau de robustez da solução x é obtido tendo em conta o número de cenários vizinhos do cenário de referência s para os quais a solução x é admissível e o número de cenários vizinhos de s para os quais os valores das funções objectivo de x são melhores do que $f^s(x)$ ou pertencem a uma certa vizinhança de $f^s(x)$.

A primeira abordagem foi testada com um problema vulgarmente usado na literatura científica da área da programação evolucionária. A segunda abordagem foi testada com um problema real, da área dos sistemas eléctricos de energia.

Capítulo 1

Introdução Geral

1. Optimização multi-objectivo

O conceito de optimização está associado à determinação de uma ou mais soluções admissíveis, as quais correspondem aos valores extremos de um ou mais objectivos. Os métodos de optimização são de grande importância na resolução de problemas práticos, em particular no domínio da engenharia, das experiências científicas e da tomada de decisões em geral.

Quando um problema de optimização modela um sistema real envolvendo apenas uma função objectivo, a tarefa de determinar a solução óptima é denominada de *optimização mono-objectivo*. Desde a II Guerra Mundial, a maioria dos esforços feitos no campo da optimização foi para desenvolver e aplicar métodos de optimização mono-objectivo, utilizando, por exemplo, técnicas de pesquisa baseadas em gradientes ou em heurísticas. Além dos princípios de pesquisa determinística envolvida num algoritmo, também existem princípios de pesquisa estocástica usados em algoritmos de optimização para determinar, com mais segurança, soluções óptimas globais. Para estender a aplicação de um algoritmo de optimização a domínios de diferentes problemas, os princípios naturais e físicos são imitados para desenvolver algoritmos de optimização poderosos, como é o caso dos algoritmos evolucionários.

Os problemas complexos que surgiram no seio das sociedades modernas tecnologicamente desenvolvidas envolvem aspectos de avaliação múltiplos, conflituosos e incomensuráveis para determinar o mérito dos planos de acção alternativos. Assim, os

modelos matemáticos de apoio à decisão tornam-se mais representativos do contexto da decisão real, se esses aspectos de avaliação forem explicitamente tidos em conta, em vez de os agregar num único indicador de custo ou benefício.

Em modelos de programação multi-objectivo esses aspectos de avaliação são traduzidos pelas funções objectivo a optimizar, sujeitas a um conjunto de restrições. Os modelos multi-objectivo permitem compreender a natureza conflituosa dos objectivos e dos compromissos[1] a serem construídos para identificar as soluções satisfatórias, estabelecendo uma base para racionalizar a comparação entre as soluções não dominadas. Uma solução não dominada (eficiente, óptima de Pareto) é uma solução admissível para a qual não é possível melhorar simultaneamente todas as funções objectivo; isto é, a melhoria numa função objectivo apenas pode ser alcançada aceitando piorar, pelo menos, uma outra função objectivo.

Para além de contribuir para a construção de um modelo mais realista perante problemas reais, uma abordagem multi-objectivo possui ainda intrinsecamente um papel importante na modelação dos processos de decisão, suportando a reflexão e a criatividade dos Agentes de Decisão (AD) face a um extenso universo de potenciais soluções, tendo em mente a sua implementação prática, uma vez que não existe uma solução melhor do que todas as outras em todos os aspectos de avaliação.

O estudo de um problema de optimização multi-objectivo envolve geralmente a caracterização do conjunto de soluções não dominadas, através da realização de um cálculo exaustivo destas soluções ou então calculando uma amostra representativa daquele conjunto. Desta forma, em optimização multi-objectivo não se pretende determinar uma solução óptima (como em optimização mono-objectivo), mas sim determinar um conjunto de soluções não dominadas (frente não dominada, frente óptima de Pareto ou simplesmente frente de Pareto).

Os métodos para resolver problemas de optimização multi-objectivo devem obedecer ao seguinte princípio: determinar um conjunto de soluções de compromisso, de forma a caracterizar o mais extensamente possível as gamas de valores alcançáveis para cada função objectivo.

2. Optimização evolucionária multi-objectivo

A capacidade de trabalhar em cada geração com uma população de potenciais soluções faz com que as abordagens evolucionárias se ajustem bem aos problemas de optimização multi-objectivo, particularmente a problemas de natureza combinatória

[1] Tradução da palavra inglesa "tradeoffs".

complexos, nos quais deve ser identificado um conjunto de soluções não dominadas em vez de uma única solução óptima.

Dado que no contexto da optimização evolucionária, em particular, e das meta-heurísticas, em geral, normalmente apenas é possível determinar se uma solução é dominada após o cálculo de uma outra que a domina, as soluções não dominadas deveriam em rigor designar-se por soluções potencialmente não dominadas. No entanto, para simplificação da linguagem utilizar-se-à a designação de solução não dominada com este último sentido, quer corresponda ou não a uma solução realmente não dominada.

Os algoritmos evolucionários são uma classe das técnicas de optimização estocástica, que consistem em imitar alguns dos mecanismos da evolução natural de populações de espécies no contexto de problemas de optimização. Estes métodos são iterativos, não garantindo a convergência para as melhores soluções, uma vez que se baseiam em heurísticas. Apesar de existirem vários tipos de algoritmos evolucionários, os mais usados são as Estratégias Evolucionárias (Rechenberg (1964), Rechenberg (1973)), a Programação Evolucionária (Fogel et al. (1966)) e os Algoritmos Genéticos (Holland (1975)). As Estratégias Evolucionárias caracterizam-se por originalmente considerarem apenas o operador genético mutação na geração de novas soluções, apesar de versões mais recentes (Schwefel (1995)) considerarem também o operador genético cruzamento. A Programação Evolucionária caracteriza-se por a representação das soluções ser livre e seguir a natureza do problema, e por o operador mutação ser o único responsável pela geração de descendentes. Os Algoritmos Genéticos (AGs) caracterizam-se por envolver todas as componentes do processo evolucionário (isto é, selecção, cruzamento, mutação e substituição). Destes três tipos de algoritmos, os AGs são os utilizados no trabalho que serviu de base a esta dissertação.

Os AGs usados para determinar a frente óptima de Pareto em problemas de optimização multi-objectivo são caracterizados pelo tipo de função de aptidão utilizado para avaliar as soluções obtidas, pelo tipo de mecanismo que usam na preservação da diversidade na frente de Pareto e pelo uso, ou não, de estratégias elitistas. Relativamente ao tipo de função de aptidão, o mais utilizado é baseado na dominância de Pareto introduzida por Goldberg (1989). Os mecanismos de preservação da diversidade, que garantem a diversificação das soluções e contribuem para a obtenção de uma distribuição uniforme das soluções, podem ser baseados nas técnicas de partilha[2] do valor de aptidão, de multidões[3], de agrupamentos[4], etc.. As estratégias elitistas utilizam, na sua maioria,

[2] Tradução da palavra inglesa "sharing".

[3] Tradução da palavra inglesa "crowding".

[4] Tradução da palavra inglesa "clustering".

uma população externa na qual guardam as melhores soluções que vão sendo obtidas pelo AG.

Os AGs podem ser classificados de acordo com a utilização ou não de estratégias elitistas. Do grupo dos AG que não usam estratégias elitistas, destacam-se os seguintes: VEGA (Schaffer (1984)), WBGA (Hajela e Lin (1992)), MOGA (Fonseca e Fleming (1993)), NSGA (Srinivas e Deb (1994)) e NPGA (Horn et al. (1994)). Do grupo dos AGs que utilizam uma estratégia elitista, destacam-se os seguintes: SPEA (Zitzler e Thiele (1998)), NSGA-II (Deb et al. (2000)) e SPEA 2 (Zitzler et al. (2001)).

Na resolução de um problema de optimização multi-objectivo, através de um destes algoritmos, pretende-se então determinar uma frente óptima de Pareto, a qual deve, por um lado, convergir o mais possível para a frente óptima de Pareto real e, por outro, garantir uma boa diversidade das soluções que a compõem. No entanto, algumas destas soluções, que poderão ser de interesse para um AD como soluções de compromisso aceitáveis no sentido que apresentam um equilíbrio satisfatório entre os aspectos de avaliação operacionalizados pelas funções objectivo, podem ser muito sensíveis a perturbações. Isto é, quando uma dada solução não dominada, seleccionada usando uma qualquer abordagem, é implementada na prática, algumas alterações relativamente aos valores estimados para os dados do modelo (coeficientes e parâmetros das funções objectivo, coeficientes e parâmetros das restrições, limites das variáveis de decisão, etc.) podem levar a uma importante degradação no desempenho das funções objectivo. Desta forma, surge a necessidade de oferecer ao AD soluções que sejam relativamente insensíveis às alterações dos dados do modelo; isto é, os algoritmos devem procurar soluções robustas.

A modelação de muitos problemas de optimização multi-objectivo (e também mono-objectivo) depende de informação que, em geral, não se conhece com exactidão, normalmente porque envolve previsões ou estimativas de fenómenos naturais, económicos ou sociais. Em alguns casos esta incerteza é simplesmente ignorada (trabalha-se com a melhor estimativa dos dados), mas é comum a construção de modelos que procuram incorporar a incerteza (baseados em cenários, probabilísticos, difusos, etc.). Sobre estes modelos aplicam-se normalmente regras ou paradigmas de optimização com carácter decisivo, que nem sempre incorporam devidamente as preferências do AD.

Desta forma, os modelos construídos para resolver problemas multi-objectivo devem incorporar explicitamente a incerteza, a qual não resulta apenas da constante variabilidade dos fenómenos mencionados, mas também da imprecisão e das variações associadas aos dados do problema, das inevitáveis imprecisões na fase de modelação, ou do desenvolvimento da estrutura de preferências do AD durante o processo interactivo de

decisão. A análise da robustez das soluções de compromisso consideradas satisfatórias pelo AD é uma questão de grande importância, devendo constituir uma componente fundamental de ferramentas computacionais interactivas de apoio à decisão.

3. Análise de robustez em optimização

A incerteza é uma característica intrínseca dos problemas reais surgindo de múltiplas origens e possuindo natureza distinta. A incerteza emerge da cada vez maior complexidade das interacções no interior dos sistemas sociais, económicos e técnicos, caracterizados por uma rápida evolução tecnológica, alterações nas estruturas de mercado e de novas relações sociais. Geralmente, é impraticável que os modelos de apoio à decisão possam capturar todos os fenómenos interrelacionados relevantes presentes, chegar até toda a informação necessária e também dar conta das alterações e/ou hesitações relacionadas com a expressão das preferências de quem decide.

O tratamento da incerteza inclui construir, primeiro, os modelos que incorporem a incerteza e, depois, os métodos para determinar as soluções destes modelos. Os modelos mais usados para tratar a incerteza em problemas de optimização são essencialmente de quatro tipos: probabilísticos, difusos, intervalares e usando cenários. Relativamente aos métodos para efectuar o tratamento da incerteza em programação matemática, existem, essencialmente, cinco classes: programação difusa, programação intervalar, análise de sensibilidade, programação estocástica e programação robusta.

Além da incerteza estrutural, relacionada com o conhecimento global acerca do sistema a ser modelado, também os dados do problema podem sofrer de imprecisões, de imperfeições ou estarem sujeitos a alterações. Neste contexto, é importante fornecer ao AD soluções robustas. Apesar da definição de robustez não estar uniformizada na literatura, uma solução robusta está geralmente ligada a um certo grau de "imunidade" em relação à incerteza dos dados, a uma capacidade de adaptabilidade ou flexibilidade relativamente a um futuro incerto ou a preferências mal especificadas, garantindo um desempenho aceitável mesmo sob alteração das condições (tais como os dados do modelo flutuando sobre os "dados nominais").

Segundo Vincke (2003), existem quatro conceitos diferentes de robustez: decisão robusta (Gupta e Rosenhead (1968)), solução robusta (Mulvey et al. (1995) e Kouvelis e Yu (1997)), conclusão robusta (Roy (1998) e Dias e Clímaco (1999)) e método robusto (Vincke (1999-a) e Vincke (1999-b)).

O primeiro foi aplicado a problemas de decisão, em que uma decisão é robusta se continuar, o mais possível, receptiva a "bons" planos no futuro. O segundo foi aplicado em problemas de optimização, em particular combinatória, onde robustez significa "boa

em todos ou na maioria dos cenários" (um cenário é um conjunto de valores possíveis para os dados do modelo), onde usaram a definição de pior cenário, o qual foi definido de acordo com os critérios *min-max* ou *min-max regret*. No terceiro caso, uma conclusão robusta significa "válida em todos ou na maioria dos cenários" (um cenário é um conjunto de valores aceitáveis para os dados do modelo). No último caso, um método robusto significa "que fornece resultados válidos em todos ou na maioria dos cenários" (um cenário é um conjunto de valores possíveis para os dados do modelo e para os parâmetros do método).

4. Análise de robustez em optimização evolucionária

Os algoritmos evolucionários permitem resolver muitos problemas de optimização, cuja modelação de cada um deles foi construída a partir de informação com algum nível de incerteza. Em optimização evolucionária, é usual a incerteza ser representada através da função de aptidão.

Segundo Jin e Branke (2005), os métodos para tratamento da incerteza em optimização evolucionária podem ser categorizados nas seguintes quatro classes: função de aptidão perturbada, função de aptidão aproximada, função de aptidão com variação temporal (ou em ambientes dinâmicos) e análise de robustez.

Relativamente à classe dos métodos para determinar soluções robustas (análise de robustez), que é a que mais interessa no âmbito deste trabalho, os métodos existentes podem ainda ser subdivididos em dois grupos: os que optimizam o valor esperado da função de aptidão e os multi-objectivo. Apesar de existirem reportadas na literatura algumas abordagens evolucionárias para determinar soluções robustas em optimização, mono e multi-objectivo, a maioria das abordagens está vocacionada para resolver problemas de optimização mono-objectivo, ou então converter o problema multi-objectivo original num mono-objectivo.

A maioria das abordagens em optimização multi-objectivo consiste em optimizar a média e o desvio padrão da função de aptidão associada ao problema em estudo, as quais são obtidas através de uma amostra de pontos na vizinhança de uma dada solução. Apenas algumas abordagens tratam o caso como multi-objectivo, mas em que consideram como funções objectivo a optimizar o desempenho e a robustez, e não como um verdadeiro problema multi-objectivo. Poucas abordagens consideram explicitamente o caso multi-objectivo (por exemplo, a apresentada por Deb e Gupta (2005)).

No trabalho realizado no âmbito desta dissertação, é usado o conceito de solução robusta, a qual é avaliada nos espaços dos dados do modelo susceptíveis de conterem um certo grau de incerteza (espaços das soluções, e dos dados associados às funções

objectivo e às restrições) e no espaço dos objectivos. Neste trabalho usa-se o conceito de cenário de referência, o qual corresponde aos valores nominais associados aos dados do modelo que contêm incerteza, isto é, que estão sujeitos a perturbações.

5. Objectivos do trabalho

Pretende-se com este trabalho construir uma metodologia para apoiar a tomada de decisão em problemas de optimização multi-objectivo, cujos valores das variáveis de decisão das soluções e dos dados (associados às funções objectivo e às restrições) do problema, estão sujeitos a perturbações, utilizando a análise de robustez. Para tal, desenvolveram-se duas abordagens evolucionárias envolvendo a análise de robustez em problemas de optimização multi-objectivo, as quais utilizam uma nova definição de robustez, designada por grau de robustez.

Estas abordagens baseiam-se no comportamento das soluções nas respectivas vizinhanças quando sujeitas a perturbações no espaço das variáveis de decisão ou nos espaços dos dados associados às funções objectivo e às restrições do problema, avaliado no espaço dos objectivos (ver também o conceito de solução robusta do tipo II em Deb e Gupta (2004) e Deb e Gupta (2005)).

O conceito de grau de robustez aqui introduzido permite ao AD exercer um controlo sobre o nível de robustez desejado das soluções obtidas. O AD pode especificar o tamanho da vizinhança das soluções, quer no espaço dos dados sujeitos a perturbações, quer no espaço das funções objectivo (normalmente o espaço com que o AD está mais familiarizado).

Na implementação destas abordagens, foram construídos dois algoritmos evolucionários (genéticos), nos quais o conceito de grau de robustez foi embebido no processo evolucionário, em especial na avaliação da aptidão (desempenho) de cada indivíduo. A ideia principal é influenciar o processo evolucionário no sentido de alcançar as soluções não dominadas mais robustas; isto é, soluções para as quais os desempenhos das funções objectivo são mais imunes a perturbações nos valores das variáveis de decisão das soluções, e nos dados associados às funções objectivo e às restrições.

Com a aplicação destes algoritmos evolucionários, pretende-se determinar uma frente óptima de Pareto, cujas soluções, para além de serem as mais robustas, assegurem também uma boa diversidade ao longo desta frente.

6. Organização da tese

Esta dissertação encontra-se dividida em sete capítulos, cada um dos quais referentes a aspectos diferentes do trabalho realizado. Neste primeiro capítulo é feita uma breve descrição do enquadramento do trabalho realizado e dos objectivos propostos.

No capítulo 2 é feita uma descrição mais pormenorizada dos conceitos essenciais (em especial o de dominância) associados à optimização multi-objectivo, passando pelos métodos existentes para resolver problemas deste tipo, dos quais se destacam os algoritmos evolucionários. Neste mesmo capítulo é também apresentada uma descrição dos algoritmos evolucionários, com particular relevo para os Algoritmos Genéticos. Por fim, são descritas as abordagens evolucionárias mais importantes para resolver problemas de optimização multi-objectivo.

O capítulo 3 está reservado ao estado da arte da Análise de Robustez em problemas de optimização, o qual começa por caracterizar os tipos de incerteza, as várias definições de robustez, e a análise de robustez em problemas de optimização (em especial optimização combinatória) e usando meta-heurísticas.

O capítulo 4 é dedicado ao estado da arte da Análise de Robustez em optimização evolucionária multi-objectivo, no qual são apresentadas uma caracterização da optimização evolucionária em ambientes incertos e as abordagens propostas de optimização robusta em ambientes sujeitos a perturbações relativamente aos seus valores nominais, incluindo as definições de grau de robustez em cada uma daquelas abordagens. Neste capítulo são também descritos os aspectos principais dos algoritmos evolucionários usados para testar o desempenho das abordagens propostas.

No capítulo 5 é apresentado um conjunto de testes e de resultados obtidos com um dos algoritmos evolucionários implementados e aplicado a um problema de teste bi-objectivo, que é normalmente usado na literatura da área da programação evolucionária, considerando perturbações nos valores das variáveis de decisão das soluções, servindo para avaliar o desempenho de uma das abordagens propostas neste trabalho.

No capítulo 6 é apresentado um estudo da aplicabilidade de uma das abordagens propostas a um caso real, mais concretamente ao problema da compensação do factor de potência numa rede eléctrica de distribuição do tipo radial. Este problema é bi-objectivo e considera-se que alguns dos dados do modelo associados às funções objectivo e às restrições estão sujeitos a perturbações.

O capítulo 7 está destinado às conclusões do trabalho realizado e às pistas de desenvolvimento a que este pode conduzir no futuro.

7. Contribuições científicas da tese

O trabalho realizado que serviu de base a esta dissertação deu origem a várias publicações nacionais e internacionais:

- dois relatórios de investigação
  - Research Report N.5/2004, INESC – Coimbra (Antunes et al. (2004)),
  - Research Report N.3/2006, INESC – Coimbra (Barrico e Antunes (2006-d);
- quatro conferências internacionais (com arbitragem pelos pares)
  - The 6th Metaheuristics International Conference (MIC2005), Viena, Áustria (Antunes et al. (2005)),
  - II European-Latin-American Workshop on Engineering Systems (SELASI II), Porto, Portugal (Barrico e Antunes (2006-a)),
  - The 7th International Conference on Multi-Objective Programming and Goal Programming (MOPGP 2006), Tours, France (Barrico e Antunes (2006-b)),
  - 2006 IEEE World Congress on Computational Intelligence (WCCI 2006), Vancouver, Canadá (Barrico e Antunes (2006-c));
- um capítulo de livro (com arbitragem pelos pares)
  - Shengxiang Yang, Yew-Soon Ong, Yaochu Jin (Eds.), "Evolutionary Computation in Dynamic and Uncertain Environments", Series on Studies in Computational Intelligence (SCI), Volume 51, Springer (Barrico e Antunes (2007)).

Capítulo 2

Optimização Evolucionária Multi-Objectivo

1. Optimização mono-objectivo

Um problema de optimização mono-objectivo é composto por uma função objectivo (a minimizar ou a maximizar) e, normalmente, por várias restrições que todas as soluções admissíveis têm de satisfazer. De uma maneira geral, um problema de optimização mono-objectivo pode ser formulado da seguinte forma:

Minimizar $f(x)$,

sujeito a $g_j(x) \geq 0$, $j = 1, 2, ..., J$ (1)

$h_k(x) = 0$, $k = 1, 2, ..., K$ (2)

$x_i^{inf} \leq x_i \leq x_i^{sup}$, $i = 1, 2, ..., n.$ (3)

O problema tem associado J+K restrições funcionais, das quais J são de desigualdade, $g_j(x)$, e K de igualdade, $h_k(x)$. As restrições de desigualdade podem ser do tipo "≥" ou "≤", apesar da forma geral apresentar apenas restrições do tipo "≥", o que não implica perda de generalidade.

Uma solução x é um vector de n variáveis de decisão: $x = (x_1, ..., x_n)$. As restrições do último conjunto denominam-se por *limites das variáveis de decisão*, as quais obrigam a que cada variável de decisão x_i assuma um valor entre um limite inferior, x_i^{inf}, e um limite superior, x_i^{sup}. Estes limites constituem o *espaço das variáveis de decisão*, *espaço das decisões* ou *espaço das soluções*.

Uma solução *x* que satisfaça todas as J+K restrições funcionais e todos os 2n limites das variáveis de decisão denomina-se por *solução admissível*. Ao conjunto de todas as soluções admissíveis dá-se o nome de *região admissível,* X*.*

Se a função objectivo e todas as restrições de um problema são funções lineares em relação a *x*, então este problema denomina-se de *linear*. Pelo contrário, se a função objectivo ou alguma das restrições são não lineares em relação a *x*, então este problema denomina-se de *não linear*.

Num problema de optimização contínua, a pesquisa da solução óptima é feita num conjunto infinito de pontos (o espaço de pesquisa é infinito). Neste tipo de problemas, as variáveis de decisão são normalmente reais.

Quando a pesquisa da solução óptima é feita num conjunto finito de pontos (o espaço de pesquisa é finito), então diz-se que se está perante um problema de optimização discreta ou combinatória.

Na resolução de um problema de optimização mono-objectivo, pretende-se determinar a *solução óptima*, ou seja, a solução admissível que optimize a função objectivo, cujo valor é único, mesmo que existam soluções óptimas alternativas.

Os dados do problema são todos os coeficientes e parâmetros associados à função objectivo e às restrições funcionais. Se todos estes dados são valores únicos, então o problema designa-se de *determinístico*. Se isto não acontece, então existe alguma incerteza a envolver alguns (ou todos) estes dados e o problema designa-se de *não determinístico*.

2. Optimização multi-objectivo

Um problema de optimização é do tipo multi-objectivo se existem duas ou mais funções objectivo associadas ao problema. Neste tipo de problemas, o conceito de solução óptima, característico de problemas de optimização mono-objectivo, não se pode aplicar, uma vez que uma solução admissível que optimize um dos objectivos, não optimiza, em geral, os restantes objectivos, quando estes estão em conflito.

2.1. Formulação de um problema de optimização multi-objectivo

Um problema de optimização multi-objectivo é composto por duas ou mais funções objectivo, que se pretendem maximizar ou minimizar e, normalmente, por várias restrições funcionais que todas as soluções admissíveis têm de satisfazer.

De uma maneira geral, um problema de optimização multi-objectivo pode ser formulado da seguinte forma:

Minimizar $f_m(x)$, $m = 1, 2, ..., M$

sujeito a $x \in X$ ((1) - (3)).

Cada uma das M funções objectivo, $f(x) = (f_1(x), f_2(x), ..., f_M(x))^T$, com $M \geq 2$, pode ser a minimizar ou a maximizar. Como foi referido anteriormente, esta formulação não implica perda de generalidade.

Uma das diferenças fundamentais entre optimização mono-objectivo e optimização multi-objectivo é que, ao espaço das variáveis de decisão há que acrescentar um outro espaço multi-dimensional gerado pelas funções objectivo, denominado de *espaço das funções objectivo*, ou simplesmente, *espaço dos objectivos*. Para cada solução $x = (x_1,..., x_n)$ no espaço das variáveis de decisão, existe um ponto que lhe corresponde no espaço dos objectivos, denotado por $f(x) = z = (z_1,..., z_M)^T$, com $z_m = f_m(x)$ e $m = 1,...,M$.

A região admissível no espaço das funções objectivo (o conjunto de todas as imagens dos pontos em X), pode então ser definida da seguinte forma:

$$Z = \{ z \in \mathbb{R}^M : z = \left(f_1(x), f_2(x), ..., f_M(x)\right),\ x \in X \}.$$

2.2. Princípios de optimização multi-objectivo

Na resolução de problemas multi-objectivo, pretende-se encontrar a *"melhor" solução de compromisso* para o AD, que possa constituir uma *solução final* do problema de decisão, ou então, fazer a caracterização, exaustiva ou não, da região não dominada. Desta forma, a noção de solução óptima usada em problemas de optimização mono-objectivo é substituída pela noção de *solução não dominada* (também designada por *óptima de Pareto*, *eficiente* ou *não inferior*).

Uma solução não dominada é uma solução admissível para a qual não é possível melhorar simultaneamente todas as funções objectivo; isto é, a melhoria numa função objectivo apenas pode ser alcançada por degradação de pelo menos uma das outras. Por outro lado, uma solução admissível diz-se dominada por outra, se ao passar-se da primeira para a segunda existir melhoria de pelo menos uma função objectivo, permanecendo inalterados as restantes (Clímaco et al. (2003)).

No conjunto das soluções não dominadas, também denominado por frente óptima de Pareto ou apenas frente de Pareto, têm de ser satisfeitas as duas condições seguintes:

1) quaisquer duas soluções deste conjunto têm de ser não dominadas entre si;
2) qualquer solução que não pertença a este conjunto é dominada por, pelo menos, uma solução deste conjunto.

Quando as funções objectivo de um problema de optimização multi-objectivo estão em conflito, normalmente o conjunto de soluções não dominadas é enorme. Desta forma, é difícil escolher uma solução deste conjunto, sem dispor de informação adicional relativa ao problema a resolver. No entanto, na ausência de tal informação, as soluções não dominadas não são comparáveis entre si, o que motiva as abordagens a determinar o número máximo possível de soluções não dominadas do problema. Assim, pode-se caracterizar o problema de optimização multi-objectivo como a determinação de um conjunto de soluções não dominadas com as seguintes características:

1) esteja o mais próximo possível da frente óptima de Pareto real,
2) seja o mais diversificado possível.

A primeira característica é inerente a qualquer tarefa de optimização; também em optimização mono-objectivo se pretende determinar uma solução admissível que garanta o valor óptimo para o modelo matemático.

A segunda característica é específica da optimização multi-objectivo. Apenas com um conjunto de soluções dispersas é possível garantir a existência de um bom conjunto de soluções de compromisso entre os objectivos. Uma vez que a optimização multi-objectivo actua em dois espaços, das variáveis de decisão e dos objectivos, a diversidade das soluções pode ser definida nestes dois espaços. Apesar de, na maioria dos problemas, a diversidade num dos espaços significar, normalmente, a diversidade no outro espaço, isto pode não acontecer em alguns problemas.

2.3. Diferenças com a optimização mono-objectivo

Existem algumas diferenças fundamentais entre a optimização multi-objectivo e a optimização mono-objectivo, as quais estão relacionadas com as metas a atingir, com os números de espaços de pesquisa onde se actua e pela necessidade de considerar funções escalares substitutas nalguns processos de cálculo (ver também Deb (2001)).

Em optimização mono-objectivo existe apenas uma meta a atingir: determinar uma solução óptima global (apesar de no espaço de pesquisa poderem existir soluções óptimas locais, é sempre uma solução óptima global que se pretende). No entanto, no caso da optimização multi-modal pretende-se determinar vários óptimos locais e globais.

Em optimização multi-objectivo pretendem-se atingir duas metas: fazer convergir o conjunto de soluções não dominadas obtido em direcção à frente óptima de Pareto real e manter um conjunto diversificado de soluções na frente obtida. Utilizando diferentes formas de compensação dos objectivos pode obter-se, em princípio, uma maior variedade de soluções não dominadas, obtendo-se assim um conjunto diversificado de soluções próximo da frente óptima de Pareto real.

Em optimização mono-objectivo existe apenas um espaço de pesquisa multi-dimensional, o espaço das variáveis de decisão, onde o algoritmo actua, aceitando e rejeitando soluções de acordo com os respectivos valores da função objectivo.

Em optimização multi-objectivo, para além do espaço das variáveis de decisão, é necessário considerar também o espaço dos objectivos, os quais, embora estejam relacionados, muitas vezes as suas propriedades não são semelhantes. Por exemplo, o nível de proximidade existente entre duas soluções pode ser muito diferente nos dois espaços. Assim, para garantir a diversidade no conjunto de soluções obtidas é importante decidir em que espaço deve ser alcançada a diversidade.

Apesar da pesquisa de soluções, por parte de qualquer algoritmo de optimização, ser efectuada no espaço das variáveis de decisão, o seu comportamento neste espaço pode ser mapeado no espaço dos objectivos. Nalguns algoritmos, o comportamento definido no espaço dos objectivos é usado para guiar a pesquisa no espaço das variáveis de decisão. Neste caso, o comportamento do algoritmo deve ser coordenado em ambos os espaços, de tal modo que a geração de novas soluções no espaço das variáveis de decisão possa contribuir para a necessária diversidade no espaço dos objectivos.

A resolução de problemas multi-objectivo envolve, geralmente, a utilização de funções escalares substitutas que agregam (temporariamente) as múltiplas funções objectivo numa única dimensão, de tal modo que a solução óptima da função escalar é não dominada do problema multi-objectivo original (Clímaco et al. (2003)). As funções escalares substitutas mais usadas consistem em somas pesadas, selecção de uma das funções considerando as outras como restrições e distância a um ponto de referência.

No caso da soma pesada, é criada uma única função objectivo composta, resultante da soma ponderada dos vários objectivos. O segundo caso consiste em escolher uma das funções objectivo para optimização e transformar as restantes em restrições do problema, limitando cada uma destas restrições a certos limiares predefinidos. As funções escalares baseadas na distância a um ponto de referência permitem obter soluções não dominadas minimizando a distância, segundo uma determinada métrica, da região admissível a um qualquer ponto de referência do espaço dos objectivos.

2.4. Dominância e optimalidade de Pareto

A maioria dos algoritmos de optimização multi-objectivo usa, no seu processo de pesquisa, o conceito de dominância. Nesta secção, vão ser apresentados os conceitos de dominância e de alguns termos relacionados, assim como algumas técnicas para identificar soluções não dominadas numa população de soluções de tamanho finito.

2.4.1. Pontos especiais

Os algoritmos de optimização multi-objectivo usam, muitas vezes, conceitos associados a alguns pontos ("soluções") especiais no espaço dos objectivos: ponto ideal e ponto nadir.

O ponto definido no espaço dos objectivos cujas componentes são o valor óptimo de cada função objectivo na região admissível, quando optimizadas separadamente, designa-se por *ponto ideal*, z^*. Geralmente, o ponto ideal não corresponde a uma solução admissível, uma vez que a solução óptima para cada função objectivo não é a mesma (os M objectivos estão, em geral, em conflito). No entanto, o ponto ideal tem grande utilidade, pois é usado em muitos algoritmos de optimização multi-objectivo como solução de referência.

O *ponto nadir*, z^{nad}, representa os piores valores de cada função objectivo no conjunto óptimo de Pareto. No entanto, dada a dificuldade computacional associada ao cálculo deste ponto, muitas vezes é usado como substituto de z^{nad} o ponto definido pelos piores valores de cada função objectivo entre todas as soluções que optimizam individualmente cada função (a chamada tabela de "pay-off").

2.4.2. Conceitos de dominância e de optimalidade de Pareto

A maioria dos algoritmos de optimização multi-objectivo usa o conceito de dominância, em particular quando há necessidade de comparar duas soluções, para averiguar se há dominância de uma sobre a outra.

Uma solução $x^1 \in X$ *domina* uma outra solução $x^2 \in X$, se e só se x^1 não é pior do que x^2 para todos os objectivos e x^1 é estritamente melhor do que x^2 para pelo menos um dos M objectivos.

Apesar de serem diferentes os conceitos de solução eficiente e de solução não dominada, quando utilizados de forma genérica nesta dissertação não é feita qualquer distinção entre eles. Enquanto que o conceito de solução eficiência se refere, geralmente, a pontos do espaço de decisão, o conceito de solução não dominada é utilizada para pontos do espaço dos objectivos; isto é, uma solução não dominada é a imagem de uma solução eficiente.

Matematicamente, e considerando todas as funções objectivo a minimizar, tem-se (Clímaco et al. (2003)):

a) Uma solução $x^1 \in X$ é *eficiente* se e só se não existe uma outra solução $x^2 \in X$ tal que $z_m(x^2) \leq z_m(x^1)$ para todo o m (m = 1, 2, ..., M) e $z_m(x^2) < z_m(x^1)$ para pelo menos um m.

b) Um ponto do espaço dos objectivos $z^1 \in Z$, com $z^1 = (z_1(x^1), ..., z_M(x^1))$ e $x^1 \in X$, diz-se *não dominado* se e só se x^1 é uma solução eficiente.

Uma solução de compromisso satisfatória para o problema multi-objectivo deverá ser não dominada, cujos valores das funções objectivo são satisfatórios para o AD e de tal modo que seja aceitável como solução final. Desta forma, é apenas sobre o conjunto das soluções não dominadas que deve recair a atenção do analista[5] e do AD.

Entre quaisquer duas soluções não dominadas a uma melhoria em pelo menos um dos objectivos encontra-se sempre associado um sacrifício em pelo menos um dos outros objectivos. Isto é, verifica-se sempre uma compensação entre objectivos no conjunto das soluções não dominadas.

A relação de dominância entre duas soluções, tal como foi definida atrás, é muitas vezes referida como uma relação de dominância *fraca*: x^1 *domina fracamente* uma solução x^2, se x^1 não é pior do que x^2 em todos os M objectivos e é estritamente melhor do que x^2 em pelo menos um dos M objectivos. A partir da definição de *dominância fraca*, pode-se obter a definição de *dominância estrita (forte)*, da forma que se segue.

Uma solução x^1 *domina estritamente (fortemente)* uma solução x^2, se a solução x^1 é estritamente melhor do que a solução x^2 em todos os M objectivos. Desta forma, se uma solução x^1 *domina estritamente (fortemente)* uma solução x^2, a solução x^1 também *domina fracamente* a solução x^2, mas o recíproco não é verdadeiro.

Tal como existem soluções óptimas globais e locais em optimização mono-objectivo, também poderão existir *frentes óptimas de Pareto globais* e *locais* em optimização multi-objectivo.

Ao conjunto de soluções não dominadas de toda a região admissível dá-se o nome de *frente óptima de Pareto global*. A frente óptima de Pareto global é referida simplesmente como *frente óptima de Pareto* ou *frente de Pareto*. Uma vez que as soluções deste conjunto são não dominadas em relação a qualquer solução da região admissível, elas são as melhores soluções do problema de optimização multi-objectivo.

Define-se, também, frente óptima de Pareto local da seguinte forma (Deb (1999) e Miettinen (1999)): se para cada solução x de um conjunto P'' não existe nenhuma solução y (na vizinhança de x, tal que $\|y-x\|_\infty \leqslant \varepsilon$, em que ε é um número positivo muito pequeno), que domine qualquer solução de P'', então as soluções que pertencem ao conjunto P'' constituem um conjunto óptimo de Pareto local.

[5] Entidade responsável pela análise ou estudo do problema, com conhecimentos técnicos sobre programação multi-objectivo, que serve de mediador entre o AD e as componentes metodológica e computacional do processo de apoio à decisão (Clímaco et al. (2003)).

De acordo com esta definição, uma frente óptima de Pareto global é também uma frente óptima de Pareto local.

A definição de dominância forte pode ser usada para definir *frente fracamente não dominada*. Dado um conjunto de soluções, P, a frente fracamente não dominada, P', é formada pelas soluções que não são fortemente dominadas relativamente a qualquer outra solução do conjunto P (Deb (2001)).

As definições de frente fracamente não dominada global e local podem também ser feitas de forma semelhante, a partir da definição de frente fracamente não dominada.

2.4.3. Determinação de uma frente não dominada

Existem vários métodos para determinar uma frente não dominada, a partir de uma dada população de soluções. A diferença entre estes métodos reside essencialmente na complexidade computacional de cada um deles.

Um método consiste em comparar cada solução *x* de uma população com todas as outras soluções desta população, para verificar se *x* é dominada por alguma delas.

Um outro método consiste em comparar cada solução da população com as soluções de uma outra população parcialmente ocupada com soluções não dominadas entre si. O processo começa por inserir a primeira solução de P num conjunto vazio P'. Depois, compara-se cada solução *x* de P com todas as soluções do conjunto P', uma a uma. Se a solução *x* domina alguma solução de P', então todas as soluções de P' que são dominadas por *x* são removidas de P' e *x* é inserida em P'. Pelo contrário, se a solução *x* é dominada por uma solução de P', então a solução *x* é ignorada. Quando todas as soluções da população forem comparadas, as soluções do conjunto P' formam a frente não dominada.

2.4.4. Determinação de todas as frentes não dominadas

A maioria dos algoritmos de optimização multi-objectivo permite determinar apenas a frente não dominada de uma população. Estes algoritmos classificam a população de soluções apenas em dois conjuntos (frentes): o primeiro constituído pelas soluções não dominadas e o segundo pelas dominadas. Neste caso, todas as soluções dominadas são consideradas em pé de igualdade.

No entanto, existem outros algoritmos que classificam a população por níveis de dominância. As soluções da população são ordenadas crescentemente por níveis de dominância: no primeiro nível (menor índice) estão as melhores soluções e no último nível (maior índice) estão as piores soluções, em termos de dominância. Portanto, qualquer solução de um certo nível de dominância, é dominada por pelo menos uma

solução de cada nível mais baixo (Figura 2.1). Neste caso, as soluções mais importantes são as não dominadas (nível 1 na Figura 2.1), mas entre as dominadas podem existir vários níveis de dominância (níveis 2, 3, 4 e 5 na Figura 2.1). As soluções classificadas no nível 2 podem ser consideradas mais interessantes do que as do nível 3, e assim sucessivamente, até às do nível 5.

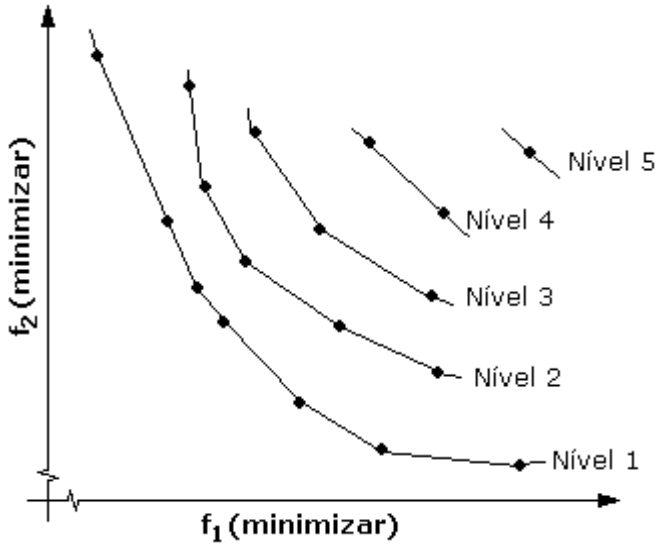

Figura 2.1: Classificação das soluções por níveis de dominância.

A diferença fundamental entre os vários algoritmos existentes para determinar as várias frentes não dominadas de uma população está na complexidade de cada um deles, quer em termos do espaço de armazenamento dos dados, quer em termos da velocidade de execução.

Um dos processos utilizados para determinar as várias frentes associadas aos vários níveis de dominância de uma população consiste em determinar primeiro a frente de nível 1 (composta pelas soluções não dominadas da população) e, depois, as restantes frentes para os restantes níveis (compostas pelos soluções dominadas da população).

Para determinar as soluções da frente de nível 1, basta tomar as soluções não dominadas da população. Para determinar as soluções de cada frente de nível k > 1, é necessário efectuar os dois passos seguintes:

1º) ignorar temporariamente todas as soluções das frentes dos níveis anteriores,
2º) determinar as soluções não dominadas entre as restantes soluções da população, as quais constituem a frente não dominada de nível k.

Estes dois passos repetem-se até todas as soluções da população estarem inseridas numa frente não dominada.

Um outro processo consiste em determinar, inicialmente, para cada solução x^i um contador n_i com o número de soluções que dominam x^i e um conjunto P_i com as soluções que são dominadas por x^i. No final desta fase do processo, todas as soluções que tenham os respectivos contadores a zero são aquelas que pertencem à frente de nível 1 (constituída pelas solução não dominadas da população). Depois, para cada uma destas soluções x^i, com $n_i = 0$, analisa-se o respectivo conjunto P_i de tal forma que, para cada uma das soluções x^j deste conjunto sejam reduzidos os respectivos contadores n_j em uma unidade e transferidas as soluções cujos contadores se tornaram nulos para um conjunto P'. Desta forma, as soluções que estão em P' têm todas os respectivos contadores a zero, o que significa que estas soluções constituem a frente de nível 2. Este processo é repetido até todas as soluções da populações estarem classificadas nos vários níveis de dominância (e distribuídas pelas diversas frentes).

Se na população também existirem solução não admissíveis, então estas devem pertencer a um terceiro grupo de frentes. Desta forma, são determinados três grupos de frentes: o primeiro composto apenas por uma frente (com as soluções não dominadas da população), um segundo grupo que pode conter várias frentes (com soluções admissíveis dominadas), e um terceiro grupo que pode conter também várias frentes (com soluções não admissíveis).

3. Métodos para resolver problemas de optimização MO

Nos últimos anos tem sido dedicado um esforço considerável ao desenvolvimento de métodos para resolução de problemas de optimização multi-objectivo, tendo em atenção o contributo que o AD pode fornecer na procura das soluções não dominadas do problema, através da expressão do seu sistema de preferências. Estes métodos podem ser divididos em dois grupos: métodos clássicos e meta-heurísticos (onde se incluem os algoritmos evolucionários).

3.1. Métodos clássicos

Ao longo dos anos, alguns autores têm procurado classificar os métodos dedicados a problemas de optimização multi-objectivo de acordo com vários aspectos, como, por exemplo (Clímaco et al. (2003)): grau de intervenção do AD, tipo de modelação das preferências do AD, número de ADs, certeza/incerteza na determinação dos dados do modelo, e dados do modelo requeridos e/ou resultados obtidos.

No entanto, a classificação dos métodos baseada no grau de intervenção do AD é a que tem sido mais usada na literatura sobre os problemas de optimização multi-

objectivo, a qual agrupa os métodos em três tipos (Clímaco et al. (2003)): *articulação a posteriori das preferências* (*geradores*), *articulação a priori das preferências* e *articulação progressiva das preferências* (*interactivos*). Uma classificação deste tipo é também considerada por Cohon (1978) e Steuer (1986).

Nos *métodos a posteriori* são calculadas algumas, ou mesmo todas (quando isto é possível), as soluções não dominadas do problema, que são depois colocadas à disposição do AD para serem avaliadas. Estes métodos exigem um elevado esforço computacional, pois utilizam um cálculo exaustivo para caracterizar o conjunto de soluções não dominadas e não têm em atenção as preferências do AD que poderiam servir de apoio na resolução do problema (identificação de uma solução de compromisso aceitável como solução final).

Nos *métodos a priori*, o AD começa por indicar as suas preferências, a partir das quais é possível transformar o problema multi-objectivo inicial num problema mono-objectivo, por exemplo, através da construção de uma função utilidade. Desta forma, apesar de existir modelação multi-objectivo do problema, pretende-se optimizar a função utilidade, cuja solução óptima será a solução final. A maior dificuldade deste processo está em obter os parâmetros para construir uma função utilidade que agregue, numa única dimensão, todos os objectivos em análise.

Nos *métodos progressivos*, o AD expressa as suas preferências através de um processo de diálogo com a componente procedimental, de forma a conduzir a pesquisa para a zona da região admissível onde se localizam as soluções que melhor correspondem ao seu sistema de preferências (isto é, existe uma articulação progressiva das preferências). Desta forma, estes métodos, ao aproveitarem a intervenção do AD, reduzem o âmbito da pesquisa, de forma a minimizar o esforço quer computacional, quer do próprio AD no processamento da informação.

Nos métodos de articulação *a posteriori* das preferências (métodos geradores) podem, por exemplo, usar-se processos de cálculo de soluções não dominadas baseados na optimização de funções escalares soma ponderada das múltiplas funções objectivo ou na escolha de uma das funções para optimizar considerando as restantes M-1 como restrições. A intenção é sempre fazer uma caracterização o mais exaustiva possível do conjunto de soluções não dominadas, sem requerer a intervenção do AD. Por exemplo, em problemas de programação linear multi-objectivo a intenção é geralmente calcular todos os vértices (soluções básicas) não dominados, a partir dos quais podem ser caracterizadas as faces, de dimensão superior, não dominadas.

Nos métodos de articulação *a priori* das preferências podem, por exemplo, ser construídas funções de utilidade que se destinem a representar analiticamente as preferências do AD, que devem ser válidas em todo o espaço de pesquisa admissível,

baseadas na minimização da distância (de acordo com uma dada métrica) à solução ideal (ou outro ponto de referência), em processos lexicográficos ou em programação por metas.

A principal característica dos métodos interactivos é o envolvimento do AD no decurso do processo de optimização, usando a expressão das suas preferências face às soluções que lhe vão sendo apresentadas para orientar a pesquisa, alternando assim fases de cálculo de soluções não dominadas e fases de diálogo para recolha de informação que é usada na fase de cálculo seguinte. O facto de o AD ser envolvido no processo de optimização, tornou estes métodos populares para a resolução de problemas práticos. Em Miettinen (1999) e Clímaco et al. (2003) encontra-se informação mais detalhada sobre estes métodos.

3.2. Meta-heurísticas

Devido à dificuldade computacional associada à resolução de alguns problemas de optimização, particularmente de natureza combinatória e em que existem múltiplos óptimos locais, usando os métodos clássicos de programação matemática, foram desenvolvidas técnicas meta-heurísticas para tentar atenuar esta dificuldade.

Reeves (1995) definiu *heurística*, para problemas mono-objectivo, como uma técnica que procura obter boas soluções (isto é, quase óptimas), com um custo computacional razoável, sem ser capaz de garantir, quer a admissibilidade, quer a optimalidade, e até mesmo, em muitos casos, determinar o quanto uma solução admissível obtida está próxima do óptimo.

Ao contrário dos métodos clássicos, as heurísticas são "menos ambiciosas", na medida em que vão calculando o valor de uma função objectivo em pontos julgados promissores, os quais vão sendo obtidos ao longo de um processo iterativo. Terminado o processo de optimização (para o caso mono-objectivo) deverá ter-se encontrado uma solução "quase óptima" (esta solução até poderá ser o óptimo global, mas não há certeza disso). A solução obtida, apesar de possivelmente não ser a óptima, define um limite superior (problemas de minimização) ou inferior (problemas de maximização) para o valor óptimo.

As heurísticas são técnicas em que, embora a exploração seja feita de forma algorítmica, o progresso é obtido pela avaliação puramente empírica do resultado. O uso de técnicas heurísticas é vantajoso em certas circunstâncias, tais como quando os dados do problema são escassos e incertos, quando não existem métodos exactos ou quando existindo requerem um elevado esforço computacional, ou quando o AD precisa de obter rapidamente um vasto conjunto de soluções para escolher uma delas.

A maioria das heurísticas surgiu para resolver tipos específicos de problemas, mas deixam de ser aplicáveis logo que algum elemento particular perturbe o modelo a que se destinam. Desta forma, a necessidade de existirem técnicas sólidas e flexíveis, que pudessem ser facilmente adaptadas a um vasto conjunto de problemas, levou ao aparecimento das técnicas *meta-heurísticas*.

O conceito de *meta-heurística* foi introduzido por Glover (1986), em simultâneo com o conceito de *pesquisa tabu* (que é uma das meta-heurísticas mais utilizadas). Uma *meta-heurística* é um processo iterativo de geração de soluções, que utiliza uma ou mais heurísticas subordinadas, combinando diferentes formas de exploração do espaço de pesquisa (a geração de novas soluções em regiões ainda não testadas — "exploration" — ou a concentração na vizinhança de soluções já conhecidas — "exploitation").

Uma *meta-heurística* pode também ser definida como um conjunto de técnicas que podem ser usadas para definir heurísticas, as quais podem ser aplicadas a um vasto conjunto de problemas diferentes. Por outras palavras, uma *meta-heurística* pode ser vista como um esquema algorítmico geral que pode ser aplicado a diferentes problemas de optimização, em que são necessárias relativamente poucas modificações para que se adapte a um problema específico (Sörensen (2003)). Ou seja, uma meta-heurística é um processo de mais alto nível que fornece linhas directrizes no desenvolvimento de métodos para a resolução de um problema específico.

Alguns das características que favorecem a utilização de *meta-heurísticas* são as seguintes:

- enquanto que as heurísticas são dependentes da especificidade do problema que tenta resolver, o domínio de aplicação das *meta-heurísticas* é mais amplo;
- enquanto que as heurísticas assentam em processos iterativos que geralmente terminam quando não for encontrada uma solução que melhore a anterior, as *meta-heurísticas* incorporam estratégias de exploração do espaço das soluções para além da optimalidade local;
- as meta-heurísticas inspiram-se em processos estudados em áreas tão diversas como as ciências sociais, a física, a biologia, etc..

Existem vários tipos de *meta-heurísticas*, dependendo das técnicas de exploração do espaço de pesquisa, e que, segundo Sörensen (2003), podem ser divididas em três grandes classes: baseadas na vizinhança, baseadas em populações e híbridas.

As *meta-heurísticas baseadas na vizinhança*, também denominadas por *meta-heurísticas de pesquisa local*, usam uma operação chamada *movimento* para, iterativamente, se deslocarem de uma solução para uma outra. Ao conjunto de todas as

soluções que podem ser atingidas pelo movimento de uma dada solução *x*, dá-se o nome de *vizinhança*, V(*x*), daquela solução. Estas soluções podem ser geradas aleatoriamente.

Geralmente, um operador de *pesquisa local* tenderá a encontrar apenas soluções óptimas locais, isto é, soluções em que na sua vizinhança não há soluções melhores. O principal objectivo do esquema meta-heurístico é dar a possibilidade de o operador de pesquisa local se afastar de um óptimo local, pois, geralmente, este óptimo não é o óptimo global.

Alguns exemplos de *meta-heurísticas* de pesquisa local são as seguintes: *pesquisa tabu*, *arrefecimento simulado*[6], GRASP ("*Greedy Randomized Adaptive Search Procedure*") e *pesquisa iterativa local*.

As *meta-heurísticas baseadas em populações* mantêm um conjunto (população) de soluções. Estas meta-heurísticas tentam melhorar as soluções de uma população, permitindo que elas se combinem entre si. Estas combinações são efectuadas até que seja satisfeito um certo critério de paragem.

Como exemplos deste tipo de meta-heurísticas podem-se referir os seguintes: *algoritmos evolucionários*, *colónias de formigas*, *pesquisa por dispersão*[7] e *algoritmos meméticos* (também podem ser considerados como meta-heurísticas híbridas).

As meta-heurísticas híbridas utilizam técnicas das meta-heurísticas baseadas na vizinhança e em populações no mesmo processo de pesquisa; por exemplo, um algoritmo genético que utiliza a pesquisa tabu para melhorar uma solução da sua população. Alguns trabalhos publicados que usam meta-heurísticas híbridas são: algoritmo genético com pesquisa local (Prins (2004)), algoritmo genético com arrefecimento simulado (Mahfoud e Goldberg (1995)), e algoritmo genético com pesquisa local e pesquisa tabu (Fleurent e Ferland (1994)).

4. Algoritmos Evolucionários

Os Algoritmos Evolucionários são procedimentos meta-heurísticos, que tentam abstrair e imitar alguns dos mecanismos da evolução natural, de forma a resolverem problemas que exijam adaptação, pesquisa e optimização. Estes métodos são estocásticos e iterativos, não garantindo a convergência para a melhor solução, uma vez que se baseiam em heurísticas. Estes algoritmos operam sobre um conjunto de indivíduos (população), em que cada indivíduo representa uma potencial solução do problema de optimização.

[6] Tradução do termo inglês "simulated annealing".

[7] Tradução do termo inglês "scatter search".

O processo evolucionário começa por gerar, aleatoriamente ou usando técnicas específicas para esse efeito, a primeira população (também denominada por população inicial), seguindo-se a avaliação de cada solução desta população usando uma função de aptidão. A função de aptidão atribui um valor a cada uma das soluções da população (valor de aptidão — "fitness"), o qual é uma medida da qualidade da solução do problema em estudo e é utilizado para orientar na pesquisa de novas soluções.

O processo evolucionário prossegue, aplicando às soluções da população principal (pode ou não ser a inicial), um procedimento para gerar uma nova população constituído pelas quatro etapas principais: selecção, cruzamento, mutação e substituição. A selecção consiste em criar uma população provisória com algumas das soluções da população principal. As soluções com maior valor de aptidão têm maior probabilidade de estarem presentes na população provisória do que as soluções com menor valor de aptidão. A estas soluções é então aplicado o operador genético cruzamento seguido de mutação, de forma a gerar uma nova população. A seguir, as soluções da população principal são substituídas pelas novas soluções, fazendo com que, normalmente, as soluções com melhores valores de aptidão se mantenham e as com os piores valores de aptidão sejam removidas. Este processo só termina, geralmente, quando for atingido um número predefinido de gerações, ou quando não for detectada uma melhoria significativa nas últimas gerações.

Existem vários tipos de Algoritmos Evolucionários, devendo-se a classificação destas técnicas a factores históricos relacionados com o aparecimento de diferentes linhas de investigação. No entanto, as técnicas mais representativas são as Estratégias Evolucionárias, a Programação Evolucionária e os Algoritmos Genéticos.

Tem sido amplamente reconhecida a vocação das abordagens evolucionárias para tratar problemas multi-objectivo (Deb (2001), Coello et al. (2002) e Osyczka (2002)), pela sua capacidade de trabalhar com uma população de soluções (não dominadas).

4.1. Estratégias Evolucionárias

As Estratégias Evolucionárias surgiram a partir do trabalho de Rechenberg (1964) (citado em Knowles e Corne (2000) e Costa e Oliveira (2002)), tendo sido desenvolvidas, inicialmente, por Rechenberg (1973) (citado em Bäck et al. (1997)) com a finalidade de resolver problemas de optimização. Posteriormente foram desenvolvidos novos esquemas evolucionários, mas seguindo os mesmos princípios, por Schwefel (1981). Estas técnicas foram desenvolvidas para serem aplicadas a problemas de engenharia.

A primeira versão desta técnica usa apenas duas soluções, uma progenitora e uma descendente, as quais são codificadas usando uma representação decimal. Esta versão considera apenas o operador genético mutação e um mecanismo de selecção (não

considera o operador cruzamento). Este tipo de Estratégias Evolucionárias designa-se por EE-(1+1), na actual notação de Estratégias Evolucionárias (Bäck et al. (1991)).

Surgiram, entretanto, algumas generalizações desta técnica, nomeadamente a sua aplicação a mais do que uma solução, nas denominadas EE-(μ+1). Nesta variante consideram-se μ soluções progenitoras, das quais é seleccionada apenas uma delas para a geração de uma descendente.

Mais tarde, desenvolveram-se outras variantes, designadas por EE-(μ+λ) e EE-(μ,λ), nas quais são geradas por mutação λ descendentes a partir de μ progenitoras e aplicados esquemas de selecção diferentes para escolher as μ melhores para a geração seguinte. No método EE-(μ+λ) as soluções da população para a geração seguinte são escolhidas entre as μ progenitoras e as λ descendentes. No método EE-(μ,λ) as soluções são escolhidas unicamente das λ descendentes (assumindo que $\lambda > \mu$).

Apesar das Estratégias Evolucionárias originais usarem apenas o operador genético mutação para gerar novos indivíduos, foi introduzido, posteriormente, o operador genético cruzamento, o qual foi aplicado juntamente com a mutação (Schwefel (1995)).

As Estratégias Evolucionárias revelaram-se algoritmos de optimização sólidos e eficientes, não exigindo nenhuma condição relativa à continuidade e convexidade do espaço de pesquisa, ao contrário de outros algoritmos de optimização (Schwefel (1995)).

4.2. Programação Evolucionária

A Programação Evolucionária foi introduzida por Fogel et al. (1966), com a finalidade de gerar máquinas de estados finitos para tarefas de pré-edição de sequências de símbolos. Actualmente, estes algoritmos são utilizados sobretudo em problemas de optimização numérica.

A principal característica da Programação Evolucionária está relacionada com a forma como os descendentes são gerados: o operador genético mutação é o único responsável pela geração de descendentes. O operador mutação é definido consoante o problema e está sujeito a adaptação durante o processo evolucionário.

Como não existe operador genético cruzamento, é possível usar qualquer representação para as soluções (está dependente do tipo de problema que se queira resolver), desde que se defina um operador mutação adequado ao problema. Na versão original, por exemplo, cada solução era uma máquina de estados finitos representada por um grafo. Em problemas de optimização numérica, as soluções são representadas por vectores de valores reais e para o problema do caixeiro-viajante, por exemplo, são utilizadas listas ordenadas.

A Programação Evolucionária simula a evolução natural como um processo baseado nas soluções, realçando o comportamento na ligação entre os progenitores e os descendentes, em vez da ligação genética, como acontece com os restantes Algoritmos Evolucionários.

4.3. Algoritmos Genéticos

Os Algoritmos Genéticos (AGs) são o grupo dos Algoritmos Evolucionários mais utilizados na resolução de problemas de pesquisa e de optimização. Estes algoritmos são métodos adaptativos inspirados nos mecanismos de evolução de populações de seres vivos, as quais evoluem de acordo com os princípios da selecção natural, descritos pela primeira vez por Charles Darwin[8].

Os princípios básicos dos AGs foram introduzidos por Holland (1975) e desenvolvidos por ele e pelos seus colaboradores (alunos e colegas) na Universidade de Michigan, nas décadas de 1960 e 1970. Os objectivos da pesquisa de Holland foram os seguintes (Goldberg (1989)):

a) abstrair e explanar com rigor os processos adaptativos dos sistemas naturais, e
b) construir *software* para sistemas artificiais que retenham os mecanismos importantes dos sistemas naturais.

No entanto, foram os trabalhos realizados por um dos alunos de Holland, David Goldberg, que popularizaram estes algoritmos, quando este aplicou esta técnica para resolver um problema complexo de controlo no transporte de gás natural em gasodutos (no âmbito da sua tese de doutoramento). O algoritmo genético original, proposto por Holland, designa-se por AG canónico ou simples (Goldberg (1989)).

Os AGs são diferentes dos mecanismos clássicos de optimização e de pesquisa, nos seguintes aspectos (Goldberg (1989)):

a) usam uma codificação do conjunto de parâmetros e não os próprios parâmetros;
b) pesquisam a partir de uma população de pontos e não de um único ponto;
c) utilizam a informação relativa vinda da função objectivo, e não de derivações ou de outro tipo de conhecimento auxiliar;
d) utilizam regras de transição probabilísticas, e não determinísticas.

É normal, na terminologia dos AGs, usar-se vários sinónimos para os mesmos elementos (Eiben e Smith (2003)). No contexto do problema original, pode-se usar qualquer um dos termos *solução*, *fenótipo* ou *indivíduo* para designar os pontos do espaço das soluções (também designado por *espaço do fenótipo*). No contexto dos AGs, pode-se usar qualquer um dos termos *genótipo*, *cromossoma* e também *indivíduo* para

[8] Charles Darwin (1809-1882) escreveu a obra "A Origem das Espécies", publicada em 1859.

designar os pontos do espaço onde se realiza a pesquisa evolucionária (também designado por *espaço do genótipo*). Também os elementos dos indivíduos (lugar e objecto) têm vários sinónimos: um *lugar* denomina-se normalmente por *variável*, *locus* (plural: loci), *posição*, ou *gene*; um *objecto* num lugar pode-se chamar de *valor* ou *alelo*. É vulgar usarem-se os termos solução e indivíduo (no espaço dos genótipos) como sinónimos, uma vez que um indivíduo é uma representação da solução.

Nesta dissertação os termos *solução* e *indivíduo* são usados como sinónimos, em que cada solução/indivíduo é composta por um conjunto de *variáveis* ou *genes*, cada uma com um certo *valor*.

Os AGs usam uma população de indivíduos (que representam as soluções do problema), sobre a qual aplicam mecanismos de selecção e de substituição, e operadores genéticos cruzamento (recombinação) e mutação, criando gerações sucessivas de indivíduos cada vez mais aptos. A pesquisa é orientada apenas pelo valor de aptidão de cada indivíduo da população: aqueles com maior valor de aptidão terão maiores probabilidades de serem seleccionados para reprodução (passando, assim, a sua informação genética para os descendentes nas gerações futuras), do que os com menor valor de aptidão (cuja informação genética tende a desaparecer nas gerações futuras). As características essenciais dos AGs são as seguintes:

- usam uma população de soluções do problema;
- cada solução tem associado um valor de aptidão que está ligado à sua capacidade de sobrevivência e de reprodução;
- as soluções com os melhores valores de aptidão têm mais oportunidades de se combinarem com outras soluções da população;
- na reprodução são geradas novas soluções que herdam algumas das características de cada um dos seus progenitores;
- a combinação de algumas boas características dos progenitores permite obter uma geração com melhores soluções.

A estrutura básica de um AG, considerando uma população principal de tamanho POP, pode ser descrita da seguinte forma:

1) t = 0;
2) Início: gerar a primeira população principal (inicial), P_t;
3) Avaliação: calcular o valor de aptidão de cada solução da população P_t;
4) Construir a nova população (associada à próxima geração), P_{t+1}:
    a) Seleccionar POP soluções (progenitores) da população P_t para reprodução (estas POP soluções constituem uma população provisória);

b) Aplicar o operador genético cruzamento às POP soluções seleccionadas (aplicado a pares de soluções), donde resultam POP novas soluções (descendentes);
c) Aplicar o operador genético mutação às POP novas soluções (descendentes);
d) Inserir estas POP novas soluções na nova população P_{t+1};

5) t = t + 1;
6) Se o critério de paragem é verificado, então terminar; senão regressar ao passo 3).

No desenvolvimento de um AG para resolver um certo problema real é normalmente necessário considerar a representação (codificação) das soluções e especificar a função de aptidão, já que estas são as únicas entidades dependentes do problema. Desta forma, é necessário considerar no processo evolucionário os seguintes aspectos: representação (codificação) das soluções, medida de aptidão, início do processo, população, mecanismos de selecção, operadores genéticos (cruzamento e mutação), mecanismos de substituição e critérios de paragem.

4.3.1. Representação (codificação) dos indivíduos

O primeiro passo na definição de um AG é estabelecer uma ponte entre o contexto do problema original e o espaço de resolução do problema onde ocorre o processo evolucionário. Este primeiro passo é geralmente designado por *representação*, que significa especificar uma correspondência entre um conjunto de soluções do problema original (espaço dos fenótipos) e um conjunto de indivíduos (no espaço dos genótipos) (Eiben e Smith (2003)). Ou seja, a *representação* consiste em estabelecer uma codificação adequada para as possíveis soluções do problema (*fenótipos*) ao nível do material genético de cada indivíduo (*genótipos*). A escolha de um tipo de representação que facilite o tratamento das soluções do problema por parte do AG requer, normalmente, um grande conhecimento das características do problema.

No entanto, o termo representação pode ser usado nos dois sentidos (Eiben e Smith (2003)): como codificação e como descodificação. A *codificação* está relacionada com a representação de um fenótipo no espaço dos genótipos (por exemplo, pode-se falar em representação ou codificação binária das soluções). A *descodificação* está relacionada com a correspondência inversa, isto é, do genótipo para o fenótipo, a qual deve exigir que para cada indivíduo exista no máximo uma solução correspondente.

A representação mais comum, e que coincide com a ideia original de Holland, é aquela que utiliza o alfabeto binário para codificar os valores possíveis de cada variável do indivíduo. No entanto, outras representações podem ser usadas, as quais devem ser apropriadas às características do problema em estudo, tais como: números reais, números inteiros e até mesmo caracteres. De facto, se a resolução de um problema

necessitar de valores com elevada precisão, então a utilização da codificação binária implica a utilização de um elevado número de *bits*, o que exige um maior esforço computacional para explorar o espaço das soluções, que pode ser enorme. Desta forma, uma representação mais natural para este tipo de problemas, é a utilização de valores reais para representar as soluções. Este tipo de representação é usado em trabalhos apresentados por Michalewicz (1996).

No entanto, o tipo de codificação mais utilizada é a binária, quer seja na sua versão tradicional, quer numa versão designada por código *gray*. O código *gray* é uma representação que se caracteriza pela propriedade de adjacência entre dois valores inteiros consecutivos; isto é, dois números inteiros consecutivos ou adjacentes têm codificações binárias que diferem apenas em um *bit* (ver Tabela 1). A vantagem do código *gray*, relativamente à codificação binária tradicional, está em que, devido à propriedade de adjacência, quando um dado indivíduo é alterado em apenas um dos genes (de 0 para 1 ou vice-versa), o indivíduo resultante não é muito diferente do anterior, o que traduz uma pequena perturbação na solução associada ao indivíduo.

Número inteiro	Tradicional	*Gray*
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Tabela 1: Comparação entre as codificações binárias tradicional e *gray*.

Os AGs usados na resolução de alguns problemas de optimização combinatória apresentam dificuldades em usar a representação binária, como é o caso do problema do caixeiro-viajante. Nestes problemas, as representações mais adequadas são aquelas que se baseiam em permutações, nas quais cada indivíduo é definido como uma sequência ordenada de todos os símbolos, sem repetições de símbolos.

4.3.2. Início do processo

A primeira etapa do processo evolucionário consiste em criar a primeira população (também denominada de população inicial), cujos indivíduos são, normalmente, gerados aleatoriamente.

No entanto, é importante que já exista nesta população um certo nível de diversidade entre os seus indivíduos, de forma a permitir uma exploração mais alargada no espaço de pesquisa. Para tal, pode-se usar um gerador de valores aleatórios de distribuição uniforme, ou então, utilizar algumas metodologias construídas para o efeito (ver Poles et al. (2006)).

A maioria dos algoritmos evolucionários multi-objectivo gera as suas populações iniciais de forma aleatória. No entanto, se o espaço de pesquisa contiver poucas soluções admissíveis, é de esperar que a população inicial, quando gerada desta forma, possa ser composta apenas por soluções não admissíveis. Por isso, a finalidade do processo evolucionário não é apenas convergir para a frente óptima de Pareto real, mas também conduzir a pesquisa para a região admissível. Desta forma, pode ser tão complicado encontrar a região admissível como melhorar uma solução admissível, no sentido de determinar a frente óptima de Pareto.

Podem também ser usadas heurísticas específicas do problema para se chegar a uma população inicial de indivíduos com elevada aptidão (Eiben e Smith (2003)). No entanto, apesar desta estratégia implicar um elevado esforço computacional, poderá trazer vantagens, se acelerar a convergência do algoritmo ou evitar que esta seja atingida prematuramente.

Affenzeller e Wagner (2004) demonstraram que a geração da população inicial é um aspecto muito importante na construção de um AG, uma vez que uma população inicial mal construída pode levar o AG a convergir prematuramente, ficando retido em óptimos locais. Isto acontece se a informação genética guardada nos indivíduos de uma população não contiver a informação necessária para melhorar a qualidade da solução. Uma das razões para que parte desta informação genética essencial (informação genética associada à solução óptima global) esteja perdida (o que leva à convergência prematura do AG), pode ser por esta informação não estar representada na população inicial do AG. É frequente acontecer, especialmente na fase inicial do processo de pesquisa, que a informação genética essencial esteja escondida em indivíduos com maus valores de aptidão, os quais são eliminados no processo de selecção.

Por outro lado, também Haubelt et al. (2005) e Hill e Hiremath (2005) provaram que a construção de uma população inicial bem distribuída acelera a convergência para a frente óptima de Pareto real, independentemente do AG e do problema a resolver. Este resultado é muito importante, tendo em conta que o processo de optimização pode consumir um elevado tempo de execução. Haubelt et al. (2005) propuseram a incorporação de um método para construção de populações iniciais em algoritmos evolucionários multi-objectivos já existentes, baseado na abordagem denominada de

"Pareto-Front-Arithmetics" (PFA), a qual permite uma rápida aproximação à frente óptima de Pareto real (Haubelt e Teick (2003)).

4.3.3. Função de avaliação (aptidão)

O papel da função de avaliação é representar as condições de adaptabilidade dos indivíduos da população, constituindo o suporte do mecanismo de selecção e facilitando, assim, o melhoramento da população (ou seja, definindo o que significa melhoramento). Tecnicamente, a função de avaliação é uma função ou procedimento que atribui uma medida de qualidade aos indivíduos (genótipos). Tipicamente, esta função é composta por uma medida de qualidade no espaço das soluções possíveis (dos fenótipos) e pela representação inversa (descodificação): por exemplo, o valor de aptidão do genótipo 10010 é f(18), em que f é a função a optimizar (Eiben e Smith (2003)).

A função da aptidão atribui um valor de aptidão a cada indivíduo (genótipo), o qual é o elemento fundamental para o processo de selecção. Para se calcular o valor de aptidão dos indivíduos, utiliza-se, normalmente, uma função de avaliação específica do problema em estudo. A função de avaliação devolve, para cada indivíduo, um valor numérico que reflecte o respectivo mérito. Note-se que um valor de aptidão pode estar associado a mais do que uma solução (fenótipo), e um indivíduo (genótipo) está associado a apenas uma solução (fenótipo) e um só valor de aptidão.

Normalmente, as noções de avaliação e de aptidão são iguais, mas nalguns casos isto não acontece. A função de avaliação fornece uma medida de desempenho relativamente a um conjunto particular de parâmetros, em que a avaliação de um indivíduo é independente das avaliações dos outros. A função de aptidão transforma aquela medida de desempenho na atribuição de oportunidades de reprodução, sendo, desta forma, dependente dos outros indivíduos da população.

Ao definir-se uma função de aptidão, é de extrema importância que esta seja adequada ao problema para uma execução correcta do AG, uma vez que a função de aptidão representa o ambiente do problema. O ideal seria a função de aptidão comportar-se de um modo suave e regular, no sentido de indivíduos semelhantes e com as mesmas características (que tenham em comum uma parte importante do material genético), terem valores de aptidão muito próximos.

4.3.4. População

Uma população é um conjunto de indivíduos (genótipos), com a particularidade de permitir a existência de elementos repetidos. A definição de uma população pode consistir apenas em especificar o número máximo de indivíduos que suporta, isto é, o seu tamanho.

O tamanho das populações é um aspecto muito importante na implementação de AG, uma vez que este valor vai afectar, quer a qualidade das soluções, quer o tempo de processamento do algoritmo. Populações pequenas têm como consequência uma fraca diversidade genética por parte dos seus elementos, o que leva a uma menor cobertura do espaço das soluções e, consequentemente, à convergência prematura (obtenção de soluções de menor qualidade). Grandes populações garantem uma maior diversidade de soluções na população, devido a uma maior cobertura do espaço das soluções, e permitem prevenir a convergência prematura, mas tal é conseguido à custa de um maior esforço computacional. Os AGs podem usar populações de quaisquer tamanhos, sendo normal este tamanho ser constante (não se alterando durante o processo evolucionário).

Um aspecto essencial para o bom funcionamento de um AG é a existência de uma boa *diversidade* entre os indivíduos da população. A *diversidade populacional* é uma medida associada à diferenciação das soluções no espaço das soluções e/ou no espaço dos objectivos. Se o nível de diversidade populacional for baixo, o que significa que os indivíduos são muito semelhantes, o operador genético cruzamento perde muito a capacidade de troca de informações úteis entre os indivíduos da população, o que faz com que a pesquisa possa progredir muito lentamente ou praticamente estacionar. Se o nível de diversidade populacional for alto, isto permitirá explorar melhor o espaço de pesquisa, o que é importante para escapar à optimalidade local.

A necessidade de controlar a diversidade populacional implica que as populações tenham tamanhos finitos e não muito grandes. Se isto não acontecer, o AG poderia tornar-se não aplicável na prática, visto que controlar uma população absurdamente grande poderia ocasionar um esforço computacional muito elevado.

Não existe uma medida única para a *diversidade populacional* pois, geralmente, pode referir-se à gama dos valores de aptidão, ao número de soluções (fenótipos) diferentes ou ao número de indivíduos (genótipos) diferentes (Eiben e Smith (2003)).

4.3.5. Mecanismos de selecção

Os mecanismos de selecção baseiam-se no princípio Darwiniano da "sobrevivência dos mais aptos": os indivíduos com melhores valores de aptidão têm maiores probabilidades de serem escolhidos para reprodução. Portanto, para além destes

mecanismos permitirem determinar que indivíduos (progenitores) são escolhidos para cruzamento, também permitem escolher que indivíduos (descendentes) devem sobreviver para a geração seguinte.

Existem basicamente dois tipos de mecanismos de selecção: baseados na proporção dos valores de aptidão e baseados na ordenação dos valores de aptidão. Na selecção baseada na proporção, os indivíduos são escolhidos de acordo com os seus valores de aptidão e da relação com os valores de aptidão dos outros indivíduos da população. Na selecção baseada na ordenação, os indivíduos são escolhidos de acordo com as respectivas posições na população ordenada e não de acordo com os seus valores de aptidão.

Alguns dos mecanismos de selecção baseados na proporção mais utilizados em problemas de optimização, são o método da roleta e a amostra universal estocástica.

O *método da roleta* (Goldberg (1989)) é um método estocástico que consiste em associar os indivíduos a porções contíguas de uma roleta, em que cada porção é proporcional à aptidão do indivíduo que lhe está associado (os indivíduos com maior valor de aptidão têm maiores probabilidades de serem escolhidos). São então realizados vários lançamentos da roleta, sendo seleccionados os indivíduos associados às porções atingidas por cada um destes lançamentos. Este mecanismo apresenta essencialmente duas desvantagens: redução da diversidade e convergência prematura (os mais aptos podem ser seleccionados muitas vezes), e estagnação da população (ao fim de algumas gerações, o valor médio da aptidão pode ser muito próximo dos melhores valores de aptidão).

No mecanismo denominado por a*mostra universal estocástica* (Baker (1987)), os indivíduos são associados a porções proporcionais aos seus valores de aptidão e contíguas de uma roleta, tal como no método da roleta. A diferença é que aqui a roleta tem N ponteiros equidistantes — N é o número de indivíduos a seleccionar — e gira apenas uma vez, sendo escolhidos os indivíduos marcados pelos N ponteiros (assim, os indivíduos menos aptos têm mais possibilidades de serem escolhidos do que no método anterior).

Entre os mecanismos de selecção baseados na ordenação mais usados, estão a ordenação linear, a selecção por truncatura e a selecção por torneio.

No mecanismo designado por *ordenação linear* (Baker (1985)), os indivíduos da população são ordenados de acordo com os seus valores da função de aptidão, sendo atribuído a cada indivíduo um valor que corresponde à sua posição na população ordenada. Desta forma, ao pior indivíduo é atribuído o valor 1 e ao melhor o valor POP (em que POP é o tamanho da população). Depois, a cada indivíduo é atribuída uma probabilidade de selecção calculada com base numa dada distribuição (as mais usuais

são a linear e a exponencial). Esta técnica trava a convergência prematura do algoritmo (não há favorecimento dos melhores indivíduos) e evita, em gerações avançadas, a estagnação da população.

No mecanismo de *selecção por truncatura*, os indivíduos são ordenados de acordo com os seus valores de aptidão e, de seguida, são seleccionados aqueles indivíduos cujos valores de aptidão sejam maiores do que um limiar predefinido (só são seleccionados os melhores indivíduos).

O mecanismo denominado por *selecção por torneio* (Goldberg (1989)) consiste em escolher aleatoriamente um certo número de indivíduos da população (designado por dimensão do torneio) e fazer um torneio entre eles. Cada torneio consiste em comparar os valores de aptidão dos indivíduos envolvidos, sendo o vencedor (e o seleccionado) aquele com melhor valor de aptidão. O número de torneios realizados é igual ao número de indivíduos a serem seleccionados, ou seja, igual ao tamanho da população. Esta técnica não conduz à convergência prematura (desde que a dimensão dos torneios seja pequena), combate a estagnação da população, é simples de implementar e não requer grande esforço computacional. Este é talvez o mecanismo de selecção mais utilizado na resolução de problemas de optimização.

Uma técnica que está intimamente ligada ao mecanismo de selecção é o *elitismo*. O *elitismo* foi introduzido por De Jong (1975) e consiste em reter na população os seus melhores indivíduos, os quais passam directamente para a próxima geração. Muitos investigadores têm encontrado no *elitismo* vantagens significativas para o desempenho dos AG (Mitchell (1997)). Com esta técnica, pretende-se, por um lado, garantir que os melhores indivíduos de cada geração não sejam destruídos pela acção dos operadores genéticos cruzamento e mutação (a definir mais à frente), e por outro, acelerar a convergência do algoritmo.

A desvantagem do *elitismo* está em quando os melhores indivíduos encontrados são os mesmos durante um certo número de gerações consecutivas, o que pode implicar a presença de várias cópias dos mesmos indivíduos em cada geração. Desta forma, pode haver perda de diversidade entre os indivíduos da população e forçar a pesquisa na direcção de um óptimo local. Assim sendo, e para combater este efeito negativo, o elitismo não deve ser utilizado de forma constante, mas sim esporadicamente ao longo do processo evolucionário.

O número de indivíduos que formam a elite constitui um dos parâmetros mais importantes dos AGs, pois se este valor não for adequado pode influenciar negativamente a convergência do AG. Em geral, este é um número baixo (no máximo 10% da população) e a sua amostragem pode ser directa (os E melhores) ou por sorteio (os E melhores entre os E' melhores da população).

4.3.6. Operador genético cruzamento

O operador genético *cruzamento* (ou recombinação) consiste em efectuar trocas de genes entre dois indivíduos. Neste processo são gerados dois novos indivíduos (descendentes), resultantes da combinação de informação contida num par de indivíduos (progenitores). O sucesso do AG está apoiado na expectativa de que o resultado do cruzamento entre indivíduos (progenitores) com melhores valores de aptidão gere novos indivíduos (descendentes) ainda de melhor qualidade (relativamente aos progenitores).

É importante que o operador cruzamento possibilite a propagação, ao longo das gerações, da informação genética dos melhores indivíduos, para conseguir explorar bem o espaço das soluções, assim como reunir a melhor informação genética num indivíduo que representará uma solução final do problema de optimização em questão.

Existem vários tipos de cruzamento, os quais dependem do tipo de representação usada na codificação dos indivíduos. Quando os indivíduos são representados por cadeias de dimensão fixa (por exemplo, alfabeto binário), os tipos de cruzamento usados com mais frequência são os que a seguir se descrevem (Baker (1985)).

O *cruzamento com um ponto de corte* consiste em determinar aleatoriamente uma posição do indivíduo (um ponto de corte), separando em dois, e no mesmo ponto, cada um dos dois indivíduos seleccionados. Desta forma, são formadas quatro sequências de genes dos indivíduos seleccionados, as quais aparecerão cruzadas nos descendentes, que recebem uma sequência de cada um dos progenitores. Se A = (A1, A2) e B = (B1, B2), então A + B = {(A1, B2), (B1, A2)}.

O *cruzamento com dois pontos de corte* consiste em determinar aleatoriamente duas posições do indivíduo (dois pontos de corte), separando em três, e nos mesmos pontos, cada um dos dois indivíduos seleccionados. Desta forma, são formadas seis sequências de genes dos indivíduos seleccionados, em que as sequências do meio aparecerão trocadas nos descendentes, que recebem as duas sequências dos extremos de um dos progenitores e a do meio do outro. Se A = (A1, A2, A3) e B = (B1, B2, B3), então A + B = {(A1, B2, A3), (B1, A2, B3)}.

O *cruzamento com vários pontos de corte* é semelhante ao anterior, mas com mais do que dois pontos de corte, em que cada um dos descendentes recebe de um dos progenitores as sequências de genes de índice ímpar e do outro as sequências de genes de índice par, e o outro descendente recebe as restantes — se A = (A1, A2, A3, A4, ...) e B = (B1, B2, B3, B4, ...), então A + B = {(A1, B2, A3, B4, ...), (B1, A2, B3, A4, ...)}.

O *cruzamento uniforme* consiste na utilização de uma máscara binária, gerada aleatoriamente e de comprimento igual ao dos indivíduos. Depois, um dos progenitores herda os genes de um dos pais para as posições da máscara com valor 0, e do outro pai para as posições da máscara com valor 1; o mesmo acontece com o outro descendente,

mas agora para os valores da máscara trocados. Este tipo de cruzamento pode não ser aplicável nalgumas representações, como por exemplo nas baseadas em permutações (usadas normalmente em problemas como o caixeiro-viajante). Nestes casos, é necessário definir outros tipos de cruzamento mais específicos e que se enquadrem no alfabeto escolhido.

O desempenho obtido por cada um dos tipos de cruzamento depende muito da especificidade dos problemas a resolver, uma vez que nem sempre é o mesmo tipo de cruzamento que permite o melhor desempenho de um algoritmo quando este é aplicado na resolução de diferentes problemas.

Um aspecto importante a ter em conta quando se utiliza este operador está na atribuição de uma *probabilidade de cruzamento*, a qual se define como a medida da possibilidade de aplicação do operador cruzamento a um dado par de indivíduos. Quanto maior for este valor, maior é a possibilidade de entrarem novos indivíduos na população e em maior número. Os valores mais usuais para a probabilidade de cruzamento variam entre 0.6 e 1.0.

4.3.7. Operador genético mutação

O operador genético mutação consiste em alterar aleatoriamente o valor de um dos genes de um indivíduo. A utilização deste operador genético nos AGs serve para, por um lado, fazer regressar à população os valores dos genes perdidos durante o processo de selecção, de modo a que possam ser testados num novo contexto, e por outro, proporcionar a entrada de novos genes que não estavam presentes nas populações anteriores.

Quando são utilizadas as representações com conjuntos de valores inteiros (por exemplo, o binário), o operador mutação consiste em determinar aleatoriamente uma posição do indivíduo, para depois substituir o valor que se encontra nessa posição por um dos outros valores do conjunto associado à representação usada (se existir mais do que um valor possível para o substituir, então este é escolhido aleatoriamente).

Quando as representações usadas são baseadas em permutações, uma forma simples para este operador consiste em trocar de posição dois elementos.

No entanto, outros tipos de mutação têm sido propostos, como a *mutação múltipla* que consiste em aplicar este operador genético a todos os elementos do indivíduo. Com este tipo de mutação, podem existir diferenças significativas entre o indivíduo antes e depois da aplicação da mutação, no espaço de pesquisa, o que é vantajoso no sentido de escapar à optimalidade local.

Um outro tipo de *mutação* é a *localizada*, onde só são afectados pela mutação os genes menos significativos do indivíduo, o que se traduz em pequenas perturbações nos indivíduos.

Um dos aspectos importantes a considerar quando se utiliza o operador genético mutação em AGs, está na sua taxa de ocorrência sobre o indivíduo (*probabilidade de mutação*). Alguns estudos indicam que uma elevada probabilidade tenderá a tornar o AG num algoritmo essencialmente aleatório. Desta forma, o valor da probabilidade de mutação deve ser baixo, o suficiente para diversificar os indivíduos da população e não prejudicar a convergência do algoritmo. Nalguns casos, a probabilidade de mutação pode variar de acordo com uma função matemática monotonamente decrescente, fazendo com que a mutação ocorra com maior frequência nas primeiras gerações (quando é necessário uma pesquisa mais alargada) e mais raramente nas últimas gerações (para possibilitar a convergência do algoritmo) (Reeves (1995) e Gomes et al. (2004)). De qualquer forma, a probabilidade de mutação deve ser diferente de zero, para se poder escapar à optimalidade local.

4.3.8. Mecanismos de substituição

Depois de terem sido gerados e avaliados (através da função de aptidão), os indivíduos devem ser introduzidos na população. Deverá ser definido quantos e quais os indivíduos descendentes que serão inseridos na população da nova geração, e que indivíduos da população actual devem ser substituídos. Para tal, define-se uma taxa de substituição, que indica a proporção de indivíduos da população a ser substituída em cada geração. Quanto menor for este valor, menor será a diferenciação genética entre gerações, o que implicará uma convergência mais lenta do algoritmo.

Existem vários mecanismos de substituição, sendo o mais comum aquele que consiste em gerar tantos descendentes como o número de indivíduos da população, para depois substituir todos estes indivíduos pelos descendentes gerados. No entanto, existem outras alternativas como, por exemplo, determinar apenas um ou dois indivíduos, determinar uma certa percentagem para reprodução, ou usar elitismo.

No primeiro caso, denominado de *AG de estado estacionário*, a etapa básica do processo evolucionário consiste em seleccionar apenas dois progenitores da população corrente para reprodução (cruzamento seguido de mutação), da qual resulta apenas um ou dois novos indivíduos (Whitley (1989)). Estes novos indivíduos gerados vão substituir outros indivíduos da população corrente, os quais são escolhidos de acordo com os seus valores de aptidão (escolhem-se os com piores valores de aptidão) ou aleatoriamente. No entanto, por vezes, a substituição só é efectuada caso os novos indivíduos tenham valores de aptidão melhores do que os indivíduos da população corrente.

O segundo caso é um mecanismo de evolução intermédio (está entre os dois mecanismos anteriores), que consiste em escolher uma certa percentagem da população corrente para reprodução (De Jong e Sarma (1992)).

No terceiro caso é feito uma recolha dos melhores indivíduos da população (elitismo — ver secção 4.3.5 deste capítulo), os quais passarão imediatamente, e sem qualquer tipo de competição, para a população da geração seguinte.

De notar que o primeiro mecanismo difere dos outros três essencialmente no facto de nestes três mecanismos existir a possibilidade de competição entre indivíduos de gerações diferentes, uma vez que numa geração existem sempre indivíduos de gerações anteriores (o que não acontece no primeiro mecanismo, em que todos os indivíduos da população corrente são substituídos por novos indivíduos).

4.3.9. Critérios de paragem

A conclusão do processo evolucionário associado a um AG depende, quer das características do problema em estudo, quer do tempo de execução do algoritmo (esforço computacional). Desta forma, pode-se definir a qualidade da solução procurada de acordo com o tempo e os recursos disponíveis, e o fim a que se destina aquela solução. O critério de paragem do processo evolucionário deve reflectir todos estes aspectos. Desta forma, os critérios mais usados para este fim são geralmente os seguintes (Eiben e Smith (2003)):

- definir um tempo de CPU máximo;
- definir o número máximo de gerações (muito frequente);
- definir um valor mínimo para o desvio padrão do valor de aptidão das soluções da população;
- não se verificarem melhorias significativas das soluções durante um número de gerações consecutivas (muito frequente);
- definir um limiar mínimo de diversidade da população;
- encontrar uma "boa" solução, caso seja possível avaliar a qualidade das soluções.

5. Algoritmos genéticos multi-objectivo

Os métodos de programação matemática multi-objectivo calculam, em geral, em cada iteração uma única solução não dominada, através da optimização de funções escalares substitutas. Estas funções agregam temporariamente numa única dimensão as múltiplas funções objectivo, de tal modo que a solução óptima de uma função escalar substituta é uma solução não dominada do problema multi-objectivo. Assim, se for necessário caracterizar com alguma exaustividade o conjunto das soluções não

dominadas, estas abordagens podem exigir um esforço computacional apreciável. As características das soluções calculadas, em particular a respectiva diversidade, estão bastante dependentes dos parâmetros de informação de preferências incluídas nas funções escalares substitutas (Clímaco et al. (2003)).

Ao trabalharem com populações de soluções, os AGs têm a potencialidade de determinar várias soluções não dominadas numa só execução do algoritmo. O processo de optimização multi-objectivo deve ter em conta os três aspectos seguintes:

- minimizar a distância da frente de Pareto obtida à frente óptima de Pareto real;
- as soluções obtidas devem ter uma boa distribuição (de preferência uniforme);
- maximizar a extensão da frente de Pareto obtida, isto é, as soluções não dominadas obtidas devem abranger a maior gama de valores possíveis para cada objectivo.

Atendendo a estes três aspectos, quando se implementa um AG em optimização multi-objectivo, devem ter-se sempre em atenção as duas questões seguintes:

1) como aperfeiçoar a função de aptidão e a selecção, de forma a conduzir a pesquisa em direcção à frente óptima de Pareto real;
2) como manter uma população diversificada de soluções, de forma a prevenir a convergência prematura e alcançar uma frente de Pareto extensa e bem distribuída.

5.1. Função de aptidão e mecanismo de selecção

Ao contrário da optimização mono-objectivo, em que a função de aptidão é muitas vezes a função objectivo, na optimização multi-objectivo a função de aptidão e o mecanismo de selecção devem usar os vários objectivos do problema. Normalmente, podem-se distinguir três tipos de AGs multi-objectivo, de acordo com o tipo de função de aptidão e o mecanismo de selecção usados (Zitzler (1999)):

- os objectivos são considerados em separado;
- baseados em técnicas clássicas de agregação;
- usam directamente o conceito de dominância de Pareto.

Na *selecção por alternância dos objectivos*, durante a fase de selecção escolhe-se alternadamente um dos objectivos para determinar os valores de aptidão dos indivíduos da população. Sempre que for necessário seleccionar um indivíduo da população para reprodução, será escolhido um dos objectivos do problema, que poderá ser diferente do usado anteriormente, o qual será usado para definir que indivíduo irá para uma população temporária (composta pelos indivíduos seleccionados para reprodução).

Na *selecção por agregação com variação dos parâmetros*, os vários objectivos são agregados numa única função objectivo parametrizada, em que para cada indivíduo da população é atribuído um conjunto diferente de parâmetros desta função. Algumas

abordagens usam o método da soma pesada para parametrizar os objectivos do problema. Uma vez que cada indivíduo é avaliado usando uma combinação particular de parâmetros, isto significa que cada indivíduo da população é avaliado por uma função objectivo diferente. Desta forma, a optimização é realizada em várias direcções, em simultâneo.

A *selecção baseada no conceito de dominância*, introduzida por Goldberg (1989), consiste em determinar vários níveis de frentes (de soluções) não dominadas (ver secção 2.4.4 deste capítulo). Esta ideia foi aproveitada por vários investigadores, resultando em várias abordagens evolucionárias onde os mecanismos para atribuir valores de aptidão às soluções são baseados na dominância de Pareto (por exemplo, Fonseca e Fleming (1993), Horn et al. (1994) e Srinivas e Deb (1994)). No entanto, Fonseca e Fleming (1995), fizeram notar que o desempenho das abordagens que utilizam este tipo de selecção pode ser influenciado pela dimensão do espaço de pesquisa. Mesmo assim, estas técnicas parecem ser as mais populares no campo da optimização evolucionária multi-objectivo (Veldhuizen e Lamont (1998)).

5.2. Diversidade da população

A preservação da diversidade dos indivíduos de uma população é crucial para a eficiência de qualquer abordagem evolucionária multi-objectivo. De forma a controlar a diversidade populacional, deve existir um controlo sobre o mecanismo de determinação das probabilidades de sobrevivência dos indivíduos mais aptos. Este mecanismo denomina-se por *pressão selectiva* e está associado à probabilidade dos melhores indivíduos serem seleccionados.

Uma elevada pressão selectiva significa que a pesquisa se concentra nos indivíduos mais aptos (porque têm uma elevada probabilidade de serem seleccionados), o que implica que a população tende a ficar rapidamente homogénea e com os seus indivíduos com elevada aptidão. Uma baixa pressão selectiva permite a exploração de uma maior diversidade de soluções, pois as probabilidades dos indivíduos da população serem seleccionados são muito próximas.

A aplicação ideal da pressão selectiva seria: menor pressão selectiva no início do processo (para favorecer a diversidade) e maior pressão selectiva na etapa final do processo (para favorecer a convergência para um óptimo global). Portanto, como o aumento da *pressão selectiva* diminui a *diversidade populacional*, é importante um equilíbrio entre estes dois factores, o que é conseguido com a aplicação dos operadores genéticos.

Para tratar esta questão da diversidade da população, foram desenvolvidos vários métodos, descrevendo-se seguidamente alguns dos mais utilizados em optimização evolucionária multi-objectivo.

5.2.1. Pré-selecção

Cavicchio (1970) foi o primeiro a introduzir um mecanismo explícito de preservação da diversidade num AG, que designou por *pré-selecção*. A ideia base da *pré-selecção* é substituir um indivíduo por um semelhante. Este mecanismo consiste no seguinte: quando é gerado um descendente, este é comparado com os seus progenitores; dos três indivíduos (descendente e progenitores) os dois com melhores valores de aptidão transitam para a próxima geração.

5.2.2. Técnica de multidões

A técnica de multidões desencoraja a aglomeração de soluções em qualquer zona do espaço de pesquisa, introduzindo, desta forma, diversidade entre as soluções de uma população. No AG apresentado por De Jong (1975), o modelo de multidões utiliza o conceito de população parcialmente semelhante e uma estratégia de multidões. Neste AG apenas é permitido que uma proporção LG (denominada de desnível de geração) da população seja reproduzida em cada geração. Além disso, antes de um descendente ser introduzido na população parcialmente semelhante, são escolhidas aleatoriamente FM soluções (designado por factor de multidão) desta população, as quais são comparadas com o descendente para que seja substituída a mais *semelhante* a ele (o autor usou LG = 0.1 e FM = 2/3). Como uma solução é substituída por um descendente semelhante, este processo preserva a diversidade da população.

5.2.3. Partilha do valor de aptidão

Goldberg e Richardson (1987) sugeriram a técnica de partilha do valor de aptidão, que consiste, basicamente, em degradar o valor de aptidão de indivíduos semelhantes (mais próximos). Esta técnica é a mais usada pelos AGs para a resolução de problemas de optimização multi-objectivo, e baseia-se na ideia de que os indivíduos que se concentram numa mesma zona (nicho) têm de partilhar os recursos disponíveis nessa zona. Quanto mais indivíduos estiverem localizados na vizinhança de um certo indivíduo, maior degradação sofrerá o valor de aptidão deste indivíduo. Esta vizinhança é definida em termos de uma medida de distância (no espaço das soluções ou dos objectivos) entre dois indivíduos x e y da população, $d(x,y)$, e especificada pelo *parâmetro de* partilha (*raio do nicho*), σ_{share}. O valor de aptidão de um indivíduo x é então dividido pelo número

de indivíduos que estão a uma distância de *x* inferior a σ_{share}, isto é, ao número de indivíduos que pertencem ao nicho de centro *x* e raio σ_{share}.

O processo para determinar o valor de aptidão de um indivíduo *x* numa população P, é o seguinte (Deb (2001)):

1. Calcular o *valor da função de partilha*, sh(d(*x*,y)), para todos os indivíduos da população, da seguinte forma:

$$Sh(d(x,y)) = \begin{cases} 1-\left(\frac{d(x,y)}{\sigma_{share}}\right)^{\alpha}, & \text{se } d(x,y) \le \sigma_{share} \\ 0, & \text{caso contrário} \end{cases},$$

   em que $\alpha > 0$ e assume tipicamente o valor 1 ou 2 (Deb (1989)).

2. Calcular o *contador de nichos*, nc(*x*), para cada indivíduo *x* da população, da seguinte forma:

$$nc(x) = \sum_{y \in P} Sh(d(x,y))$$

3. Calcular o *valor de aptidão partilhado*, da seguinte forma:

$$F'(x) = \frac{f(x)}{nc(x)}, \text{ em que } f(x) \text{ é o } \textit{valor de aptidão} \text{ do indivíduo } x.$$

Depois de calculado o valor de aptidão partilhado para todos os indivíduos da população, é aplicado, normalmente, um método de selecção proporcional ao valor de aptidão partilhado para seleccionar os indivíduos para competição.

O parâmetro de partilha, σ_{share}, corresponde à distância máxima entre duas soluções, para que estas sejam consideradas como pertencentes ao mesmo nicho. No contexto da optimização multi-objectivo, o valor para o parâmetro σ_{share} deve ser escolhido de forma a obter uma distribuição uniforme das soluções da frente óptima de Pareto.

Se os nichos forem considerados no espaço dos objectivos, então usa-se a distância Euclidiana entre duas soluções. Se a frente óptima de Pareto for conhecida, então pode-se partir do princípio que as POP (tamanho da população) soluções estão distribuídas uniformemente pela frente óptima de Pareto. Desta forma, conhecendo-se o perímetro L da frente óptima de Pareto, deve-se usar σ_{share} = L/POP. Assim, cada nicho conterá apenas uma solução, para uma distribuição quase-uniforme das soluções óptimas de Pareto. No entanto, normalmente, em particular em problemas reais, a frente óptima de Pareto real não é conhecida *a priori*.

Fonseca e Fleming (1993) sugeriram um procedimento para calcular o valor de σ_{share} em cada geração. Este procedimento consiste em calcular o hiper-volume limitado pelos valores mínimos (min_m) e máximos (max_m) de cada função objectivo $f_m(x)$ (m = 1,

..., M), onde se encontram todas as POP soluções da população: $V = \Pi_{m=1}^{M}\left(\max_m - \min_m\right)$. Para cada objectivo, o respectivo termo é acrescido em σ_{share} e calcula-se um novo hiper-volume: $V' = \Pi_{m=1}^{M}\left(\max_m - \min_m + \sigma_{share}\right)$. A diferença entre estes dois hiper-volumes é $\Delta V = V' - V$ e corresponde ao hiper-volume onde idealmente cada uma das POP soluções deveria estar. Como cada nicho ocupa um hiper-volume de $(\sigma_{share})^M$ e existem POP hiper-volumes destes em ΔV, então para calcular σ_{share} usa-se a seguinte equação:

$$\Delta V = \Pi_{m=1}^{M}\left(\max_m - \min_m + \sigma_{share}\right) - \Pi_{m=1}^{M}\left(\max_m - \min_m\right) = POP(\sigma_{share})^M \quad (1).$$

Para problemas com duas funções objectivo (M = 2), esta equação reduz-se à seguinte:

$$\sigma_{share} = \frac{\left(\max_2 - \min_2\right) + \left(\max_1 - \min_1\right)}{POP - 1}.$$

Se forem usados os valores normalizados das funções objectivo (isto é, valores entre 0 e 1), a equação (1) pode ser escrita de forma a normalizar o parâmetro de partilha σ'_{share} da seguinte forma:

$$\left(1 + \sigma'_{share}\right)^M - 1 = POP\left(\sigma'_{share}\right)^M, \text{ pois } \max_m - \min_m = 1, \text{ para } m = 1, ..., M.$$

Logo, para M = 2, esta última equação reduz-se a:

$$\sigma'_{share} = \frac{2}{POP - 1}.$$

5.2.4. Técnica de agrupamentos

Esta técnica de preservação da diversidade apenas é aplicada em abordagens que usam populações externas/secundárias, pois só são aplicadas quando o número de soluções candidatas a pertencerem à população externa é superior ao seu tamanho máximo. Esta técnica consiste em determinar os agrupamentos que melhor se distribuem pela frente não dominada, em número igual ao tamanho máximo da população externa.

Este processo começa por considerar tantos agrupamentos como o número total de soluções candidatas, para de seguida calcular as distâncias entre todos os pares de agrupamentos; por exemplo, pode-se considerar a distância d_{ij} entre dois agrupamentos C_i e C_j como a distância Euclidiana média de todos os pares de soluções ($x \in C_i$, $y \in C_j$):

$$d_{ij} = \frac{1}{|C_i|\,|C_j|} \sum_{x \in C_i, y \in C_j} d(x, y).$$

A distância entre *x* e *y*, d(x,y), pode ser calculada nos espaços das soluções ou dos objectivos. Depois de calcular as distâncias entre todos os pares de agrupamentos, aqueles que estão associados à menor distância obtida são agrupados, formando um só

agrupamento (o número de agrupamentos é decrementado em uma unidade). Este processo (calcular as distâncias entre todos os pares de agrupamentos e juntar o par que está à menor distância) repete-se, até ficarem apenas tantos agrupamentos como o tamanho máximo da população externa, os quais são disjuntos entre si. Por fim, para cada agrupamento determina-se a solução que detém a menor distância média para as restantes soluções do agrupamento, a qual será a representante do agrupamento.

5.3. Principais abordagens com Algoritmos Genéticos

A primeira verdadeira implementação de um algoritmo evolucionário multi-objectivo foi sugerido por Schaffer (1984) com o AG designado por VEGA.

Mais tarde, Goldberg (1989) apresentou um procedimento de classificação das soluções por níveis de dominância. A sugestão de Goldberg foi no sentido de usar o conceito de dominância para inserir mais cópias de soluções não dominadas numa população. Para manter a diversidade, sugeriu a utilização de uma estratégia de nichos entre as soluções da mesma classe de não dominância. A partir das sugestões de Goldberg foram desenvolvidos, alguns anos depois, alguns algoritmos evolucionários multi-objectivo, dos quais se destacam os seguintes: WBGA (Hajela e Lin (1992)), MOGA (Fonseca e Fleming (1993)), NSGA (Srinivas e Deb (1994)), NPGA (Horn et al. (1994)) e NPGA 2 (Erickson et al. (2001)).

No entanto, e apesar do sucesso da aplicação das abordagens referidas, as quais não incorporam o elitismo nos seus AG, a diversos problemas de optimização multi-objectivo, alguns autores, como Rudolph (1999) e Zitzler et al. (2000), sugeriram que o elitismo pode melhorar o desempenho dos AG e, até mesmo, impedir que algumas soluções não dominadas obtidas durante o processo de pesquisa se percam. Algumas das abordagens elitistas em optimização multi-objectivo são: SPEA (Zitzler e Thiele (1998)), NSGA-II (Deb et al. (2000)) e SPEA-2 (Zitzler et al. (2001)).

Existem na literatura várias publicações que apresentam descrições de algumas das várias abordagens existentes para optimização evolucionária multi-objectivo, incluindo discussões detalhadas dos seus pontos fortes e fracos. Alguns destes artigos ou livros são: Fonseca e Fleming (1995), Veldhuizen e Lamont (1998), Veldhuizen e Lamont (2000), Fonseca e Fleming (1998-a), Fonseca e Fleming (1998-b), Deb (1999), Veldhuizen (1999), Coello (2000), Deb (2001) e Kicinger et al. (2005).

Existem também na literatura algumas publicações que, para além de descreverem alguns dos métodos existentes nesta área, apresentam também alguns dos problemas a que estas abordagens foram aplicadas: Coello (1999) e Coello et al. (2002).

Para além disto, existe também um sítio na internet que contém uma lista com muitas publicações nesta área: http://www.lania.mx/~ccoello/EMOO/EMOObib.html.

5.3.1. Abordagens Não Elitistas

Apesar de até à data terem sido sugeridas várias abordagens não elitistas em optimização evolucionária multi-objectivo, neste texto apenas serão descritas as seguintes: VEGA, WBGA, MOGA, NSGA e NPGA.

5.3.1.1. Vector Evaluated Genetic Algorithm (VEGA)

O primeiro AG para determinar um conjunto de soluções não dominadas de um problema de optimização multi-objectivo foi implementado por Schaffer (1984), ao qual deu o nome de "Vector Evaluated Genetic Algorithm" (VEGA). Posteriormente, Schaffer (1985) comparou este AG com uma técnica de pesquisa adaptativa aleatória, tendo observado um melhor desempenho por parte do VEGA.

Para tratar os M objectivos do problema, em cada geração a população é dividida, aleatoriamente, em M sub-populações de tamanhos iguais, em que a cada uma destas sub-populações é associado um único objectivo.

Depois, às soluções de cada uma das sub-populações é atribuído um valor de aptidão, de acordo com a correspondente função objectivo (*selecção por alternância dos objectivos* — ver secção 5.1 deste capítulo). É então aplicado um mecanismo de selecção (proporcional à aptidão), mas apenas entre soluções da mesma sub-população.

Por fim, são aplicados os operadores genéticos cruzamento e mutação a toda a população, para gerar as soluções da nova população. Neste algoritmo, não é usada qualquer técnica específica para controlar a diversidade da população.

Esta abordagem é particularmente útil no tratamento de problemas em que as funções objectivo tomam valores de diferentes ordens de magnitude. Uma vez que todas as soluções de uma sub-população têm associado um valor de aptidão baseado numa função objectivo particular, a restrição do mecanismo de selecção apenas às soluções desta população leva a que sejam realçadas as melhores soluções para cada função objectivo. Apesar de serem reconhecidas algumas falhas importantes (Schaffer (1985), Fonseca e Fleming (1995)), esta abordagem é um ponto de referência nesta área.

5.3.1.2. Weight-Based Genetic Algorithm (WBGA)

Hajela e Lin (1992) introduziram uma abordagem, que designaram por "Weight-Based Genetic Algorithm" (WBGA), em que cada função objectivo $f_m(x)$ é multiplicada por um peso w_m (com m = 1, ..., M), isto é as funções objectivo são ponderadas.

Para determinar os valores de aptidão dos indivíduos da população, esta abordagem baseia-se na *selecção por agregação com variação dos parâmetros* (ver secção 5.1 deste capítulo). Para tal, é usado o método da soma pesada, em que cada indivíduo da

população tem associado uma combinação de pesos normalmente diferente. Desta forma, em vez de se determinar uma única solução não dominada correspondente a uma combinação de pesos específica, a população mantém em simultâneo várias combinações de pesos, determinando, assim, várias soluções não dominadas em cada iteração do AG.

A preservação da diversidade das combinações de pesos é conseguida através de duas técnicas: usando um método baseado em nichos apenas sobre cada combinação de pesos (através da técnica de partilha do valor de aptidão — secção 5.2.3 deste capítulo — aplicada no espaço dos objectivos), ou avaliando sub-populações escolhidas adequadamente e associadas a diferentes combinações de pesos predefinidas (semelhante ao VEGA) (Deb (2001)).

O valor de aptidão de uma solução x, $F(x)$, é igual à soma pesada dos valores normalizados das funções objectivo, de acordo com a equação seguinte:

$$F(x) = \sum_{m=1}^{M} w_m \frac{f_m(x) - f_m^{min}}{f_m^{max} - f_m^{min}},$$

em que $w = (w_1, ..., w_M)$ é a combinação de pesos normalizados a partir da qual se obteve a solução x, f_m^{max} e f_m^{min} são os valores máximo e mínimo de $f_m(x)$, respectivamente.

Na técnica de partilha do valor de aptidão, para calcular o valor de aptidão partilhado, $F'(x)$, é necessário calcular as distâncias entre todos os pares de soluções da população, $d(x,y)$, o que é feito usando uma seriação das combinações de pesos associadas às soluções. Por exemplo, se forem usadas CP combinações de pesos a partir das quais se obtiveram POP soluções, a cada uma destas CP combinações é associado um valor inteiro entre 1 e CP. Desta forma, a distância entre duas soluções x e y é igual ao valor absoluto da diferença dos inteiros associados às combinações de pesos de que resultaram as soluções x e y (para mais detalhes, ver Deb (2001)). Uma vez que o valor de aptidão é degradado pela aplicação da técnica de partilha do valor de aptidão, é necessário usar-se o mecanismo de selecção proporcional ao valor de aptidão partilhado. Os operadores genéticos (cruzamento e mutação) são aplicados normalmente.

Na técnica semelhante ao VEGA, o AG começa por escolher um conjunto de CP combinações diferentes de pesos. A seguir, cada uma destas combinações de pesos é usada para determinar a aptidão pesada normalizada de todos os POP indivíduos da população, dos quais se escolhem, de acordo com estes valores de aptidão, os POP/CP melhores indivíduos para formarem uma sub-população. Portanto, são constituídas CP sub-populações, cada uma contendo os respectivos melhores POP/CP indivíduos de acordo com os valores da função de aptidão pesada normalizada associada a cada uma

das combinações de pesos. O mecanismo de selecção e os operadores genéticos (cruzamento e mutação) são restringidos às soluções de cada sub-população.

5.3.1.3. Multiple Objective Genetic Algorithm (MOGA)

Em 1993 surgiu um outro algoritmo, denominado "Multiple Objective Genetic Algorithm" (MOGA), da autoria de Fonseca e Fleming (1993). Estes autores foram os primeiros a sugerir um AG multi-objectivo que realça explicitamente a importância das soluções não dominadas, ao mesmo tempo que preserva a diversidade das soluções na população.

Para determinar os valores de aptidão das soluções da população de tamanho POP, este algoritmo utiliza o esquema de classificação das soluções da população do AG por níveis de dominância (ver secção 5.1 deste capítulo): a classificação de um certo indivíduo da população é proporcional ao número de indivíduos da mesma população que o dominam. Esta técnica começa por determinar, para cada solução x, o número de soluções que a domina, nd(x), obtendo-se a sua classificação de acordo com o seguinte valor: r(x) = 1 + nd(x). Desta forma, as soluções classificadas com valor 1, são as soluções não dominadas da população.

As soluções são então ordenadas de acordo com as respectivas classificações, das melhores (com r = 1) para as piores (com r ≤ POP). De seguida, a cada solução é atribuído um valor de *aptidão bruta*, através de uma função linear (ou outra adequada), de forma a atribuir às soluções valores entre POP (para as melhor classificadas) e 1 (para as pior classificadas). Determina-se então a média dos valores de *aptidão bruta* entre as soluções com a mesma classificação, a qual é atribuída como *valor de aptidão* a cada solução com a mesma classificação.

Finalmente, é aplicada a técnica de partilha do valor de aptidão (secção 5.2.3 deste capítulo), para determinar o valor de aptidão partilhado das soluções da população. Desta forma, embora as soluções com a mesma classificação tenham o mesmo valor de aptidão, as soluções que pertencem a um nicho com menos indivíduos têm melhor valor de aptidão partilhado.

Para garantirem a diversidade entre as soluções não dominadas da população, os autores introduziram um mecanismo baseado num esquema de nichos entre soluções com a mesma classificação. Para tal, usaram a técnica de partilha do valor de aptidão (ver secção 5.2.3 deste capítulo), com as seguintes adaptações:

- a distância entre soluções é medida no espaço dos objectivos e é normalizada;
- para cada solução x, d(x, y) é calculada para todas as soluções y (incluindo x) com a mesma classificação;

- para calcular o valor da função de partilha de uma solução *x*, é usada a equação para calcular Sh(*x*), com $\alpha = 1$ (ver secção 5.2.3 deste capítulo);
- o contador de nicho, nc(*x*), é calculado somando os valores da função de partilha, envolvendo todas as soluções com a mesma classificação de *x*.

Finalmente, são aplicados o mecanismo de selecção universal estocástica (de acordo com os valores de aptidão partilhado) e os operadores genéticos cruzamento com um ponto de corte e mutação binária simples, de forma a gerar uma nova população.

5.3.1.4. Non-dominated Sorting Genetic Algorithm (NSGA)

A ideia de Goldberg em usar o conceito de ordenação por não dominância (Goldberg (1989)), foi implementada na totalidade por Srinivas e Deb (1994), na abordagem denominada por "Non-dominated Sorting Genetic Algorithm" (NSGA). Esta abordagem utiliza a selecção baseada no conceito de dominância para classificar as soluções da população por níveis de dominância (ver secção 5.1 deste capítulo) e uma estratégia de partilha dos valores de aptidão para preservar a diversidade entre as soluções de cada frente não dominada.

O primeiro passo deste algoritmo consiste em aplicar o conceito de dominância de Pareto às soluções de uma população P, classificando-as em várias frentes de soluções não dominadas (ver secção 2.4.4 deste capítulo).

O processo para atribuir valores de aptidão às soluções da população P começa pela primeira frente não dominada (a que contém as melhores soluções, isto é, as não dominadas), até à última (que contém as piores soluções). A cada solução da primeira frente não dominada é atribuído um valor de aptidão igual ao tamanho da população, uma vez que todas estas soluções têm o mesmo nível de importância, em termos de proximidade à frente óptima de Pareto.

Para manter a diversidade da população, os valores de aptidão das soluções são degradados, de acordo com os respectivos números de soluções vizinhas. Para tal, é usado o método de partilha do valor de aptidão (ver secção 5.2.3 deste capítulo), que começa por calcular, para cada solução *x* de uma frente P_1, a distância Euclidiana normalizada para uma outra solução desta frente, no espaço das soluções. Depois, usando esta distância, calcula-se a função de partilha, Sh, com $\alpha = 2$. Qualquer solução da população que esteja a uma distância superior a σ_{share} (que é um valor fixo) de *x* não contribui para o valor da função de partilha desta solução. Depois de estarem calculados os valores de aptidão partilhados de todas as soluções da frente P_1, calculam-se os contadores de nicho de cada solução nc(*x*) da frente P_1 (nc(*x*) traduz o número de

soluções vizinhas de *x*, incluindo ela própria). Por fim, para se obter o valor de aptidão partilhado da solução *x*, divide-se o seu valor de aptidão pelo seu contador de nicho.

Depois de calculados os valores de aptidão partilhados das soluções da primeira frente, determinam-se estes valores para as soluções da segunda frente, começando-se por atribuir a cada uma destas soluções um valor de aptidão partilhado ligeiramente inferior ao valor de aptidão partilhado mínimo associado a uma solução da primeira frente. Desta forma, qualquer solução da primeira frente tem atribuído um valor de aptidão partilhado superior a qualquer solução da segunda frente. Depois, o processo continua até estar atribuído um valor de aptidão partilhado a todas as soluções desta frente. Finalmente, e depois de estar atribuído um valor de aptidão partilhado às soluções da segunda frente, o processo repete-se para as restantes frentes não dominadas.

Uma vez que o método de partilha do valor de aptidão é usado com um operador de selecção proporcional, o NSGA usa como mecanismo de selecção o método da roleta (Goldberg (1989)), que divide a roleta em tantas porções como o número de soluções da população e em que cada porção é proporcional ao valor de aptidão partilhado da respectiva solução (ver secção 4.3.5 deste capítulo). Os operadores genéticos cruzamento e mutação são aplicados a toda a população normalmente.

5.3.1.5. Niched Pareto Genetic Algorithm (NPGA)

Horn et al. (1994) propuseram uma abordagem, designada por "Niched Pareto Genetic Algorithm" (NPGA), a qual difere das abordagens anteriores, essencialmente, no tipo de mecanismo de selecção que usa.

Esta abordagem combina a técnica de partilha do valor de aptidão com um mecanismo de selecção baseado na ordenação dos valores de aptidão, a selecção por torneio (ao contrário das abordagens VEGA, NSGA e MOGA, que aplicam a selecção proporcional à aptidão). Este algoritmo começa por seleccionar aleatoriamente duas soluções *x* e *y* da população P de tamanho POP, as quais são comparadas com uma sub-população P_{dom} (de dominação), de tamanho POP_{dom} (<< POP), escolhida aleatoriamente da população P (geralmente $POP_{dom} = 0.1 \times POP$). Se uma destas soluções domina todas as soluções de P_{dom} e a outra solução é dominada por pelo menos uma solução de P_{dom}, então a vencedora do torneio é a primeira. Se esta situação não acontecer, então as soluções *x* e *y* são testadas na sub-população (incompleta) de descendentes já gerados Q e se $Q \neq \emptyset$ (se $Q = \emptyset$, então escolhe-se uma delas aleatoriamente): calculam-se os contadores de nicho das soluções *x* e *y* considerando apenas a sub-população de descendentes Q, em que o contador de nicho de *x*, nc(*x*), é calculado como o número de descendentes ($z \in Q$) que estão a uma distância d(*x*,*z*) de *x* inferior a σ_{share}, em que d(*x*,*z*) é a distância Euclidiana entre as soluções *x* e *z* no espaço

dos objectivos. Das soluções *x* e *y* a que tiver menor contador de nicho (com menos vizinhos) é a vencedora do torneio.

5.3.2. Abordagens Elitistas

A utilização de elitismo em AG levanta algumas questões importantes, como sejam (Zitzler (1999)):

- Que indivíduos guardar na elite e durante quanto tempo?
- Quando e como os indivíduos guardados devem ser reinseridos na população?

Uma forma de implementar o elitismo consiste em utilizar uma população externa (também denominada por secundária), na qual são mantidas as soluções não dominadas obtidas durante o processo de pesquisa. Neste caso, é necessário definir o mecanismo de actualização (de forma a garantir a diversidade das soluções na população), o tamanho (número máximo de soluções que pode guardar) e a estrutura da população externa, assim como o modo como as soluções são reinseridas na população.

Este tipo de abordagem (elitista com população externa) poderá ser vantajosa em problemas com muitas soluções não dominadas. Também é possível parametrizar o elitismo, fazendo variar, por exemplo, o mecanismo de actualização da população externa e o nível de elitismo. No entanto, a actualização da população externa exige um esforço computacional extra.

Apesar de existirem várias abordagens elitistas em optimização evolucionária multi-objectivo, neste texto apenas serão descritas as três abordagens mais referidas na literatura: SPEA, NSGA-II e SPEA-2.

5.3.2.1. Strength Pareto Evolutionary Algorithm

Zitzler e Thiele (1998) propuseram um AG elitista, que designaram por "Strength Pareto Evolutionary Algorithm" (SPEA), em que o elitismo é tratado através de uma população externa (ver também Zitzler e Thiele (1999)). Esta população guarda um número fixo de soluções não dominadas que vão sendo obtidas ao longo das gerações (iterações) do AG. Em cada geração, as novas soluções não dominadas obtidas são comparadas com as da população externa, sendo guardadas as soluções não dominadas resultantes. Este AG não se preocupa apenas em preservar as elites, pois estas também participam no mecanismo de selecção e na aplicação dos operadores genéticos, conjuntamente com a população corrente, na esperança de influenciar a condução da população na direcção das regiões mais interessantes do espaço de pesquisa.

O SPEA é semelhante a outras abordagens em optimização evolucionária multi-objectivo, nos seguintes aspectos (Zitzler (1999)):

- guarda algumas soluções externamente, as quais representam uma frente não dominada entre todas as soluções consideradas até então;
- usa o conceito de dominância no cálculo dos valores de aptidão das soluções;
- determina agrupamentos de soluções não dominadas para reduzir o número destas soluções guardadas na população externa, sem destruir as características da frente óptima de Pareto;

e é diferente nos seguintes aspectos:

- combina os três aspectos mencionados antes num único algoritmo;
- os valores de aptidão das soluções das populações externa e corrente são determinados usando apenas as soluções da população externa;
- todas as soluções da população externa participam no mecanismo de selecção;
- introduz um novo método baseado em nichos, de forma a preservar a diversidade da população.

O SPEA começa por gerar aleatoriamente uma população P_0 de tamanho POP e uma população externa $PE_0 = \varnothing$ de tamanho máximo POP_E. Em cada geração t, as soluções não dominadas da população P_t (nova elite) são copiadas para a nova população externa PE_t, que é actualizada. Esta actualização consiste em remover as soluções dominadas da população externa PE_t.

No entanto, continuando este processo ao longo das gerações, pode-se chegar a uma situação em que o número de soluções da população externa, depois de actualizada, é superior a seu tamanho máximo, POP_E. Desta forma, é necessário aplicar um mecanismo para escolher as "melhores" POP_E soluções da população externa PE_t para constituírem uma população externa apenas com POP_E soluções (as "melhores" soluções são as que estão menos agrupadas). Para determinar as POP_E "melhores" soluções, o SPEA utiliza um método de agrupamentos (ver secção 5.2.4 deste capítulo), que consiste em determinar os POP_E agrupamentos que melhor se distribuem pela frente não dominada. Por fim, a população externa da próxima geração, PE_{t+1}, é construída à custa das soluções que se encontram em cada um dos POP_E agrupamentos, em que cada agrupamento contribui com apenas uma solução.

Depois de obtida a nova população externa (a da próxima geração), a fase seguinte consiste em avaliar as soluções desta população e também da população corrente, o que é feito de modos diferentes. O valor de aptidão de uma solução x da população externa, $F(x)$, é um valor proporcional ao número $n(x)$ de soluções da população corrente que são

dominadas pela solução x: $F(x) = n(x) / (POP+1)$ (divide-se por POP+1 para garantir que $F(x) < 1$); desta forma, as soluções que dominam mais soluções têm maior valor de $F(x)$. O valor de aptidão de uma solução y da população corrente é obtido através da soma dos valores de aptidão das soluções da população externa que dominam a solução y: $F(y) = 1 + \sum_{x \in PE_t} F(x)$, em que x domina y ($F(y) > 1$); desta forma, os valores de aptidão das soluções da população corrente são sempre superiores aos valores de aptidão da soluções da população externa. Com este método de cálculo dos valores de aptidão, as soluções com menor valor de aptidão são as "melhores".

Por fim, aplica-se o mecanismo de selecção por torneio usando os valores de aptidão determinados antes e os operadores genéticos cruzamento e mutação às soluções da população resultante da combinação das duas populações, $PE_t \cup P_t$, de tamanho $POP + POP_E$, para criar a população associada à geração seguinte, P_{t+1}, de tamanho POP.

5.3.2.2. Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II)

Deb et al. (2000) sugeriram a abordagem denominada por "Elitist Non-dominated Sorting Genetic Algorithm" (NSGA-II), a qual usa explicitamente um mecanismo de preservação e diversificação das soluções (ver também Deb et al. (2002)).

O NSGA-II começa por gerar, aleatoriamente, uma população inicial P_0 de tamanho POP, a qual é depois classificada em diferentes níveis de dominância, donde resulta a identificação de várias frentes não dominadas. Desta forma, a cada solução é atribuído um valor de aptidão igual ao seu nível de dominância (1 para o melhor nível — ver secção 5.1 deste capítulo). A seguir, aplica-se um mecanismo de selecção por torneio baseado na técnica de multidões (a descrever à frente) de forma a preservar a diversidade da população e os operadores genéticos cruzamento e mutação, para se construir uma população de descendentes, Q_0, de tamanho POP.

É depois aplicado um processo comum a cada geração t, que começa por combinar as duas populações, de progenitores e de descendentes, $R_t = P_t \cup Q_t$, de tamanho $2 \times POP$. As soluções de R_t são então classificados por níveis de dominância, donde resulta a identificação de diferentes frentes não dominadas.

A nova população, P_{t+1}, é então construída a partir das várias frentes não dominadas obtidas (com $2 \times POP$ soluções), adicionando a primeira frente, depois a segunda, e assim sucessivamente, até que esta nova população, de tamanho POP, fique completa. Como o tamanho máximo de R_t é $2 \times POP$ e o tamanho da nova população é apenas POP, nem todas as frentes podem ser adicionadas a esta população, o que implica que as "piores" frentes sejam simplesmente ignoradas. Por outro lado, ao considerar-se a

última frente a adicionar, o número de soluções desta frente pode ser maior do que o número de soluções necessárias para completar a nova população, o que significa que nem todas as soluções desta frente podem ser adicionadas à nova população. Desta forma, para completar a nova população, o NSGA-II usa uma estratégia baseada em nichos para escolher as soluções da última frente que se localizam em regiões de menor densidade populacional nesta frente (a descrever à frente).

Por fim, é então construída a população de descendentes, Q_{t+1}, de tamanho POP, a partir da população P_{t+1}, usando selecção por torneio baseada na técnica de multidões e os operadores genéticos cruzamento e mutação.

O mecanismo de selecção por torneio baseado na técnica de multidões consiste em comparar duas soluções e declarar como vencedora aquela com melhor classificação (em termos de ordenação das frentes não dominadas — ver secção 2.4.4 deste capítulo) ou, tendo a mesma classificação, a que tiver maior distância de multidões (aquela que se localiza na área de menor densidade).

A distância de multidões de uma solução *x* pode ser calculada de várias formas, como, por exemplo, o contador de nicho (ver secção 5.2.3 deste capítulo) ou o perímetro do cubóide cujos vértices são formados pelas soluções mais próximas da solução *x* para todas as funções objectivo.

5.3.2.3. Strength Pareto Evolutionary Algorithm 2

Zitzler et al. (2001) apresentaram uma versão melhorada da abordagem "Strength Pareto Evolutionary Algorithm", a que chamaram SPEA2. As principais diferenças entre esta nova versão e a versão anterior são as seguintes:

- o mecanismo para determinar o valor de aptidão de uma solução usa uma estratégia baseada no número de soluções que a dominam e no número de soluções que são dominadas por ela;
- aquele mecanismo incorpora uma técnica adicional para estimação da densidade para discriminar as soluções com características idênticas em termos de dominância;
- o tamanho da população externa é constante, o que significa que esta população pode conter soluções dominadas;
- usa um método de truncatura aplicado à população externa para preservar as soluções extremas desta população (as soluções com os melhores valores para cada uma das funções objectivo);
- apenas as soluções da população externa participam no processo de selecção.

Os valores de aptidão das soluções das populações corrente P e externa PE são calculados da mesma forma. Na geração t, a cada solução *x* de qualquer das populações é atribuído um valor $F(x) = n(x)$, em que $n(x)$ é o número de soluções que são

dominadas por x. É então calculado o valor de aptidão bruta de x, FB(x), como a soma dos valores de n(y) de todas as soluções y das populações P_t e PE_t que dominam x ($F(x) = \sum_{y \in P_t \cup PE_t} F(y)$, em que y domina x). Desta forma, para uma solução x, FB(x) = 0 significa que x é uma solução não dominada, e FB(x) com valores elevados significa que x é uma solução dominada por muitas soluções. Este mecanismo incorpora ainda uma técnica de estimação da densidade das soluções baseada na distância para o vizinho mais próximo, de forma a discriminar as soluções com o mesmo valor de aptidão bruta. Esta técnica é uma adaptação do método do k-ésimo vizinho mais próximo (Silverman (1986)), e consiste em tomar o inverso da distância, no espaço dos objectivos, para o k-ésimo vizinho mais próximo como estimação da densidade; a densidade de uma solução x, é $D(x) = 1 / (\sigma^k(x)+2)$, em que $\sigma^k(x)$ é a distância da solução x para o k-ésimo vizinho mais próximo (segundo os autores, um valor comum é $k = \sqrt{POP + POP_E}$, em que POP e POP_E são os tamanhos das populações P e PE, respectivamente). Finalmente, o valor de aptidão da solução x é $F(x) = FB(x) + D(x)$.

O tamanho da população externa é constante, o que significa que esta deve estar sempre completa, nem que seja com soluções dominadas. Quando o número de soluções não dominadas for superior ao tamanho da população externa, aplica-se um método de truncatura, que é semelhante ao método baseado em nichos usado no SPEA, para evitar que sejam removidas as soluções não dominadas extremas.

Este método começa por inserir todas as soluções não dominadas x (com $F(x) < 1$) das populações corrente P_t e externa PE_t, na população externa da próxima geração, PE_{t+1}. Se PE_{t+1} ficar completa, então termina este processo; caso contrário, aquela população está incompleta ($|PE_{t+1}| < POP_E$) ou está sobrelotada ($|PE_{t+1}| > POP_E$).

No primeiro caso, a população fica completa com as melhores soluções dominadas y ($F(y) \geq 1$), de acordo com os seus valores de aptidão, das anteriores populações externa PE_t e corrente P_t. No segundo caso, é utilizado um procedimento para truncar a população externa, que consiste em remover soluções da população externa sobrelotada até que esta fique com o tamanho adequado ($|PE_{t+1}| = POP_E$). Em cada iteração deste processo, a solução a ser removida é aquela que detém a distância mínima para outra solução; se existir mais do que uma solução nestas circunstâncias, entre estas escolhe-se aquela que detém a segunda menor distância, e assim sucessivamente.

O AG associado ao SPEA2 começa por gerar uma população inicial P_0 e criar uma população externa $PE_0 = \varnothing$, na geração t = 0. A seguir, determinam-se os valores de aptidão das soluções das duas populações, utilizando o processo já descrito nesta secção.

O passo seguinte consiste em construir a população secundária da próxima geração, PE_{t+1}, usando o procedimento já descrito também nesta secção.

A etapa seguinte consiste em verificar se a condição de paragem é satisfeita. Se sim, então todas as soluções não dominadas da população externa, PE_{t+1} (esta população pode conter também soluções dominadas), são inseridas num conjunto A e o algoritmo termina. Caso contrário, aplica-se um mecanismo de selecção por torneio com substituição sobre a população PE_{t+1}, e de seguida aplicam-se os operadores genéticos cruzamento e mutação às soluções escolhidas pelo mecanismo de selecção. Por fim, repete-se todo o processo para a próxima geração (t = t + 1).

Capítulo 3

Análise de Robustez em Optimização

1. Caracterização da incerteza

1.1. Introdução

A tomada de decisões num contexto de incerteza é certamente uma das situações mais frequentes em problemas reais, nomeadamente no planeamento de actividades em diversos campos. A modelação da incerteza dos dados associados aos modelos matemáticos através de distribuições de probabilidade tem sido o principal meio para incorporar explicitamente informação que não é completamente conhecida na fase de construção dos modelos (ou porque não está disponível, ou é variável com o tempo ou outros factores, ou é contraditória entre diversas fontes, ou é controversa para diferentes intervenientes no processo). A incerteza pode ser proveniente de diversas fontes ou ser classificada em diferentes tipos, não sendo, em geral, adequado representar probabilisticamente todas as formas de incerteza associada a modelos matemáticos, nomeadamente se a informação disponível é para tal insuficiente (Agarwal et al. (2004)).

Tem-se assistido, nos últimos anos, a um crescimento do interesse do tratamento da incerteza em modelos matemáticos. O modelo matemático associado a um sistema real pode incluir vários tipos de incerteza, a qual pode ocorrer nos dados do modelo, na precisão do modelo usado para descrever o sistema, ou na sequência de possíveis acontecimentos que podem ocorrer num sistema de acontecimentos discretos (Oberkampf et al. (2004)).

A importância da construção de modelos que incorporem explicitamente a incerteza está no facto de a maioria dos problemas do mundo real não poderem ser modelados deterministicamente. Se isto acontecesse, bastaria "apenas" determinar a solução que optimiza um problema mono-objectivo ou escolher uma solução do conjunto de soluções não dominadas de um problema multi-objectivo, usando algoritmos específicos para tal. Como o que geralmente acontece é os dados associados aos problemas serem incertos, é necessário recorrer a modelos que incorporem a incerteza, uma vez que a qualidade das próprias soluções é também incerta. Algumas razões da necessidade deste tipo de modelos são as seguintes:

1) a natural incerteza das previsões relativas ao futuro;
2) a impossibilidade de medir os conceitos do mundo real com a precisão exigida pelo modelo matemático;
3) a impossibilidade de implementar uma solução com a precisão obtida através do modelo matemático;
4) a natural e constante alteração do mundo real onde a solução é implementada;
5) o facto das funções objectivo serem apenas traduções aproximadas dos objectivos do mundo real.

1.2. Tipos e causas da incerteza

A comunidade científica tem vindo, nos últimos anos, a diferenciar e caracterizar diferentes formas de incerteza.

Oberkampf et al. (2004) classificaram a incerteza em dois tipos: *aleatória* e *epistémica*. A *incerteza aleatória* é também designada por variabilidade, incerteza irreduzível, incerteza inerente, ou incerteza estocástica. A *incerteza epistémica* é também designada por incerteza reduzível, incerteza subjectiva, ou incerteza sobre a forma do modelo. Limbourg e Aponte (2005) também designaram a *incerteza aleatória* por *perturbação*[9] e a *epistémica* por *imprecisão*.

A *incerteza aleatória* descreve a variação inerente associada ao sistema físico ou ambiente em consideração (Oberkampf et al. (2004)), em que esta variação é causada, normalmente, pela natureza aleatória dos dados associados ao problema, podendo ser representada matematicamente por uma distribuição de probabilidade, desde que os dados experimentais disponíveis sejam suficientes (Agarwal et al. (2004)). A *incerteza aleatória* é o tipo de incerteza que tem sido alvo de mais estudos, sendo uma propriedade inerente à modelação de sistemas (ou inserida no modelo para simular este comportamento) e que não pode ser reduzida (Limbourg e Aponte (2005)).

[9] Tradução da palavra inglesa "noise".

A *incerteza epistémica* está associada a um certo nível de ignorância, ou informação incompleta, do sistema ou do ambiente que o rodeia. A *incerteza epistémica* é usada para descrever qualquer falta de conhecimento ou informação numa qualquer fase ou actividade do processo de modelação do sistema (Oberkampf et al. (2004)).

As causas da incerteza influenciam o tipo de informação associada ao modelo usado no tratamento da incerteza para traduzir o sistema físico em análise. São várias e distintas as causas da incerteza (Zimmermann (1998)): falta de informação, excesso de informação, provas em conflito, ambiguidade, medições e crença.

A *falta de informação* é provavelmente a causa mais frequente da incerteza. Existem situações em que um AD não tem qualquer informação acerca de qual dos possíveis estados naturais irá ocorrer. Noutras, o AD conhece apenas as probabilidades de ocorrência dos vários estados. Uma outra situação é aquela em que ninguém tem ou quer reunir informação suficiente para construir uma descrição exacta do acontecimento, embora isto possa ser possível.

O *excesso de informação* deve-se ao facto de ser limitada a capacidade do Ser Humano em perceber e processar, simultaneamente, grande quantidade de dados. Em muitas situações, são disponibilizados mais dados do que aqueles que podem ser processados; noutras, os fenómenos que lhe são transmitidos são definidos ou descritos com um grande número de características ou propriedades. Normalmente, estes dados são processados pelo analista que os transforma em dados perceptíveis, ou que centra a sua atenção apenas nos aspectos que lhe parecem ser os mais importantes (que podem não coincidir com os do AD) e negligenciando todos os outros dados ou informação.

Podem existir várias classes de informação (importante) disponível sobre um mesmo sistema, mas em que cada uma delas aponta para diferentes comportamentos do sistema (*provas em conflito*). Este conflito pode acontecer devido ao facto de parte da informação disponível ao analista estar errada (mas não identificável como tal pelo analista), das características da informação serem irrelevantes para o sistema, do modelo que o analista tem do sistema ser incorrecto, etc..

Se numa situação certas informações (por exemplo, linguísticas) têm significados totalmente diferentes ou têm uma correspondência de um para vários (matematicamente falando), então diz-se que existe *ambiguidade*. Todas as linguagens contêm palavras que, por diversas razões, têm diferentes significados em diferentes contextos.

O termo *medição* é aqui usado no sentido de "medição em engenharia", isto é, instrumentos para medição de aspectos físicos, tais como peso, temperatura, altura, etc.. No entanto, se uma dada propriedade exacta não puder ser medida com precisão, tem-se

alguma incerteza relativamente à medição real, conhecendo-se apenas uma medida indicativa.

Em todas as classes de causas da incerteza já referidas, sempre se considerou que a informação disponível ao analista é objectiva. No entanto, é importante considerar uma outra classe para as situações em que a informação disponível ao analista é subjectiva; estas situações são classificadas segundo um tipo de *crença* numa certa circunstância. Estas situações são, talvez, as mais duvidosas de todas, pois também podiam ser classificadas como "falta de informação" no sentido objectivo.

A utilização de dados incertos em modelos de programação matemática tem sido reconhecida como um problema central na optimização, uma vez que na maioria das situações de apoio à decisão do mundo real, o AD/analista não tem possibilidades de especificar os valores exactos dos dados do modelo, devido ao facto de a informação conter um certo grau de incerteza. Assim sendo, é conveniente considerar a extensão dos modelos matemáticos de apoio à decisão para ambientes onde predomina a incerteza, sem assumir o carácter decisivo dos dados do modelo.

1.3. Modelos que incorporam a incerteza

Tratar a incerteza inclui construir modelos que incorporem a incerteza e, posteriormente, abordagens para determinar as soluções destes modelos.

Apesar de não existir um modelo único capaz de incorporar qualquer tipo de incerteza, o modelo escolhido tem de ser adequado, quer ao tipo de incerteza quer à quantidade e qualidade da informação disponível. Os modelos mais usados para tratar a incerteza, são os seguintes (Zimmermann (1998), Agarwal et al. (2004) e Matos (2007)): probabilístico, difuso, baseado em intervalos e baseado em cenários.

O *modelo probabilístico* é geralmente usado quando existe muita informação acerca do problema resultante de análises estatísticas ou proporcionais. A situação mais frequente é ter-se distribuições independentes dos dados incertos associados ao problema. Cada dado incerto tem vários valores possíveis, os quais são independentes dos valores dos restantes dados. Se a probabilidade de cada valor possível para um dado incerto for conhecida, então diz-se que este dado é caracterizado por uma distribuição de probabilidade. Cada dado pode ter um número finito ou infinito de valores possíveis, resultando numa distribuição de probabilidade discreta ou contínua, respectivamente.

O *modelo difuso* (isto é, aquele que modela o problema usando conjuntos difusos) é geralmente usado quando não existe informação estatística disponível ou quando se está a tratar de descrições qualitativas correspondentes a declarações de peritos acerca dos dados ou do impacto das alternativas. Enquanto que no modelo clássico de conjuntos, a

relação entre um objecto e um conjunto é de pertença ou não pertença, no modelo difuso um objecto pode também pertencer parcialmente a um conjunto, existindo um grau de pertença nesta relação. A teoria de conjuntos difusos, desenvolvida por Zadeh (1965), é uma extensão à teoria clássica de conjuntos em que o grau de pertença para um elemento num conjunto toma um valor algures no intervalo [0,1], em vez de apenas 0 ou 1. Um conjunto difuso é uma classe de objectos em que não existe uma fronteira bem definida entre os objectos que pertencem à classe e os que não pertencem. Esta teoria foi desenvolvida no intuito de tornar tratável a complexidade subjacente a descrições de processos subjectivos ou mal entendidos.

O *modelo baseado em intervalos* considera que alguns ou todos os dados incertos do problema são descritos através de intervalos, em vez de um único valor numérico. Os intervalos são também usados para traduzir descrições qualitativas em valores numéricos. Na sua formulação básica, os intervalos não têm quaisquer ligações com as distribuições probabilísticas ou possibilísticas, sendo muito mais parecidos com os cenários (a descrever a seguir), pois apenas tentam capturar cada possível valor futuro dos dados relevantes. Os intervalos podem ser considerados como um caso particular de valores difusos, correspondendo a nenhuma informação adicional para além da gama de valores possível.

No *modelo baseado em cenários* os dados incertos são estimados globalmente, consideradas as suas correlações e construídos diferentes *futuros estruturados*. Apesar deste modelo não requerer uma distribuição de probabilidade, o conceito de "cenário mais provável" é muitas vezes invocado num sentido qualitativo, o que significa, geralmente, que todos os restantes cenários serão desprezados. Os cenários são possíveis (futuras) instâncias de dados do problema, sendo sempre necessário um modelo que forneça os meios para avaliar as consequências de uma decisão em cada cenário possível, para tornar esta abordagem útil. Atribuir probabilidades a cenários é uma prática usual, geralmente através de um conjunto de valores estimados por um perito.

2. Métodos associadas a modelos com incerteza

Existem, essencialmente, cinco classes de métodos em programação matemática que podem ser aplicados aos modelos que incorporam a incerteza (apresentados na secção anterior). Estas classes são as seguintes: análise de sensibilidade, programação difusa, programação intervalar, programação estocástica e programação robusta.

A *análise de sensibilidade* permite obter *a posteriori* as gamas de variação dos dados do problema, para as quais as soluções encontradas não sofrem alterações. Deste

modo, avalia-se o efeito de variações previsíveis nos dados do problema sobre as soluções do modelo, não tendo em conta *a priori* a incerteza associada aos dados do problema. Por esta razão, alguns autores não consideram a *análise de sensibilidade* como uma abordagem para modelar a incerteza, pois a incerteza não é incorporada no processo de pesquisa de soluções.

Segundo Bellman e Zadeh (1970), a tomada de decisão num ambiente difuso é considerada como um processo de decisão em que os objectivos e/ou as restrições são de natureza difusa, o que significa que os objectivos e/ou as restrições constituem classes de alternativas cujas fronteiras não estão definidas com precisão. Os objectivos e as restrições podem assim ser definidas de um modo adequado na forma de conjuntos difusos, no espaço das alternativas.

A optimização difusa pretende determinar a solução "mais satisfatória" na presença de informação incompleta, subjectiva, imprecisa e/ou vaga. A primeira abordagem para modelos de decisão em ambientes difusos foi proposta por Bellman e Zadeh (1970), na qual não existe diferenciação entre os objectivos e as restrições, pretendendo-se tornar menos rígidas as noções de restrições e de optimização das funções objectivo.

Ao contrário das abordagens clássicas de optimização, as *abordagens difusas* não apresentam uma formulação única, pois permitem muitas variações semânticas, de acordo com as características do sistema real a modelar. Vários autores consideram duas classes de abordagens difusas distintas: programação flexível e programação robusta. Na programação difusa flexível, a estrutura dos modelos é fixa (todos os dados envolvidos são conhecidos) e as relações matemáticas envolvidas são difusas (objectivos e restrições difusas). Na programação difusa robusta, a estrutura dos modelos não é conhecida com exactidão, isto é, os dados dos modelos não podem ser fornecidos com precisão (Borges e Antunes (2002)). Para informações mais detalhadas sobre esta temática, ver também Borges (2005).

Na *programação intervalar* pressupõe-se que existe informação acerca das gamas de variação de alguns (ou de todos) dados do problema, os quais permitem especificar um modelo em que os dados são intervalos. As abordagens que têm sido propostas no âmbito da programação matemática intervalar estão essencialmente dirigidas para o tratamento da incerteza apenas nos dados associados às funções objectivo (a maioria), nos dados associados às funções objectivo e aos termos independentes das restrições, ou em todos os dados associados ao modelo.

Inuiguchi e Kume (1994) e Inuiguchi e Sakawa (1995) consideram duas abordagens diferentes para tratamento da incerteza apenas com funções objectivo intervalares: abordagem de satisfação e abordagem de optimização. Na abordagem de satisfação cada função objectivo intervalar é transformada numa ou em várias funções

objectivo, de forma a obter uma solução de compromisso. A abordagem de optimização estende o conceito de eficiência usado em programação multi-objectivo, para o caso intervalar. Oliveira e Antunes (2007) contém informações mais detalhadas sobre esta temática.

A *programação estocástica* requer a existência de dados estatísticos suficientes para obter as funções de distribuição dos dados incertos do modelo ou o uso de probabilidades subjectivas, quando não existe aquele tipo de informação. Em meados dos anos 1950, Dantzig (1955) introduziu a programação estocástica como uma abordagem a modelos com incerteza nos seus dados, considerando cenários que ocorrem com diferentes probabilidades para descrever os dados do problema. As duas principais dificuldades de uma abordagem deste tipo são as seguintes:

a) não é fácil obter, na prática, a distribuição exacta para os dados, e assim, enumerar os cenários que recolhem esta distribuição;
b) o tamanho do modelo de optimização resultante aumenta drasticamente de acordo com o número de cenários, o que acarreta grandes desafios em termos computacionais.

No entanto, uma técnica normalmente usada para tratar este tipo de incerteza consiste em transformar o problema (não determinístico) num outro (determinístico), substituindo todos os parâmetros estocásticos do modelo pelos seus valores esperados (*problema do valor médio*). Desta forma, a solução "óptima" é aquela com o melhor valor esperado.

A *programação robusta* é baseada no conceito de *robustez* e nas noções relacionadas com robustez, cuja ideia base foi apresentada por Kouvelis e Yu (1997) e que corresponde ao paradigma *min-max* que pode ser enunciado da seguinte forma: *escolher a alternativa que, no pior caso, tem o melhor valor do atributo*. A formulação base da robustez ignora qualquer informação adicional, tal como distribuições de probabilidade. O conceito de robustez apesar de estar relacionado com o modelo baseado em cenários, não lhe está restrito, podendo ser usado com intervalos ou conjuntos difusos. A finalidade da *programação robusta* é determinar as soluções que estejam imunes a perturbações nos valores dos dados do modelo.

A programação robusta é uma das classes de abordagens de programação matemática que tem sido proposta com maior frequência nos últimos anos. A programação robusta tem sido usada numa grande variedade de modelos, sendo muitas vezes subdividida segundo as características mais específicas destes modelos, como é o caso dos modelos de optimização discreta robusta. Por esta razão, o estudo da *optimização discreta robusta* será também individualizado numa secção própria.

3. Conceitos de robustez

O conceito de robustez não é usado de forma uniforme por todos os investigadores. A diversidade de situações é tão grande que, provavelmente, seria necessário classificar os tipos de problemas de decisão e os tipos de incerteza, antes de propor diferentes tipos de robustez.

O primeiro conceito de robustez em problemas de decisão foi apresentado por Gupta e Rosenhead (1968). Desde então, várias interpretações de robustez têm sido introduzidas por diferentes autores.

Na literatura existem essencialmente quatro conceitos distintos, os quais podem ser pontos de partida para futuros desenvolvimentos (Vincke (2003)): *decisão robusta*, *solução robusta*, *conclusão robusta* e *método robusto*.

O conceito de *decisão robusta* num *contexto dinâmico* (Gupta e Rosenhead (1968), Rosenhead et al. (1972) e Rosenhead (1989)) pode também ser denominada de flexibilidade, uma vez que, num dado momento, a decisão é robusta se continuar, o mais possível, receptiva a planos "bons" no futuro.

O conceito de *solução robusta* em *problemas de optimização* (Rosenblatt e Lee (1987), em problemas de planeamento de instalações; Mulvey et al. (1995), em programação matemática; Kouvelis e Yu (1997), em problemas de optimização combinatória), onde robustez significa "boa em todos ou na maioria dos cenários" e em que um cenário é um conjunto de valores possíveis para os dados do modelo.

O conceito de *conclusão robusta* (Roy (1998) e Dias e Clímaco (1999)), onde robustez significa "válida em todos ou na maioria dos cenários" e em que um cenário é um conjunto de valores aceitáveis para os dados do modelo.

O conceito de *método robusto* (Vincke (1999-a), Vincke (1999-b) e Sörensen (2001)), onde robustez significa "que fornece resultados válidos em todos ou na maioria dos cenários" e onde um cenário é um conjunto de valores possíveis para os dados do modelo e para os parâmetros do método.

3.1. Decisão robusta

A noção de robustez aplicada a problemas de decisão foi inicialmente introduzida por Gupta e Rosenhead (1968) no contexto de problemas de planeamento sequencial. Neste tipo de problemas, as decisões são construídas ao longo do tempo perante diferentes fontes de incerteza. Desta forma, todas as decisões irão afectar os planos futuros, limitando o número de planos "bons" que poderão ser alcançados no futuro. A robustez de uma dada decisão baseia-se na flexibilidade que mantém. Uma decisão é

considerada mais flexível quanto menos limitar o número de planos "bons" no futuro. O ideal era ser-se capaz de tomar decisões num dado instante sem limitar as diferentes possibilidades no futuro. Quanto mais flexível for uma solução, mais robusta será. Matematicamente, Rosenhead et al. (1972) definiram robustez da forma seguinte.

Seja d_i a decisão inicial escolhida de um conjunto de decisões D. Seja S o conjunto de todos os possíveis planos realizáveis no futuro. Seja S_i um subconjunto de S de planos alcançáveis após a decisão d_i ter sido escolhida. Sejam S* e S_i* o subconjunto de S e S_i de planos "bons" ou "aceitáveis", respectivamente. Então, a robustez da decisão d_i é medida em função do subconjunto de "bons" planos, isto é,

$$r_i = \frac{num(S_i^*)}{num(S^*)},$$

onde num(S) é o número de elementos do conjunto S. Quanto maior for o valor de r_i mais robusta é a decisão.

Como referido por Rosenhead et al. (1972), este critério dá mais ênfase ao planeamento contínuo de processos do que ao próprio plano final. A robustez está mais relacionada com as possíveis (diferentes) consequências de cada decisão considerada ao longo do planeamento do que com a própria decisão.

Nesta definição de robustez é considerada apenas a qualidade da solução (a ideia que uma solução robusta é aquela que se altera ligeiramente de um cenário para outro, não é aqui tido em conta).

3.2. Solução robusta

Mulvey et al. (1995) apresentaram uma formulação geral para um modelo de programação matemática mono-objectivo (a que chamaram de "optimização robusta"), baseada em cenários. Nesta formulação, uma solução de um modelo de optimização é definida como *solução robusta* se permanece "próxima" do óptimo em todos os cenários e como *modelo robusto* se continua "quase" admissível em todos os cenários.

Kouvelis e Yu (1997) apresentaram uma noção de robustez, mais conservadora, para problemas de optimização discreta, utilizando cenários. Mais conservadora significa que a noção assume que o AD corre o menor risco possível.

Nas abordagens apresentadas por Kouvelis e Yu (1997), as soluções robustas são aquelas cujos valores são os melhores no pior cenário, em que o pior cenário pode ser definido de várias formas. Em particular, estes investigadores definiram três medidas de robustez para definirem o pior cenário, uma baseada no critério *min-max absoluto* (robustez absoluta) e duas outras baseadas no critério *min-max regret* (desvio robusto e robustez relativa).

3.3. Conclusão robusta

Roy (1998) sugeriu a aplicação do conceito de robustez, não apenas a soluções mas, mais genericamente, a *conclusões* (asserções, recomendações).

Uma *conclusão* é uma informação deduzida a partir do modelo e dada ao AD durante o processo de decisão. Esta pode ser uma *proposição* para uma solução do problema, mas também pode ser uma *propriedade* ou um *facto* que pode ser útil ao AD. Uma *conclusão* diz-se *robusta* se é válida em todos (ou quase todos) os cenários, em que um cenário é um conjunto de valores possíveis para os dados do modelo usado para resolver o problema.

Uma *proposição* formal (por exemplo, 'o valor da função objectivo da solução *x* é pelo menos igual ao da solução *y*') é definida como sendo *perfeitamente robusta* se for válida em todos os cenários. Uma *proposição* formal é *aproximadamente robusta* se é válida em todos os cenários, excepto nalguns (não necessariamente identificados com clareza), que são considerados insignificantes. Se uma *proposição* é menos formal (por exemplo, 'a qualidade da solução *x* é boa'), mas é válida em todos os cenários, então diz-se que ela é *pseudo-robusta*.

Dias e Clímaco (1999) refinaram as definições anteriores, apresentadas por Roy (1998), e propuseram a seguinte classificação.

Uma *conclusão robusta absoluta* é uma premissa intrínseca a uma das acções (soluções), que é válida em qualquer cenário. Por exemplo, 'o valor da acção *x* é maior do que 0.7'.

Uma *conclusão (relativa) binária robusta* é uma premissa associada a um par de acções, a qual é válida em qualquer cenário. Por exemplo, '*x* domina *y*' ou '*x* é superior a *y* com uma credibilidade maior do que 0.7' são possíveis conclusões binárias robustas.

Uma *conclusão (relativa) unária robusta* é uma premissa associada a uma acção mas em relação às outras, que é válida em todos os cenários. Por exemplo, '*x* é não-dominada' ou 'a acção *x* está entre as três melhores acções'.

3.4. Método robusto

Vincke (1999-a) propôs uma definição formal de robustez, utilizando os conceitos de procedimento, método e tratamento. Um *procedimento* é uma aplicação que associa uma solução a cada instância de um problema. Um *método* é uma sequência de procedimentos que permite obter diferentes soluções para diferentes execuções (por exemplo, estabelecendo diferentes níveis de preferências). Um *tratamento* corresponde a uma aplicação específica de um método sobre um cenário específico (conjunto de dados do modelo e de parâmetros do método). Uma *solução x* é *robusta*, relativamente a um

conjunto de tratamentos, se para todo o tratamento daquele conjunto a solução *y* obtida é semelhante a *x* em termos dos valores das funções objectivo. Um *método* é *robusto* para uma instância do problema, se para cada par de procedimentos deste método, as soluções obtidas são semelhantes entre si. Um *método* é *robusto para um problema* se é robusto para qualquer instância daquele problema.

A robustez de uma solução ou de um método é relativa, uma vez que depende da descrição do modelo ou do método. Isto porque a robustez não é uma qualidade intrínseca de uma solução (ou de um método), mas depende muito do contexto. O que é importante é uma certa coerência na formalização do modelo de decisão e na aplicação do método. Para a definição desta coerência pode ser útil a noção de robustez, mesmo que não seja suficiente. Desta forma, a robustez não é a única propriedade que permite caracterizar ou escolher um método, embora possa ser um meta-critério possível para avaliar os diferentes métodos.

Sörensen (2001) introduziu uma outra definição de robustez, mas relacionada com o comportamento de uma solução perante as possíveis perturbações nos dados do modelo. Se uma solução for muito sensível a pequenas perturbações naqueles dados, então esta não deve ser uma boa solução. A robustez está, assim, relacionada com a sensibilidade de uma solução em relação às perturbações que podem ocorrer nos dados do modelo. Este autor distinguiu dois tipos de robustez: *robustez de qualidade* e *robustez de solução*.

A *robustez de qualidade* de uma *solução* está relacionada com o facto da sua qualidade como solução continuar elevada, mesmo quando ocorrem alterações nos dados do modelo. Este tipo de robustez é especialmente importante em modelos em que é intratável uma frequente reoptimização do modelo, como, por exemplo, em modelos de localização de instalações (fábricas, armazéns, etc.). A *robustez de qualidade* pode ser descrita como robustez no espaço dos objectivos (Sevaux e Sörensen (2004)).

Nalguns modelos, quando ocorrem alterações nos seus dados, poderá ter que ser encontrada uma nova solução, uma vez que a solução original pode não satisfazer as restrições do novo modelo e, desta forma, o modelo terá que ser reoptimizado. Nestes casos, pode-se exigir que a nova solução seja o mais possível semelhante à original. Esta semelhança terá que ser medida no espaço das soluções e não no espaço dos objectivos. Desta forma, a *robustez de solução* pode ser descrita como robustez no espaço das soluções (Sevaux e Sörensen (2004)). Uma solução é robusta no espaço das soluções se a nova solução sofrer apenas uma ligeira modificação, quando comparada com a solução original, com as alterações nos dados do modelo.

Contrariamente à *robustez no espaço dos objectivos*, a *robustez no espaço das soluções* não é apenas uma característica da solução, mas também do método utilizado

para a determinar. No entanto, a principal diferença entre estes dois tipos de robustez é que, enquanto na primeira é a qualidade da solução que tem que se manter, na segunda é a própria solução que não pode ser alterada (Sörensen (2001)).

4. Optimização robusta

A optimização robusta estuda metodologias de modelação que, combinadas com ferramentas computacionais, processa problemas de optimização com dados incertos e em que apenas se conhece se pertencem a um determinado conjunto de incerteza.

Soyster (1973) propôs um modelo de optimização linear para construir uma solução que fosse admissível para todos os dados do problema, tal que cada um dos dados do problema incerto pudesse tomar qualquer valor de um intervalo. No entanto, esta abordagem tende a determinar soluções demasiado conservadoras.

Mulvey et al. (1995) apresentaram uma abordagem, a que chamaram de *"optimização robusta"*, a qual integra formulações de *programação por metas* em que a descrição dos dados do problema é baseada em cenários. Esta abordagem é analisada detalhadamente na próxima secção deste capítulo (secção 4.1), por ser considerada por muitos autores um marco importante da optimização robusta.

El-Ghaoui e Lebret (1997) e El-Ghaoui et al. (1998) introduziram um maior conservadorismo nas soluções robustas, considerando conjuntos de incerteza dos dados do problema como elipsóides, e propuseram algoritmos eficientes para resolver problemas de optimização convexa com dados incertos. Os autores trataram da reformulação robusta do modelo de optimização, adaptando técnicas de controlo robusto partindo da suposição de que os dados da matriz dos coeficientes podem variar no interior do elipsóide associado ao conjunto da incerteza. Em alguns problemas importantes, os modelos robustos são exacta ou aproximadamente tratáveis, sendo eficientemente resolvidos através de métodos de ponto interior.

Também Ben-Tal e Nemirovski (1998), Ben-Tal e Nemirovski (1999) e Ben-Tal e Nemirovski (2000) introduziram um maior conservadorismo nas soluções robustas, considerando os mesmos tipos de conjuntos e de algoritmos que os autores anteriores. No entanto, como as formulações robustas resultantes conduzem a problemas quadráticos cónicos (ver Ben-Tal e Nemirovski (1999)), tais métodos não podem ser aplicados directamente a problemas discretos.

Ben-Tal e Nemirovski (2002) passam em revista os principais resultados obtidos pela "Optimização Robusta" quando aplicada à programação linear incerta, cónica quadrática e semi-definida.

Bertsimas e Sim (2003) e Bertsimas e Sim (2004) propuseram uma abordagem diferente para controlar o nível de conservadorismo de uma solução, a qual tem a vantagem de conduzir a um modelo de optimização linear, e assim, poder ser aplicado directamente em modelos de optimização discreta.

4.1. Abordagem de Mulvey, Verderbel e Zenios

Mulvey et al. (1995) introduziram uma abordagem pro-activa que modela a incerteza associada aos dados de um problema de programação matemática mono-objectivo usando cenários, produzindo soluções menos instáveis, relativamente aos dados do modelo, do que as formulações clássicas de programação matemática. Nesta abordagem, os dados do problema são representados por um conjunto de cenários, em que a probabilidade de cada um deles é supostamente conhecida. Desta forma, a noção tradicional de "solução óptima" deixa de fazer sentido, uma vez que é raro que uma solução seja óptima em todos os cenários possíveis do problema. Esta abordagem consiste em gerar uma cadeia de soluções que vão sendo progressivamente menos instáveis relativamente aos dados do modelo de um conjunto de cenários. O facto de algumas soluções poderem violar as restrições do problema original nalguns cenários é explicitamente reconhecido, sendo introduzidas funções de penalização para penalizar as soluções que violam as restrições.

Os autores apresentaram dois conceitos diferentes de robustez: *solução robusta* e *modelo robusto*. A solução óptima de um problema de programação matemática é robusta, relativamente à optimalidade, se permanecer "próxima" do óptimo em qualquer cenário (*solução robusta*). A solução óptima de um problema de programação matemática é robusta, relativamente à admissibilidade, se permanecer "quase" admissível em qualquer cenário (*modelo robusto*). As noções de "próxima" e "quase" são definidas com exactidão através da escolha de normas, as quais estão relacionadas com o risco que o AD pretende correr.

O modelo apresentado tem duas componentes distintas: a *estrutural* (que é fixa e livre de qualquer perturbação nos dados do problema) e a *de controlo* (que está sujeita aos dados do problema que são perturbados). Para se definir o modelo apropriado, é necessário introduzir dois conjuntos de variáveis:

- $x \in \mathbb{R}^{n1}$ é o vector das *variáveis de decisão* cujo valor óptimo não está dependente de alterações dos dados incertos (as variáveis de *projecto*). Estas variáveis não podem ser ajustadas, uma vez que estão perante um cenário único e inalterado.
- $y \in \mathbb{R}^{n2}$ é o vector das *variáveis de decisão* que estão sujeitas a ajustamentos, uma vez que existem dados incertos (variáveis *de controlo*). Os seus valores óptimos não

dependem apenas da existência de dados incertos, mas também dos valores óptimos das variáveis do projecto.

em que n1 + n2 = n (n é o número total de variáveis de decisão do problema).

Num contexto de planeamento da produção, as variáveis do projecto determinam a estrutura do processo e o tamanho dos módulos de produção. As variáveis de controlo são usadas para ajustar o modo e o nível de produção em resposta à rotura do processo, às alterações na procura ou na quantidade produzida, etc.. O modelo de optimização tem a seguinte estrutura:

Minimizar $c^T x + d^T y, \quad x \in \mathbb{R}^{n1}, y \in \mathbb{R}^{n2}$ (1)

sujeito a

$A x = b,$ (2)

$B x + C y = e,$ (3)

$x, y \geq 0,$ (4)

A equação (2) indica as restrições estruturais cujos coeficientes são fixos e livres de perturbações. A equação (3) indica as restrições de controlo, cujos coeficientes estão sujeitos a perturbações.

Para definirem o problema de optimização robusta, os autores introduziram um conjunto de cenários $\Omega = \{1, 2, ..., S\}$, em que cada cenário $s \in \Omega$ tem associado o conjunto $\{d_s, B_s, C_s, e_s\}$ de realizações de valores das variáveis de controlo (d_s) e de coeficientes das restrições de controlo (B_s, C_s, e_s), e a probabilidade do cenário p_s (com $\sum_{s\in\Omega} p_s = 1$). A solução óptima do programa matemático (1)-(4) será robusta, relativamente à optimalidade, se permanecer "próxima" do óptimo para qualquer cenário $s \in \Omega$ (*solução robusta*). A solução é também robusta relativamente à admissibilidade, se permanecer "quase" admissível para qualquer cenário $s \in \Omega$ (*modelo robusto*).

É pouco provável que alguma solução do modelo (1)–(4) permaneça admissível e óptima em todos os cenários $s \in \Omega$. Se o sistema que está a ser modelado tem embutido redundâncias substanciais, então pode ser possível encontrar soluções que permaneçam quer admissíveis quer óptimas. Caso contrário, é necessário um modelo que permita que seja medido o compromisso entre a robustez da solução e do modelo. O modelo de optimização robusto formaliza uma forma de medir este compromisso.

O modelo de optimização robusta tem a seguinte estrutura:

Minimizar $\sigma(x, y_1, y_2, ..., y_s) + \omega\rho(z_1, z_2, ..., z_s)$

sujeito a

$A x = b,$

$B_s x + C_s y + z_s = e_s, \qquad \forall s \in \Omega$

$x, y_s \geq 0, \qquad \forall s \in \Omega$

em que { y_1, y_2, ..., y_s } é um conjunto de variáveis de controlo para cada cenário $s \in \Omega$ e { z_1, z_2, ..., z_s } é um conjunto de vectores de erro que medirão a não admissibilidade permitida pelas restrições de controlo no cenário *s*.

No caso de múltiplos cenários, a função objectivo $\xi = c^T x + d^T y$ transforma-se numa variável aleatória que assume o valor $\xi_s = c^T x + d_s^T y_s$, com probabilidade p_s. Desta forma, como função global pode-se utilizar, por exemplo, o valor esperado $\left(\sigma(.)=\sum_{s\in\Omega} p_s \xi_s\right)$, que é a função usada na programação linear estocástica, ou o valor no pior caso $\left(\sigma(.) = \max_{s\in S} \xi_s\right)$.

O segundo termo da função objectivo $\rho(z_1, z_2, ..., z_s)$ é uma função de penalização da admissibilidade, que é usada para penalizar as violações das restrições de controlo em alguns dos cenários.

Este modelo tem uma forma de objectivos múltiplos: o primeiro termo mede a robustez da optimalidade e o termo de penalização é uma medida da robustez do modelo. O peso da programação por metas ω é usado para obter um espectro de respostas às soluções de compromisso da robustez do modelo.

A introdução da função de penalização deve-se ao facto de o modelo reconhecer que nem sempre é possível obter uma solução que seja admissível em todos os cenários. A escolha específica de uma função de penalização é dependente do problema, e tem também implicações na implementação do algoritmo usado na obtenção da solução. Por exemplo, os autores consideraram duas funções de penalização alternativas:

- $\rho(z_1, z_2, ..., z_s) = \sum_{s\in\Omega} p_s z_s^T z_s$ é uma função de penalização quadrática que se aplica a problemas com restrições de igualdade, onde não são desejáveis violações das restrições de controlo, quer sejam positivas ou negativas.
- $\rho(z_1, z_2, ..., z_s) = \sum_{s\in\Omega} p_s \max\{0, z_s\}$ é uma função de penalização exacta que se aplica a restrições de controlo do tipo desigualdade, quando apenas têm interesse as violações positivas (valores negativos de z indicam inactividade das restrições de desigualdade).

4.2. Optimização discreta (combinatória) robusta

Um dos ramos mais interessantes da optimização discreta que tem emergido nos últimos anos é a optimização robusta. A optimização discreta robusta utiliza abordagens baseadas em cenários, mas em que a probabilidade de cada cenário se realizar não é conhecida nem especificada.

Na aplicação da abordagem robusta, devem ser considerados três elementos críticos inseridos num esquema sistemático de apoio à decisão, que são os seguintes (Kouvelis e Yu (1997)):

- utilização de uma Abordagem por Planeamento de Cenários para estruturar os dados incertos numa situação de decisão,
- escolha do(s) Critério(s) de Robustez apropriado(s) para a situação de decisão,
- desenvolvimento formal de um Modelo de Decisão.

Ou seja, para operacionalizar o apoio à decisão, uma abordagem robusta começa por identificar potenciais cenários para o modelo de decisão adequado à situação, sem contudo atribuir probabilidades aos vários cenários, para de seguida se ocupar em determinar a melhor decisão que está associada precisamente ao pior cenário identificado (Kouvelis e Yu (1997)).

Para se construir o pior cenário, Kouvelis e Yu (1997) definiram três medidas (critérios) diferentes de robustez, cuja escolha depende do risco que o AD pretende correr: robustez absoluta, desvio robusto e robustez relativa. Depois de criados os cenários e escolhido um critério de robustez, o problema é então formulado como um modelo de optimização que gere decisões robustas.

Desta forma, é importante identificar classes de modelos, para os quais pequenas alterações nos dados do problema levam a pequenas alterações nos resultados no pior cenário possível.

Nas definições apresentadas nesta secção, considera-se sempre que o problema em análise é de minimização.

Seja S o conjunto de todos os cenários possíveis, X o conjunto de todas as soluções admissíveis do problema e D o conjunto de todos os dados do problema. Seja X_S o conjunto de todas as soluções admissíveis para o cenário *s*. Suponhamos que a qualidade da solução $x \in X_S$ é calculada usando a função f(x, s) — note-se a dependência da função f(x) da solução x e do cenário *s*. A *solução óptima no cenário* $s \in S$, x_s^*, é tal que (Kouvelis e Yu (1997))

$$f(x_s^*, s) = \min_{x \in X_s} f(x, s).$$

Robustez baseada num critério *min-max simples*

As soluções robustas absolutas são de natureza conservadora, uma vez que admitem à partida que o pior cenário possível vai acontecer.

A *solução robusta absoluta*, x^a, é a solução que minimiza o custo total máximo, entre todas as soluções admissíveis de todos os cenários realizáveis, isto é

$$z^a = \max_{s\in S} f(x^a, s) = \min_{x \in \cap_{s\in S} X_s} \max_{s\in S} f(x, s).$$

Dada uma solução $x \in X$, o *pior cenário absoluto para x*, s^a_x, é o cenário no qual o custo da solução x é máximo, isto é (Hites (2002))

$$s^a_x = \arg \max_{s\in S} f(x, s).$$

Com base na definição anterior, Hites (2002) definiu a *solução robusta absoluta* como a solução com o menor custo no pior cenário absoluto.

Robustez baseada num critério *min-max regret*

Kouvelis e Yu (1997) utilizaram duas variantes do conceito de *min-max regret* para definirem o pior cenário. Estas noções são menos conservadoras do que a anterior, uma vez que tendem a olhar para a incerteza mais como uma oportunidade a explorar do que como um risco a correr.

A *solução desvio robusta*, x^d, é a solução que minimiza o desvio máximo da optimalidade, entre todas as soluções admissíveis de todos os cenários realizáveis x^*_s, isto é (Kouvelis e Yu (1997))

$$z^d = \max_{s\in S} \left(f(x^d, s) - f(x^*_s, s) \right) = \min_{x \in \cap_{s\in S} X_s} \max_{s\in S} \left(f(x, s) - f(x^*_s, s) \right).$$

Dada uma solução $x \in X$, o *pior cenário desvio para x*, s^d_x, é o cenário no qual é máxima a diferença entre o custo da solução x e o custo da solução óptima neste cenário, isto é (Hites (2002))

$$s^d_x = \arg \max_{s\in S} (f(x, s) - f(x^*_s, s)).$$

Denomina-se por *pior desvio para a solução x* a diferença (Hites (2002))

$$d_x = f(x, s^d_x) - f\left(x^*_{s^d_x}, s^d_x\right).$$

Com base na definição anterior, Hites (2002) definiu a *solução desvio robusta* como a solução que se desvia menos da solução óptima no seu pior cenário desvio.

A *solução robusta relativa*, x^r, é a que, em termos percentuais, minimiza o desvio máximo da optimalidade entre todas as soluções admissíveis de todos os cenários realizáveis, isto é (Kouvelis e Yu (1997))

$$z^r = \max_{s\in S}\left(\frac{f(x^r, s)-f(x^*_s, s)}{f(x^*_s, s)}\right) = \min_{x\in \cap_{s\in S} X_s} \max_{s\in S}\left(\frac{f(x, s)-f(x^*_s, s)}{f(x^*_s, s)}\right).$$

Dada uma solução $x \in X$, o *pior cenário relativo para x*, s^r_x, é o cenário no qual é máxima a relação entre o desvio da solução x para a solução óptima e o custo da solução óptima, neste cenário; isto é

$$s^r_x = \arg\max_{s\in S}\left(\frac{f(x, s)-f(x^*_s, s)}{f(x^*_s, s)}\right).$$

Denomina-se por *pior desvio relativo para a solução x***,** a relação

$$r_x = \frac{f(x, s^r_x)-f\left(x^*_{s^r_x}, s^r_x\right)}{f\left(x^*_{s^r_x}, s^r_x\right)}.$$

Com base na definição anterior, pode-se definir a *solução robusta relativa* como a solução que se desvia menos, em termos percentuais, da solução óptima no seu pior cenário relativo.

A justificação de Kouvelis e Yu (1997) para estas medidas é que, perante a incerteza, é necessário considerar todos as situações possíveis, incluindo as piores, uma vez que não se conhece qual delas pode vir a acontecer.

Kouvelis e Yu (1997) demonstraram que para certos problemas com dados intervalares, como é o caso de problemas em redes, o pior cenário desvio, para uma dada solução x, é o cenário em que o custo de todos os elementos de x correspondem aos seus limites inferiores e o custo de todos os outros elementos correspondem aos seus limites superiores. Por construção, este é um dos cenários em que a solução x se desvia mais da solução óptima no mesmo cenário.

5. Optimização robusta usando meta-heurísticas

A importância da optimização robusta usando meta-heurísticas é reconhecida por Kouvelis e Yu (1997), quando os autores escrevem: "*Acreditamos que deve ser despendido um esforço mais considerável no desenvolvimento sistemático de [...] esquemas meta-heurísticos, os quais com um esforço mínimo e adequado, podem ser aplicados a uma grande classe de problemas de optimização robusta, [...]*" (pág. 354).

Relativamente a abordagens em optimização robusta usando meta-heurísticas de pesquisa local, são conhecidas poucas na literatura.

Taillard (1990) e Battiti e Tecchiolli (1994) apresentaram técnicas para actualizar dinamicamente o tamanho da lista tabu, designadas de "Pesquisa Tabu Robusta" e "Pesquisa Tabu Reactiva", respectivamente. Estas técnicas podem-se considerar robustas porque determinam boas soluções sem exigirem extensas parametrizações do método.

Sörensen (2001) apresentou uma abordagem que consiste em adaptar a técnica de pesquisa tabu padrão para determinar soluções robustas e de boa qualidade (robustez no espaço dos objectivos – secção 3.4 deste capítulo). Esta abordagem consiste em substituir a função de avaliação associada à pesquisa tabu padrão, $f(x)$, por uma nova, $f_r(x)$, designada de função de avaliação robusta, de forma a avaliar a qualidade das soluções. Apesar desta função de avaliação robusta aceitar diferentes formas, todas elas seguem os mesmos dois princípios seguintes:

Princípio 1: A função de avaliação robusta adiciona uma certa perturbação à solução corrente antes de a avaliar. Assim, a técnica de pesquisa tabu robusta avalia $x^* = x + \delta$ em vez de x, em que x^* designa-se por solução derivada e pode ou não pertencer ao espaço das soluções original. A perturbação adicionada é dependente da perturbação esperada dos dados do modelo e deve reflectir as alterações esperadas destes dados.

Princípio 2: A função de avaliação robusta avalia várias soluções derivadas (e não apenas uma), combinando-as num único valor da função de avaliação.

A forma geral de uma função de avaliação robusta é a seguinte:

$$f_r(x) = \frac{1}{ne} \sum_{i=1}^{ne} w_i \, f(x+\delta_i)$$

em que ne é o número de soluções derivadas que são avaliadas e w_i é o peso associado ao valor da função de avaliação da i-ésima solução derivada, $x + \delta_i$. Esta função pode ser determinística ou estocástica, de acordo com a natureza da perturbação adicionada.

Sörensen (2003) apresentou uma metodologia que consiste na utilização de uma meta-heurística de qualquer tipo e substituição da sua função de avaliação padrão por uma formulação robusta, de forma a orientar a pesquisa em direcção a soluções robustas. A robustez das soluções pode ser medida quer no espaço dos objectivos, quer no espaço das soluções (ver definições de robustez na secção 3.4 deste capítulo).

A função de avaliação robusta mais apropriada é escolhida de acordo com o tipo de robustez que é exigido. No entanto, uma função de avaliação robusta respeita sempre os seguintes princípios (Sörensen (2001) e Sörensen (2003)):

Princípio 1: Cada solução é avaliada sobre uma amostra de dados incertos do problema (solução derivada).

Princípio 2: As várias avaliações das soluções derivadas são combinadas na função de avaliação robusta (avaliação derivada).

Para se medir a robustez das soluções no *espaço dos objectivos*, usa-se uma função de avaliação robusta em que a qualidade das soluções é avaliada por amostragem dos dados incertos do problema. Uma possível função de avaliação robusta é a média pesada de ne avaliações derivadas (tal como em Sörensen (2001)):

$$f_r(x) = \frac{1}{ne} \sum_{i=1}^{ne} w_i f(x, \delta_i).$$

Uma outra função de avaliação robusta, mais moderada, permite analisar o desempenho de uma solução no pior caso entre todas as avaliações derivadas (para funções a minimizar):

$$f_r(x) = \max_{i=1,\ldots,ne} f(x, \delta_i).$$

A robustez de uma solução medida no *espaço das soluções* é caracterizada pela proximidade (semelhança) a uma dada solução de referência, x_0, isto é, para a qual a *distância* à solução de referência (quando medida por alguma função de distância no espaço das soluções) é pequena (Sevaux e Sörensen (2004)). Para calcular a proximidade entre x_0 e x, utiliza-se uma qualquer métrica.

Nesta metodologia, a robustez pode ser medida apenas no espaço dos objectivos ou em ambos os espaços (dos objectivos e das soluções) em simultâneo.

Quando a robustez é medida apenas no espaço dos objectivos, a estrutura geral desta metodologia é a seguinte:

1) gerar muitas soluções diferentes, usando uma meta-heurística;
2) avaliar cada uma das soluções geradas, usando uma função de avaliação robusta;
3) seleccionar a solução com melhor desempenho, de acordo com a função de avaliação robusta.

Quando a robustez é medida nos espaços dos objectivos e das soluções em simultâneo, a estrutura geral desta metodologia é a seguinte:

1) gerar muitas soluções diferentes, usando uma meta-heurística;
2) calcular a qualidade de cada solução gerada e a distância entre a solução corrente e a solução de referência;
3) seleccionar a solução com o melhor compromisso entre os valores da robustez nos espaços dos objectivos e das soluções (usar um processo de apoio à decisão, tendo em conta as preferências do AD em relação à robustez nos dois espaços).

Relativamente às abordagens em optimização robusta usando meta-heurísticas baseadas em populações, é um assunto a tratar no próximo capítulo.

Capítulo 4

Análise de Robustez em Optimização Evolucionária Multi-objectivo

1. Introdução

Embora a principal finalidade do estudo de problemas de optimização seja, em geral, a de determinar as soluções óptimas globais (uma única função objectivo) ou as soluções óptimas de Pareto globais (múltiplas funções objectivo), muitas vezes estas soluções não têm interesse do ponto de vista da sua exploração prática se forem sensíveis a perturbações nos dados do modelo.

Desta forma, se uma solução é muito sensível a perturbações que ocorram na sua vizinhança então, quando implementada, pode resultar num conjunto de valores das funções objectivo diferentes relativamente à solução do modelo. Assim, estas soluções têm menor importância prática e deve ser dado maior relevo à determinação das denominadas *soluções robustas*, isto é soluções cujo desempenho é menos sensível a perturbações.

Embora a definição de solução robusta não seja uniforme na literatura, existe um ponto comum: uma solução robusta deverá comportar-se razoavelmente bem em condições (ligeiramente) diferentes, o que significa que tem mais possibilidades de ser imune a pequenas alterações nas condições para que foi projectada.

2. Optimização evolucionária em ambientes incertos

Pelas suas próprias característica os algoritmos evolucionários permitem resolver problemas de optimização na presença de diversas fontes de incertezas. Jin e Branke

(2005) apresentaram recentemente uma revisão das abordagens propostas de optimização evolucionária em ambientes incertos, na qual os métodos para tratamento da incerteza em optimização evolucionária podem ser categorizados nas seguintes quatro classes: *função de aptidão perturbada*, *função de aptidão aproximada*, *função de aptidão com variação temporal* (ou em *ambientes dinâmicos*) e *análise de robustez*.

2.1. Função de aptidão perturbada

O facto de os dados do problema serem incertos leva a que a avaliação da aptidão de um indivíduo esteja sujeita a perturbações. Estas perturbações podem ter diferentes origens, tais como erros nas medições sensoriais ou simulações aleatórias.

Matematicamente, uma função de aptidão perturbada pode ser formulada da seguinte forma (Jin e Branke (2005)):

$$F(x)=\int_{-\infty}^{\infty} \left(f(x)+\mu\right) p(\mu)\, d\mu = f(x), \quad \mu \sim N\left(0,\sigma^2\right) \tag{1}$$

em que x é um vector de parâmetros que podem ser alterados pelo algoritmo (muitas vezes designado como *variáveis de projecto*), $f(x)$ é uma função de aptidão sem variação temporal e μ é a perturbação adicionada (que muitas vezes se considera seguir uma distribuição normal de média 0 e variância σ^2).

Também se pode considerar uma perturbação não Gaussiana, como é o caso da perturbação distribuída de Cauchy (Arnold e Beyer (2003)). No entanto, não se observou qualquer diferença qualitativa no desempenho de uma estratégia evolucionária na presença de perturbação Gaussiana ou de Cauchy (Jin e Branke (2005)).

O ideal seria os algoritmos evolucionários trabalharem com a função de aptidão esperada $F(x)$ e não serem "enganados" devido à presença de perturbações. No entanto, durante a optimização, o único valor de aptidão mensurável é o estocástico $f(x) + \mu$. Na prática, porém, a função de aptidão esperada (1) é muitas vezes aproximada por uma média de um número de amostras aleatórias

$$\hat{F}(x) = \frac{1}{N}\sum_{i=1}^{N}\left(f(x)+\mu_i\right) \tag{2}$$

em que N é o tamanho da amostra e $\hat{F}(x)$ é uma estimativa de $f(x)$.

A função de aptidão perturbada é usada em muitas abordagens evolucionárias, em áreas tão diferentes como, por exemplo, redes neuronais, robótica e optimização de processos.

A aplicação de algoritmos evolucionários em ambientes com perturbação tem sido o tema de muitos artigos de investigação. Pode-se encontrar uma análise teórica mais detalhada da influência da perturbação no desempenho das estratégias evolucionárias em Arnold (2002) e Beyer (2000), assim como em Arnold (2001) que apresenta um breve

resumo dos trabalhos existentes. Existe também disponível na internet uma lista de referências a artigos relacionados com este tópico, com actualização permanente, no sítio: http://www.soft-computing.de/noise_robust.html.

Em termos metodológicos, foram adoptadas as seguintes abordagens para tratar as funções de aptidão perturbadas (Jin e Branke (2005)): *cálculo explícito da média*, *cálculo implícito da média* e *modificação do mecanismo de selecção*.

2.1.1. Cálculo explícito da média

Uma abordagem usual para reduzir a influência da perturbação é estimar a aptidão através do cálculo da média de um número de amostras obtidas ao longo do tempo (*cálculo da média no tempo*). Aumentar o tamanho da amostra é equivalente a reduzir a variância da aptidão estimada. Em abordagens simples, o número de unidades da amostra (tamanho da amostra) para cada indivíduo é predefinido e constante. Uma vez que cada avaliação da amostra (cálculo da aptidão) é muito dispendiosa, é preferível reduzir o tamanho da amostra o mais possível sem, no entanto, degradar o desempenho dos algoritmos associados a estas abordagens.

Aizawa e Wah (1993) e Aizawa e Wah (1994) foram dos primeiros a sugerir que o tamanho da amostra deveria ser adaptado durante a execução do algoritmo, propondo dois tipos de adaptação: aumentar o tamanho da amostra em função do número de gerações e usar o tamanho da amostra mais elevado para os indivíduos com maior variância estimada.

Branke e Schmidh (2003) e Branke e Schmidh (2004) propuseram também abordagens de adaptação dos tamanhos das amostras, que consistem em tomar várias amostras dos indivíduos que participam na selecção por torneio até que a diferença normalizada do valor de aptidão entre estes dois indivíduos (calculada dividindo a diferença dos valores de aptidão obtidos pelo desvio padrão da diferença) esteja abaixo de um certo limiar. Desta forma, os tamanhos das amostras vão sendo reduzidos até um valor mínimo requerido que permita fazer a distinção entre os indivíduos que participam na selecção por torneio.

Cantú-Paz (2004) propôs uma abordagem semelhante às de Branke e Schmidh, que se distinguem por tomar, uma de cada vez, as amostras do indivíduo com maior variância observada e por usar testes estatísticos padrão para seleccionar o vencedor do torneio com certa probabilidade.

Visto que a redução do tamanho da amostra leva também à redução do custo computacional do cálculo da média no tempo, uma abordagem alternativa é calcular a aptidão através do cálculo da média na vizinhança do ponto a ser avaliado (*cálculo da média no espaço*) (Branke et al. (2001), Sano e Kita (2000), Sano e Kita (2002) e Sano

et al. (2000)). Para este efeito, têm também sido construídos modelos de pesquisa local. Os motivos para substituir o cálculo da média no tempo pelo cálculo da média no espaço, é que a perturbação na vizinhança tem as mesmas características da perturbação no ponto a ser avaliado e o relevo da função de aptidão é localmente regular.

2.1.2. Cálculo implícito da média

Uma vez que as áreas promissoras do espaço de pesquisa são testadas repetidamente pelo algoritmo evolucionário e existem, normalmente, muitas soluções semelhantes na população, quando o tamanho da população é elevado é muito provável que a influência da perturbação na avaliação de um indivíduo seja compensada pela perturbação de um indivíduo semelhante. Este efeito pode ser visto como um tipo de *cálculo implícito da média*. Desta forma, uma abordagem simples para reduzir a influência da perturbação na optimização é usar uma população de grande dimensão (Fitzpatrick e Grefenstette (1988)). Miller e Goldberg (1996) mostraram que quando o tamanho da população é infinito, a selecção proporcional à aptidão não é afectada pela perturbação. Rattray e Shapiro (1997) estudaram a utilização de algoritmos genéticos com população de tamanho finito e mostraram que o aumento do tamanho da população reduz os efeitos da perturbação Gaussiana na selecção de Boltzmann[10].

Foram também desenvolvidos modelos teóricos que permitem optimizar, simultaneamente, o tamanho da população e o tamanho da amostra (Miller e Goldberg (1996) e Miller (1997)), os quais são baseados no trabalho de Goldberg et al. (1992).

2.1.3. Selecção modificada

Muitos autores têm sugerido modificar o processo de selecção de forma a combater a perturbação. Um exemplo é impor, nas estratégias evolutivas, um limiar durante a selecção determinística (Markon et al. (2001)), isto é, um descendente será aceite se e só se a sua aptidão for melhor do que as dos seus pais em, pelo menos, um limiar predefinido.

A incerteza causada pela perturbação tem sido também estudada em optimização multi-objectivo, na qual é usada a dominância de Pareto na selecção dos indivíduos.

Teich (2001) propôs substituir a ordenação de Pareto de um indivíduo pela sua probabilidade de ser dominado. Para tal, o autor estendeu técnicas de exploração do espaço baseadas na dominância de Pareto, a casos onde uma ou várias das funções objectivo estão sujeitas a incertezas modeladas por intervalos.

[10] O mecanismo de selecção de Boltzmann caracteriza-se por ser aleatório e com distribuição proporcional.

Hughes (2001) propôs também substituir a ordenação de Pareto de um indivíduo pela sua probabilidade de ser dominado. O autor calculou o erro esperado a ser usado na dominância de Pareto determinístico, de acordo com a perturbação nos dados associados às funções objectivo.

2.1.4. Casos relacionados

Têm sido estudados importantes casos relacionados com a pesquisa eficiente, a convergência e a auto-adaptação, tanto de algoritmos evolucionários como de outros algoritmos heurísticos de pesquisa, na resolução de problemas sujeitos a perturbações. Baum et al. (1995) provaram que, para uma classe de problemas simples sujeitos a perturbações, os Algoritmos Genéticos têm melhor desempenho do que muitos métodos de pesquisa local. Arnold e Beyer (2003) compararam as estratégias evolucionárias com outras meta-heurísticas de pesquisa local e mostraram que as estratégias evolucionárias têm clara vantagem em ambientes perturbados.

Enquanto que a maioria dos investigadores se concentrou na redução da influência da perturbação nas avaliações da aptidão, alguns artigos descrevem que a perturbação, até um certo nível, pode ajudar a melhorar o desempenho dos Algoritmos Evolucionários e de outros métodos de pesquisa heurísticos, porque permitem que os algoritmos saiam de óptimos locais.

As funções de aptidão perturbadas têm também sido investigadas com outros tipos de meta-heurísticas, como é o caso do arrefecimento simulado, colónia de formigas e pesquisa tabu (ver Jin e Branke (2005)).

Goh e Tan (2006) investigaram as consequências da perturbação em algoritmos evolucionários multi-objectivo, dando particular atenção a populações com aptidão e diversidade dinâmicas. Partindo de investigações empíricas, os autores propuseram três características para a manipulação da perturbação, de forma a melhorar a robustez e o desempenho dos algoritmos evolucionários multi-objectivo: um operador de perturbação dirigido pelo conhecimento baseado na experiência, uma estratégia de selecção e adaptação do gene, e um modelo de armazenamento possibilístico. O operador adapta a magnitude e a direcção da variação de acordo com as experiências do passado para acelerar a convergência. A estratégia melhora a pesquisa evolucionária para fugir à convergência prematura em ambientes com perturbação. O modelo de armazenamento possibilístico é baseado no conceito de medidas de possibilidades e necessidades (usando conjuntos difusos) para tratar problemas com incertezas. Neste estudo, a perturbação adicionada é implementada como uma distribuição normal de média zero, no espaço dos objectivos. Isto é, $F(x) = f(x) + N(0,\sigma^2)$, onde σ^2 representa o nível de perturbação

presente, N denota a função de distribuição normal, F(x) e f(x) denotam a função objectivo com e sem perturbação adicionada, respectivamente.

2.2. Função de aptidão aproximada

Quando o cálculo da função de aptidão é muito dispendioso ou não está disponível uma função de aptidão analítica, as funções de aptidão são muitas vezes aproximadas e baseadas nos dados gerados pelas experiências ou simulações. As funções de aptidão aproximadas são também conhecidas como *meta-modelos* ou *substitutas*. Como sugeriu Jin e Sendhoff (2002), um meta-modelo deveria ser usado, normalmente, juntamente com a função de aptidão original. Neste caso, a função de aptidão a ser optimizada pelos algoritmos evolucionários tornar-se-ia (Jin e Branke (2005))

$$F(x) = \begin{cases} f(x), & \text{se é usada a função de aptidão original} \\ f(x)+E(x), & \text{se é usado um meta}-\text{modelo} \end{cases} \quad (3)$$

em que E(x) é o erro de aproximação do meta-modelo.

A diferença mais importante entre uma função de aptidão perturbada e uma função de aptidão aproximada é que o erro na função de aptidão aproximada é determinístico (uma vez que o meta-modelo foi idealizado) e regular (isto é, com média diferente de zero). No entanto, o erro não pode ser reduzido recalculando a função de aptidão aproximada. Em vez disso, o erro tem de ser tratado usando a verdadeira função de aptidão em vez da aproximada. Desta forma, o desafio é encontrar o equilíbrio certo entre as avaliações com a função aptidão aproximada, que são de baixo custo mas com erros, e com a função de aptidão verdadeira, que são dispendiosas mas precisas.

Nos últimos anos, tem-se assistido, na computação evolucionária, a um aumento de interesse na utilização de funções de aptidão aproximada. Para um estudo mais aprofundado acerca deste tópico, consultar Jin e Sendhoff (2002) e Jin (2005). Está também disponível na Internet uma lista de referências a artigos relativos a este tópico, em http://www.soft-computing.de/amec.html.

O uso de uma função de aptidão aproximada é motivado, principalmente, pelas seguintes razões (Jin e Branke (2005)):

- Cada avaliação da aptidão é extremamente demorada.
- Não é fácil encontrar disponível uma função de aptidão analítica total. Na computação evolucionária, a estimação da aptidão é uma consequência inerente ao facto de um indivíduo codificar apenas parte da solução e, desta forma, a qualidade do indivíduo não poder ser calculada com exactidão.
- Necessidade de avaliações adicionais da aptidão, por exemplo, no tratamento com funções de aptidão perturbadas ou na procura de soluções robustas. Os modelos

aproximados são muito úteis, no sentido de reduzir o custo computacional destas avaliações adicionais da aptidão.

- O relevo da função de aptidão é irregular. A hipótese básica é que se pode construir um meta-modelo global, capaz de aplanar os óptimos locais da função de aptidão original, de relevo irregular, sem alterar a localização do óptimo global.

2.2.1. Métodos de aproximação

Em computação evolucionária, podem ser adoptados vários níveis ou estratégias de aproximação para chegar à aptidão aproximada: *aproximação ao problema*, *aproximação funcional conduzida pelos dados*, e *herança, imitação e afectação da aptidão*.

A *aproximação ao problema* tenta substituir o enunciado original do problema por um que seja aproximadamente o mesmo, mas mais fácil de resolver.

Na *aproximação funcional*, é construída uma expressão alternativa e explícita para a função objectivo *baseada nos dados*, descrevendo a correspondência entre as variáveis de decisão e a qualidade da decisão. Neste caso, os modelos obtidos a partir dos dados são, muitas vezes, conhecidos como meta-modelos ou substitutos.

A *herança*, *imitação* e *afectação da aptidão* é um tipo de abordagem específica dos algoritmos evolucionários que consiste em guardar as avaliações da aptidão, em que a aptidão de um indivíduo é estimada a partir de outros indivíduos semelhantes. Na subclasse desta categoria conhecida como *herança da aptidão*, a aptidão é simplesmente herdada a partir dos pais. Na *imitação da aptidão*, os indivíduos são reunidos em vários grupos, em que apenas o indivíduo que representa cada grupo será avaliado usando a função de aptidão. Os valores da aptidão dos restantes indivíduos do grupo são estimados a partir do indivíduo representativo do grupo, com base numa medida de distância.

Nos algoritmos co-evolucionários de dois níveis (com duas populações), os indivíduos de uma das duas populações codificam apenas parte do problema e os seus valores de aptidão dependem sempre dos valores dos indivíduos da outra população. Para este caso, foram desenvolvidos métodos para estimar os valores de aptidão em sistemas co-evolucionários colaborativos, os quais são conhecidos como métodos de *afectação da aptidão*.

2.2.2. Mecanismos para incorporar meta-modelos

Os modelos que incorporam meta-modelos podem levar vantagem em quase todos os elementos dos algoritmos evolucionários, incluindo iniciação, cruzamento (recombinação), mutação e avaliação da aptidão (Jin e Branke (2005)):

- Utilização de modelos de aptidão aproximada na população inicial (Rasheed (2000)).
- Utilização de modelos de aptidão aproximada para reduzir a aleatoriedade nos operadores genéticos cruzamento e mutação (Abboud e Schoenauer (2002), Anderson e Hsu (1999) e Mutoh et al. (2003)).
- Utilização de modelos de aptidão aproximada nas avaliações da aptidão. Na maioria dos trabalhos de investigação, o modelo aproximado tem sido usado directamente nas avaliações da aptidão de forma a reduzir o número de cálculos da aptidão (Emmerich et al. (2002), El-Beltagy (1999), Jin et al. (2000) e Jin et al. (2002)). As avaliações da aptidão aproximada têm sido também aplicadas em optimização evolucionária multi-objectivo (Chafekar et al. (2004), Chung e Alonso (2004), Farina (2000) e Farina (2002)).

2.2.3. Controlo da evolução ou gestão de meta-modelos

A motivação fundamental para o uso de meta-modelos nas avaliações da aptidão é reduzir o elevado número destas avaliações, sem degradar a qualidade da solução final obtida. Para alcançar esta meta, os meta-modelos deverão ser combinados com a própria função de aptidão original, o que muitas vezes é conhecido como *controlo da evolução* ou *gestão do modelo*.

Muitas vezes, supõe-se que o modelo aproximado é de elevada precisão e, assim, a função de aptidão original não é totalmente usada no processo evolucionário. No entanto, a evolução que usa meta-modelos pode correr o risco de uma incorrecta convergência, se não controlar a evolução que usa a função de aptidão verdadeira (Jin et al. (2000)). Desta forma, deve-se dar maior atenção às estratégias de gestão dos modelos que podem controlar a evolução convenientemente.

As metodologias existentes para o controlo da evolução podem ser divididas, de um modo geral, em duas categorias: *baseadas no indivíduo* e *baseadas na geração*.

O *controlo da evolução baseado no indivíduo* consiste em controlar a evolução de forma que, na mesma geração, alguns indivíduos usem os meta-modelos para determinar os seus valores de aptidão e os restantes usem a verdadeira função de aptidão para o mesmo fim (ver Jin et al. (2000)). A questão principal destas metodologias é determinar que indivíduos devem usar o meta-modelo e quais devem usar a verdadeira função de aptidão.

No *controlo da evolução baseado na geração*, todos os indivíduos de uma mesma geração são avaliados, ou pela verdadeira função de aptidão, ou pelo meta-modelo. A questão principal destas metodologias é determinar em que gerações deve ser usada a verdadeira função de aptidão e em quais deve ser usado o meta-modelo.

Tem sido desenvolvida uma grande variedade de metodologias para controlo da evolução baseado no indivíduo, cujo passo comum consiste em pré-avaliar todos os indivíduos usando o meta-modelo, para depois se escolher um número de indivíduos para serem reavaliados (controlados) usando a função de aptidão verdadeira.

Estas metodologias podem ser classificadas de acordo com o tipo de escolha sobre os indivíduos a reavaliar, nas seguintes classes (Jin e Branke (2005)):

- *Escolha aleatória dos indivíduos*. Neste método, é seleccionado aleatoriamente um número de indivíduos da população. Neste caso, não é necessária uma pré-avaliação usando o meta-modelo.
- *Escolha dos melhores indivíduos de acordo com a pré-avaliação feita pelo meta-modelo*. Se for assumido que é melhor fazer-se a escolha usando o meta-modelo do que aleatoriamente, o que é uma hipótese muito débil pela qualidade do meta-modelo, é natural escolher-se os melhores indivíduos (isto é, os mais promissores) para serem reavaliados usando a função de aptidão verdadeira (Jin et al. (2000), Jin et al. (2002)). Normalmente, o número de indivíduos a serem reavaliados é predefinido e constante durante o processo evolucionário.
- *Escolha dos indivíduos mais incertos*. Por um lado, ao reavaliar os indivíduos mais incertos, a incerteza introduzida pela utilização dos meta-modelos poderá ser reduzida o máximo possível. Por outro lado, os indivíduos com elevado grau de incerteza estão muitas vezes localizados em áreas não exploradas. Desta forma, o uso deste tipo de selecção incentiva a pesquisa em áreas ainda não testadas. Uma medida de incerteza pode muitas vezes ser obtida calculando a distância do indivíduo considerado ao ponto mais próximo necessário à criação do meta-modelo. Uma outra medida pode ser escolher os indivíduos com as maiores probabilidades de melhoramento, em vez dos indivíduos com os melhores valores de aptidão. Pode também ser usada uma combinação dos critérios qualidade e incerteza na escolha dos indivíduos a serem reavaliados (Branke e Schmidh (2005)).
- *Métodos de pré-selecção*. Se o tamanho da população é POP, então o número de indivíduos escolhidos para serem reavaliados (POP_R) é, normalmente, menor ou igual a POP. No entanto, isto não acontece nos métodos denominados por *pré-selecção*, onde o número de descendentes gerados na estratégia evolucionária (POP_P) é maior do que POP. Estes POP_P indivíduos são todos avaliados usando o meta-modelo, sendo depois escolhidos POP para reavaliação usando a função de

aptidão verdadeira de acordo com uma combinação dos critérios qualidade e incerteza.

O método de pré-selecção é basicamente igual aos métodos que usam o meta-modelo para reduzir a aleatoriedade dos operadores genéticos cruzamento e mutação.

Um outro método interessante é aquele que consiste em escolher os indivíduos mais "representativos" para reavaliação, usando a verdadeira função de aptidão de acordo com a localização dos indivíduos. A abordagem básica consiste em agrupar os indivíduos da população num certo número de agrupamentos e depois escolher os indivíduos mais próximos do centro do respectivo agrupamento como seu representante (Jin e Sendhoff (2004)). Uma vantagem destes métodos é que pode ser obtido um bom equilíbrio entre pesquisa em regiões não testadas e pesquisa nas vizinhanças ("*exploration*" vs. "*exploitation*").

As metodologias de controlo da evolução baseadas na geração podem ser classificados de acordo com o tipo de frequência do controlo, nas seguintes classes (Jin e Branke (2005)):

- *Frequência do controlo constante*. O controlo da evolução baseado na geração tem sido, muitas vezes, usado de tal modo que a função de aptidão verdadeira é aplicada em cada k gerações, onde k é um valor predefinido e constante durante o processo evolucionário. Não é muito prático usar a frequência do controlo constante, porque a precisão do modelo aproximado pode variar significativamente durante o processo de optimização. De facto, a predefinição da frequência do controlo da evolução pode causar fortes oscilações durante o processo de optimização devido ao grande número de erros do modelo.

  Uma outra abordagem consiste em executar a optimização no meta-modelo até o processo evolucionário convergir. A solução de convergência é então reavaliada usando a função de aptidão verdadeira. Um inconveniente deste método é que o processo evolucionário pode facilmente convergir para um óptimo local.
- *Frequência do controlo adaptativa*. Em geral, a frequência do controlo da evolução deve depender da precisão do modelo aproximado. Foi sugerido por Nair e Keane (1998) um método para ajustar a frequência do controlo da evolução, baseado em metodologias de regiões promissoras, na qual é usada a abordagem baseada na geração. Foi também sugerido em Jin et al. (2001-b) uma metodologia para gestão de modelos aproximados, a qual foi aplicada a um problema de optimização bidimensional na área da aerodinâmica (Jin e Sendhoff (2002)).

Jin et al. (2000) efectuaram um estudo comparativo das estratégias de controlo da evolução, no qual usaram duas estratégias baseadas no indivíduo e uma baseada na geração. Estes autores verificaram que é mais eficaz escolher os melhores indivíduos de acordo com a aptidão aproximada para a reavaliação usando a função de aptidão verdadeira, do que os escolher aleatoriamente. Por outro lado, na escolha dos melhores indivíduos, verificaram que ambas as estratégias apresentam desempenhos semelhantes.

Jin e Sendhoff (2004) mostraram ser vantajoso, em comparação com as melhores estratégias, usar-se uma estratégia baseada no indivíduo na qual os indivíduos são escolhidos para reavaliação através de um algoritmo de agrupamentos.

Branke e Schmidh (2005) estudaram o uso de dois esquemas de aproximação local padrão, nomeadamente a interpolação e a regressão, os quais se baseiam nos indivíduos vizinhos já avaliados. Em cada geração, uma percentagem fixa da população é avaliada com a função objectivo real, enquanto que a aptidão dos restantes indivíduos é estimada. Os indivíduos a serem avaliados com exactidão são determinados de acordo com os seus valores de aptidão estimados e a incerteza. Os testes efectuados mostraram que este esquema de aproximação acelera a convergência, produzindo uma solução com a mesma qualidade mas com significativamente menos avaliações da função de aptidão verdadeira.

2.2.4. Convergência

Embora a optimização evolucionária que usa funções de aptidão aproximadas tenha tido sucesso em vários trabalhos de investigação, ainda falta fazer uma análise rigorosa à convergência destes métodos. Jin et al. (2000) orientaram um estudo empírico para a convergência dos algoritmos evolucionários, o qual mostrou que pode ser observada consistentemente uma correcta convergência, se forem efectuadas mais de 50% das avaliações da aptidão com a verdadeira função de aptidão. Também mostraram que as estratégias evolutivas convergem correctamente, na maioria dos casos, quando são controlados apenas 30% dos indivíduos escolhidos de acordo com um algoritmo de agrupamentos (Jin e Sendhoff (2004)).

2.3. Ambientes dinâmicos

Em problemas de optimização dinâmicos mono-objectivo, a função de aptidão apesar de ser determinística em qualquer instante, depende do momento t; isto é,

$$F(x) = f_t(x) \tag{4}$$

Como consequência, também o óptimo se altera ao longo do tempo (os ambientes são dinâmicos).

Da mesma maneira, em problemas de optimização dinâmicos multi-objectivo, é de esperar que a frente óptima de Pareto resultante também se altere com o tempo (ou com as iterações do processo evolucionário).

Desta forma, o algoritmo evolucionário deve ser capaz de acompanhar, continuamente, a variação do óptimo (mono-objectivo) ou a frente óptima de Pareto (multi-objectivo), em vez de exigir repetidamente o reinício do processo de optimização. O desafio é reutilizar a informação vinda dos ambientes anteriores para acelerar o processo de optimização após ocorrer uma alteração.

2.3.1. Considerações gerais

Em muitos problemas de optimização do mundo real, a função objectivo, a instância do problema, ou as restrições podem alterar-se em qualquer instante e, assim, o óptimo daquele problema pode também variar. Se alguma destas entidades incertas for considerada no processo de optimização, então diz-se que é um *problema dinâmico* (Jin e Branke (2005)).

A forma mais simples de reagir a uma alteração do ambiente é olhar para cada alteração como se tratasse da chegada de um novo problema de optimização, que tem de ser resolvido imediatamente. Se houver tempo suficiente, esta é certamente uma alternativa viável. No entanto, muitas vezes o tempo para reoptimização é muito curto, e uma abordagem de reinício explícito assume que tem de ser identificada a entidade alterada, o que nem sempre é o caso.

Uma tentativa natural para acelerar a optimização após ocorrer uma alteração, poderá consistir em usar, de modo adequado, o conhecimento acerca do espaço de pesquisa anterior para avançar na pesquisa após a alteração. Se, por exemplo, for possível assumir-se que a nova solução final está "próxima" da antiga, será certamente benéfico restringir a procura à vizinhança da solução final antiga. Se é ou não vantajoso reutilizar a informação que vem do passado, tal depende da natureza da alteração. Se a alteração é radical e o novo problema tem poucas semelhanças com o anterior, então o reinício do processo poderá ser a única opção viável, e qualquer reutilização da informação recebida do antigo problema poderá ser prejudicial em vez de ajudar na pesquisa. No entanto, espera-se que, para a maioria dos problemas do mundo real, as alterações sejam muito ligeiras e, assim, seja possível ganhar-se um pouco ao transferir conhecimento do passado. A questão difícil de resolver é saber que informação deve ser guardada e como deve ser usada para acelerar a procura, após a alteração do ambiente.

Mas mesmo quando pode ser transferida informação útil, tem que ser garantido que o algoritmo de optimização é suficientemente flexível para responder às alterações. A maioria das meta-heurísticas converge durante a execução, pelo menos nos instantes em

que o ambiente é estático, perdendo assim a sua capacidade adaptativa. Assim, e apesar da transferência de conhecimento, uma boa meta-heurística para problemas de optimização dinâmica tem de permitir manter a adaptabilidade.

2.3.2. Tratamento de problemas de optimização dinâmica com AE

As primeiras aplicações de Algoritmos Evolucionários em ambientes dinâmicos foram descritas em Fogel et al. (1966). No entanto, foi só a partir do final da década de 1980 que este tópico se tornou um centro de interesse para muitos investigadores, tendo então surgido um grande número de publicações. Embora existam também outras meta-heurísticas de inspiração evolucionária aplicadas a problemas de optimização dinâmica, os Algoritmos Evolucionários são os mais utilizados.

Normalmente, a informação vinda do espaço de pesquisa anterior é transferida simplesmente guardando os indivíduos da população. Nalguns casos, quando a dimensão do problema se altera, os indivíduos têm de ser adaptados após a alteração ter ocorrido. Apesar destas necessárias alterações do genótipo, verificam-se melhorias significativas na velocidade de convergência e na qualidade das soluções, quando os indivíduos alterados (antigos) são reutilizados (Lin et al. (1997), Bierwirth e Mattfeld (1999)).

Na construção das gerações nos Algoritmos Evolucionários não é problemático a informação da aptidão estar desactualizada, devido às alterações do problema, porque geralmente nenhum indivíduo sobrevive para além de uma geração (todos os progenitores são substituídos pelos seus descendentes). Se os pais competissem com os filhos pela sobrevivência, uma possibilidade seria reavaliar todos os antigos indivíduos após ocorrer uma alteração (Smith e Vavak (1999)).

As principais abordagens que conduzem à convergência podem ser agrupadas nas quatro categorias seguintes (Jin e Branke (2005)):

- *Gerar diversidade após a alteração.* O Algoritmo Evolucionário é executado normalmente mas, logo que é detectada uma alteração no ambiente, são tomadas acções explícitas para aumentar a diversidade e, assim, facilitar a deslocação para a nova solução final. Uma representação típica desta abordagem é a hiper-mutação, onde a probabilidade de mutação é aumentada bruscamente por um grande número de gerações após o ambiente ter sido alterado. Na pesquisa local variável, a probabilidade de mutação é aumentada gradualmente. O problema destas abordagens é que aumentar a diversidade é basicamente equivalente a substituir informação dos indivíduos eleitos anteriormente por informação aleatória. Desta forma, é difícil determinar uma quantidade útil de diversidade: enquanto demasiada assemelha-se a reinício, muito pouca não resolve o problema da convergência.

- *Manter a diversidade durante toda a execução.* A convergência é evitada durante todo o tempo e é esperado que uma população alargada possa adaptar-se às alterações mais facilmente. Os mecanismos de partilha ou de multidão têm um modo mais elaborado de assegurar a diversidade. Embora existam algumas abordagens muito interessantes para este caso, a excessiva focalização na diversidade atrasa o processo de optimização.
- *Abordagens baseadas na memória.* O Algoritmo Evolucionário é provido de uma memória para ser capaz de retirar informação útil das gerações anteriores, o que parece especialmente útil quando a pesquisa regressa repetidamente aos locais anteriores. Este tipo de abordagem pode ser dividido em *memória explícita* e *memória implícita*. No caso de *memória explícita*, são definidas estratégias específicas para armazenar e recuperar informação. Nas abordagens com *memória implícita*, o Algoritmo Evolucionário é simplesmente equipado com uma representação redundante da qual fará o uso conveniente.

  Como foi notado por Branke (1999), e depois confirmado por outros investigadores, a memória é muito dependente da diversidade e deverá, assim, ser usada em combinação com as técnicas de preservação da diversidade.
- *Abordagens com várias populações.* A divisão da população em várias sub-populações permite localizar múltiplos picos no traçado da função de aptidão. Desta forma, as diferentes sub-populações mantêm informação acerca de várias regiões promissoras do espaço de pesquisa, podendo este processo ser visto como um tipo de memória diversa e auto-adaptativa.

Descrições mais detalhadas sobre Algoritmos Evolucionários aplicados a ambientes dinâmicos podem ser encontradas em Branke (2001-a), Morrison (2004) e Weicker (2003). Encontra-se também disponível uma vasta lista de referências sobre este tipo de abordagem em http://www.aifb.uni-karlsruhe.de/~jbr/EvoDOP.

2.3.3. Algumas abordagens

Enquanto que a maioria dos trabalhos iniciais eram de natureza empírica, ultimamente os autores tentam olhar para o problema segundo um ponto de vista teórico.

A primeira abordagem onde o problema é alvo de um estudo teórico encontra-se em Stanhope e Daida (1999), no qual foram derivadas as equações das probabilidades de transição de um AE (1+1) para o problema de *"dynamic bit matching"*. Branke e Wang (2003) também consideraram este problema, e compararam analiticamente diferentes estratégias para tratar as alterações ambientais em *uma* geração (em vez de ser *entre* duas gerações) baseadas em métodos semelhantes, como em Stanhope e Daida (1999).

Bui et al. (2005) estudaram o uso de métodos de optimização evolucionária multi-objectivo, os quais são altamente eficientes na preservação da diversidade genética e na resolução de problemas de optimização mono-objectivo em ambientes dinâmicos. Um grande número de autores propuseram o uso de optimização evolucionária multi-objectivo para manterem a diversidade em problemas de optimização mono-objectivo, em que transformam o problema original (mono-objectivo) num outro de optimização multi-objectivo, adicionando uma função objectivo artificial. Na formulação do problema como multi-objectivo, usaram-se duas funções objectivo: a primeira é o objectivo original dinâmico e a segunda é um objectivo artificial para promover a diversidade (foram examinados seis possibilidades diferentes: dependentes do tempo, gerar valores aleatórios, inverter o sentido da optimização do objectivo, distância ao vizinho mais próximo, distância média para todos os indivíduos da população e distância para o melhor indivíduo da população). Os autores usaram o NSGA-II (ver secção 5.3.2.2 do capítulo 2). Todos os resultados foram comparados com um AG tradicional e duas variantes clássicas para ambientes dinâmicos (imigrantes aleatórios e uma variação do algoritmo de hiper-mutação). Os testes demonstraram que as abordagens propostas são competitivas, e que as que usaram como objectivo artificial a distância para todos os indivíduos e a distância para os melhores indivíduos têm muito melhor desempenho quando comparadas com as outras.

Surgiram, entretanto, também alguns estudos envolvendo outras meta-heurísticas aplicadas a problemas de optimização dinâmica. Por exemplo, a meta-heurística de colónia de formigas foi usada para resolver problemas de optimização do tipo caixeiro-viajante dinâmico (Guntsch e Middendorf (2000), Guntsch et al. (2000) e Guntsch et al. (2001)), e a optimização por enxames de partículas foi aplicada em ambientes dinâmicos (Kennedy e Eberhart (2001) e Mendes et al. (2004)).

Farina et al. (2004) desenvolveram alguns problemas de teste para optimização multi-objectivo dinâmica, tanto contínua como discreta, assim como sugeriram um algoritmo para resolver estes problemas. Estes problemas de teste foram construídos a partir de problemas de teste para optimização multi-objectivo estática (problemas em que os respectivos dados se mantêm constantes ao longo de todo o processo de optimização) e de parâmetros dependentes do tempo. Estes autores também sugeriram um algoritmo básico para procurar resolver tais problemas.

Neste artigo, também é discutido um problema de controlo adaptativo de um sistema temporal, onde o controlador final é dependente do tempo, uma vez que as propriedades do sistema são dependentes do tempo. Além disso, são considerados múltiplos objectivos para a optimização do controlador dinâmico, assim como é fornecida uma formulação para o problema de optimização multi-objectivo dinâmico resultante.

2.4. Robustez

A procura de soluções óptimas robustas é de grande importância para um grande número de aplicações reais (Paenke et al. (2006)):

- no processo de fabricação — como normalmente é impossível produzir um item exactamente conforme as especificações planeadas, o planeamento tem de permitir tolerâncias na produção (Greiner (1996) e Wiesmann et al. (1998));
- no escalonamento — um escalonamento deve ser capaz de tolerar pequenos desvios nos tempos de processamento estimados, ou ser capaz de adaptar as máquinas que falham (Jensen (2003) e Leon et al. (1994));
- no planeamento de circuitos — os circuitos devem trabalhar sob uma ampla gama de condições ambientais, tais como sob diferentes temperaturas (Thompson (1996));
- no planeamento das pás de turbinas — a turbina deve funcionar bem perante uma gama de condições; por exemplo, deve trabalhar com eficiência em diferentes velocidades (Ray e Tsai (2004) e Yamaguchi e Arima (2002)).

A maioria das abordagens evolucionárias para pesquisar soluções robustas consiste em optimizar a função de aptidão esperada, dada uma distribuição de probabilidade de perturbações. Apenas alguns estudos consideram o problema como multi-objectivo, consistindo a maioria deles em optimizar, separadamente, duas funções objectivo: o desempenho e a robustez. Segundo Jin e Branke (2005), a análise de robustez em optimização evolucionária pode ser dividida nas seguintes duas classes de abordagens: *optimização da função esperada* e *abordagens multi-objectivo*.

2.4.1. Optimização da função de aptidão esperada

Como as abordagens que usam esta estratégia são aplicadas apenas a problemas com um só objectivo, são também denominadas por optimização robusta mono-objectivo (Paenke et al. (2006)).

Apenas se vai estudar os casos em que as perturbações incidem sobre as variáveis de decisão do modelo. Formalmente, se x denota um vector de variáveis de decisão (solução) de dimensão n e $f(x)$ é a função de aptidão, então a *função de aptidão esperada* é definida da seguinte forma (Paenke et al. (2006)):

$$f_{esp}(x) = \int_{-\infty}^{\infty} f(x+\delta)\, p(\delta)\, d\delta \qquad (5)$$

onde δ é uma perturbação que é distribuída de acordo com a função de densidade com probabilidade $p(\delta)$ — $f_{esp}(x)$ é também denominada por *função de aptidão efectiva* (Tsutsui et al. (1996)).

Do mesmo modo, a *variância da função de aptidão* pode ser definida da seguinte forma (Paenke et al. (2006)):

$$f_{var}(x) = \int_{-\infty}^{\infty} \left(f(x+\delta) - f_{esp}(x) \right)^2 p(\delta)\, d\delta. \quad (6)$$

Para problemas razoavelmente complexos, a função de aptidão esperada (5) e a variância da função de aptidão (6) não podem ser calculadas analiticamente, normalmente porque a função de aptidão f(x) não tem disponível uma forma analítica fechada. Como alternativa, $f_{esp}(x)$ e $f_{var}(x)$ podem ser estimadas usando a simulação de Monte Carlo, isto é, construindo amostras sobre um certo número NR de realizações de δ. Por exemplo, para a função de aptidão esperada $f_{esp}(x)$ a correspondente simulação de Monte Carlo pode ser a seguinte (Jin e Branke (2005)):

$$\hat{f}_{esp}(x) = \frac{1}{NR} \sum_{i=1}^{NR} f(x+\delta_i). \quad (7)$$

No entanto, como cada amostra corresponde a uma avaliação da aptidão, esta abordagem não é viável, quando aquelas avaliações são muito dispendiosas (em termos de tempo de execução).

Apesar desta expressão (7) se parecer muito com a descrita em (2), para a função de aptidão perturbada, e destes dois casos serem muito semelhantes, existem algumas diferenças significativas. Enquanto que no caso da função de aptidão perturbada se supõe, normalmente, que a perturbação incide sobre os valores de aptidão ($f(x) + \delta_i$), na pesquisa de soluções robustas a incerteza encontra-se nas variáveis de decisão ($x + \delta_i$).

Uma solução óptima robusta não é necessariamente um óptimo de f(x), mas existe, normalmente, um compromisso entre a qualidade e a robustez da solução.

Por outro lado, a dificuldade dos métodos de pesquisa de soluções robustas está em estimar o integral da equação (5) sobre todas as perturbações possíveis, uma vez que se supõe que a função de aptidão é conhecida e determinística, e que a incerteza é avaliada apenas *após* o processo de optimização ter terminado. Neste caso, é possível escolher adequadamente as perturbações δ usadas na aproximação em (7), as quais têm consequências nas estratégias da solução.

Como, na maioria dos casos, a formulação matemática da função de aptidão esperada (5) não está disponível numa forma fechada, o valor da função de aptidão esperada de um indivíduo tem de ser estimado baseado em f(x). Isto é muito semelhante à estimação da função de aptidão em ambientes perturbados, o que implica que podem ser aplicadas abordagens semelhantes: *cálculo explícito da média* e *cálculo implícito da média*.

2.4.1.1. Cálculo explícito da média

A forma mais simples de estimar a função de aptidão esperada é através da simulação de Monte Carlo, isto é, através da média dos valores da função de aptidão sobre um número de amostras de perturbações geradas aleatoriamente na vizinhança da solução x a ser avaliada, $f(x+\delta)$ (Greiner (1996), Sebald e Fogel (1992), Thompson (1998) e Wiesmann et al. (1998)).

As abordagens deste tipo necessitam de um grande número de avaliações da aptidão adicionais, o que se torna impraticável na maioria dos problemas do mundo real. No entanto, têm sido propostos na literatura métodos para reduzir este número de avaliações (Paenke et al. (2006)):

- *Técnicas para redução da variância* — escolhendo as perturbações da amostra de forma adequada, em vez de aleatória, reduz-se a variância do estimador, permitindo uma estimativa mais adequada com menos amostras. Em Loughlin e Ranjithan (1999), Markon et al. (2001) e Branke (2001-b), é proposto o "*Latin Hypercube Sampling*" juntamente com a ideia de se usar as mesmas perturbações para todos os indivíduos duma geração.
- *Avaliando os indivíduos mais importantes várias vezes* — em Branke (1998), é sugerido avaliar os bons indivíduos muito mais vezes do que os maus, uma vez que os bons indivíduos têm mais possibilidades de sobreviver, e por esse motivo é benéfica uma estimativa mais precisa. Em Branke (2001-a) foi proposto que os indivíduos com elevada variação da aptidão devem ser avaliados várias vezes.
- *Usando outros indivíduos da vizinhança* — uma vez que as regiões promissoras do espaço de procura são pesquisadas várias vezes, é possível usar a informação relativa a outros indivíduos da vizinhança para estimar a aptidão esperada de um dado indivíduo. Em particular, Branke (1998) propôs registar a *história* de uma *evolução*, isto é, armazenar numa base de dados todos os indivíduos de uma execução do processo evolucionário com os correspondentes valores da aptidão, e usar a aptidão média ponderada dos indivíduos vizinhos que estão no histórico.

Provavelmente, a abordagem mais comum consiste em construir amostras de um número de pontos escolhidos aleatoriamente na vizinhança da solução x a ser avaliada, e depois tomar a média dos pontos desta amostra como o valor estimado da aptidão esperada de x (ver Branke (1998), Greiner (1996) e Wiesmann et al. (1998)).

Paenke et al. (2006) apresentaram uma abordagem que consiste em usar um modelo de aproximação para estimar a robustez de uma solução. Em vez de usarem a verdadeira função de aptidão na simulação de Monte Carlo, os autores usaram o modelo de aproximação para este fim. Em princípio, esta ideia poderia ser usada combinada com

qualquer modelo de aproximação apropriado, como redes neuronais artificiais, modelos de Kriging[11] ou processos Gaussianos. No entanto, nesta abordagem, os autores usaram modelos de aproximação local, os quais têm a vantagem de serem relativamente fáceis de construir e parecem também apropriados para aproximar o desempenho de uma solução perante uma distribuição de perturbações locais.

A ideia principal é calcular a aptidão média em abordagens mono-objectivo ou a variância da aptidão em abordagens multi-objectivo, baseando-se directamente nas soluções vizinhas da população (geração) corrente ou em toda a história da evolução. Usar as soluções vizinhas existentes para calcular a média e a variância da aptidão é apenas uma aproximação muito grosseira da simulação de Monte Carlo. Para resolver este problema, os autores sugeriram construir modelos computacionalmente eficientes usando as soluções disponíveis para substituir a verdadeira função de aptidão (que é dispendiosa) no cálculo da média e da variância dos valores de aptidão.

O algoritmo evolucionário proposto para procurar soluções robustas começa com um *historial* vazio e por coleccionar os dados durante a execução. Em cada geração, a função de aptidão verdadeira é calculada na localização dos indivíduos actuais, sendo depois estes dados armazenados no *historial*. Com esta estratégia, o número total de avaliações da função de aptidão verdadeira é igual ao número de gerações multiplicado pelo tamanho da população, que é o mesmo valor requerido para o algoritmo evolucionário padrão. Desta forma, não são necessárias avaliações adicionais da aptidão para avaliar a robustez.

Para a aproximação à aptidão, os autores usaram os métodos de interpolação e de regressão local, os quais são combinados com uma variedade de modelos de distribuição. Uma vez que os modelos de aproximação local são apenas de confiança numa pequena vizinhança do seu ponto de ajustamento, são necessários vários modelos para avaliar uma população. Os autores usaram duas estratégias simples: construir um modelo em redor de cada indivíduo da população e construir um modelo em redor de cada ponto da amostra. Apesar da segunda estratégia ser a mais adequada, também é muito mais dispendiosa em termos computacionais.

[11]O modelo de Kriging é um interpolador estocástico que assume que os dados recolhidos de uma determinada população se encontram correlacionados no espaço. Isto é, se determinado ponto p tem um atributo r, é muito provável que se encontrem resultados muito próximos de r quanto mais próximos se estiver do ponto p. Porém, a partir de determinada distância de p, certamente não se encontrarão valores aproximados do atributo r porque a correlação espacial pode deixar de existir.

2.4.1.2. Cálculo implícito da média

Enquanto que todos os métodos referidos na secção anterior *explicitam* a média sobre um número de avaliações da aptidão, Tsutsui et al. (1996) e Tsutsui e Ghosh (1997) apresentaram a ideia de perturbar, de forma simples, as características dos fenótipos antes de avaliar a aptidão de um indivíduo. Usando o teorema do esquema ("schema theorem" de Goldberg (1989)) estes autores mostraram que, para algoritmos genéticos com uma população de tamanho infinito e usando selecção proporcional à aptidão, adicionar perturbações aleatórias às variáveis de decisão em cada geração é equivalente a optimizar a função de aptidão esperada. Tal como no cálculo da média implícita para o caso de funções de aptidão perturbada, o aumento do tamanho da população pode reduzir a influência da perturbação, e assim, garantir uma correcta convergência do algoritmo.

Tsutsui e Ghosh (1997) propuseram uma metodologia que estende a aplicação dos AGs a domínios que requerem a identificação de soluções robustas. Esta técnica, denominada por GA/RS^3 ("*Genetic Algorithms with a Robust Solution Searching Scheme"*), determina soluções robustas, adicionando perturbações aos valores dos parâmetros do fenótipo, enquanto faz a avaliação dos indivíduos com a função de aptidão.

Na Natureza, a expressão do fenótipo de um organismo é determinada, em parte, pela descodificação do código genótipo dos genes dos cromossomas. Durante este processo de descodificação, podem existir algumas perturbações, por exemplo, causadas por uma temperatura anormal, por um desequilíbrio nutricional, pela existência de substância nociva, etc.. Se um indivíduo tiver fraca aptidão devido a estas características fenotípicas perturbadas, então não sobreviverá para gerar filhos. Desta forma, indivíduos e populações reprodutivas, mesmo tendo "bom" material genótipo, poderão extinguir-se se forem altamente sensíveis às perturbações das características fenotípicas. Por outro lado, em ambientes perturbados, as unidades reprodutoras que são robustas para aquelas perturbações poderão ter mais hipóteses de sobreviver. Tsutsui e Ghosh (1997) desenvolveram o GAS/RS^3 com o objectivo de localizar soluções robustas usando esta metáfora de selecção da genética natural.

Existem abordagens que consideram a existência de perturbações no cálculo dos valores de aptidão, em que as perturbações são adicionadas à função de aptidão. Se $x = (x_1, x_2, \ldots, x_m)$ é um vector de parâmetros fenótipos, f(x) é a função de aptidão e δ uma perturbação escalar, então a aptidão do individuo será f(x) + δ. Quando se pretende detectar soluções robustas, adiciona-se a perturbação durante o processo de descodificação do código genótipo dos parâmetros do fenótipo. Por isso, para se adicionar

perturbação aos parâmetros do fenótipo x, parece razoável usar-se uma função de avaliação da forma $f(x + \delta)$, onde $\delta = (\delta_1, \delta_2, \ldots, \delta_m)$ é um vector aleatório.

Ong et al. (2006) apresentaram um algoritmo evolucionário baseado numa combinação de uma estratégia de optimização *max-min* com uma metodologia Baldwiniana sobre a região promissora (combinação de algoritmos evolucionários com técnicas de aproximação). Para reduzir o elevado custo computacional associado ao algoritmo evolucionário robusto, foram introduzidos modelos de substituição local para estimar a aptidão esperada e/ou variância da aptidão das possíveis soluções, em vez dos valores de aptidão verdadeiros.

2.4.2. Abordagens multi-objectivo

Das (2000) e Jin e Sendhoff (2003) demonstraram que, nalguns casos, optimizar a função de aptidão média (aptidão esperada) não é suficiente. Com a aptidão esperada como único objectivo, os desvios positivos ou negativos da aptidão verdadeira podem anular-se mutuamente na vizinhança de um ponto alvo. Assim, uma solução com elevada variabilidade da aptidão pode ser considerada robusta. No entanto, pode ser vantajoso considerar a aptidão esperada e a variância da aptidão como critérios de optimização separados, o que permite procurar soluções com diferentes compromissos entre desempenho e robustez.

Existem poucos estudos dedicados à optimização robusta multi-objectivo reportados na literatura. Alguns destes estudos são descritos a seguir.

Ray (2002) e Ray e Tsai (2004) propuseram abordagens para tratar problemas de optimização robusta com restrições, nas quais a robustez de uma solução é considerada durante o processo de pesquisa e não após determinar as soluções finais. Nestas abordagens, a procura de soluções robustas é tratada como um problema de optimização tri-objectivo, cujos objectivos são: o desempenho de uma solução, o desempenho médio dos seus vizinhos e o desvio padrão do desempenho dos seus vizinhos, em que o desempenho de uma solução é avaliado através do seu valor de aptidão verdadeiro. De modo a tratar a admissibilidade, Ray (2002) introduziu um esquema de tratamento das restrições baseado no conceito de dominância de Pareto que considera a admissibilidade da própria solução e a admissibilidade na sua vizinhança. Com este método resultam, eventualmente, soluções que são admissíveis e têm vizinhos admissíveis.

Jin e Sendhoff (2003) consideraram um problema de optimização bi-objectivo, cujos objectivos a maximizar são a robustez e o desempenho (através da função de aptidão verdadeira). Desta forma, os autores pretendiam determinar as soluções robustas de um problema de optimização com um só objectivo. A medida de robustez é

definida como a razão entre o desvio padrão da aptidão e a média do desvio padrão das variáveis de decisão. Para estimar a medida de robustez, são estimadas a aptidão média e o desvio padrão usando as soluções vizinhas da geração corrente, não sendo necessárias quaisquer avaliações adicionais da aptidão, mas apenas usando os indivíduos da geração corrente para estimar a variância local da aptidão. Isto torna-se possível porque a diversidade local da população é mantida usando o método dinâmico de agregação ponderada para optimização multi-objectivo (Jin et al. (2001-a)).

Li et al. (2005) apresentaram uma abordagem determinística para estudar o compromisso entre desempenho e robustez das soluções óptimas de Pareto de um problema multi-objectivo. A finalidade desta abordagem é optimizar, em simultâneo, duas funções objectivo: um valor de aptidão e um índice de robustez. O valor de aptidão serve como medida do desempenho das soluções óptimas de Pareto relativamente aos vários objectivos e da admissibilidade do problema original (considera os objectivos e as restrições do problema original). Estes valores de aptidão podem ser obtidos a partir de qualquer abordagem evolucionária multi-objectivo, como é o caso do NSGA (Srinivas e Deb (1994)). O índice de robustez é uma medida que avalia quantitativamente a robustez das soluções obtidas. O índice de robustez de uma solução é calculado com base na estimação da sensibilidade dos dados incertos e o raio da hiper-esfera exterior da região de tolerância normalizada (a região de tolerância é um hiper-rectângulo no espaço formado por um conjunto de valores das variações dos dados incertos, em relação aos quais o AD pretende que uma solução robusta seja insensível).

Lim et al. (2005) apresentaram uma abordagem de optimização evolucionária para pesquisar soluções com desempenhos (função de aptidão) nominais não dominados e robustas. Usando *a priori* a informação sobre a robustez desejada para a solução final do modelo, o algoritmo deve convergir para um conjunto de soluções que apresentem bons desempenhos nominais e que exibam robustez máxima. Um aspecto importante desta metodologia é que não requer a construção de qualquer estrutura da incerteza envolvida, sendo, por isso, particularmente útil quando é escassa a informação disponível sobre a incerteza. Os autores usaram o NSGA (Srinivas e Deb (1994)) para testar esta metodologia.

Deb e Gupta (2005) (ver também Deb e Gupta (2004)) apresentaram duas abordagens para determinar soluções robustas em problemas de optimização multi-objectivo. A primeira consiste em optimizar as funções objectivo efectivas médias, as quais são calculadas através da média de um conjunto representativo de soluções vizinhas. A segunda consiste em optimizar as funções objectivo originais, mas considerando uma restrição para limitar as alterações nos valores das funções objectivo a um valor definido pelo utilizador.

Gupta e Deb (2005) estenderam os conceitos apresentados por Deb e Gupta (2005) para determinar soluções robustas em problemas de optimização multi-objectivo com restrições efectivas. Os autores usaram o modelo desenvolvido por Deb e Gupta (2005) (em optimização multi-objectivo sem restrições), e mostraram que a inclusão de restrições resultou num maior deslocamento da fronteira robusta, uma vez que as soluções nos limites da região admissível (embora sejam as melhores) são geralmente não robustas. A estratégia de manipulação de restrições proposta foi incorporada nas duas abordagens descritas em Deb e Gupta (2005).

Como as abordagens apresentadas por Deb e Gupta (2004) e Deb e Gupta (2005) foram as primeiras a tratar os problemas como multi-objectivo tendo, assim, maior relevância no âmbito desta tese, vão ser alvo de uma análise mais detalhada na secção seguinte.

3. Abordagem de Deb e Gupta

Deb e Gupta (2004) e Deb e Gupta (2005) apresentaram duas abordagens diferentes de optimização robusta multi-objectivo, uma utilizando as funções objectivo efectivas médias e a outra utilizando as funções objectivo originais.

Na primeira abordagem, os autores estenderam uma abordagem já existente para determinar soluções robustas de um problema de optimização com um só objectivo à optimização multi-objectivo. No essencial, em vez de optimizar as funções objectivo originais, optimizam-se as funções efectivas médias, calculadas através da média de um conjunto representativo de soluções vizinhas. As soluções que são menos sensíveis a perturbações locais serão favorecidas em termos de valor da função efectiva média, o que leva a que a frente eficiente resultante seja constituída por soluções robustas.

A segunda abordagem consiste em optimizar as funções objectivo do problema original, mas adicionando uma (nova) restrição para limitar as dimensões das alterações dos valores das funções objectivo, causadas por perturbações locais, a um valor definido pelo utilizador. Desta forma, esta abordagem é mais pragmática do que a primeira, tendo o utilizador um controlo no nível de robustez desejado das soluções obtidas.

No entanto, estas definições de robustez apenas são utilizadas em problemas onde a incerteza se encontra nos valores das variáveis de decisão das soluções.

3.1. Robustez em optimização mono-objectivo

Considere-se o problema de optimização com um só objectivo, cuja formalização foi apresentada na secção 1 do capítulo 2, mas que pode ser simplificada do seguinte modo:

$$\begin{aligned} &\text{Minimizar} \quad f(x) \\ &\text{sujeito a} \quad x \in X \end{aligned} \tag{1}$$

em que X é o espaço de pesquisa admissível.

Na minimização de uma função objectivo f(x), a solução x^* é chamada de *solução robusta do tipo I* se é o mínimo global da função efectiva média $f^{ef}(x)$ definida em relação a uma δ-vizinhança, da forma seguinte (Deb e Gupta (2004) e Deb e Gupta (2005)):

$$\begin{aligned} &\text{Minimizar} \quad f^{ef}(x) = \frac{1}{|B_\delta(x)|} \int_{y \in B_\delta(x)} f(y)\, dy \\ &\text{sujeito a} \quad x \in X \end{aligned} \tag{2}$$

em que $B_\delta(x)$ é a δ-vizinhança da solução x e $|B_\delta(x)|$ é o hiper-volume da vizinhança.

Para se analisar a vizinhança de uma solução, considera-se um conjunto finito de h soluções, as quais podem ser escolhidas aleatoriamente (ou de outra forma estruturada) dentro de uma δ-vizinhança de uma solução x, $B_\delta(x)$, no espaço das variáveis, e usar-se a função efectiva média, $f^{ef}(x)$, como função de aptidão de um algoritmo evolucionário. Desta forma, em vez de a um indivíduo estar associado um valor da própria função f(x), está associado um valor do objectivo "aglomerado", o qual depende da sua vizinhança.

A outra abordagem consiste em calcular a diferença normalizada entre o valor da função perturbada, $f^p(x)$ e o valor da função original f(x). Nesta abordagem, uma solução é considerada robusta se a diferença normalizada (relativa) for menor do que um certo limiar, η, previamente escolhido.

Na minimização de uma função objectivo f(x), uma solução x^* é chamada de *solução robusta do tipo II* se é uma solução mínima global do seguinte problema (Deb e Gupta (2004) e Deb e Gupta (2005)):

$$\begin{aligned} &\text{Minimizar} \quad f(x) \\ &\text{sujeito a} \quad \frac{\|f^p(x) - f(x)\|}{\|f(x)\|} \le \eta, \\ &\qquad x \in X \end{aligned} \tag{3}$$

O valor da função perturbada $f^p(x)$ pode ser escolhido, por exemplo, como o valor da função efectiva média $f^{ef}(x)$ ou como o pior valor da função na vizinhança (entre h soluções escolhidas aleatoriamente). Este método requer também o cálculo de h soluções vizinhas. O operador $\|.\|$ pode ser qualquer medida adequada. Considera-se o limiar η constante e previamente definido pelo utilizador. Nalgumas aplicações, pode-se utilizar a diferença normalizada absoluta, $\|f^p(x) - f(x)\| \le \eta$, em vez da relativa.

3.2. Robustez em optimização multi-objectivo

Considere-se o problema de optimização multi-objectivo, cuja formalização foi apresentada na secção 2.1 do capítulo 2, mas que pode ser simplificada do seguinte modo:

$$\begin{aligned} &\text{Minimizar} \quad \left(f_1(x), f_2(x), \ldots, f_M(x)\right) \\ &\text{sujeito a} \quad x \in X \end{aligned} \tag{4}$$

em que se consideram M funções objectivo, geralmente em conflito e não comensuráveis.

Os dois conceitos de robustez apresentados na secção anterior, para o caso mono-objectivo, podem ser estendidos para o caso multi-objectivo. No entanto, para que uma solução óptima de Pareto seja classificada como robusta, tem agora que demonstrar a sua "imunidade" em todas as funções objectivo relativamente a pequenas perturbações nos valores das variáveis de decisão. Esta imunidade tem agora que ser estabelecida relativamente a todos os M objectivos (ou a um subconjunto que o AD considere mais relevantes) — isto é, tem que ser usado um efeito combinado das variações em todos os M objectivos como uma *medida* de imunidade face a perturbações de diferentes dimensões. Adicionalmente, existem muitas soluções a serem pesquisadas com vista à avaliação da sua robustez, em vez de uma solução óptima como no caso da optimização com um único objectivo (ou mais do que uma no caso de existirem soluções óptimas alternativas).

A partir da definição de solução robusta do tipo I, para o caso mono-objectivo, pode definir-se solução robusta multi-objectivo do tipo I.

Uma solução x^* é designada por *solução robusta multi-objectivo do tipo I* se é uma solução óptima de Pareto global para o seguinte problema multi-objectivo, definido relativamente a uma δ-vizinhança, $B_\delta(x)$, de uma solução x (Deb e Gupta (2004) e Deb e Gupta (2005)):

$$\begin{aligned} &\text{Minimizar} \quad \left(f_1^{ef}(x), f_2^{ef}(x), \ldots, f_M^{ef}(x)\right) \\ &\text{sujeito a} \quad x \in X \end{aligned} \tag{5}$$

em que

$$f_j^{ef} = \frac{1}{|B_\delta(x)|} \int_{y \in B_\delta(x)} f_j(y)\, dy, \text{ para } j = 1, \ldots, M. \tag{6}$$

Devido aos diferentes níveis de imunidade relacionada com os vários objectivos, uma parte da frente eficiente global original pode não ser classificada como frente robusta. Nalgumas situações, a frente eficiente global original (correspondente ao problema (4)) pode ser totalmente não-robusta e uma frente local original ou uma frente sub-óptima original pode tornar-se robusta. Dependendo de quão robusta (do tipo I) é

uma frente eficiente global original, podem ocorrer os seguintes quatro casos principais (Deb e Gupta (2004) e Deb e Gupta (2005)):

1) toda a frente eficiente original é robusta;
2) uma parte da frente eficiente original é não robusta;
3) toda a frente eficiente original é não robusta, mas é robusta uma frente eficiente local;
4) uma parte da frente eficiente original é robusta, assim como uma parte de uma frente eficiente local.

A partir da definição de solução robusta de tipo II, para o caso mono-objectivo, pode também definir-se solução robusta multi-objectivo do tipo II.

Uma solução x^* é designada por *solução robusta multi-objectivo do tipo II* se é solução óptima de Pareto global do seguinte problema multi-objectivo (Deb e Gupta (2004) e Deb e Gupta (2005)):

$$\begin{aligned} &\text{Minimizar } (f_1(x), f_2(x), \ldots, f_M(x)) \\ &\text{sujeito a } \frac{\|f^p(x)-f(x)\|}{\|f(x)\|} \le \eta \\ &\quad x \in X \end{aligned} \tag{7}$$

Pode usar-se qualquer função para $f^p(x)$ e qualquer norma para o operador $\|.\|$, desde que sejam adequadas ao problema; por exemplo, pode usar-se a função efectiva média $f^{ef}(x)$ e a norma Euclidiana, respectivamente. Considera-se o limiar η constante e previamente definido pelo utilizador.

4. Uma nova abordagem à optimização robusta multi-objectivo

As definições apresentadas na secção anterior permitem classificar as soluções apenas como robustas ou não-robustas. No entanto, a "diferença" entre uma solução robusta e uma não-robusta pode ser muito pequena. Por outro lado, uma solução pode ser mais robusta do que outra e, no entanto, esta segunda ser considerada "mais interessante", para o AD, do que a primeira, face aos valores que apresenta para as funções objectivo.

Desta forma, é mais interessante, do ponto de vista operacional do apoio à decisão, utilizar as definições de robustez de forma a qualificar as soluções tendo em conta a respectiva vizinhança; por exemplo, definindo um grau de robustez que qualifique cada solução, cujo valor irá interferir directamente no cálculo do valor da função de aptidão de cada uma delas e, desta forma, contribuir para a evolução das soluções determinadas pelo processo evolucionário.

Estas novas definições de robustez podem ser usadas na construção de abordagens evolucionárias para problemas sujeitos a incerteza (perturbações), quer nos valores das variáveis de decisão, quer nos dados (coeficientes e parâmetros) associados às restrições e às funções objectivo do problema em estudo. O problema em estudo é modelado deterministicamente (através dos dados "nominais" associados ao problema), sendo as perturbações adicionadas às respectivas entidades do modelo no decorrer do processo evolucionário, aquando do cálculo dos graus de robustez das soluções.

Quando apenas os valores das variáveis de decisão estão sujeitos a perturbações, diz-se que estas incidem sobre o espaço das variáveis de decisão. Quando apenas os dados associados às restrições estão sujeitos a perturbações, diz-se que estas incidem sobre a região admissível. Quando apenas os dados associados às funções objectivo estão sujeitos a perturbações, diz-se que estas incidem sobre o espaço dos objectivos.

Para avaliar a robustez das soluções de um problema em que os dados associados às restrições e às funções objectivo estão sujeitas a perturbações, é necessário introduzir o conceito de cenário. Um *cenário* é um conjunto de valores possíveis para os dados do problema sujeitos a perturbações. Ao conjunto dos valores iniciais (nominais) para estes dados dá-se o nome de *cenário de referência*. Designa-se por *espaço dos cenários* o conjunto de todos os cenários possíveis.

4.1. Perturbações nos valores das variáveis de decisão

Assume-se que as perturbações sobre uma solução podem ocorrer em qualquer dimensão do espaço das soluções (x_1, x_2, ..., x_n). O cálculo do grau de robustez de uma solução x depende do comportamento das soluções que pertencem a uma vizinhança de raio δ em redor de x, no espaço das soluções e no espaço dos objectivos. Desta forma, o grau de robustez de uma solução x depende:

a) do número de soluções admissíveis da δ-vizinhança de x;
b) do número de soluções da δ-vizinhança de x, cujos valores das funções objectivo são melhores do que os valores das funções objectivo para x, f(x), ou pertencem a uma η-vizinhança em redor de f(x).

A ideia subjacente é determinar um conjunto de k vizinhanças em redor da solução x, cujos raios são valores múltiplos de um valor δ ($k\delta$), tal que as soluções que pertencem a estas vizinhanças são admissíveis e as suas imagens são melhores do que f(x) ou pertencem a uma vizinhança de raio η em redor de f(x).

O processo começa por analisar h soluções pertencentes à δ-vizinhança de uma solução x, as quais são geradas aleatoriamente e se distribuem uniformemente em redor de x (ver também Deb e Gupta (2004)). Esta vizinhança é então progressivamente

alargada em múltiplos de δ (δ, 2δ, ...), até que a percentagem das h soluções que são admissíveis seja inferior a um limiar predefinido, ou cujos valores das funções objectivo que são melhores do que f(x) ou que pertencem à vizinhança de raio η em redor de f(x) seja inferior a um outro limiar predefinido. Isto permite atribuir um grau de robustez às soluções de acordo com o número de vizinhanças de raios múltiplos de δ construídas em que a condição anterior é satisfeita.

Desta forma, o grau de robustez k de uma solução x é calculado gradualmente, com k a aumentar a amplitude das vizinhanças (δ, 2δ, ..., $k\delta$) e o número de soluções vizinhas de x (h, h+qh, ..., h+(k-1)qh), tal que na tδ-vizinhança de x são analisadas h+(t-1)qh soluções vizinhas — $t \in \{1, ..., k\}$ e $0 \leq q \leq 1$ (parâmetro descrito na próxima secção) — ver Figura 4.1.

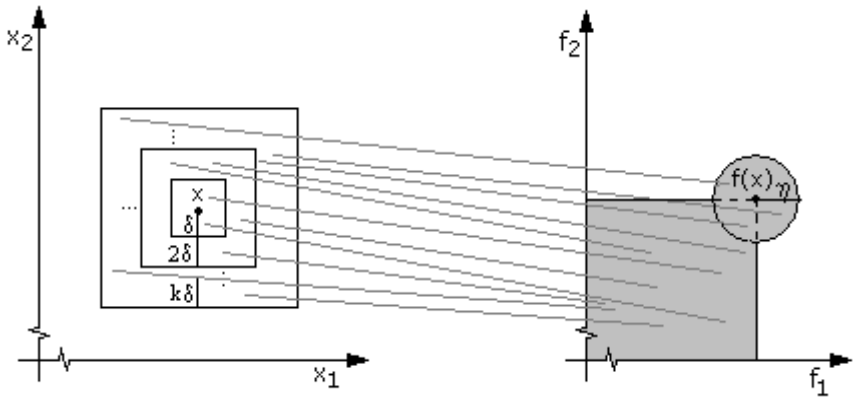

Figura 4.1: Definições das vizinhanças nos espaços das variáveis de decisão e das funções objectivo (para espaços bidimensionais e todas as funções a minimizar).

O grau de robustez uma solução x define-se como um valor inteiro positivo k, tal que (Figura 4.1):

a) a percentagem de soluções admissíveis da $k\delta$-vizinhança de x é maior ou igual a um limiar predefinido p1;
b) a percentagem de soluções da $k\delta$-vizinhança de x, cujos valores das funções objectivo que são melhores do que f(x) ou que pertencem à η-vizinhança de f(x) é maior ou igual a um limiar predefinido p2*;*
c) a percentagem de soluções admissíveis da (k+1)δ-vizinhança de x é inferior a p1; *ou*
d) a percentagem de soluções da (k+1)δ-vizinhança de x, cujos valores das funções objectivo que são melhores do que f(x) ou que pertencem à η-vizinhança de f(x) é inferior a p2.

A δ-vizinhança de x pode ser traduzida por uma hiper-esfera de raio δ ou por um hipercubo de arestas 2δ centrados em x, assim como a η-vizinhança de f(x) pode ser uma

hiper-esfera de raio η ou um hipercubo de arestas 2η centrados em f(x) (ver Figura 4.1 para o caso bidimensional).

Para o caso , a δ-vizinhança de x é traduzida por um intervalo de amplitude 2δ centrado em x ($x \pm \delta$), assim como a η-vizinhança de f(x) pode ser traduzida por um intervalo de amplitude 2η centrado em f(x) (f(x) $\pm \eta$).

4.2. Perturbações nos dados associados às funções objectivo

Para este caso, é necessário utilizar o conceito de cenário, introduzido no início desta secção. Um cenário é um conjunto de possíveis valores para os dados associados às funções objectivo do modelo sujeitos a perturbações. O cálculo do grau de robustez de uma solução x envolve a análise do comportamento de x para os cenários pertencentes à vizinhança de raio δ em redor do cenário de referência s, numa vizinhança de raio η em redor dos valores das funções objectivo da solução x para o cenário de referência s, $f^s(x)$.

A ideia fundamental é determinar um conjunto de k vizinhanças em redor do cenário de referência s, cujos raios são múltiplos de δ ($k\delta$), tal que os valores das funções objectivo de x para estes cenários são melhores do que $f^s(x)$ ou pertencem a uma η-vizinhança predefinida em redor de $f^s(x)$.

O processo começa por analisar um conjunto de h cenários pertencentes a uma vizinhança de raio δ em redor do cenário de referência s, os quais são geradas aleatoriamente e se distribuem uniformemente em redor de s (ver também Deb e Gupta (2004)). Esta vizinhança é então progressivamente alargada em múltiplos de δ (δ, 2δ, ...), até que a percentagem dos h cenários para os quais os valores das funções objectivo de x para estes cenários são melhores do que $f^s(x)$ ou pertencem a uma η-vizinhança de $f^s(x)$ seja inferior a um limiar predefinido. Isto permite atribuir um grau de robustez às soluções de acordo com o número de vizinhanças de raios múltiplos de δ construídas em que a condição anterior é satisfeita.

Desta forma, o grau de robustez k de uma solução x é calculado gradualmente, com k a aumentar a amplitude das vizinhanças (δ, 2δ, ..., $k\delta$) e o número de cenários vizinhos do cenário de referência s (h, h+qh, ..., h+(k-1)qh), tal que na tδ-vizinhança de s são analisados h+(t-1)qh cenários vizinhos — $t \in \{1, ..., k\}$ e $0 \leq q \leq 1$ (parâmetro descrito na próxima secção) — ver Figura 4.2.

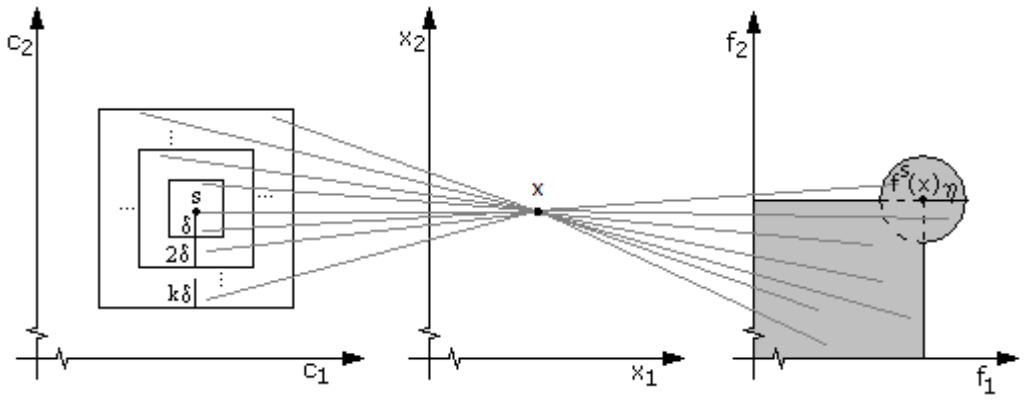

Figura 4.2: Definições das vizinhanças nos espaços dos cenários e das funções objectivo (para espaços bidimensionais e todas as funções a minimizar).

O grau de robustez de uma solução x define-se como um valor inteiro positivo k, tal que (Figura 4.2):

a) a percentagem de cenários da $k\delta$-vizinhança de s, cujos valores das funções objectivo de x para estes cenários que são melhores do que $f^s(x)$ ou que pertencem à η-vizinhança de $f^s(x)$ é maior ou igual a um limiar predefinido p2;

b) a percentagem de cenários da $(k+1)\delta$-vizinhança de s, cujos valores das funções objectivo de x para aqueles cenários que são melhores do que $f^s(x)$ ou que pertencem à η-vizinhança de $f^s(x)$ é inferior a p2.

A δ-vizinhança de s pode ser traduzida por uma hiper-esfera de raio δ ou um hipercubo de arestas 2δ centrados em s, assim como a η-vizinhança de $f^s(x)$ pode ser uma hiper-esfera de raio η ou um hipercubo de arestas 2η centrados em $f^s(x)$ (ver Figura 4.2, para o caso bidimensional).

Para o caso , a δ-vizinhança de s é traduzida por um intervalo de amplitude 2δ centrado em s ($s \pm \delta$), assim como η-vizinhança de $f^s(x)$ pode ser traduzida por um intervalo de amplitude 2η centrado em $f^s(x)$ ($f^s(x) \pm \eta$).

4.3. Perturbações nos dados associados às restrições

Também para este caso é necessário utilizar o conceito de cenário, introduzido no início desta secção. Um cenário é um conjunto de possíveis valores para os dados associados às restrições do modelo (pode-se também incluir as 2n restrições associadas aos limites das variáveis de decisão) sujeitos a perturbações.

O cálculo do grau de robustez de uma solução x envolve a análise da vizinhança de raio δ em redor do cenário de referência s, em termos de admissibilidade de x para estes cenários. Note-se que $f(x)$ é a imagem de x para o cenário de referência s, assim como

para todos os cenários possíveis. Pelo contrário, a admissibilidade da solução *x* pode ser diferente para cenários diferentes, uma vez que cada cenário pode corresponder a uma região admissível diferente.

Pretende-se determinar um conjunto de *k* vizinhanças em redor do cenário de referência *s*, cujos raios são valores múltiplos de um valor δ ($k\delta$), tal que a solução *x* é admissível para estes cenários.

O processo começa por analisar h cenários pertencentes a uma vizinhança de raio δ em redor do cenário de referência *s*, os quais são geradas aleatoriamente e se distribuem uniformemente em redor de *s* (ver também Deb e Gupta (2004)). Esta vizinhança é então progressivamente alargada em múltiplos de δ (δ, 2δ, ...), até que a percentagem de cenários para os quais a solução *x* é admissível seja inferior a um limiar predefinido. Isto permite atribuir um grau de robustez às soluções de acordo com o número de vizinhanças de raios múltiplos de δ construídas em que a condição anterior é satisfeita.

Desta forma, o grau de robustez *k* de uma solução *x* é calculado gradualmente, com *k* a aumentar a amplitude das vizinhanças (δ, 2δ, ..., $k\delta$) e o número de cenários vizinhos do cenário de referência *s* (h, h+qh, ..., h+(*k*-1)qh), tal que na tδ-vizinhança de *s* são analisados h+(t-1)qh cenários vizinhos — $t \in \{1, ..., k\}$ e $0 \leq q \leq 1$ (parâmetro descrito na próxima secção).

O grau de robustez de uma solução *x* define-se como um valor inteiro positivo *k*, tal que:

a) a percentagem de cenários da $k\delta$-vizinhança de *s*, para os quais a solução *x* é admissível, é maior ou igual a um limiar predefinido p1;
b) a percentagem de cenários da $(k+1)\delta$-vizinhança de *s*, para os quais a solução *x* é admissível, é inferior a p1.

A δ-vizinhança de *s* pode ser traduzida por uma hiper-esfera de raio δ ou um hipercubo de arestas 2δ centrados em *s*, assim como a η-vizinhança de $f^s(x)$ pode ser uma hiper-esfera de raio η ou um hipercubo de arestas 2η centrados em $f^s(x)$.

Para o caso , a δ-vizinhança de *s* é traduzida por um intervalo de amplitude 2δ centrado em *s* ($s \pm \delta$), assim como a η-vizinhança de f(*x*) pode ser traduzida por um intervalo de amplitude 2η centrado em f(*x*) (f(*x*) $\pm \eta$).

4.4. Parâmetros de robustez

Existem vários parâmetros que estão associados à análise de robustez das soluções, cada um deles com uma função específica, os quais podem receber valores diferentes consoante a vontade do AD e a fiabilidade pretendida no cálculo da solução escolhida (em termos de robustez).

O parâmetro h está associado ao número de pontos que pertencem à vizinhança de uma solução ou cenário de referência que são gerados e analisados. Quanto maior for este valor, mais fiável é a análise efectuada (no entanto, maior é o tempo de execução do algoritmo).

O parâmetro δ está associado ao raio base das vizinhanças das soluções ou cenários de referência. O valor para o raio de cada vizinhança é um múltiplo de δ (δ, 2δ, ...). Este valor é indicado pelo AD e deve reflectir as suas preferências quanto à dimensão base da vizinhança das soluções no espaço das soluções, ou dos cenários de referência no espaço dos cenários, que ele aceite como região alternativa a cada solução, ou cenário.

O parâmetro η pode ser entendido como um indicador da robustez requerida pelo AD, mas de natureza diferente dos parâmetros p1 e p2 (descritos a seguir). Este parâmetro é usado como limite superior para a distância entre duas soluções (de qualquer tipo) no espaço dos objectivos (em problemas em que a perturbação incide sobre os valores das variáveis de decisão) ou entre as imagens de uma solução *x* em dois cenários diferentes, em que um deles é o cenário de referência (em problemas em que os dados do modelo associados às funções objectivo estão sujeitos a perturbações).

Este parâmetro reflecte os limiares de indiferença relativamente aos valores de cada função objectivo. Com o aumento do valor de η, o que significa que o AD é mais tolerante relativamente às diferenças nos valores das funções objectivo, a tendência é que o número de soluções com graus de robustez mais elevados também aumente.

Os parâmetros p1 e p2 podem ser entendidos como os principais indicadores da robustez requerida.

Em problemas em que os valores das variáveis de decisão estão sujeitos a perturbações, se o grau de robustez de uma solução *x* é *k*, então p1 = 100% e p2 = 100% significa que em todas as tδ-vizinhanças (t $\in$ {1,...,k}) de *x*, todos os h pontos vizinhos de *x* analisados são admissíveis e os respectivos valores das funções objectivo são melhores do que f(*x*) ou pertencem à η-vizinhança de f(*x*). Se p1 = 100% e p2 = 90%, então todos os h pontos vizinhos de *x* são admissíveis e a percentagem destes h pontos para os quais os respectivos valores das funções objectivo são melhores do que f(*x*) ou pertencem à η-vizinhança de f(*x*) é pelo menos 90%. Se p1 = 90% e p2 = 90%, então pelo menos 90% dos h pontos vizinhos de *x* são admissíveis e a percentagem destes h pontos para os quais os respectivos valores das funções objectivo são melhores do que f(*x*) ou pertencem à η-vizinhança de f(*x*) é pelo menos 90%.

Para problemas em que os dados associados às funções objectivo estão sujeitos a perturbações, se o grau de robustez de uma solução *x* é *k*, então p2 = 100% significa que em todas as tδ-vizinhanças (t $\in$ {1, ..., *k*}) do cenário de referência *s*, os valores das funções objectivo de *x* para todos os h cenários vizinhos *s'* de *s*, $f^{s'}(x)$, são melhores do

que $f^s(x)$ ou pertencem à η-vizinhança predefinida de $f^s(x)$. Se p2 = 90%, então a percentagem dos h cenários vizinhos *s'* de *s*, para os quais os valores das funções objectivo de *x*, $f^{s'}(x)$, são melhores do que $f^s(x)$ ou pertencem à η-vizinhança de $f^s(x)$ é pelo menos 90%.

Para problemas em que os dados associados às restrições estão sujeitos a perturbações, se o grau de robustez de uma solução *x* é *k*, então p1 = 100% significa que em todas as tδ-vizinhanças ($t \in \{1, ..., k\}$) do cenário de referência *s*, a solução *x* é admissível para todos os h cenários vizinhos de *s*. Se p1 = 90% então a solução *x* é admissível em pelo menos 90% dos h cenários vizinhos de *s*.

Desta forma, é mais provável que uma solução *x* com grau de robustez *k* obtido com p1 = 100% e/ou p2 = 100% tenha, de facto, este valor como grau de robustez do que quando obtido com valores inferiores para p1 e/ou p2, pois isto significa um relaxamento dos valores destes parâmetros (menor exigência implica menor fiabilidade do grau de robustez obtido para a solução e, portanto, maior risco associado à sua classificação com um certo grau de robustez).

O valor do parâmetro q é um coeficiente ($0 \leq q \leq 1$) associado ao aumento do número de pontos vizinhos a serem analisados nos sucessivos alargamentos das vizinhanças de uma solução ou cenário de referência. Por exemplo, se q = 0.5 então o número de pontos vizinhos a serem analisados na *k*-ésima ($k > 1$) vizinhança é aumentado em $0.5 \times h$ em relação ao número de pontos vizinhos analisados na vizinhança anterior: δ (h pontos), 2δ ($h+0.5 \times h$ pontos), 3δ ($h+0.5 \times h+0.5 \times h$ pontos), ...

No entanto, devido ao considerável tempo de execução dos algoritmos evolucionários implementados no âmbito deste trabalho (a descrever na secção 4.6 deste capítulo), cada um dos parâmetros p1, p2, η e q pode ter valores diferentes, de acordo com o tipo de soluções em causa (se são soluções admissíveis não dominadas, admissíveis dominadas ou não admissíveis), de forma a que este tempo se reduza.

Relativamente aos valores dos parâmetros p1 e p2, recomenda-se que estes sejam sempre muito elevados (100%) para soluções admissíveis dominadas e não admissíveis, para que o grau de robustez máximo obtido por uma solução de qualquer destes tipos seja baixo (o que implica mais baixos custos computacionais); para soluções admissíveis não dominadas, estes parâmetros assumem os valores que o AD achar mais adequados (tendo em conta que parâmetros mais/menos elevados corresponde a maior/menor exigência quanto à robustez). Da mesma forma, recomenda-se que as soluções admissíveis dominadas e não admissíveis tenham valores baixos para os parâmetros η e q; para as soluções admissíveis não dominadas, os valores para estes parâmetros devem ser o que AD achar mais adequados. A razão fundamental é que as soluções admissíveis não dominadas são muito mais importantes do que as soluções admissíveis dominadas e

não admissíveis, sendo, por isso, necessária uma análise mais exaustiva e mais precisa para as soluções admissíveis não dominadas.

4.5. Métricas

Para se verificar se, para uma solução *y* pertencente à δ-vizinhança da solução *x*, *y* também pertence à η-vizinhança de *x* no espaço dos objectivos, é necessário calcular a distância entre as imagens das soluções *x* e *y* no espaço dos objectivos, f(*x*) e f(*y*).

Da mesma forma, para se verificar se, para um cenário *s'* pertencente à δ-vizinhança do cenário de referência *s*, $f^{s'}(x)$ pertence à η-vizinhança de $f^{s}(x)$, é necessário calcular a distância entre as imagens da solução *x* nos cenários *s'* e *s*, $f^{s'}(x)$ e $f^{s}(x)$, no espaço dos objectivos.

Para calcular a distância normalizada entre f(*x*) e f(*y*) pode usar-se a expressão $\|f(y)-f(x)\|/\|f(x)\|$ (relativa) ou $\|f(y)-f(x)\|$ (absoluta). Para calcular a distância normalizada entre $f^{s'}(x)$ e $f^{s}(x)$ pode usar-se a expressão $\|f^{s'}(x)-f^{s}(x)\|/\|f^{s}(x)\|$ (relativa) ou $\|f^{s'}(x)-f^{s}(x)\|$ (absoluta). Para ambos os casos, o operador $\|.\|$ pode ser qualquer métrica adequada, tal como a Euclidiana (L_2) ou de Chebycheff (L_∞).

Pode também usar-se a distância não normalizada entre f(*x*) e f(*y*), ou entre $f^{s'}(x)$ e $f^{s}(x)$. Nestes casos, $\eta = (\eta_1, ..., \eta_M)$, em que M é o número de funções objectivo do problema.

4.6. Abordagem evolucionária

4.6.1. Construção das vizinhanças

Foi implementado um algoritmo evolucionário, mais propriamente um AG, no qual foram incorporadas as definições de grau de robustez propostas neste capítulo, para determinar as soluções óptimas de Pareto robustas de um problema de optimização multi-objectivo.

Neste algoritmo, um dos aspectos fundamentais está na construção das vizinhanças de uma solução *x* (espaço das soluções) ou de um cenário *s* (espaço dos cenários). Para tal, é necessário considerar dois parâmetros:

- a dimensão da vizinhança (δ) considerada para as soluções (no espaço das variáveis de decisão) ou para os cenários (no espaço dos cenários);
- número de h pontos (soluções ou cenários) vizinhos usados no cálculo do grau de robustez de uma solução *x*.

Relativamente à dimensão da vizinhança de uma solução ou de um cenário, ela pode ter valores diferentes para cada uma das suas componentes: $\delta = (\delta_1, \delta_2, ..., \delta_{nc})$, em que nc é o número de componentes dos dados sujeitos a perturbações. Isto significa que para $i \neq j$ pode acontecer que $\delta_i \neq \delta_j$. Estes valores são predefinidos e fornecidos pelo AD, pois é ele que conhece o problema e sabe que valores reflectem as suas preferências quanto às diferenças significativas nas várias dimensões.

Há várias formas de gerar as h soluções (ou cenários) vizinhas(os) de uma solução *x* (ou cenário *s*) (Branke (2000)). A estratégia mais simples consiste em criar aleatoriamente h soluções (ou cenários) na vizinhança da solução *x* (ou cenário *s*). No entanto, tal introduz aleatoriedade adicional ao analisar a vizinhança da mesma solução (ou cenário) mais do que uma vez. Desta forma, deve criar-se um padrão aleatório de soluções (ou cenários) em redor de uma solução *x* (ou cenário *s*) no início de uma simulação e usar o mesmo padrão para todas as soluções (ou cenários).

Para criar sistematicamente um padrão (Figura 4.3), divide-se o intervalo de variação da vizinhança de cada uma das nc componentes sujeitas a perturbações, $[-\delta_i, \delta_i]$ (com i = 1, 2, ..., nc), em exactamente h grelhas iguais, dividindo assim a δ-vizinhança em h^{nc} pequenas hiper-caixas. Depois, escolhe-se, de forma aleatória, exactamente h hiper-caixas das h^{nc} possíveis para que, em cada dimensão, todas as h grelhas distintas estejam representadas.

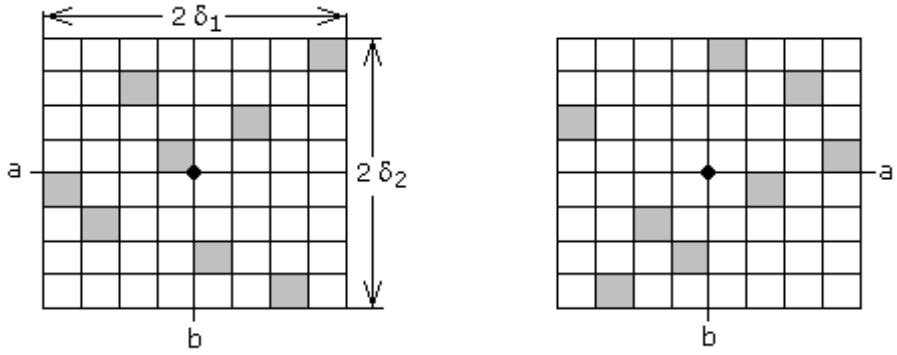

Figura 4.3: Forma de gerar h=8 pontos vizinhos em redor de uma solução x=(a, b).

A Figura 4.3 apresenta dois padrões diferentes associados a uma mesma solução (ou cenário) de um problema em que duas componentes dos dados do problema estão sujeitos a perturbações (nc = 2). Depois de as hiper-caixas serem identificadas, é escolhida aleatoriamente uma solução (ou cenário) vizinha(o) em cada hiper-caixa, a qual é usada no cálculo do grau de robustez daquela solução. Este procedimento garante também que as h soluções (ou cenários) vizinhas(os) determinadas(os) estão todas(os) bem distribuídas(os) em redor de uma solução *x* (ou cenário *s*).

4.6.2. Algoritmo genético

Foi concebido e implementado um AG envolvendo o cálculo do grau de robustez associado a cada solução, o qual usa uma estratégia elitista com uma população secundária (apenas com soluções admissíveis não dominadas) de tamanho máximo constante. A estratégia elitista permite aumentar o desempenho do AG, quer acelerando a convergência para a frente não dominada, quer garantindo que as soluções encontradas são de facto não dominadas e estão bem distribuídas pela frente. Este é um aspecto importante em problemas reais (ver Gomes et al. (2004) e Costa (2003)), já que é necessário fornecer ao AD um conjunto de soluções não dominadas bem distribuídas e diversificadas, para que ele fique bem preparado para tomar uma decisão final.

De modo a influenciar o processo evolucionário na procura das soluções mais robustas, o conceito de grau de robustez apresentado é embebido na avaliação da função de aptidão de cada indivíduo (solução), juntamente com o teste de dominância. Em cada nível de não dominância, as soluções mais robustas são as que mais contribuem para a próxima geração.

Os passos principais do AG construído para testar a metodologia aqui proposta, são os seguintes:

- calcular o grau de robustez de cada indivíduo da população principal;
- calcular o valor da função de aptidão de cada indivíduo da população principal, usando o respectivo grau de robustez calculando antes;
- seleccionar por torneio POP-E indivíduos da população principal (POP é o tamanho da população principal e E é o número de indivíduos que compõem a elite);
- construir uma nova população formada por POP-E filhos gerados por cruzamento e mutação e pelos E indivíduos (elite) mais robustos da população secundária;
- avaliar os indivíduos da nova população, utilizando um teste de dominância e os respectivos graus de robustez, definindo, desta forma, uma aproximação à fronteira não dominada;
- determinar os indivíduos candidatos a pertencerem à população secundária (soluções não dominadas) e processá-los de forma a actualizar a população secundária usando uma técnica de manutenção da diversidade, se necessário (se o número de indivíduos candidatos for superior ao tamanho máximo da população secundária).

4.6.3. Função de aptidão

Dada uma população de soluções de qualquer tipo, o valor da função de aptidão de uma qualquer solução desta população depende, quer do teste de dominância aplicado às soluções da população, quer do grau de robustez da própria solução.

Para calcular o valor da função de aptidão de cada solução da população, utiliza-se uma técnica que consiste em determinar vários níveis de dominância, tal como no algoritmo "NSGA-II" (Deb (2001) e Deb et al. (2002)). Este método consiste em determinar 3 grupos de frentes não dominadas (ver secção 2.4.4 do capítulo 2):

- 1º grupo: todas as soluções admissíveis não dominadas (1 frente);
- 2º grupo: todas as soluções admissíveis dominadas;
- 3º grupo: todas as soluções não admissíveis.

Os principais passos do processo de cálculo do valor da função de aptidão das soluções (indivíduos) de uma população, são os seguintes:

1) Atribuir a cada solução da primeira frente, que é composta por todas as soluções não dominadas da população (1º grupo), um valor mínimo da função de aptidão igual a POPx(GrauMax+1), em que GrauMax é um valor predefinido que corresponde ao valor máximo que se pode atribuir para o grau de robustez de uma solução (por conveniência do algoritmo);
2) Para cada uma destas soluções, adicionar o seu grau de robustez ao seu valor da função de aptidão;
3) Ignorar temporariamente as soluções da primeira frente e aplicar às restantes soluções admissíveis (as soluções dominadas) um teste de dominância (as soluções consideradas não dominadas pertencerão à segunda frente de soluções);
4) Determinar o valor mínimo da função de aptidão para as soluções desta nova frente, subtraindo o valor "GrauMax+1" ao valor mínimo da frente anterior, e atribuí-lo a cada uma das soluções desta nova frente;
5) Para cada solução da frente actual, adicionar ao seu valor da função de aptidão o seu grau de robustez;
6) Este processo continua até todas as soluções admissíveis dominadas (2º grupo) terem associado um valor da função de aptidão;
7) Repetir o processo para todas as soluções não admissíveis (3º grupo), até todas elas terem associado um valor da função de aptidão.

Se duas soluções tiverem o mesmo valor da função de aptidão, então a melhor solução é aquela com menos soluções na sua vizinhança, de acordo com um raio predefinido. Para tal, define-se um nicho de raio df centrado em cada solução em causa, em que df é a distância máxima necessária entre soluções para obter uma frente bem distribuída. Por exemplo, em problemas bi-objectivo, o valor de df é igual a MaxDist/POP, em que MaxDist é a distância entre os pontos ideais (obtidos considerando os melhores e os piores valores de cada função objectivo na população principal — ver secção 2.4.1 do capítulo 2) e POP é o tamanho da população que contém as soluções em questão.

Este processo garante que todas as soluções de um mesmo nível de dominância têm um valor da função de aptidão maior do que qualquer solução dos níveis seguintes. Desta forma, as soluções com maior valor da função de aptidão são as soluções admissíveis não dominadas (1º grupo) e as com pior valor são as soluções não admissíveis (3º grupo). Este processo garante também que, no mesmo nível de dominância, as soluções com maior grau de robustez têm maior valor da função de aptidão. Finalmente, num grupo de soluções do mesmo nível e com igual grau de robustez, as melhores são aquelas com menos vizinhos na população, de acordo com um raio predefinido df.

4.6.4. Mecanismo de manutenção da diversidade

Para actualizar a população secundária de forma a manter a diversidade na população secundária, é usado um mecanismo de partilha que utiliza um esquema de nichos, em que os valores dos raios são dinâmicos. Este mecanismo é aplicado após determinar todas as soluções não dominadas candidatas a pertencerem à população secundária. Estas são todas as soluções não dominadas do conjunto formado pela população secundária e pela população principal. Este mecanismo é apenas aplicado quando o número de soluções candidatas à população secundária (NCPS) é superior ao tamanho máximo da população secundária (NPS).

O mecanismo de partilha usado no AG elaborado consiste nos seguintes passos (adaptado a problemas bi-objectivo):

1) Inserir as soluções extremas (aquelas com os melhores valores para cada função objectivo);
2) Calcular o primeiro valor para o raio dos nichos, ds, como o quociente entre a distância normalizada entre as soluções extremas e o tamanho máximo da população secundária (isto é, $ds = \sqrt{2} / NPS$);
3) Do conjunto das soluções candidatas, inserir (transferir) aquelas que estão a uma distância superior a ds das que já pertencem à nova população secundária, usando o grau de robustez como critério de prioridade;
4) Actualizar o valor do raio dos nichos, ds, reduzindo-o em 10%;
5) Se a população secundária ainda não está completa, então regressar ao passo 3.

A Tabela 2 apresenta os resultados obtidos de um estudo efectuado ao desempenho do mecanismo apresentado de manutenção da diversidade duma população de soluções. Neste estudo foram realizadas 10 execuções de um AG caracterizado por uma população secundária de tamanho máximo igual a 50 (NPS = 50) e por executar 20.000 iterações (gerações). O campo "actualizados" indica o número de vezes que a população

secundária foi actualizada em cada uma das 10 execuções do AG (o máximo possível em cada execução é 20.000 actualizações), assim como os correspondentes valores mínimo, máximo e médio. O campo "colocados" indica o número de soluções que foram colocadas com o valor inicial do raio do nicho ($ds = \sqrt{2} / NPS$) em cada uma das 10 execuções do AG, assim como os correspondentes valores mínimo, máximo e médio. Os campos "melhor", "pior" e "média" indicam, para cada uma das 10 execuções, as percentagens mais elevada, mais baixa e média das soluções colocadas com o valor inicial do raio do nicho ($ds = \sqrt{2} / NPS$), assim como as percentagens mínima, máxima e média das 10 execuções.

	Actualizados	**Colocados**	**Melhor (%)**	**Pior (%)**	**Média (%)**
1	19.699	786.659	86	58	80
2	19.881	740.463	82	56	74
3	19.966	779.385	82	56	78
4	19.934	803.047	84	60	81
5	19.971	771.801	84	54	77
6	19.523	759.757	84	54	78
7	19.826	803.817	86	64	81
8	19.112	760.970	84	56	80
9	19.794	778.142	88	58	79
10	19.958	813.674	84	56	82
Min	19.112	740.463	82	54	74
Max	19.971	813.674	88	64	81
Média	19.767	779.772	85	58	** **Expressão incorrecta** **

Tabela 2: Resultados da aplicação do mecanismo de preservação da diversidade.

Os resultados obtidos com este mecanismo de partilha podem-se considerar bons, uma vez que são colocados com o valor inicial do raio do nicho, em média, 79% (registaram-se médias entre 74% e 81%) das NPS soluções da frente de Pareto (Tabela 2), o que indicia uma boa distribuição pela frente de Pareto logo com o valor inicial do raio do nicho. Por outro lado, a vantagem deste mecanismo, em relação aos geralmente usados (ver secção 5 do capítulo 2), está na simplicidade da sua implementação.

A Figura 4.4 mostra um exemplo da ordem de inserção das soluções numa frente de Pareto com 30 soluções, de acordo com a variação do valor do raio do nicho: para

$ds = \sqrt{2}/30$ foram inseridas 23 soluções, isto é, 77% do total de soluções *(a)*, reduzindo ds em 10% foi inserida mais 1 solução *(b)*, reduzindo novamente ds em 10% foram inseridas mais 4 soluções *(c)* e, voltando a reduzir, pela última vez, ds em 10% foram inseridas mais 2 soluções *(d)*.

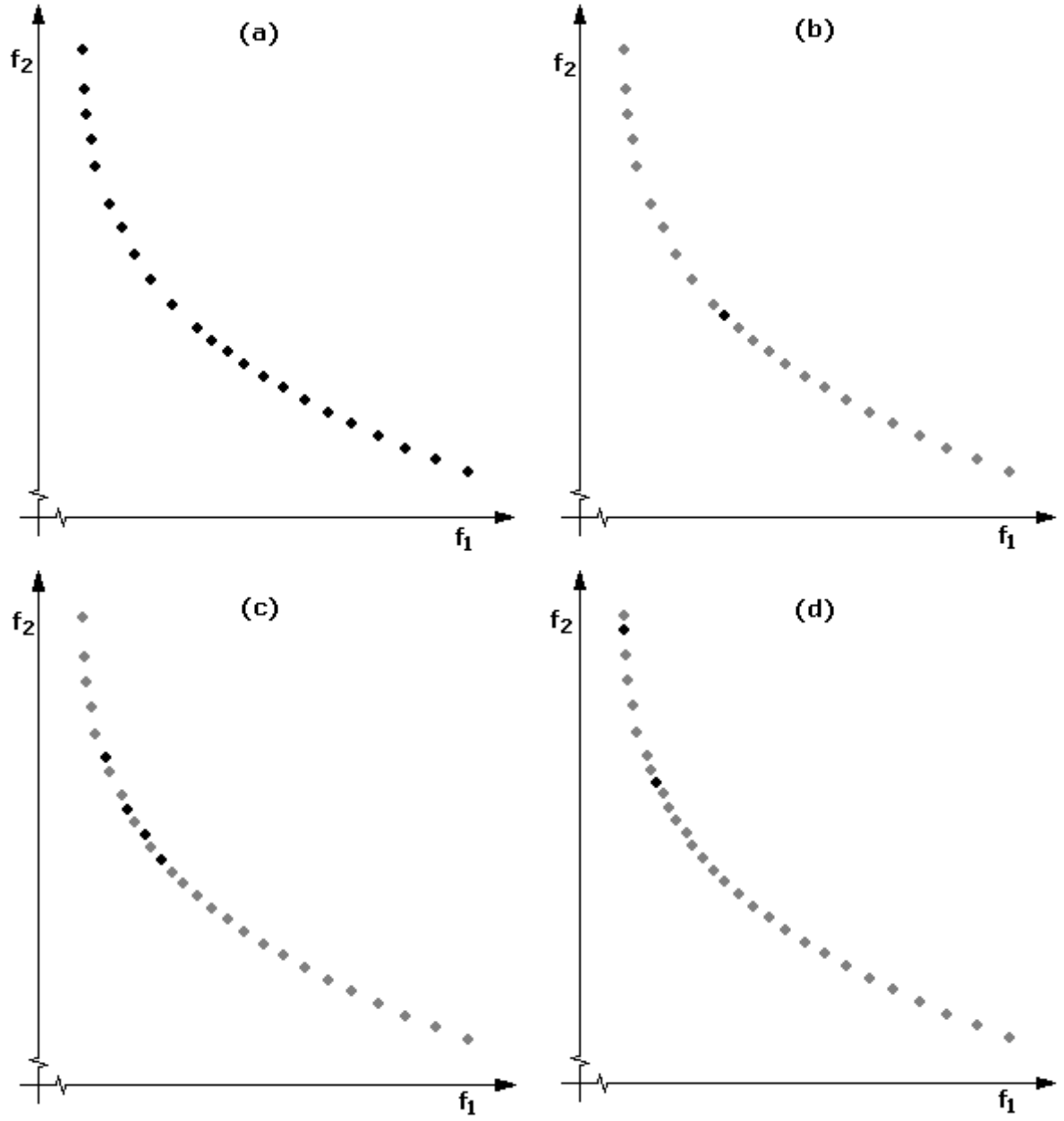

Figura 4.4: Mecanismo de preservação da diversidade populacional.

4.6.5. População inicial

A estratégia usada para determinar a população inicial consiste em gerar aleatoriamente apenas soluções admissíveis não dominadas. Este processo consiste nos seguintes passos:

1) Gerar uma solução admissível;
2) Se esta solução for não dominada relativamente à população inicial, então inseri-la nesta população e actualizar a população inicial (aplicando o teste de dominância);
3) Se a população inicial ainda não está completa, então regressar ao passo 1).

No entanto, em problemas com poucas soluções admissíveis não dominadas, a população inicial pode também conter soluções admissíveis dominadas. Para tal, usa-se um contador de soluções admissíveis geradas. Quando este contador atingir um número predefinido, se a população inicial ainda não estiver completa, então as próximas soluções admissíveis geradas são automaticamente inseridas na população inicial até esta ficar completa.

4.6.6. Descrição do algoritmo

O AG desenvolvido para testar a metodologia proposta consiste nos seguintes passos:

1) Iniciação: gerar aleatoriamente a população inicial com POP soluções admissíveis não dominadas (secção 4.6.5);
2) Determinar o grau de robustez de cada solução da população inicial;
3) Avaliação: calcular o valor da função de aptidão de cada solução da população inicial, usando o teste de dominância e o respectivo grau de robustez (secção 4.6.3);
4) Determinar a população secundária (inicial) de tamanho máximo NPS a partir da população inicial: se NPS ≥ POP então copiar todas as soluções da população inicial para a população secundária; caso contrário, aplicar o mecanismo de partilha às soluções da população inicial para seleccionar apenas NPS soluções (secção 4.6.4);
5) População actual ← População inicial;
6) Construir a população (principal) de tamanho POP, associada à nova geração:
    a) Introduzir as E soluções mais robustas da população secundária (elite) na população principal;
    b) Seleccionar por torneio 2 soluções da população actual (em cada torneio, são escolhidas aleatoriamente 10% das soluções da população actual, das quais é escolhida aquela com maior valor da função de aptidão);

c) Aplicar os operadores genéticos, cruzamento e mutação, às 2 soluções antes seleccionadas, donde resultam 2 novas soluções;
d) Inserir estas 2 novas soluções na população principal;
e) Se a população principal ainda não contém POP soluções, então regressar ao passo b);

7) Determinar o grau de robustez de cada solução da população principal;
8) Avaliação: calcular o valor da função de aptidão de cada solução da população principal, usando o teste de dominância e o respectivo grau de robustez (secção 4.6.3);
9) Determinar as NCPS soluções candidatas a pertencerem à população secundária;
10) Actualizar a população secundária: se NPS ≥ NCPS então copiar todas as NCPS soluções não dominadas candidatas à população secundária; caso contrário, aplicar o mecanismo de partilha às NCPS soluções não dominadas determinadas no passo 9) para seleccionar apenas NPS soluções (secção 4.6.4);
11) População actual ← População principal;
12) Se foi atingido o número de iterações predefinido, então STOP; caso contrário, regressar ao passo 6).

Capítulo 5

Testes Computacionais e Análise de Resultados

1. Introdução

Nesta secção é apresentado um estudo efectuado com um problema de teste, para ilustrar a aplicação do conceito de grau de robustez em problemas sujeitos a perturbações nos valores das variáveis de decisão das soluções do problema (Barrico e Antunes (2006-b), Barrico e Antunes (2006-c), Barrico e Antunes (2006-d) e Barrico e Antunes (2007)). Para tal, utilizou-se o algoritmo genético descrito na secção 4.6.2 do capítulo 4, o qual foi adaptado a um problema de teste utilizado com muita regularidade na literatura para testar algumas técnicas associadas à computação evolucionária.

O objectivo principal do algoritmo evolucionário elaborado é caracterizar a frente não dominada (eficiente) contendo as soluções não dominadas mais robustas e garantindo, também, a diversidade ao longo da frente.

Cada solução x tem associado um grau de robustez, o qual depende do comportamento das soluções pertencentes às vizinhanças de x no espaço dos objectivos, em termos de proximidade ao valor da imagem de x, $f(x)$. O AD pode especificar o tamanho da vizinhança base das soluções, tanto no espaço das variáveis de decisão, como no espaço das funções objectivo.

Com este estudo, pretende-se analisar a influência de alguns dos parâmetros de robustez, como p1, p2 e η, na determinação da frente não dominada robusta. Estes parâmetros estão associados ao nível de robustez exigido e ao limiar de indiferença entre os valores da função objectivo especificados pelo AD.

Os indivíduos da população usada na implementação do algoritmo associado ao problema de teste analisado nesta secção, são representados por uma cadeia de $B = B_1 + ... + B_n$ valores binários, em que n é o número de variáveis de decisão e B_i é o tamanho binário da representação da variável de decisão x_i.

2. Formalização matemática do problema de teste

Este problema de teste é regularmente usado na literatura de optimização em problemas de engenharia (ver Deb (2001) e Li et al. (2005)):

Minimizar $f_1(x) = x_1\sqrt{16+x_3^2} + x_2\sqrt{1+x_3^2}$

Minimizar $f_2(x) = 20\,\frac{\sqrt{16+x_3^2}}{x_1 x_3}$

Sujeito a

$20\sqrt{16+x_3^2} - 100\,x_3 x_1 \leqslant 0,$

$80\sqrt{1+x_3^2} - 100\,x_3 x_2 \leqslant 0,$

$x_1 > 0\ ,$

$x_2 \geq 0\ ,$

$1 \leq x_3 \leq 3.$

Na representação binária das variáveis de decisão, $B_1 = B_2 = 17$ e $B_3 = 6$, e Δx = (0.0001, 0.0001, 0.05) em que Δx é a variação mínima dos valores das variáveis de decisão. Desta forma, por construção B_i zeros corresponde ao valor mínimo que a variável de decisão x_i (com i = 1, 2, 3) pode assumir (isto é, (Δx_1, 0, 1), pois $x_1 > 0$), e B_i uns corresponde ao valor máximo que x_i pode assumir (isto é, (13.1072, 13.1071, 4.15)), no algoritmo genético (valores por conveniência do AG). Estes valores máximos são os considerados satisfatórios para se fazer uma análise adequada, uma vez que não se podem considerar todos os valores do domínio das variáveis de decisão x_1 e x_2 ($x_1 > 0$ e $x_2 \geq 0$); relativamente a x_3, com aquele valor máximo, algumas soluções geradas são não admissíveis.

Neste algoritmo genético é usado o cruzamento uniforme e a mutação binária, com probabilidades pc e pm, respectivamente.

Para determinar a frente não dominada, este algoritmo genético usa os seguintes valores para os parâmetros: POP = 120; NPS = 60; E = 0.1×NPS; pc = 0.95; pm = 0.1; número de iterações = 20.

3. Frente de Pareto obtida com e sem análise de robustez

A Figura 5.1 mostra a frente não dominada obtida sem considerar a análise de robustez. Nota-se uma maior concentração de soluções na região da frente não dominada onde os valores para as duas funções objectivo estão ambos próximos dos respectivos valores óptimos.

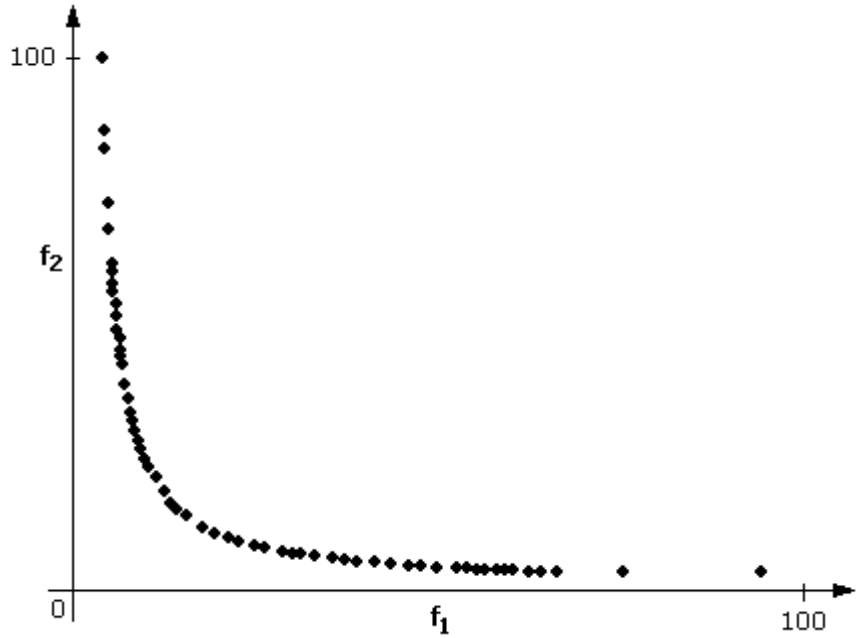

Figura 5.1: Frente de Pareto (análise sem robustez).

A Figura 5.2 mostra a frente não dominada obtida com os seguintes valores dos parâmetros associados à análise de robustez: p1 = p2 = 100%; h = 500; $\delta = (\delta_1, \delta_2, \delta_3) = (0.1, 0.02, 0.05)$; $\eta = 1.5$, o qual é usado como o limite superior para a distância normalizada absoluta entre duas soluções (de qualquer tipo) no espaço dos objectivos; e q = 1, para todo o tipo de soluções.

Nesta frente (Figura 5.2), as soluções não dominadas estão categorizadas em 3 graus de robustez (0, 1 e 2). Esta informação é relevante no ponto de vista do apoio à decisão, uma vez que o AD poderá dar preferência a uma solução não dominada de compromisso com o maior grau de robustez. Neste caso, em que existe o nível máximo de exigência (p1 = p2 = 100%), as soluções mais robustas (com grau de robustez 2) estão localizadas na região "central" da frente não dominada (zona de equilíbrio dentro da frente de soluções não dominadas), estendendo-se ligeiramente ao longo da frente não dominada em direcção a um ligeiro melhoramento de f_2 à custa de uma degradação do valor de f_1.

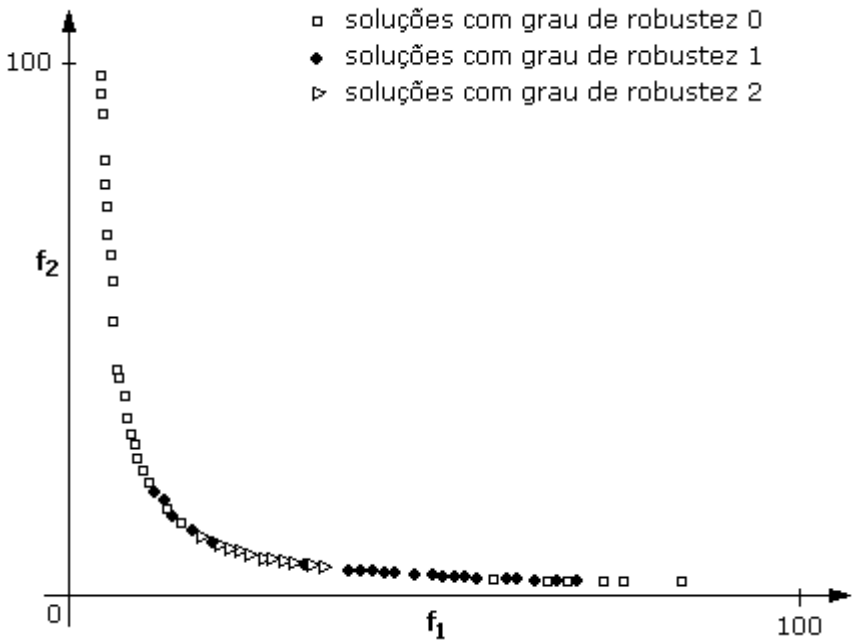

Figura 5.2: Frente de Pareto (com análise de robustez, p1 = p2 = 100% e η = 1.5).

4. Análise do parâmetro p1

O parâmetro de robustez p1 está relacionado com o número mínimo de pontos vizinhos de um dado ponto de referência (correspondente a uma solução do problema), que devem ser pontos admissíveis do problema em estudo.

A finalidade da análise isolada ao parâmetro de robustez p1 é averiguar o seu comportamento e a sua contribuição para a determinação dos graus de robustez das soluções que compõem a frente não dominada. Desta forma, o parâmetro p1 vai assumir os seguintes valores: 95% (Figura 5.3) e 85% (Figura 5.4). Todos os outros valores para os parâmetros de robustez são os mesmos usados para obter a frente representada na Figura 5.2 (p1 = 100%).

Quer a Figura 5.3 (p1 = 95%), quer a Figura 5.4 (p1 = 85%), não apresentam diferenças significativas, o que indicia uma muito baixa influência deste parâmetro no cálculo dos graus de robustez das soluções.

De qualquer forma, e como seria de esperar, as soluções com grau de robustez mais elevado encontram-se na zona "central" da frente óptima de Pareto obtida, estendendo-se ligeiramente em direcção a um ligeiro melhoramento de f_2 à custa de uma degradação do valor de f_1.

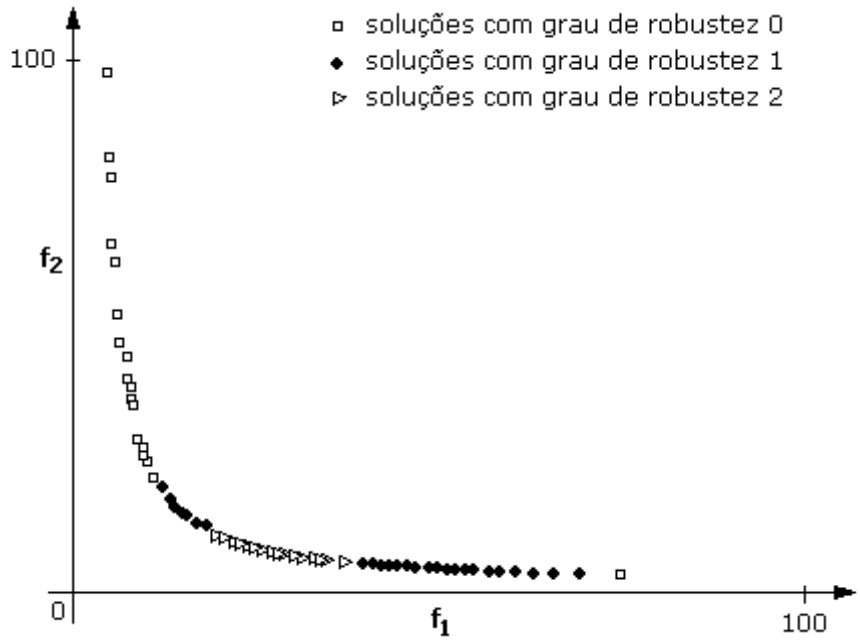

Figura 5.3: Frente de Pareto (p1 = 95%; p2 = 100%; η = 1.5).

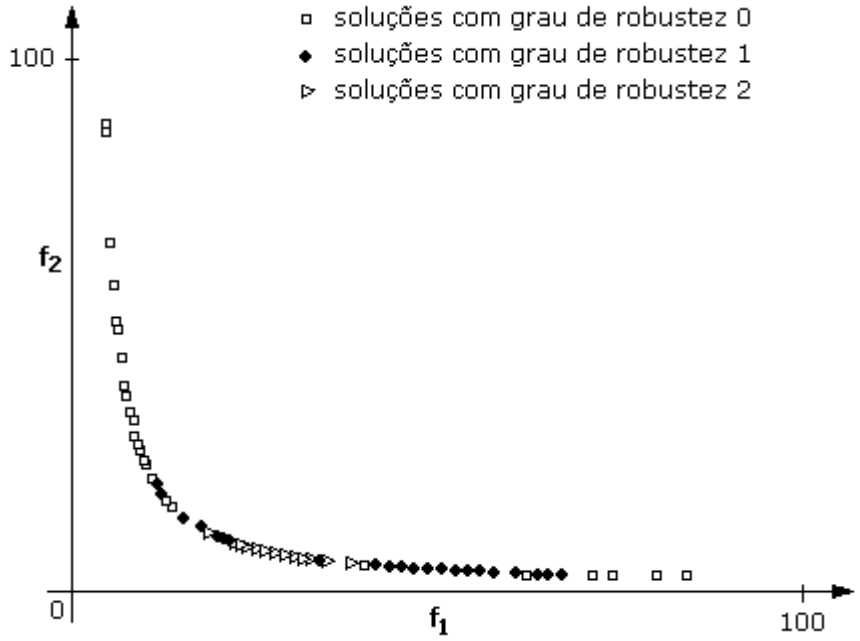

Figura 5.4: Frente de Pareto (p1 = 85%; p2 = 100%; η = 1.5).

5. Análise do parâmetro p2

Nesta secção o parâmetro de robustez p2 vai ser analisado de forma isolada, no sentido de averiguar o seu comportamento e a sua contribuição para a determinação dos graus de robustez das soluções que compõem a frente não dominada. Todos os outros valores para os parâmetros de robustez são os mesmos usados para obter a frente representada na Figura 5.2 (com p2 = 100%).

As Figura 5.5 - Figura 5.7 mostram as soluções não dominadas obtidas com diferentes valores de p2 para as soluções não dominadas (para as soluções admissíveis dominadas e não admissíveis, p2 = 100%): 95% (Figura 5.5), 85% (Figura 5.6) e 75% (Figura 5.7). Como em todos estes casos p1 = 100%, isto significa que todos os pontos vizinhos têm que ser admissíveis.

As soluções não dominadas encontradas com p2 = 95% (Figura 5.5) estão categorizadas em 3 graus de robustez (tal como para p2 = 100%), com p2 = 85% (Figura 5.6) estão categorizadas em 4 diferentes graus de robustez, e o mesmo acontecendo para p2 = 75% (Figura 5.7).

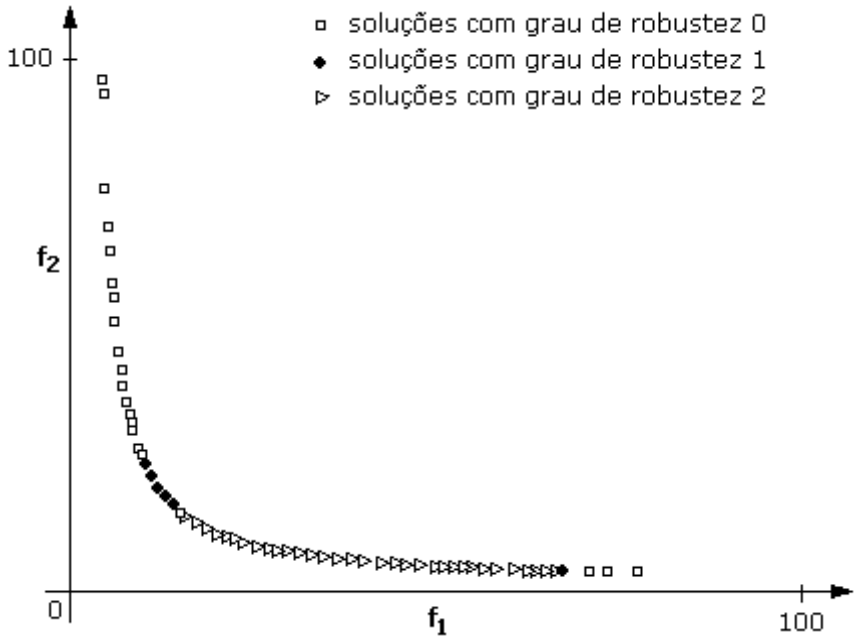

Figura 5.5: Frente de Pareto (p2 = 95%).

Ao diminuir-se o valor do parâmetro p2 de 100% para 95%, verifica-se na frente de Pareto obtida um aumento de soluções com grau de robustez 2 à custa da diminuição das soluções com grau de robustez 1, mantendo-se as zonas onde existiam soluções de grau de robustez 0 com soluções do mesmo grau e reforçando a tendência de estender as soluções mais robustas ao longo da fronteira não dominada em direcção a um ligeiro

melhoramento de f_2 (numa região onde os valores de f_2 não estão muito longe dos seus óptimos) à custa de uma degradação dos valores de f_1 (Figura 5.5).

Com a diminuição do valor de p2 de 95% para 85%, as soluções não dominadas com o grau de robustez mais elevado (grau 3) estão localizadas na região "central" da frente, mas estendendo-se ligeiramente em direcção ao óptimo de f_2 (Figura 5.6). Neste caso, existem soluções não dominadas possuindo um grau de robustez 3, o que não acontecia com os valores anteriores de p2 testados (100% e 95%).

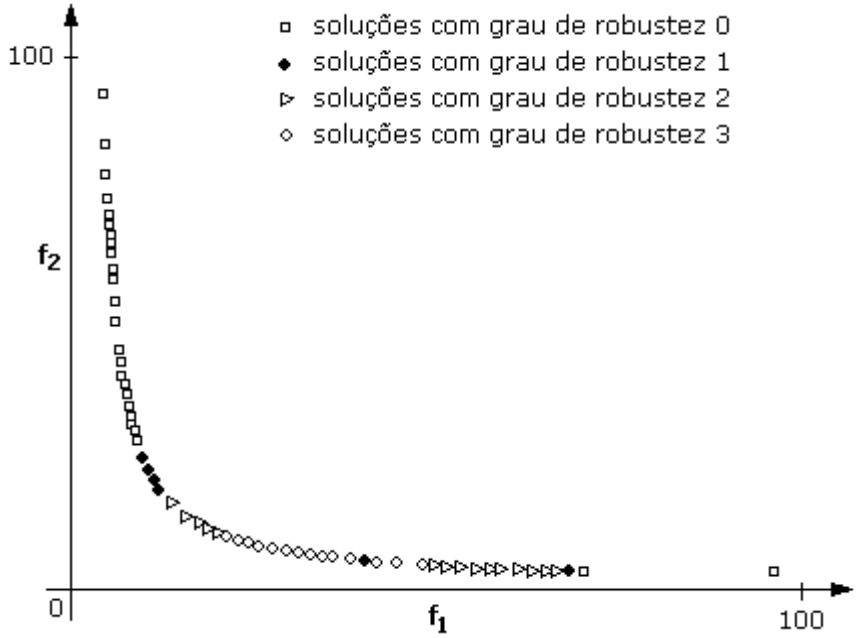

Figura 5.6: Frente de Pareto (p2 = 85%).

Finalmente, na transição de p2 de 85% para 75% (Figura 5.7), nota-se claramente um aumento do número de soluções com grau de robustez 3, essencialmente à custa da diminuição das soluções com grau de robustez 2, as quais continuam a localizar-se próximas do óptimo de f_2, mas agora algumas delas ainda mais próximas deste valor.

Nesta frente aparecem algumas, embora poucas, soluções com grau de robustez 2, as quais se encontram na região "central" da frente.

Um estudo deste tipo pode fornecer ao AD informação detalhada relativamente à escolha de uma solução de compromisso (entre os valores das funções objectivo), tendo em conta o grau de robustez das soluções. Neste exemplo, as soluções não dominadas que privilegiam f_1 possuem um grau de robustez baixo. O grau de robustez mais elevado encontra-se numa região de soluções bem equilibradas, mas mais próximas do óptimo de f_2. Esta informação é relevante para o AD quando avalia o mérito das soluções não

dominadas em competição, não se limitando apenas aos valores das funções objectivo, mas também aos seus graus de robustez.

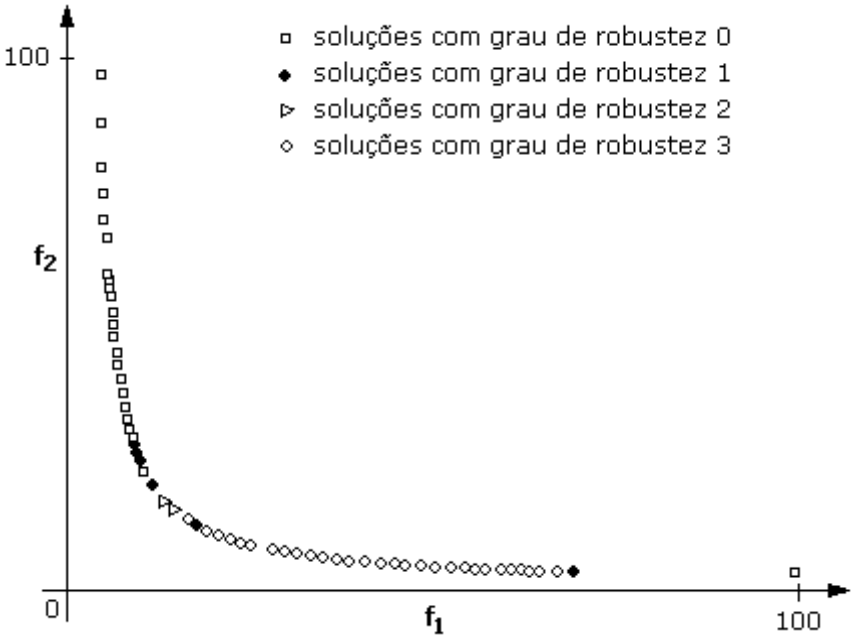

Figura 5.7: Frente de Pareto (p2 = 75%).

6. Análise dos parâmetros p1 e p2 em conjunto

Nesta secção, pretende-se estudar o comportamento e a influência dos parâmetros de robustez p1 e p2 em conjunto, isto é, manipulados como um só parâmetro (designado por p1⊕p2). Desta forma, os pontos analisados que não satisfaçam as condições associadas a p1 e p2 são contabilizados uma só vez, e não duas, como aconteceria se estes parâmetros fossem analisados em separado.

Considera-se que uma solução *y*, pertencente à δ-vizinhança de *x*, não cumpre os requisitos impostos, se é não admissível ou, sendo admissível, a sua imagem no espaço dos objectivos não pertence à η-vizinhança da imagem de *x*.

Desta forma, pretende-se estudar o comportamento da frente óptima de Pareto resultante da aplicação do AG, em que p1⊕p2 assume vários valores (na Figura 5.2, p1 = p2 = 100%, o que significa que p1⊕p2 = 100%): 95% (Figura 5.8) e 85% (Figura 5.9).

Analisando a Figura 5.8 (p1⊕p2 = 95%) e comparando-a com a frente da Figura 5.2 (p1⊕p2 = 100%), pode-se verificar que, apesar do grau de robustez mais elevado (2) presente em soluções da frente ser igual ao apresentado pelas soluções da frente da Figura 5.2, há, no entanto, um aumento do número de soluções com grau de robustez 2 (o mais elevado) à custa da diminuição das soluções com grau de robustez 1.

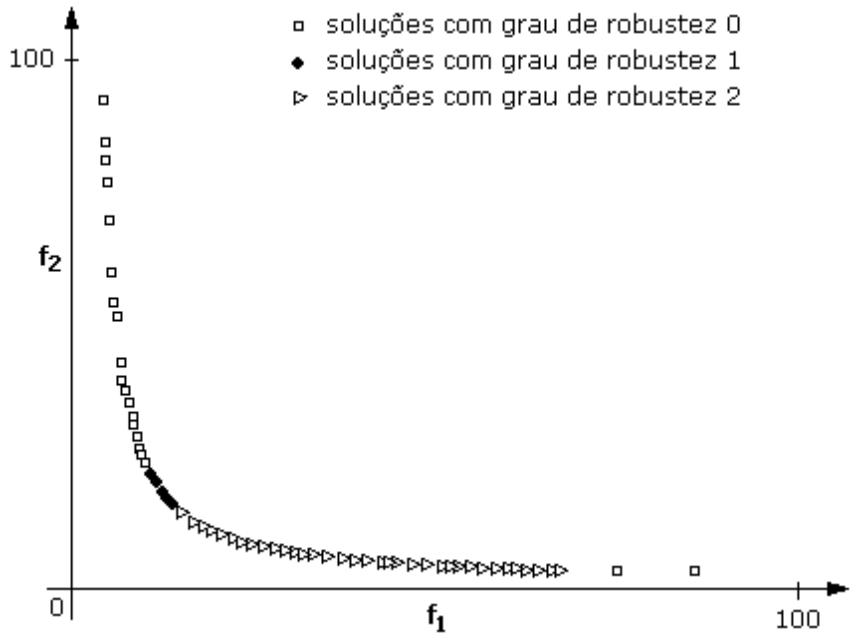

Figura 5.8: Frente de Pareto (p1⊕p2 = 95%).

De referir que estas soluções (com grau de robustez mais elevado) continuam a encontrar-se na zona "central" da frente, estendendo-se muito para zonas próximas do óptimo de f_2. As soluções de grau de robustez 1 existem em número limitado e encontram-se apenas numa zona "central", mas a estenderem-se ligeiramente em direcção ao óptimo de f_1. As soluções de grau de robustez 0 mantém-se tal como na Figura 5.2, quer em número, quer nas zonas que ocupam (estas soluções localizam-se nas regiões extremas, isto é, junto aos óptimos individuais de cada função objectivo, onde os valores da outra função objectivo são piores, embora a grande maioria se estenda por zonas próximas do óptimo de f_1).

Ao diminuir-se o valor de p1⊕p2 de 95% para 85% (Figura 5.9), verifica-se o aparecimento de soluções com grau de robustez máximo igual a 3, superior ao grau de robustez máximo associado às soluções da frente obtida para p1⊕p2 = 95% (Figura 5.8).

As soluções com grau de robustez mais elevado (3) surgiram numa zona onde predominavam soluções de grau de robustez 2 na frente obtida com p1⊕p2 = 95%. As soluções de grau de robustez 3 foram obtidas à custa da diminuição das soluções de grau de robustez 2.

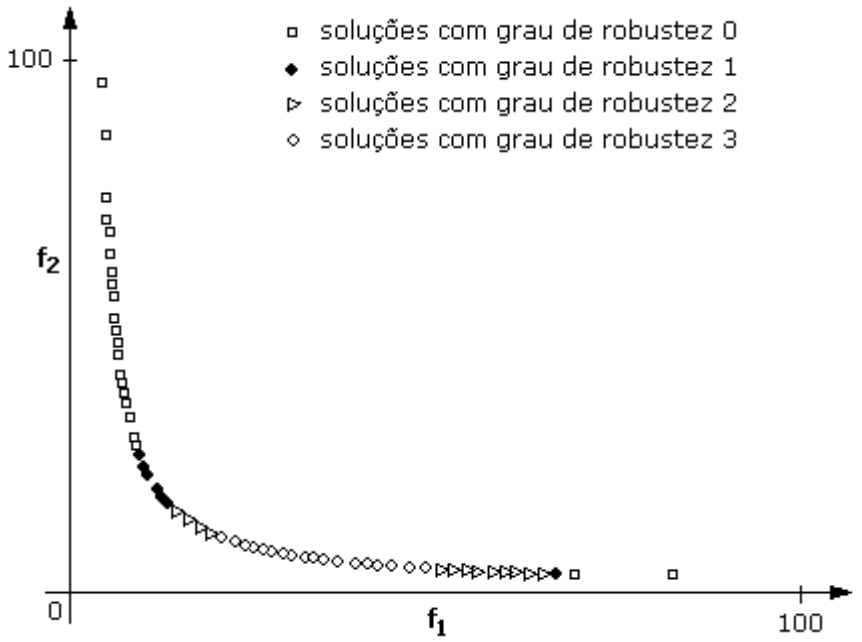

Figura 5.9: Frente de Pareto (p1⊕p2 = 85%).

As soluções de grau de robustez 3 (o mais elevado) ocupam agora uma zona da frente entre as soluções de grau de robustez 2: uma zona da frente ligeiramente fora da zona "central", e estendendo-se em direcção ao óptimo de f_2, e uma outra estendendo-se ainda mais em direcção ao valor óptimo de f_2, mas mais afastada do valor óptimo de f_1.

As soluções de grau de robustez 2 ocupam agora duas zonas: uma central e uma outra que se estende em direcção ao valor de óptimo de f_2 (ainda mais afastado do valor óptimo de f_1 do que as soluções de grau de robustez 3).

As soluções de graus de robustez 1 e 0 mantêm em número e zonas de ocupação, tal como na frente obtida com p1⊕p2 = 95% (Figura 5.8).

7. Análise do parâmetro η

Nesta secção vai ser analisada a influência do parâmetro η no cálculo do grau de robustez associado às soluções. O valor do parâmetro η corresponde ao limite superior da distância normalizada absoluta entre duas soluções (de qualquer tipo) no espaço dos objectivos. Note-se que a especificação do parâmetro η não impõe uma excessiva sobrecarga ao AD, uma vez que o espaço das funções objectivo é o que lhe é mais familiar. Em princípio, o AD é capaz de fornecer informação sobre estes limiares para os quais a diferença entre as soluções em cada dimensão não é relevante para o estabelecimento de uma preferência clara.

As Figura 5.10 - Figura 5.13 mostram as frentes não dominadas obtidas com diferentes valores do parâmetro η: 1.0 (Figura 5.10), 0.5 (Figura 5.11), 2.0 (Figura 5.12) e 2.5 (Figura 5.13). Todos os outros parâmetros associados à análise de robustez têm os mesmos valores usados na obtenção da frente apresentada na Figura 5.2 (η = 1.5).

Ao alterar-se o valor de η de 1.5 para 1.0 (Figura 5.10), a frente não dominada obtida é composta por soluções com 2 graus de robustez (0 e 1), ao passo que na frente obtida com η = 1.5 (Figura 5.2) existem soluções com 3 graus de robustez (0, 1 e 2). Verifica-se, na frente agora obtida, um aumento do número de soluções de grau de robustez 1, ocupando as zonas onde estavam as soluções de graus de robustez 1 e 2.

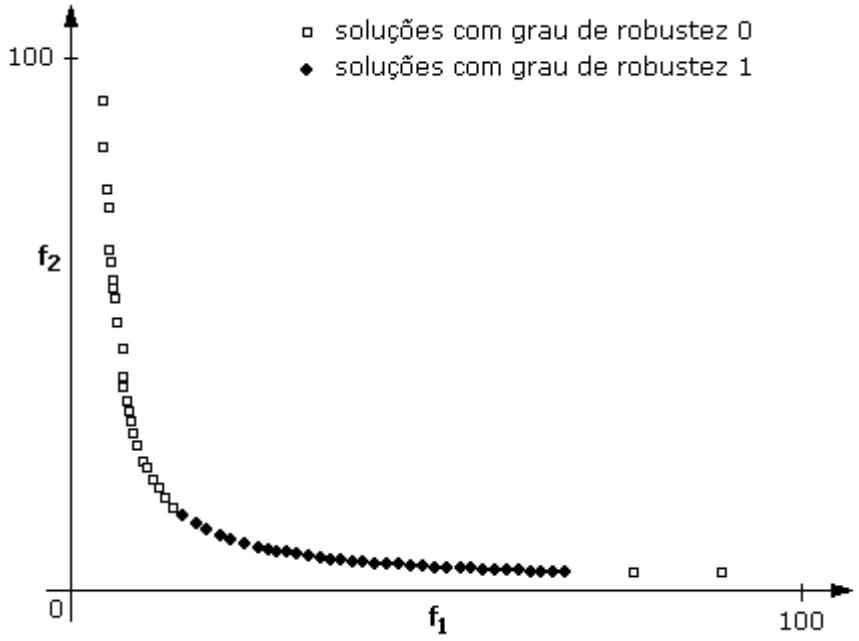

Figura 5.10: Frente de Pareto (η = 1.0).

Isto acontece porque a diminuição do valor do parâmetro η implica um acréscimo da importância associada às diferenças entre os valores das funções objectivo (acima de um limiar) e, assim, à contracção da vizinhança na qual o AD é indiferente entre soluções não dominadas localizadas dentro desta vizinhança.

Contraindo ainda mais a vizinhança das soluções no espaço dos objectivos, passando para η = 0.5 (Figura 5.11), verifica-se que todas as soluções não dominadas da frente obtida têm o mesmo grau de robustez (0). Neste caso, as diferenças nos valores das funções objectivo tornam-se mais importantes para o AD.

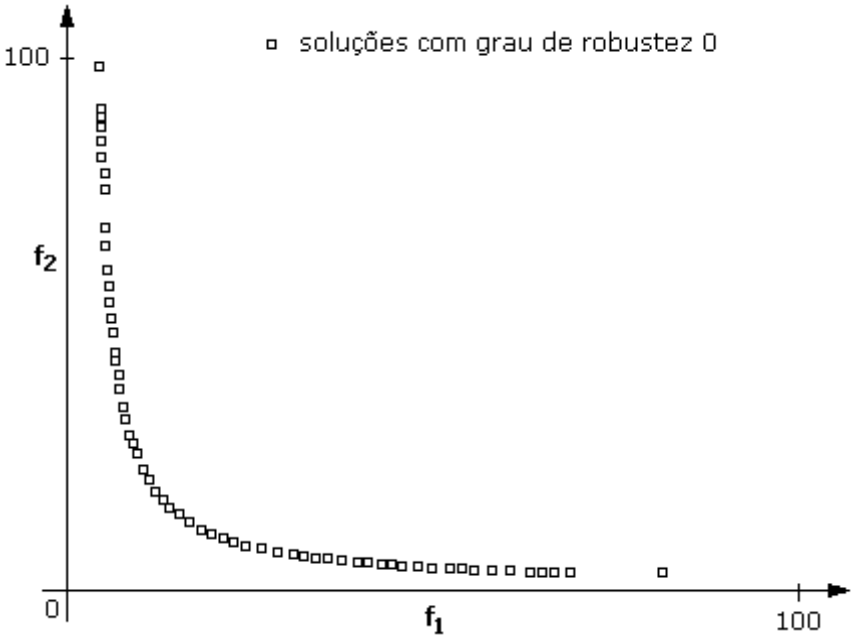

Figura 5.11: Frente de Pareto (η = 0.5).

De salientar que o aumento da exigência, traduzido pela contracção da vizinhança das soluções no espaço dos objectivos, pode ter como consequência uma maior dificuldade na escolha das "melhores" soluções, uma vez que provoca, em geral, uma diminuição do número das soluções mais robustas, podendo levar a uma frente uniforme em termos de robustez.

Na transição do valor de η de 1.5 para 2.0 (Figura 5.12), obteve-se uma frente não dominada em que o número de soluções com grau de robustez mais elevado (2) aumentou muito, essencialmente à custa da diminuição do número de soluções com grau de robustez 1.

Na frente obtida com η = 2.0 (Figura 5.12), as soluções com maior grau de robustez (2) encontram-se na zona típica das soluções mais robustas para este problema, isto é, na região "central" e a estenderem-se muito em direcção ao valor óptimo de f_2. As soluções com grau de robustez 1 encontram-se praticamente todas na região "central" da frente, mas ligeiramente deslocadas na direcção do valor óptimo de f_1. Por fim, as soluções de menor grau de robustez localizam-se, praticamente todas, junto ao óptimo de f_1, possuindo também a solução que optimiza f_2 grau de robustez 0.

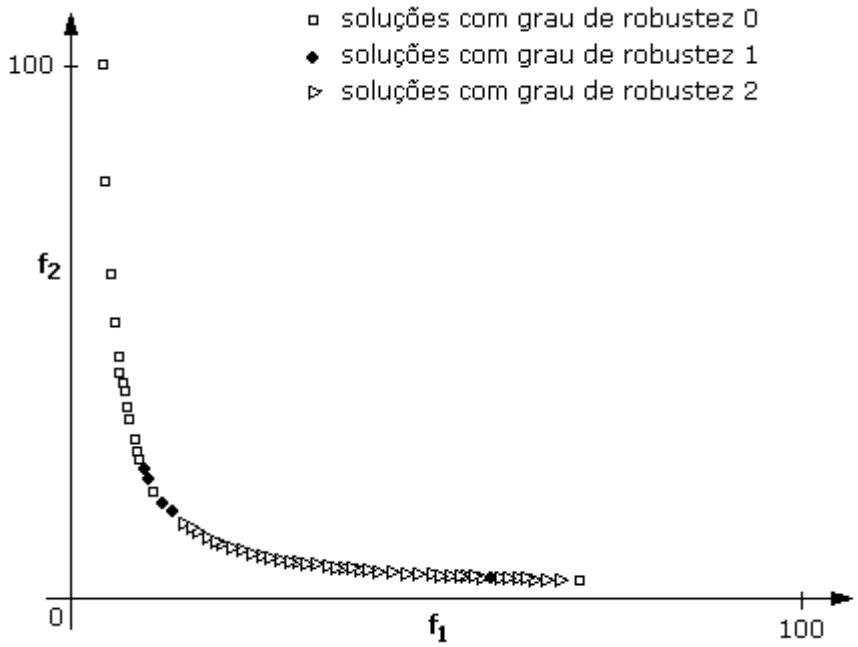

Figura 5.12: Frente de Pareto (η = 2.0).

Quando o valor de η passa de 2.0 para 2.5 (Figura 5.13), verifica-se o aparecimento de soluções com grau de robustez 3, as quais derivam da diminuição do número de soluções com grau de robustez 2 (da frente obtida com η = 2.0, Figura 5.12). Isto acontece porque o aumento do valor do parâmetro η implica um relaxamento na importância associada às diferenças entre os valores das funções objectivo (abaixo de um limiar) e, assim, ao alargamento da vizinhança na qual o AD é indiferente entre soluções não dominadas localizadas dentro desta vizinhança.

Nesta frente, as soluções de grau de robustez mais elevado (3) ocupam uma zona da frente de Pareto entre as soluções de grau de robustez 2: uma zona da frente ligeiramente fora da zona "central" e a estender-se em direcção ao valor óptimo de f_2, e uma outra ainda mais próxima do óptimo de f_2 mas muito afastada do valor óptimo de f_1.

As soluções de grau de robustez 2 ocupam agora duas zonas: uma central e uma outra que se estende em direcção ao valor de óptimo de f_2, mais afastada do valor óptimo de f_1 do que a das soluções de grau de robustez 3.

As soluções de graus de robustez 1 e 0, mantêm-se em número e zonas de ocupação, tal como na frente obtida com η = 2.0 (Figura 5.12). As soluções com grau de robustez 1 encontram-se na zona "central", a estenderem-se ligeiramente em direcção ao óptimo de f_1. As soluções com grau de robustez 0 encontram-se nas extremidades da frente de Pareto.

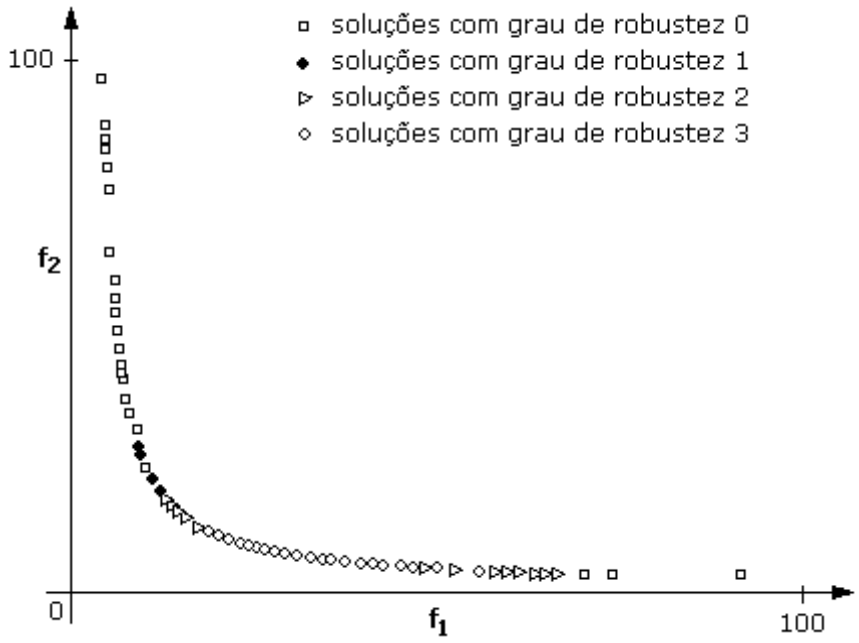

Figura 5.13: Frente de Pareto (η = 2.5).

Em resumo, podemos concluir que as soluções com o grau de robustez mais elevado estão geralmente localizadas numa região da frente não dominada que apresenta valores para f_2 não muito longe do seu valor óptimo e para valores de f_1 no centro da gama dos valores de f_1 alcançados dentro da frente não dominada. Este tipo de conhecimento, proporcionado pela atribuição de um grau de robustez às soluções não dominadas, é uma informação relevante para um AD avaliar o respectivo mérito na escolha de uma solução de compromisso satisfatória.

8. População secundária só com soluções robustas

Nesta secção, pretende-se averiguar a influência do uso da População Secundária apenas com soluções não dominadas robustas (com graus de robustez superior a 0). Testaram-se vários casos, todos com o parâmetro η = 1.5, mas para valores diferentes de p1 e p2: p1 = p2 = 100% (Figura 5.14), p1 = p2 = 95% (Figura 5.15), e p1 = 95% e p2 = 85% (Figura 5.16). Os restantes parâmetros têm os mesmos valores usados para obter a frente de Pareto apresentada na Figura 5.2.

Comparando as frentes de Pareto só com soluções robustas obtidas para os três testes efectuados (Figura 5.14 - Figura 5.16), verifica-se que à medida que se diminuem os valores de p1 e p2, o que implica um relaxamento, por parte do AD, na fiabilidade dos graus de robustez calculados, aumenta o número de soluções com o grau de robustez mais elevado. De facto, ao passar-se da frente da Figura 5.14 para a frente da Figura

5.15, há um aumento do número de soluções com grau de robustez mais elevado (2). Ao passar-se desta última frente para a frente apresentada na Figura 5.16, verifica-se o aparecimento de soluções com grau de robustez 3 (o mais elevado nesta frente), que foi conseguido à custa da diminuição do número de soluções com grau de robustez 2.

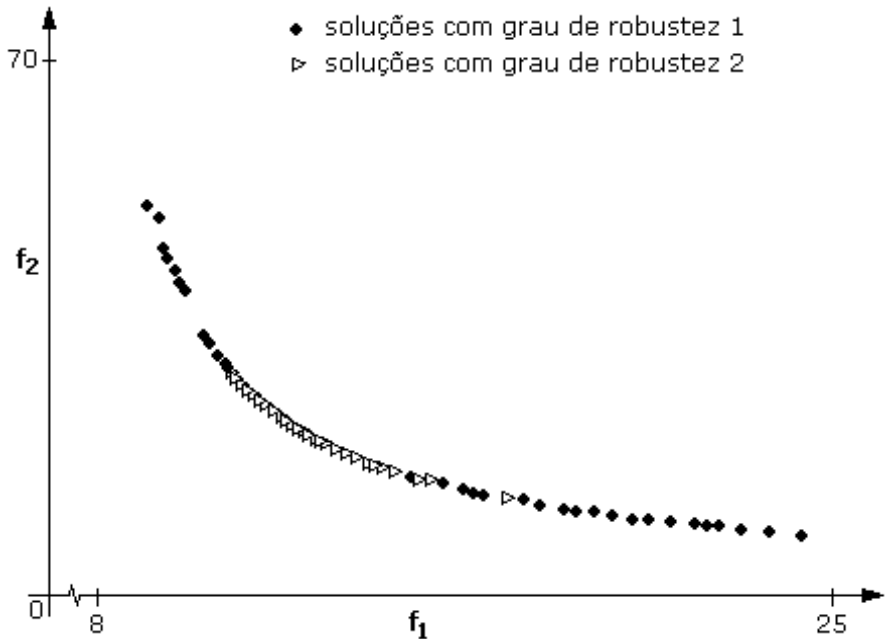

Figura 5.14: Frente de Pareto só com soluções robustas (p1 = p2 = 100%).

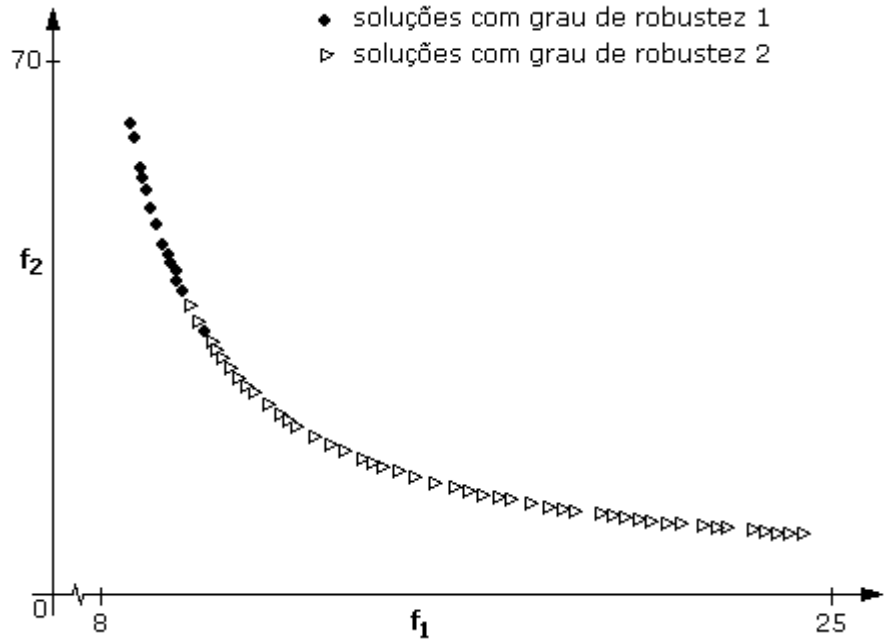

Figura 5.15: Frente de Pareto só com soluções robustas (p1 = p2 = 95%).

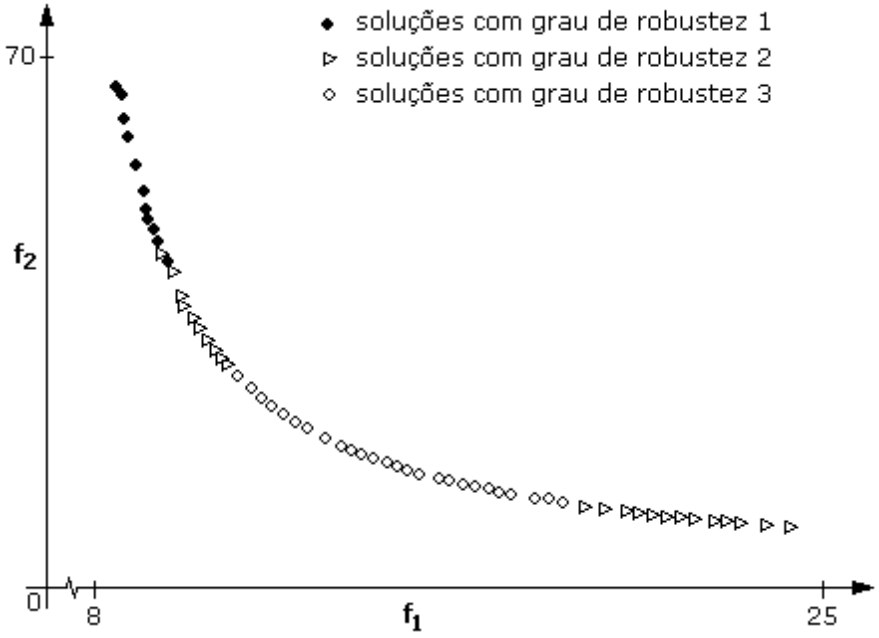

Figura 5.16: Frente de Pareto só com soluções robustas (p1 = 95%; p2 = 85%).

Analisando as frentes de Pareto só com soluções robustas obtidas, podemos concluir que as soluções mais robustas se situam na zona "central" das frentes a descair para uma zona que apresenta valores muito próximos do óptimo de f_2. Esta tendência é nítida nas frentes representadas na Figura 5.14 e Figura 5.16, embora não o seja totalmente na frente representada na Figura 5.15, o que leva a reforçar a ideia de que uma escolha adequada dos parâmetros é essencial para diferenciar as soluções de uma frente óptima de Pareto, em termos de robustez.

9. Utilização de métricas

Nos exemplos anteriormente apresentados, a métrica usada na determinação da distância entre duas soluções (de qualquer tipo) no espaço dos objectivos foi a distância normalizada absoluta. No entanto, outros tipos de métricas podem ser usados, tais como a distância normalizada relativa ou a distância não normalizada (absoluta ou relativa).

A Figura 5.17 apresenta a frente não dominada obtida usando como métrica a distância normalizada relativa. Os valores dos parâmetros são os mesmos usados na obtenção da frente apresentada na Figura 5.2, com excepção do valor do parâmetro η (η = 0.05, que significa 5% do valor de referência). Neste caso, as soluções mais robustas estão localizadas próximas dos melhores valores de f_2, embora à custa de uma pequena degradação dos valores de f_1.

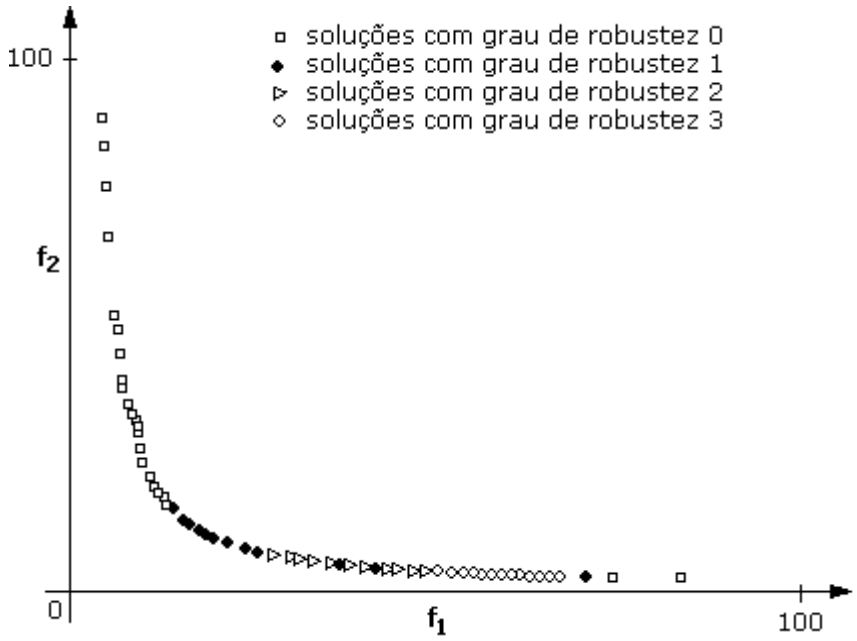

Figura 5.17: Frente de Pareto (usando a distância normalizada relativa).

A Figura 5.18 mostra a frente de Pareto obtida usando como métrica a distância não normalizada absoluta. Os valores dos parâmetros são os mesmos usados na obtenção da frente apresentada na Figura 5.2, com excepção do valor de η: $\eta = (\eta_1, \eta_2) = (1.5, 1.5)$. Nesta frente, as soluções mais robustas possuem as mesmas características mencionadas antes (frente apresentada na Figura 5.17): as soluções apresentam valores muito bons para f_2 e valores de f_1 no centro da sua gama na frente não dominada.

A Figura 5.19 mostra a frente não dominada obtida usando como métrica a distância não normalizada relativa. Os valores dos parâmetros são os mesmos usados na obtenção da frente apresentada na Figura 5.2, com excepção para o valor do parâmetro η. Para soluções admissíveis não dominadas, $\eta = (\eta_1, \eta_2) = (0.06, 0.08)$, o que significa 6% e 8% dos valores de referência de f_1 e f_2, respectivamente, e para as soluções admissíveis dominadas e não admissíveis $\eta = (\eta_1, \eta_2) = (0.03, 0.04)$. Nesta frente, as soluções mais robustas possuem as mesmas características mencionadas antes (frente apresentada na Figura 5.17): as soluções apresentam valores para f_2 próximos do seu óptimo e valores de f_1 no centro da sua gama na frente não dominada (embora a tender para os piores valores de f_1).

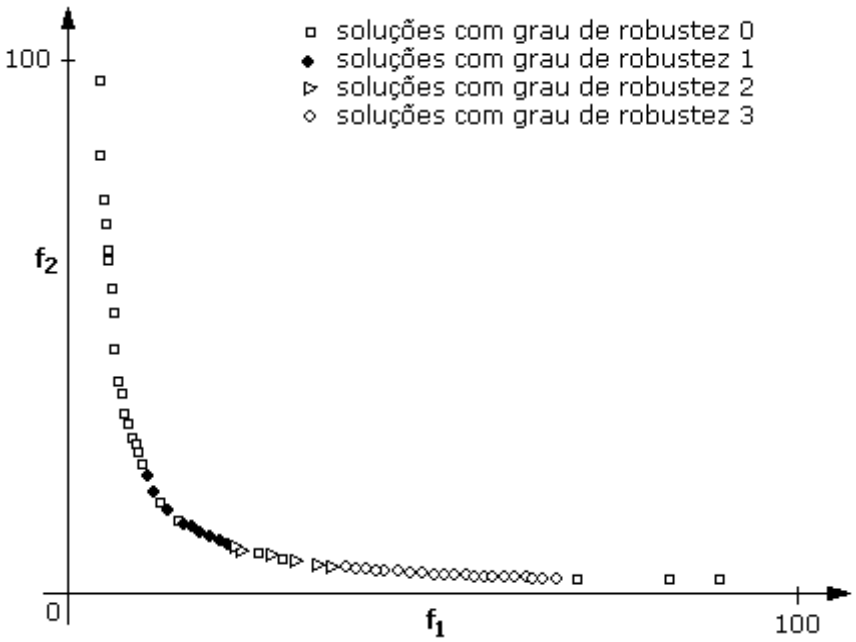

Figura 5.19: Frente de Pareto (usando a distância não normalizada relativa).

Capítulo 6

Estudo de Caso — Compensação de energia reactiva em redes radiais de distribuição

1. Introdução

Este capítulo apresenta duas abordagens de optimização evolucionária multi-objectivo a um problema de compensação de energia reactiva em redes radiais de distribuição de energia eléctrica. O modelo matemático usado para resolver este problema considera explicitamente duas funções objectivo, a minimização do investimento requerido para o equipamento de compensação e a minimização das perdas (ver Pires (2001) e Pires et al. (2005)).

Na primeira abordagem os valores nominais dos coeficientes das funções objectivo mantém-se inalterados ao longo do processo de obtenção da frente não dominada (ver Antunes et al. (2004) e Antunes et al. (2005)).

Na segunda abordagem faz-se uma análise de robustez das soluções, considerando que os valores nominais de alguns dos dados associados às funções objectivos e às restrições estão sujeitos a perturbações (ver Barrico e Antunes (2006-a) e Barrico e Antunes (2006-d)).

2. O problema da compensação de energia reactiva em redes de distribuição

2.1. Introdução

A compensação de potência reactiva é um importante problema nos sistemas de energia eléctrica, o qual envolve aspectos operacionais, económicos e de qualidade de serviço. Os consumidores de energia eléctrica (domésticos, industriais, sector dos serviços, etc.) necessitam de potência activa e reactiva, de acordo com as características das cargas.

A função principal de uma rede de distribuição de energia eléctrica é a de fornecer, em cada instante, as potências activa e reactiva requerida pelas cargas, respeitando condições técnicas e de qualidade de serviço. A potência activa é convertida em energia "útil", tal como luz ou calor. A potência reactiva é produzida ou absorvida pela maioria dos componentes do sistema de energia, como geradores, elementos de ligação (linhas e transformadores), cargas e dispositivos de compensação.

A procura de energia reactiva deve-se ao seu consumo por alguns tipos de carga e às perdas do próprio sistema. A energia reactiva deve ser, então, compensada para garantir uma distribuição eficiente da energia activa pelas cargas, reduzindo as perdas e melhorando o factor de potência e o perfil de tensões do sistema. A obtenção destas metas depende da dimensão e da localização dos condensadores (fontes de potência reactiva).

Os dispositivos de compensação mais utilizados em redes de distribuição são os condensadores "shunt" (isto é, condensadores ligados em paralelo com as linhas) fixos ou mecanicamente comutados, uma vez que são os que acarretam menos custos. A função fundamental dos condensadores usados nas redes de distribuição, instalados como unidade única ou em banco, é a de regular a tensão e o fluxo de potência reactiva no ponto onde estão instalados. Os condensadores "shunt" conseguem-no através da mudança do factor de potência da carga, enquanto que os em série fazem-no por compensação directa da reactância indutiva do circuito ao qual estão aplicados. Os condensadores "shunt" fornecem a corrente reactiva requerida pelas cargas indutivas responsáveis pela componente desfasada da corrente do sistema.

Em termos tecnológicos, pode-se transferir potência activa ao longo de grandes distâncias, o que não deve acontecer com a potência reactiva. A potência reactiva deve ser injectada, preferencialmente, a pequenas distâncias do local de consumo, uma vez que o seu transporte implica um aumento nas perdas de transmissão. Assim sendo, a transferência de potência reactiva deve ser minimizada, para reduzir a corrente e evitar

quedas de tensão. Correntes menores implicam perdas activas e reactivas também menores, e a redução de perdas activas representa um ganho económico directo devido à redução do custo de produção.

Assim, é justificada a opção habitual de colocar os condensadores junto dos pontos de carga, uma vez que não é viável, em termos económicos, gerar potência reactiva na central de produção de energia eléctrica para ser entregue a longa distância, nas redes de distribuição, junto da carga.

Dada uma configuração de carga num determinado instante, a análise do trânsito de potência no sistema permite conhecer as grandezas eléctricas em cada ponto do sistema. Deste modo, é possível analisar, quer as condições de funcionamento nesse instante, quer as modificações resultantes de qualquer alteração (por exemplo, a resultante da colocação de baterias de condensadores nalguns nós da rede).

2.2. Colocação de condensadores em sistemas eléctricos

Na literatura científica existe um grande número de modelos e de abordagens metodológicas dedicados ao problema da compensação de energia reactiva. Entre os vários tipos de abordagens para tratar este problema, incluem-se os algoritmos de programação matemática e, mais recentemente, as meta-heurísticas. Em particular, tem sido dada especial atenção a algoritmos genéticos e a abordagens evolucionárias, cujas características são adequadas à obtenção de soluções para problemas com estrutura de rede: Iba (1994), Lee et al. (1995), Kim e You (1999), Levitin et al. (2000), Delfanti et al. (2000), Baran et al. (2001) e Pires et al. (2005).

Nesta secção é apresentada uma abordagem evolucionária para determinar um conjunto de soluções não dominadas do problema da compensação de potência reactiva em redes radiais de distribuição eléctrica.

O modelo matemático usado considera explicitamente duas funções objectivo: minimização dos custos de instalação dos condensadores e minimização das perdas resistivas do sistema. As restrições do modelo estão relacionadas com o nível de tensão requerida nos nós (qualidade de serviço), trânsito de potência (leis físicas em redes eléctricas) e impossibilidade da colocar condensadores nalguns nós (restrições técnicas). A finalidade é identificar as soluções não dominadas do modelo, através da determinação do número e dos locais onde instalar as baterias de condensadores, assim como da dimensão de cada bateria de condensadores a ser instalada, de forma a reduzir as perdas do sistema e melhorar o perfil de tensões. No entanto, é necessário um equilíbrio entre os custos (associados à instalação de novos condensadores) e os aspectos de avaliação técnica e de qualidade de serviço (ver Baran e Wu (1989-a), Baran e Wu (1989-b), Das

et al. (1994) e Pires et al. (2005)). Para mais detalhes sobre este modelo matemático consultar o Anexo.

Existe uma dupla motivação para se usar esta abordagem metodológica. Por um lado, a capacidade de trabalhar, em cada geração, com uma população de potenciais soluções torna as abordagens evolucionárias adequadas a problemas de optimização multi-objectivo, nas quais se pretende identificar um conjunto de soluções não dominadas em vez de uma única solução óptima. Por outro, este problema de natureza combinatória é muito complexo para ser resolvido com ferramentas de programação matemática, uma vez que, para além de considerar múltiplas funções objectivo, é não linear com variáveis contínuas, inteiras e binárias.

A abordagem evolucionária implementada inclui uma estratégia elitista com o objectivo de acelerar a convergência para a frente óptima de Pareto, garantindo que as soluções atingidas são de facto não dominadas e que estão bem distribuídas pela frente. Este aspecto é muito importante em problemas reais, uma vez que é necessário fornecer ao AD soluções bem distribuídas e diversificadas que lhe permita diferentes compromissos para alcançar uma decisão final bem fundamentada.

3. Abordagem sem análise de robustez

3.1. Algoritmo genético

A população usada na implementação do algoritmo genético consiste num conjunto de indivíduos representados por uma cadeia de NN valores inteiros, em que NN é o número de nós da rede eléctrica em estudo. Cada índice da cadeia corresponde a um nó da rede e o valor que lhe está associado identifica o tipo de condensador a instalar nesse nó (o tipo de condensador está indexado por uma cadeia de valores entre 0 e Y, em que 0 significa que não existe qualquer condensador instalado naquele nó; isto é, existem Y condensadores de dimensões diferentes que podem ser instalados na rede).

Usou-se o operador genético cruzamento de dois pontos de corte com uma probabilidade pc. Os dois pontos de corte são escolhidos aleatoriamente com a restrição de estarem a uma distância entre si superior a 1/4 do tamanho do cromossoma (o tamanho de cada cromossoma é NN, ou seja, o número de nós da rede).

Como o tipo de condensador é identificado por um valor inteiro entre 0 e Y, o operador genético mutação consiste em modificar, com uma probabilidade pm, o valor actual por um dos outros Y valores possíveis.

Relativamente à população inicial, à função de aptidão e ao mecanismo de partilha usados, estes são os descritos na secção 4.6.2 do capítulo 4, com as devidas alterações, uma vez que nesta primeira abordagem não é considerada a análise de robustez.

Na implementação do algoritmo genético dedicado ao problema da compensação de potência reactiva, é usada uma estratégia elitista com uma população secundária de tamanho máximo constante, consistindo nos seguintes passos:

1) Iniciação: gerar aleatoriamente a população inicial com POP soluções não dominadas;
2) Avaliação: determinar o valor da função de aptidão de cada indivíduo da população inicial;
3) Determinar a população secundária inicial de tamanho máximo NPS a partir da população inicial: se NPS ≥ POP, então a população secundária é composta por todos os indivíduos da população inicial; caso contrário, aplicar um mecanismo de partilha aos indivíduos da população inicial para seleccionar NPS soluções;
4) População actual ← População inicial
5) Construir a população (principal) de tamanho POP, associada à nova geração:
    a) Introduzir E indivíduos da população secundária (elite) directamente na população principal;
    b) seleccionar, por torneio, 2 indivíduos da população actual (em cada torneio, escolhem-se aleatoriamente 10% dos indivíduos da população actual e determina-se o melhor deles);
    c) Aplicar os operadores genéticos cruzamento e mutação aos 2 indivíduos seleccionados e inseri-los na população principal;
    d) Se a população principal ainda não está completa então regressar ao passo b);
6) Avaliação: aplicar o teste de dominância e calcular o valor da função de aptidão de cada indivíduo da população principal;
7) Determinar as soluções não dominadas candidatas a pertencerem à população secundária (são todas as soluções não dominadas da população principal que não são dominadas por soluções da população secundária, e vice-versa);
8) Actualizar a população secundária, aplicando um mecanismo de partilha se o número de soluções candidatas for superior ao tamanho máximo da população secundária;
9) População actual ← População principal;
10) Se foi atingido o número de iterações predefinido, então STOP; caso contrário, regressar ao passo 5).

3.2. Rede testada

A metodologia, descrita na secção anterior, para caracterizar a frente óptima de Pareto e ajudar à tomada de decisão foi aplicada a uma rede eléctrica do tipo radial do sistema de distribuição português com 94 nós e 24 alimentadores, distribuídos por um único alimentador principal com 33 nós e 23 alimentadores laterais (Pires (2001)).

Esta rede caracteriza-se por, em períodos de sobrecarga (como é o que está em estudo), apresentar perfis de tensões com valores abaixo dos 0.93 p.u. para a maioria dos nós da rede (apenas 10 dos nós apresentam valores acima de 0.93 p.u.). Este valor é, segundo a legislação portuguesa, o limite inferior do intervalo admissível para o valor da tensão (o valor da tensão de um qualquer nó deve estar entre 0.93 e 1.07), o que significa que as soluções de custo nulo são não admissíveis para esta rede.

As perdas apresentadas por esta rede, quando não há compensação (sem qualquer condensador instalado) e para o cenário de carga considerado para estudo (períodos de sobrecarga), são 0.3204 MW.

3.3. Frente óptima de Pareto

Nos testes realizados neste estudo, consideraram-se apenas três tipos de condensadores para possível instalação (Y = 3).

Realizaram-se vários testes com diferentes conjuntos de valores para os parâmetros associados ao algoritmo evolucionário. Os melhores resultados foram obtidos com o seguinte conjunto de valores: POP = 30, NPS = 40, E = 4, pm = 0.1, pc = 1, e número de iterações = 7500.

Cada solução da população secundária está associada a um esquema de compensação definido pela seguinte informação: número e dimensão dos condensadores, nós da rede onde os condensadores são instalados, bem como os correspondentes custo de instalação e perdas resistivas.

A Figura 6.1 mostra a frente óptima de Pareto associada à população secundária final, a qual corresponde ao resultado do processo evolucionário pretendido.

A Figura 6.2 permite fazer uma comparação entre a população inicial (que coincide com a população secundária inicial, devido à característica do algoritmo genético usado) e a população secundária final (que corresponde à frente óptima de Pareto).

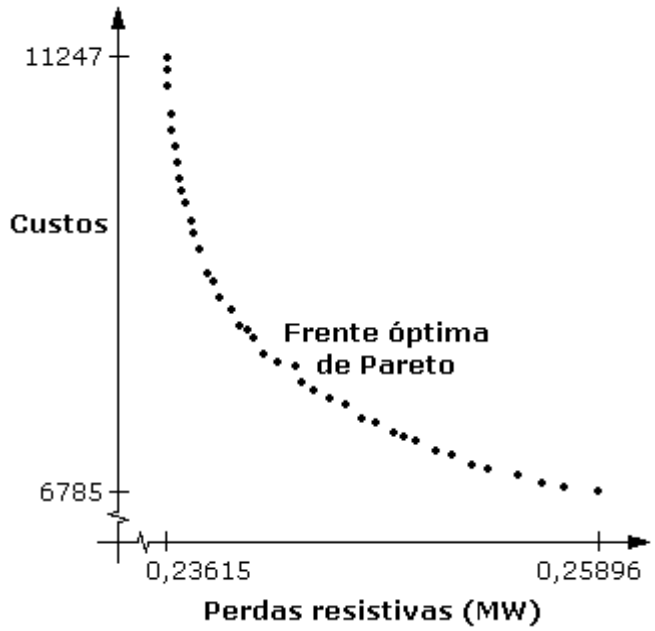

Figura 6.1: População secundária final (frente óptima de Pareto).

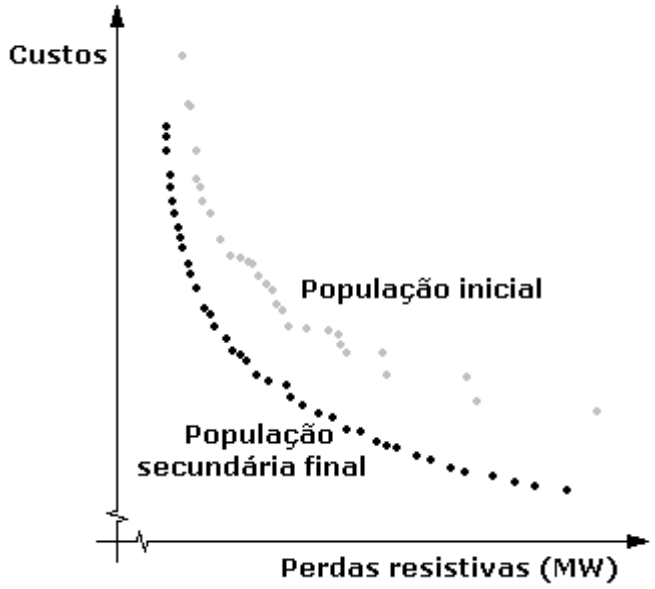

Figura 6.2: População secundária final *versus* população inicial.

3.4. População inicial

De forma a afinar os valores dos parâmetros associados ao algoritmo genético de acordo com as características do caso em estudo, foram implementadas e testadas algumas variantes das técnicas usadas neste algoritmo. Em particular, fizeram-se

experiências usando diferentes estratégias para construir a população inicial. Foram analisadas 3 estratégias diferentes, relacionadas com o tipo de soluções que pertencem à população:

(A) soluções de qualquer tipo (admissíveis e não admissíveis),

(B) apenas soluções admissíveis (dominadas e não dominadas),

(C) apenas soluções admissíveis não dominadas.

Os melhores resultados foram obtidas com a população inicial apenas com soluções admissíveis não dominadas (tipo C). A frente óptima de Pareto obtida pelo algoritmo quando este usa a população inicial do tipo (C) é melhor do que as frentes obtidas quando usa a população inicial do tipo (A) ou (B), quer em termos de dominância, quer em termos de diversidade da frente, como mostram a Figura 6.3 e a Figura 6.4.

A Figura 6.3 mostra as frentes óptimas de Pareto obtidas com as três estratégias diferentes para construção da população inicial. Estas três frentes foram obtidas usando o mesmo conjunto de valores para os parâmetros genéticos (POP = 30, NPS = 40, E = 4, pm = 0.1, pc = 1, e número de iterações = 7500).

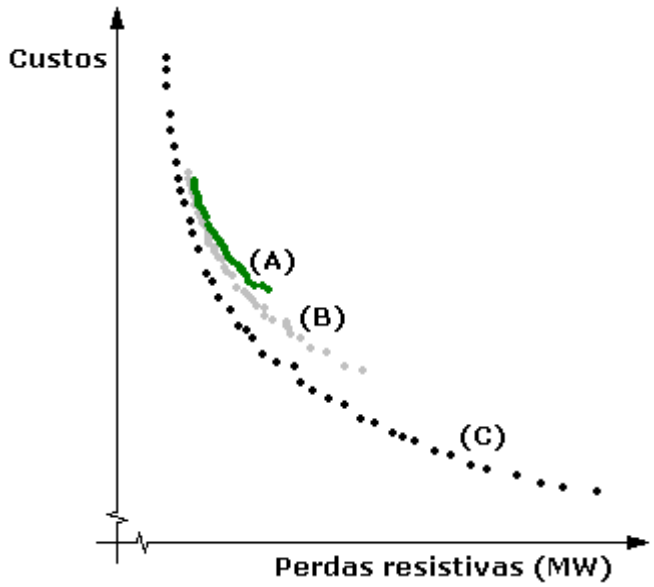

Figura 6.3: Frentes de Pareto (mesmos valores para os parâmetros genéticos).

A Figura 6.4 mostra as três frentes óptimas de Pareto obtidas usando os melhores conjuntos de valores para os parâmetros genéticos em cada uma das estratégias de construção da população inicial (entre todos os conjuntos testados):

- (A): POP = 100; NPS = 40; E = 6; pm = 0.1; pc = 1; número de iterações = 10000;
- (B): POP = 40; NPS = 50; E = 4; pm = 0.1; pc = 1; número de iterações = 7500;
- (C): POP = 30; NPS = 40; E = 4; pm = 0.1; pc = 1; número de iterações = 7500.

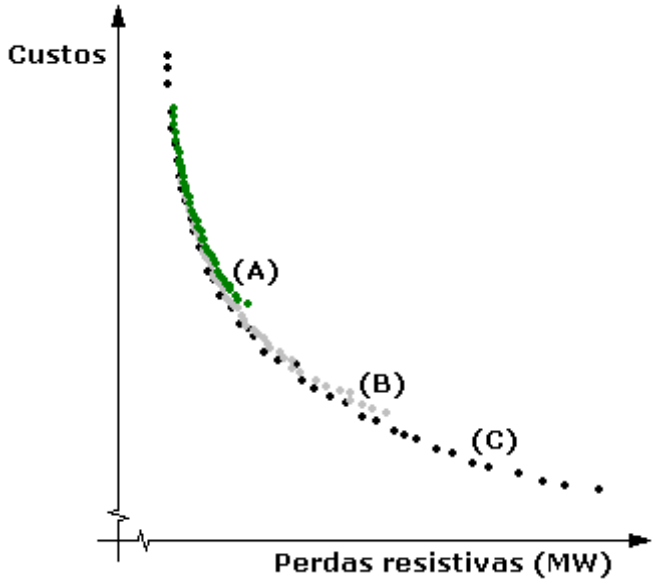

Figura 6.4: Populações secundárias finais (os melhores resultados).

Para a rede de distribuição com 94 nós, os tempos de execução (num Pentium III, 1 GHz, 512 MB de RAM) do algoritmo genético para cada uma das três estratégias de construção da população inicial (em todas os testes foi usado o cruzamento com 2 pontos de corte), foram os seguintes:

- mesmo conjunto de valores (Figura 6.3): 51 s (A), 57 s (B) e 387 s (C);
- melhores resultados obtidos (Figura 6.4): 267 s (A), 100 s (B) e 387 s (C).

No entanto, embora os tempos de execução obtidos com a população inicial do tipo (C) sejam elevados, o tempo de execução médio (sobre 12 execuções) foi de 260 s, o melhor tempo registado foi de 96 s e o pior de 406 s. De referir que os tempos de execução registados incluem o cálculo do trânsito de potência.

4. Abordagem com análise de robustez

Esta abordagem utiliza o conceito de cenário e de cenário de referência (ver secção 4.2 do capítulo 4), baseando-se no comportamento das soluções na vizinhança do cenário de referência, tendo em conta que os dados associados às funções objectivo e às restrições podem sofrer alterações relativamente aos respectivos valores nominais.

A abordagem evolucionária descrita na secção 4.6.2 do capítulo 4 foi adaptada ao problema de compensação de potência reactiva em redes de distribuição eléctrica (ver Antunes et al. (2005)), incorporando o conceito de grau de robustez. O AD pode especificar o tamanho da vizinhança do cenário de referência relativamente, quer aos dados associados às funções objectivo, quer aos próprios valores das funções objectivo.

Os dados associados ao modelo são os seguintes: os custos de instalação e o valor da capacidade de cada tipo de condensador a instalar, a resistência e a indutância de cada ramo (arco) da rede de distribuição e a potência activa e reactiva (carga) em cada nó da rede de distribuição. De referir que os dados associados à função objectivo perdas resistivas também influenciam a admissibilidade das soluções do modelo (restrição 12 do modelo matemático descrito na secção 1 do Anexo). Desta forma, esta abordagem usa uma conjugação das definições de grau de robustez apresentadas para os casos em que as perturbações incidem sobre os dados associados às funções objectivo e às restrições do modelo (secções 4.2 e 4.3 do capítulo 4).

Neste estudo, assume-se que os dados associados aos custos e às capacidades dos condensadores a instalar se mantêm inalterados ao longo do processo. Os restantes dados (resistência e indutância dos arcos e potência activa e reactiva nos nós) estão sujeitos a pequenas perturbações, devido às incertezas inerentes às medições e estimativas. Desta forma, a incerteza apenas incide sobre a função objectivo perdas resistivas, pois os custos de instalação dos condensadores associados a uma dada solução são iguais para todos os cenários possíveis, o que não acontece com o valor das perdas resistivas, que podem ser diferentes de cenário para cenário. No entanto, e apesar destes últimos dados estarem sujeitos a perturbações, o processo inicia-se com um conjunto de valores nominais para estes dados, que constituem o denominado cenário de referência, sobre o qual incidem as perturbações.

4.1. Algoritmo genético

A população usada na implementação do algoritmo genético é do mesmo tipo usado para o algoritmo genético associado à abordagem sem robustez (secção 3.1 deste capítulo).

Este algoritmo usa o operador genético cruzamento do tipo uniforme, com probabilidade pc, em que para cada geração (iteração) do algoritmo evolucionário é criada uma nova máscara. O operador genético mutação consiste em modificar, com uma probabilidade pm, o valor actual por um dos outros Y valores possíveis (tal como na abordagem sem robustez — ver secção 3.1 deste capítulo).

A abordagem evolucionária incorporando análise de robustez para caracterizar a frente óptima de Pareto foi aplicada a uma rede radial do sistema de distribuição indiano com 28 nós (Das et al. (1994)), a qual tem uma topologia simples, constituída por apenas um alimentador principal com 10 nós e 5 alimentadores laterais.

O objectivo deste estudo é analisar a influência de alguns dos parâmetros associados à análise de robustez (p1, p2 e η) na determinação da frente óptima de Pareto com soluções robustas. Estes parâmetros estão associados ao nível de exigência

da robustez, o qual é especificado pelo AD/analista. O parâmetro η é usado como limite superior para a distância não normalizada absoluta entre os valores das funções objectivo de uma solução (de qualquer tipo) determinadas para cenários diferentes (de referência e vizinho deste). Os parâmetros p1 e p2 estão associados às percentagens mínimas de cenários vizinhos do cenário de referência que devem ser admissíveis e em que os valores das funções objectivo de uma solução para estes cenários devem pertencer a uma vizinhança dos valores das funções objectivo daquela solução para o cenário de referência, respectivamente.

Os valores de alguns dos parâmetros associados ao algoritmo evolucionário foram os mesmos em todos os testes efectuados. Estes valores são os seguintes (Figura 6.5 - Figura 6.15): POP = 30, NPS = 30, E = 0.1×NPS, pc = 0.95; pm = 0.05; e número de iterações = 2000.

4.2. Frente não dominada

A Figura 6.5 mostra a frente óptima de Pareto obtida sem considerar a análise de robustez.

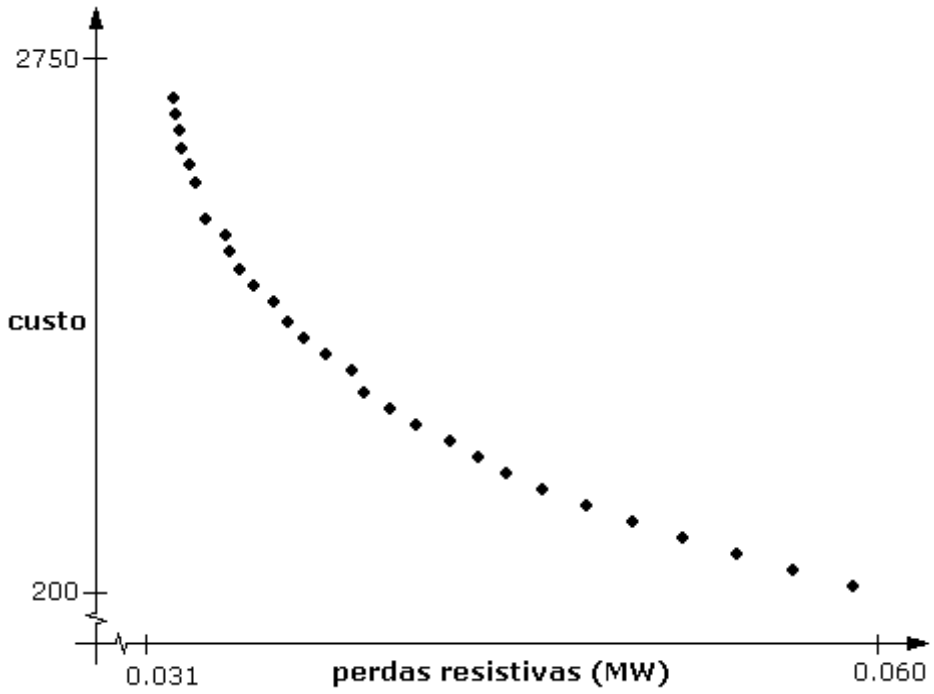

Figura 6.5: Frente óptima de Pareto (sem considerar análise de robustez).

A Figura 6.6 mostra a frente não dominada obtida com os seguintes valores para os parâmetros associados à análise de robustez:

- p1 = 100% e p2 = 100%, para qualquer tipo de soluções,
- h = 150,
- δ = 0.003, para a resistência e a reactância dos troços,
- δ = 0.03, para as potências activa e reactiva nos nós,
- η = 0.004, indica a diferença máxima permitida entre os valores das perdas de uma solução em dois cenários distintos, em que um deles é o cenário de referência,
- q = 1, para qualquer tipo de solução.

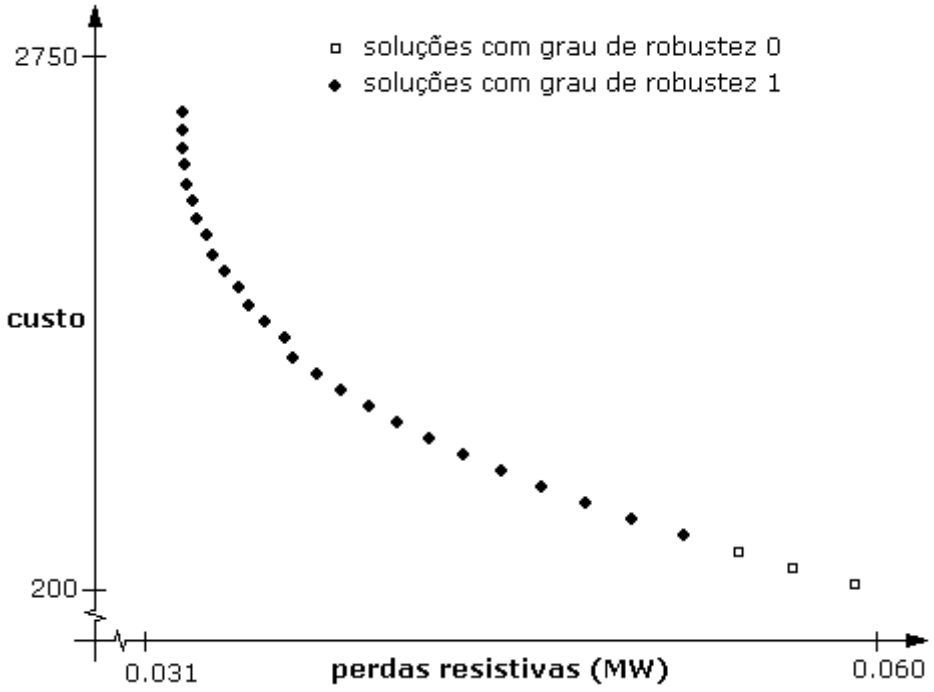

Figura 6.6: Frente de Pareto (com análise de robustez: p1 = p2 = 100% e η = 0.004).

Para p1 = p2 = 100% (Figura 6.6) a maioria das soluções tem um grau de robustez de valor 1 e apenas um pequeno grupo de soluções próximas do valor óptimo da função objectivo custo tem um grau de robustez 0.

4.3. Análise do parâmetro p1

Nesta secção é analisada a influência e o comportamento do parâmetro p1, o qual está relacionado com a percentagem mínima dos h cenários vizinhos da δ-vizinhança do cenário de referência para os quais uma solução tem que ser admissível. Para tal, é feito variar o valor do parâmetro p1 para as soluções admissíveis não dominadas (p1 = 95% e p1 = 90%), mas mantendo este parâmetro com o mesmo valor (p1 = 100%) para as soluções admissíveis dominadas e não admissíveis. Os restantes parâmetros associados à

análise de robustez têm os mesmos valores que foram utilizados para obter a frente apresentada na Figura 6.6 (onde p1 = 100% para qualquer tipo de soluções).

Ao relaxar o valor do parâmetro p1 = 100% (Figura 6.6) para p1 = 95% (Figura 6.7), verifica-se uma diminuição do número de soluções de grau de robustez 0 (na zona onde se encontravam estão agora soluções com grau de robustez 1) e também o aparecimento de soluções com grau de robustez 2, numa zona onde antes existiam apenas soluções com grau de robustez 1.

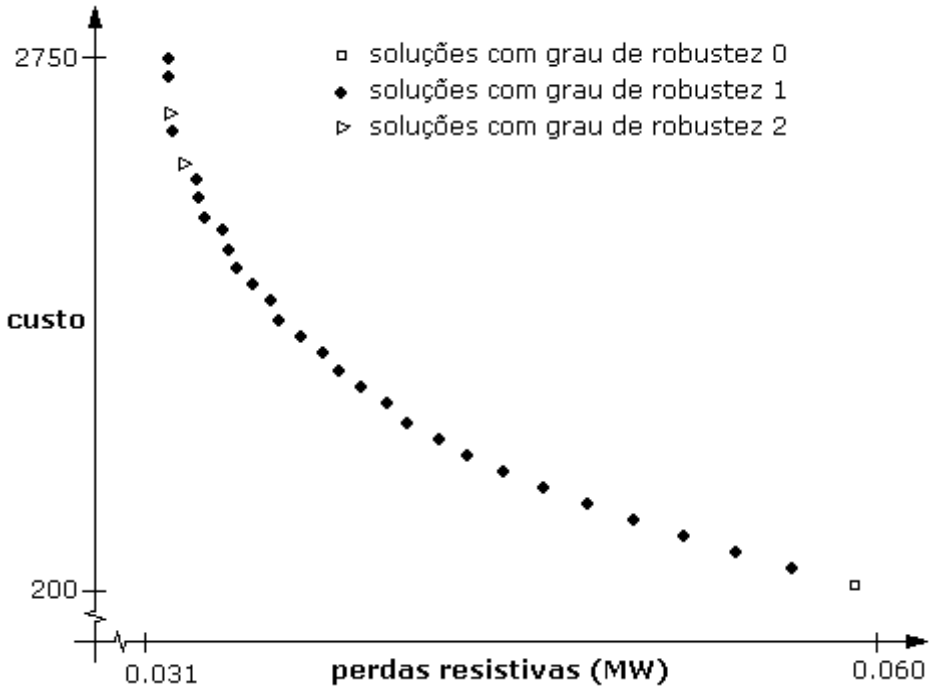

Figura 6.7: Frente de Pareto (com análise de robustez: p1 = 95%).

Note-se também que, tal como para p1 = 100%, também agora a zona das soluções com menor grau de robustez se situa na extremidade da frente óptima de Pareto junto ao valor óptimo do custo (isto é, numa zona onde se encontram os piores valores obtidos para as perdas resistivas). Pelo contrário, as soluções com maior grau de robustez (2) surgem na outra extremidade da frente junto aos melhores valores obtidos para as perdas resistivas, isto é, para os piores valores obtidos para o custo.

Relaxando ainda mais o valor do parâmetro p1 para 90% (Figura 6.8), verifica-se apenas, relativamente à frente obtida com p1 = 95%, um ligeiro aumento das soluções com grau de robustez 2, à custa das soluções com grau de robustez 1.

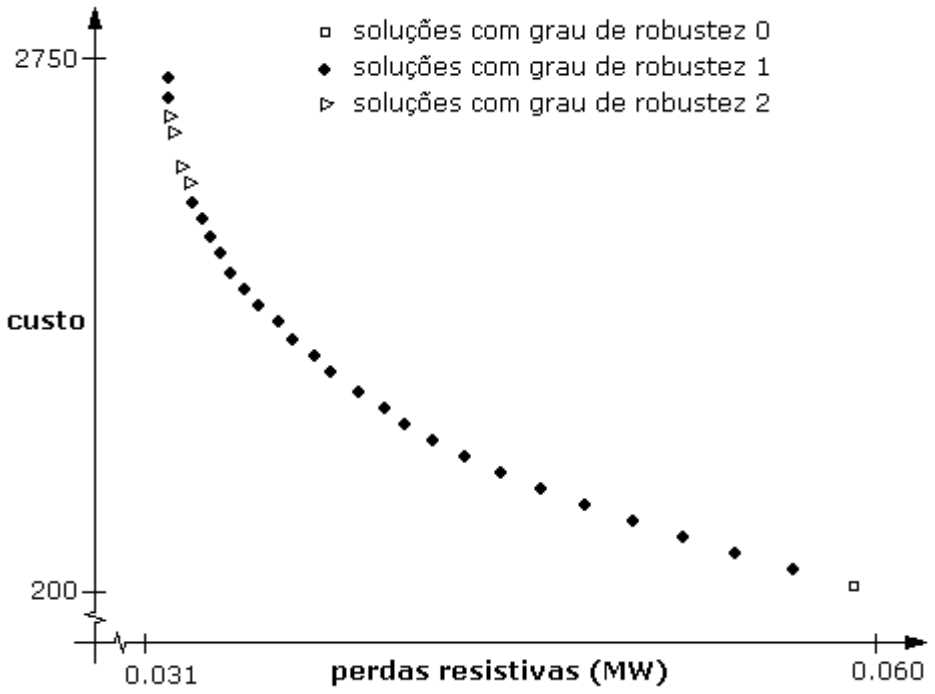

Figura 6.8: Frente de Pareto (com análise de robustez: p1 = 90%).

4.4. Análise do parâmetro p2

A Figura 6.9 e Figura 6.10 apresentam as frentes óptimas de Pareto obtidas para valores predefinidos diferentes para o parâmetro p2, associados apenas às soluções admissíveis não dominadas: 95% e 90%, respectivamente. Para as soluções admissíveis dominadas e não admissíveis, p2 = 100%. Os valores para os restantes parâmetros associados à análise de robustez são os mesmos que foram utilizados para obter a frente apresentada na Figura 6.6 (onde p2 = 100% para qualquer tipo de soluções).

Analisando a frente obtida com p2 = 95% (Figura 6.9), verifica-se que se mantém, quer o número de soluções de menor grau de robustez (0), quer as suas localizações na frente óptima de Pareto, o que já não acontece com as soluções de grau de robustez 1. De facto, praticamente em toda a região da frente óptima de Pareto onde se encontravam soluções com grau de robustez 1, estão agora soluções com grau de robustez 2 (apenas foi calculada uma solução com grau de robustez 1, que se situa na extremidade da frente junto ao valor óptimo do custo).

Ao relaxar-se ainda mais o valor do parâmetro p2 para 90% (Figura 6.10), a frente óptima de Pareto obtida apresenta várias soluções com grau de robustez 3, o que não acontecia nas frentes anteriores (Figura 6.6 e Figura 6.9). Estas soluções com grau de robustez 3 ocupam grande parte da frente óptima de Pareto, desde a extremidade junto

aos melhores valores obtidos para as perdas resistivas. As soluções menos robustas encontram-se mais próximas dos melhores valores obtidos para o custo.

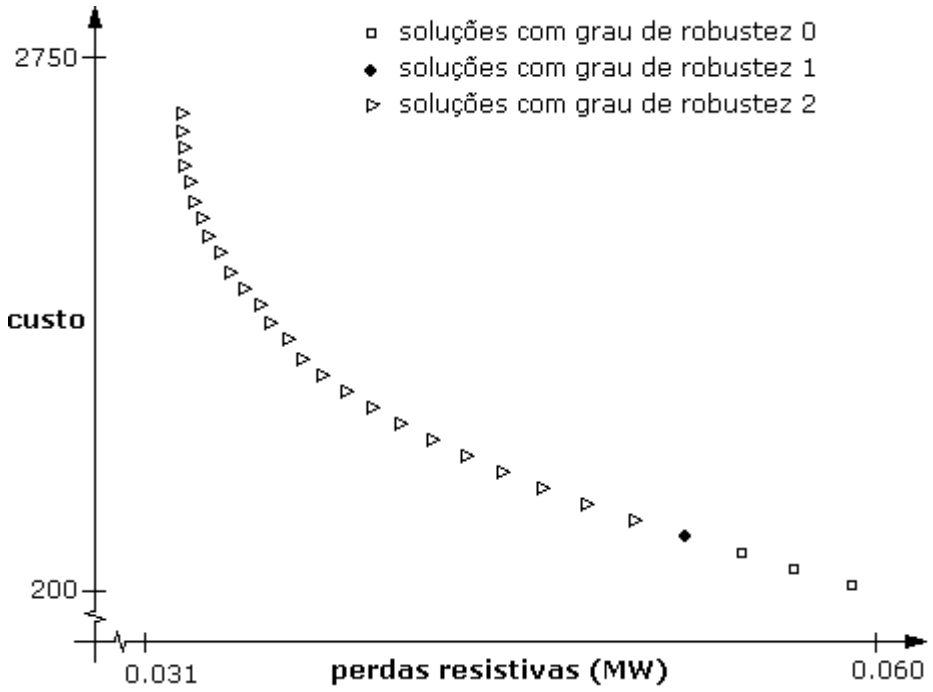

Figura 6.9: Frente de Pareto (com análise de robustez: p2 = 95%).

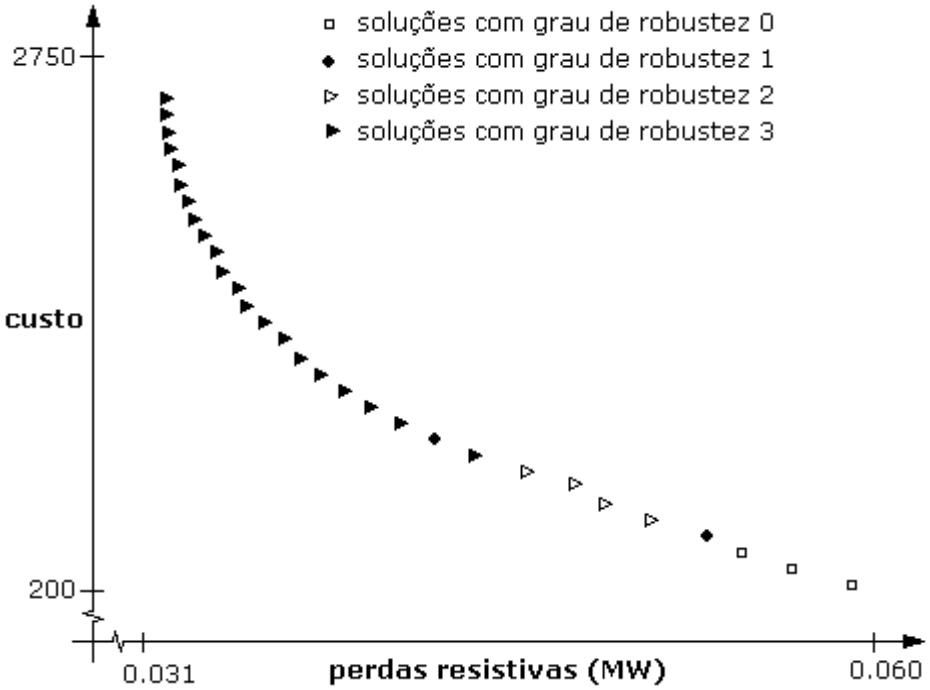

Figura 6.10: Frente de Pareto (com análise de robustez: p2 = 90%).

Analisando as frentes obtidas com a variação do parâmetro p2 (Figura 6.6, Figura 6.9 e Figura 6.10), pode-se concluir que as soluções não dominadas mais robustas estão localizadas próximas dos melhores valores para a função objectivo associada às perdas resistivas, e as menos robustas próximas do melhores valores do custo. O relaxamento do valor do parâmetro p2, de uma maneira geral, permite obter uma melhor discriminação, no tocante à robustez, para as soluções da frente óptima de Pareto.

Com a diminuição do valor de p2, o nível de exigência da robustez também decresce, fazendo com que as soluções pertencentes às frentes obtidas possam apresentar um espectro maior de graus de robustez, o que pode tornar-se interessante para facilitar o processo de identificação de uma solução de compromisso. No entanto, quanto menor for o valor de p2, menor é a fiabilidade do cálculo dos graus de robustez das soluções (isto é, maior é o risco destas terem na realidade um comportamento diferente daquele que é ditado pelo modelo).

4.5. Análise dos parâmetros p1 e p2 em conjunto

Nesta secção pretende-se analisar o comportamento dos parâmetros p1 e p2 quando visto como um só (p1⊕p2). Isto é, considera-se que um cenário *s'* pertencente à δ-vizinhança do cenário de referência *s*, não cumpre com os requisitos impostos, se uma solução *x* é não admissível para o cenário *s'* ou a sua imagem para o cenário *s'* não pertence à η-vizinhança da imagem de *x* para o cenário *s*.

A frente obtida com p1⊕p2 = 95% (Figura 6.11) é composta por soluções com quatro graus de robustez. No entanto, enquanto que existem poucas soluções com grau de robustez 0, 1 (que se situam em zonas da frente próximas dos melhores valores obtidos para o custo) e 3 (numa zona da frente próxima dos melhores valores obtidos para a função objectivo perdas resistivas), as soluções com grau de robustez 2 são em grande número, ocupando cerca de 70% da extensão da frente de Pareto obtida.

Fazendo p1⊕p2 = 90% (Figura 6.12), obtém-se uma frente óptima de Pareto constituída por soluções com graus de robustez de quatro tipos (tal como para o caso anterior). Em comparação com a frente obtida com p1⊕p2 = 95% (Figura 6.11), verifica-se que o número de soluções com graus de robustez mais baixos (0 e 1) é semelhante. Por outro lado, verifica-se um aumento substancial de soluções com o grau de robustez 3 (cerca de 75%), que foi alcançado à custa de uma diminuição abrupta do número de soluções com grau de robustez 2.

As soluções com grau de robustez mais elevado (3) ocupam uma zona que vai desde a extremidade próxima dos melhores valores obtidos para a função objectivo perdas resistivas, até uma zona já próxima dos melhores valores obtidos para o custo. As

soluções de menor grau de robustez (0 e 1) continuam a ocupar as zonas extremas da frente de Pareto junto ao melhores valores obtidos para a função objectivo custo.

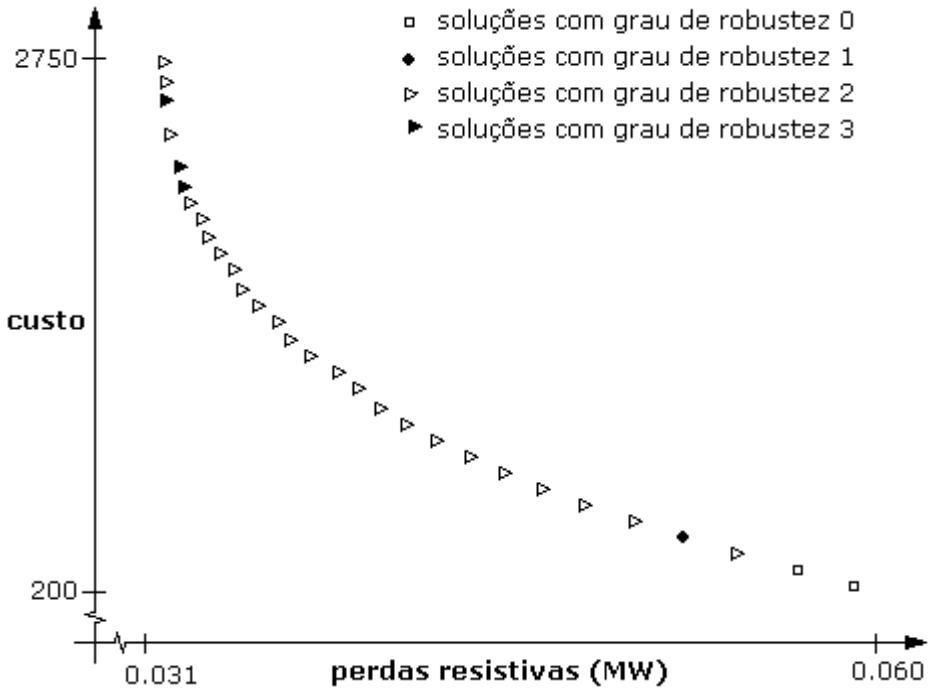

Figura 6.11: Frente de Pareto (com análise de robustez: p1 ⊕ p2 = 95%).

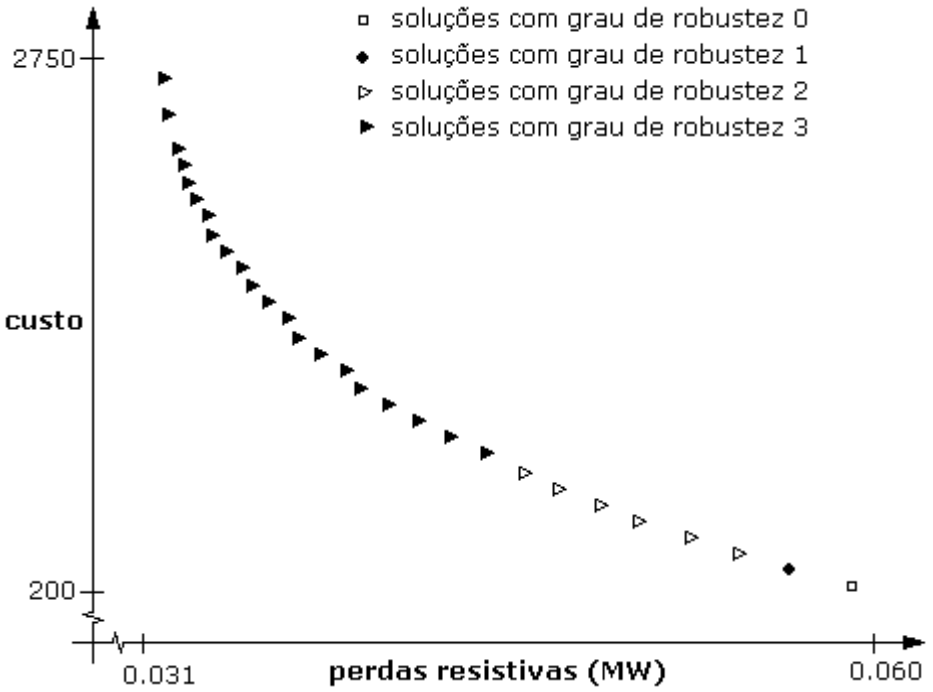

Figura 6.12: Frente de Pareto (com análise de robustez: p1 ⊕ p2 = 90%).

4.6. Análise do parâmetro η

As Figura 6.13 - Figura 6.15 mostram as soluções não dominadas obtidas com diferentes valores predefinidos associados ao parâmetro $\eta = (\eta_1, \eta_2)$ associado a qualquer tipo de solução (admissível não dominada e dominada, e não admissível): (0.003, 150), (0.005, 150) e (0.006, 150) — os custos de instalação dos condensadores não sofrem qualquer tipo de perturbação. Para os restantes parâmetros associados à análise de robustez, os seus valores são os mesmos que foram usados para obter a frente não dominada da Figura 6.6, onde $\eta = (\eta_1, \eta_2) = (0.004, 150)$.

Neste caso pode-se concluir, pela análise das frentes obtidas, que tal como acontece com o relaxamento dos valores para os parâmetros p1 e p2, ao relaxar-se (aumentar) o valor de η verifica-se também um aumento do número de soluções com graus de robustez mais elevados ou o aparecimento de soluções com maior grau de robustez (Figura 6.14 e Figura 6.15) na frente de Pareto obtida. Pelo contrário, ao introduzir um maior nível de exigência do valor do parâmetro η (correspondendo à diminuição da dimensão da vizinhança no espaço dos objectivos), a frente óptima de Pareto obtida contém menos soluções com os graus mais elevados e mais soluções com os graus de robustez mais baixos (Figura 6.13, com $\eta = (0.003, 150)$).

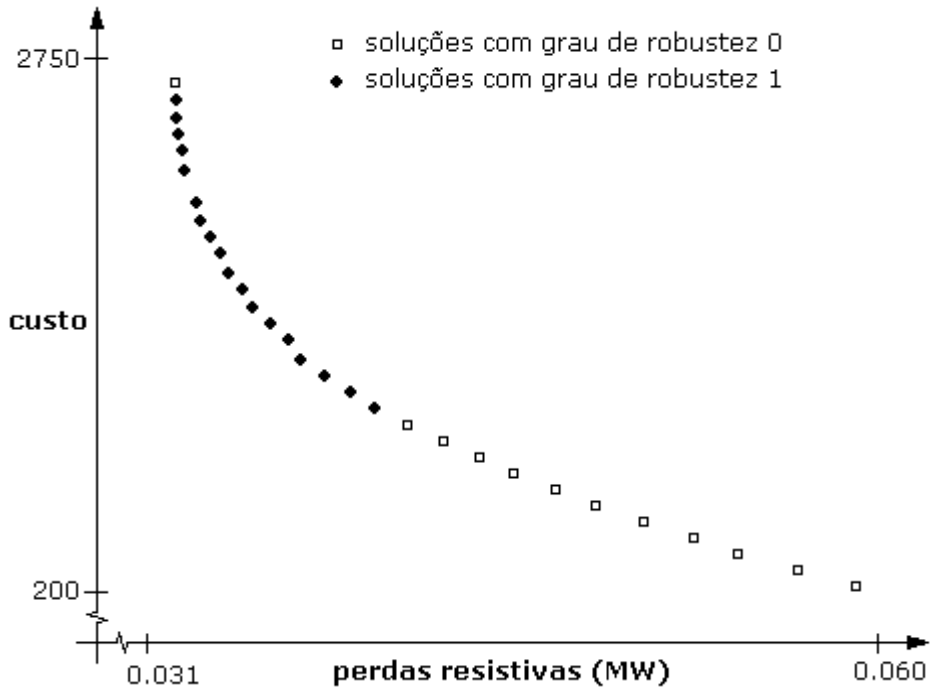

Figura 6.13: Frente de Pareto (com análise de robustez: $\eta = (0.003, 150)$).

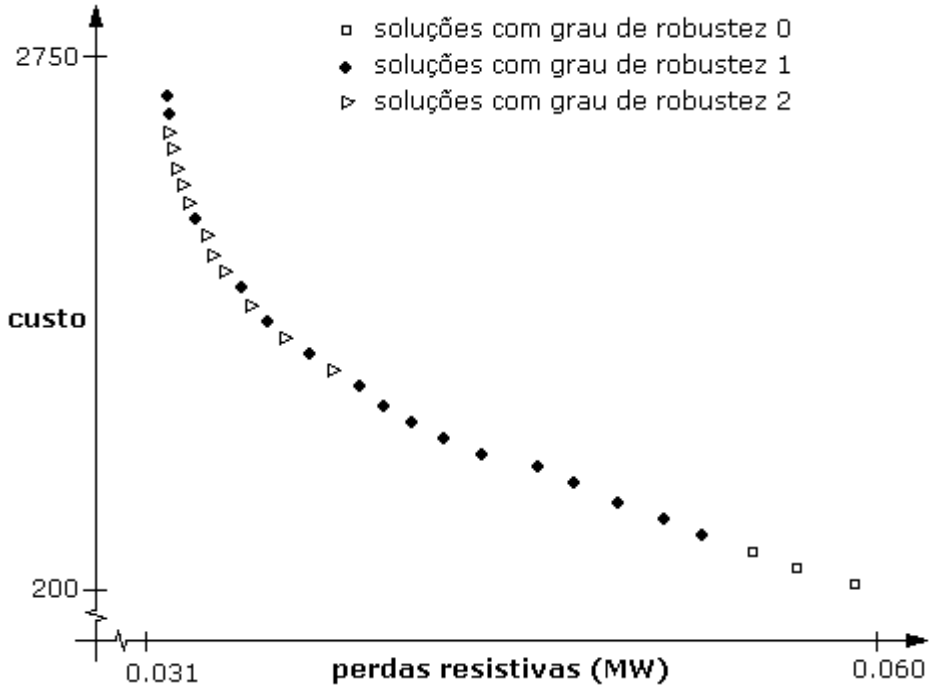

Figura 6.15: Frente de Pareto (com análise de robustez: (η = 0.006, 150)).

Tal como acontece com a variação dos valores para os parâmetros p1 e p2, também neste caso as soluções com os graus de robustez mais elevados se localizam na extremidade da frente óptima de Pareto, próximas dos melhores valores obtidos para as perdas resistivas, e as soluções com graus de robustez mais baixos se situam mais próximas dos melhores valores obtidos para o custo.

4.7. Conclusão

Neste capítulo pretendeu-se avaliar a importância da realização de uma análise de robustez a um problema real, usando a abordagem apresentada na secção 4 do capítulo 4. Para tal, implementou-se um algoritmo genético para resolver um problema de compensação de potência reactiva num sistema radial de distribuição de energia eléctrica, usando o mesmo modelo que foi utilizado para a abordagem anterior (que não considerava a análise de robustez).

Foram realizadas várias experiências com diversas combinações de valores dos parâmetros associados à análise de robustez (p1, p2 e η). Na primeira, correspondente ao nível máximo de exigência do AD (p1 = p2 = 100% e η = 0.004), a frente óptima de Pareto obtida apresenta uma tendência para que as soluções menos robustas se situem na extremidade da frente com valores muito próximos dos melhores valores da função objectivo custo.

Com o relaxamento dos valores para os parâmetros p1 e p2, esta tendência persiste e nota-se agora que as soluções mais robustas se situam numa zona muito próxima da outra extremidade onde predominam os melhores valores para a função objectivo perdas resistivas. No entanto, quando se relaxa p1 e p2 para os mesmos valores (90% e 95%), verifica-se que, para o primeiro caso (Figura 6.7 e Figura 6.8), o número de soluções de grau máximo da frente óptima de Pareto é menor do que para o segundo caso (Figura 6.9 e Figura 6.10).

Quando se relaxam os valores dos parâmetros p1 e p2 em conjunto (p1⊕p2), nota-se que as características da frente óptima de Pareto (Figura 6.11 e Figura 6.12) são semelhantes às da frente óptima de Pareto obtida quando se relaxa apenas o parâmetro p2. Este facto indicia que a influência do parâmetro p2, na classificação das soluções da frente de Pareto por graus de robustez, é mais acentuada do que a influência de p1.

O mesmo acontece quando se relaxa (aumenta) o valor do parâmetro η, em que a frente óptima de Pareto apresenta um aumento do número de soluções com graus de robustez mais elevados (Figura 6.14 e Figura 6.15). Pelo contrário, quando se aumenta o nível de exigência (diminuição do valor do parâmetro η), verifica-se que a frente óptima de Pareto apresenta um aumento do número de soluções de menor grau de robustez (Figura 6.13). Em termos de localização das soluções na frente de Pareto, a tendência é a mesma que foi verificada para os casos de p1 e p2, isto é, as soluções mais robustas situam-se numa zona muito próxima da extremidade da frente onde predominam os melhores valores para a função objectivo perdas resistivas.

Capítulo 7

Conclusões Finais e Pistas para Futuros Desenvolvimentos

1. Conclusões Finais

Nesta dissertação, foram apresentadas duas abordagens de optimização evolucionária multi-objectivo robusta, ambas baseadas na classificação das soluções do modelo por níveis de robustez. Estas abordagens foram concebidas supondo que, quer os valores das variáveis de decisão, quer os dados do modelo associados às funções objectivo e às restrições estão sujeitos a perturbações, em relação aos respectivos valores "nominais". Para tal, foi introduzido um novo conceito de robustez, designado por grau de robustez, o qual foi incorporado num algoritmo evolucionário (mais especificamente, num algoritmo genético). O grau de robustez intervém no cálculo dos valores da função de aptidão das soluções do modelo e no mecanismo de selecção das soluções de uma população, de modo a influenciar que as soluções mais robustas não sejam eliminadas (isto é, possui uma perspectiva elitista) e que contribuam para a construção das gerações seguintes. Estas abordagens permitem caracterizar as soluções da frente óptima de Pareto, classificando-as de acordo com os respectivos graus de robustez (a frente óptima de Pareto pode conter soluções com diferentes graus de robustez), e não apenas classificá-las como robustas ou não robustas, como acontece noutras abordagens.

É importante, para um AD, que a informação que tem disponível sobre as soluções da frente óptima de Pareto obtida não seja apenas relativa aos valores das funções objectivo, mas também contenha informação sobre os respectivos níveis (graus) de

robustez. Desta forma, uma decisão final pode ser mais fundamentada, uma vez que, para além de ter em conta os valores das funções objectivo para identificar uma boa solução de compromisso, tem também em consideração o grau de estabilidade exibido pela solução face a possíveis perturbações que possam ocorrer nos valores da própria solução ou nos dados do modelo. A informação adicional sobre a robustez da solução facilita a tarefa do AD na escolha da solução final do modelo (que corresponderá à solução a implementar no quadro do problema real) tornando o processo de apoio à decisão mais efectivo, dado que estas soluções exibem algum nível de diferenciação entre si em termos de robustez.

Estas abordagens foram construídas tendo em conta esta necessidade de diferenciar as soluções da frente óptima de Pareto, classificando-as por graus de robustez, através da introdução de parâmetros de robustez que possam ser ajustados pelo AD ou ao longo do processo de cálculo. Estes parâmetros podem ser manipulados pelo AD com a finalidade de obter uma classificação das soluções da frente óptima de Pareto em termos de robustez, tornando, assim, a escolha da solução final do modelo por parte do AD mais bem fundamentada, ao possibilitar introduzir diferentes níveis de exigência quanto à robustez das soluções. Por um lado, o AD, ao dispor de um conjunto de soluções óptimas de Pareto que apresentam diferentes graus de robustez, fica mais confortável para escolher uma solução final, pois tem mais uma forma de diferenciar as soluções, para além dos respectivos valores das funções objectivo. Por outro lado, a escolha de soluções com maior grau de robustez, em detrimento das com menor grau, garante, em princípio, uma maior estabilidade das soluções implementadas, ou seja maior imunidade a perturbações que ocorram nos dados do modelo e que ultrapassem as previsões iniciais do AD.

Nas abordagens apresentadas neste texto, os valores dos parâmetros associados à análise de robustez das soluções são definidos pelo AD, em princípio auxiliado por um analista, traduzindo as suas preferências em termos do limiar de indiferença relativamente aos valores das funções objectivo (η), da dimensão prevista das perturbações que podem ocorrer nos valores das variáveis de decisão ou nos dados do modelo (δ) e da fiabilidade com que pretende que sejam obtidos os graus de robustez (p1 e p2).

Em problemas onde a incerteza incide sobre os valores das variáveis de decisão das soluções, o grau de robustez de uma solução x ser k significa que as percentagens de soluções da kδ-vizinhança de x admissíveis e cujos valores das funções objectivo que são melhores do que f(x) ou que pertencem à η-vizinhança de f(x) são maiores ou iguais a p1 e p2, respectivamente.

Quando a incerteza incide sobre os valores nominais dos dados associados às funções objectivo de um problema (traduzido pelo cenário de referência s), o grau de robustez de uma solução x ser k significa que a percentagem de cenários da kδ-vizinhança de s cujos valores de f(x) para estes cenários que são melhores do que f(x) para s ou que pertencem à η-vizinhança de f(x) para s é maior ou igual a p2.

Em problemas onde a incerteza incide sobre os valores nominais dos dados associados às restrições (traduzido pelo cenário de referência s), o grau de robustez de uma solução x ser k significa que a percentagem de cenários da kδ-vizinhança de s que são admissíveis é maior ou igual a p1.

Dado as estimativas com que o modelo foi construído poderem revelarem-se incorrectas, as soluções implementadas podem tornar-se inadequadas para o problema. Desta forma, o AD deve escolher uma solução que possa estar preparada para esta eventualidade, isto é, que tenha um grau de robustez elevado. Nas abordagens que apenas classificam as soluções como robustas ou não robustas (isto é, apenas com um nível de robustez) tal não é devidamente tido em conta, o que pode obrigar a frequentes reoptimizações.

Podemos, então, concluir que estas abordagens constituem importantes ferramentas no apoio à tomada da decisão em modelos cujos dados estão sujeitos a perturbações.

Efectuou-se também um estudo no sentido de avaliar a influência e o comportamento de alguns dos parâmetros de robustez. Deste estudo, pode-se concluir que quando o AD é mais tolerante em relação às diferenças nos valores das funções objectivo, que se traduz num valor elevado para o parâmetro η, a tendência é para se obter uma frente óptima de Pareto com maior número de soluções robustas, assim como com soluções com graus de robustez mais elevados.

Da mesma maneira, o relaxamento (diminuição) do valor dos parâmetros p1 e p2 implica a obtenção de uma frente óptima de Pareto com mais soluções robustas e de soluções com graus de robustez mais altos. No entanto, o relaxamento do valor de qualquer um destes parâmetros implica uma menor fiabilidade dos graus de robustez determinados.

2. Pistas para Futuros Desenvolvimentos

No seguimento do trabalho descrito nesta dissertação, já se encontra em curso trabalho de investigação para estender estas abordagens a problemas onde a incerteza é inerente ao modelo, isto é, está situada nos dados do modelo e pode ser representada através de números intervalares. Assim, tenta-se modelar os dados do problema como

desconhecidos, mas limitando-os sem ser necessário especificar as distribuições de probabilidade (como na programação estocástica) ou de possibilidades (como na programação difusa).

Uma parametrização (por exemplo, as probabilidades de cruzamento e de mutação) adequada dos algoritmos evolucionários usados no estudo de um problema é essencial para o bom desempenho deste tipo de algoritmos. Todavia, devido ao seu carácter probabilístico, a partir de várias simulações (com os mesmos valores dos parâmetros) de um algoritmo evolucionário obtêm-se, normalmente, diferentes conjuntos de soluções não dominadas. Para além disso, em ambientes em que a parametrização do problema apresenta um comportamento dinâmico torna-se necessário analisar a robustez das parametrizações do algoritmo, no sentido de avaliar a qualidade dos resultados obtidos em função de um certo conjunto de parâmetros do algoritmo evolucionário e de um certo conjunto de dados do problema.

Desta forma, uma linha de investigação será utilizar a análise de robustez, em particular o conceito de grau de robustez proposto nesta dissertação, de forma a contornar as dificuldades de parametrização dos algoritmos evolucionários, sobretudo em ambientes multi-objectivo, avaliando o comportamento de um certo conjunto de parâmetros do algoritmo evolucionário perante diferentes conjuntos de dados do problema. Desta forma, pretende-se estudar o controlo adaptativo dos parâmetros do algoritmo evolucionário tendo em conta a robustez de cada conjunto destes parâmetros, o qual é avaliado através do seu comportamento perante diferentes conjuntos de dados do problema.

Referências

Abboud e Schoenauer (2002)

K. Abboud e M. Schoenauer (2002). "Surrogate deterministic mutation". Artificial Evolution, Lecture Notes in Computer Science 2310. Springer, 103-115.

Affenzeller e Wagner (2004)

M. Affenzeller e S. Wagner (2004). "The influence of population genetics for the redesign of genetic algorithm". Journal of Systems Science 30 (4), 41-49.

Agarwal et al. (2004)

H. Agarwal, J.E. Renaud, E.L. Preston e D. Padmanabhan (2004). "Uncertainty quantification using evidence theory in multidisciplinary design optimization". Reliability Engineering and System Safety 85. Elsevier, 281–294.

Aizawa e Wah (1993)

A.N. Aizawa e B.W. Wah (1993). "Dynamic control of genetic algorithms in a noisy environment". Proceedings of the Fifth International Conference on Genetic Algorithms. S. Forrest (Ed.). Morgan Kaufmann Publishers, 48-55.

Aizawa e Wah (1994)

A.N. Aizawa e B.W. Wah (1994). "Scheduling of genetic algorithms in a noisy environment". Evolutionary Computation 2 (2), 97-122.

Anderson e Hsu (1999)

K. Anderson e Y. Hsu (1999). "Genetic crossover strategy using an approximation concept". Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99) 1. IEEE Press, 527-533.

Antunes et al. (2004)

C.H. Antunes, C. Barrico, A. Gomes, D. Pires e A. Martins (2004). "An evolutionary algorithm for reactive power compensation in radial distribution networks". Research Report N.5/2004. INESC Coimbra (URL: http://www.inescc.pt/documentos/AG_VAR_AOR.pdf).

Antunes et al. (2005)

C.H. Antunes, C. Barrico, A. Gomes, D. Pires e A. Martins (2005). "On the use of evolutionary algorithms for reactive power compensation in electrical distribution networks experiments on a case study". Proceedings of the 6th Metaheuristics International Conference (MIC2005), 514-519.

Arnold (2001)

D.V. Arnold (2001). "Evolution strategies in noisy environments — A survey of existing work". Theoretical Aspects of Evolutionary Computing. L. Kallel, B. Naudts e A. Rogers (Eds.). Springer, 239-250.

Arnold (2002)

D.V. Arnold (2002). "Noisy optimization with evolution strategies". Kluwer Academic Publishers.

Arnold e Beyer (2003)

D.V. Arnold e H.-G. Beyer (2003). "On the effects of outliers on evolutionary optimization". Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science 2690. Springer, 151-160.

Bäck et al. (1991)

T. Bäck, F. Hoffmeister e H.-P. Schwefel (1991). "A survey of evolution strategies". Proceedings of the Fourth International Conference on Genetic Algorithms. R.K. Belew e L.B. Booker (Eds.). Morgan Kaufmann Publishers, 2-9.

Bäck et al. (1997)

T. Bäck, U. Hammel e H.-P. Schwefel (1997). "Evolutionary computation: Comments on the history and current state". IEEE Transactions on Evolutionary Computation 1 (1). IEEE Press, 3-17.

Baker (1985)

J.E. Baker (1985). "Adaptive selection methods for genetic algorithms". Proceedings of the First International Conference on Genetic Algorithms and Their Applications. J.J. Grefenstette (Ed.). Lawrence Erlbaum Associates, 101-111.

Baker (1987)

J.E. Baker (1987). "Reducing bias and inefficiency in the selection algorithm". Proceedings of the Second International Conference on Genetic Algorithms and Their Applications. J.J. Grefenstette (Ed.). Lawrence Erlbaum Associates, 14-21.

Baran e Wu (1989-a)

M.E. Baran e F.F. Wu (1989). "Optimal capacitor placement on radial distribution system". IEEE Transactions on Power Delivery 4 (1). IEEE Press, 725-734.

Baran e Wu (1989-b)

M.E. Baran e F.F. Wu (1989). "Optimal sizing of capacitors placed on a radial distribution system". IEEE Transactions on Power Delivery 4 (1). IEEE Press, 735-743.

Baran et al. (2001)

B. Baran, J. Vallejos, R. Ramos e U. Fernandez (2001). "Reactive power compensation using a multi-objective evolutionary algorithm". Proceedings of the 2001 IEEE Porto Power Tech 2. IEEE Press.

Barrico e Antunes (2006-a)

C. Barrico e C.H. Antunes (2006). "Robustness analysis in evolutionary multi-objective optimization — With a case study in electrical distribution networks". Proceedings of the II European-Latin-American Workshop on Engineering Systems (SELASI II).

Barrico e Antunes (2006-b)

C. Barrico e C.H. Antunes (2006). "A new approach to robustness analysis in multi-objective optimization". Proceedings of the 7th International Conference on Multi-Objective Programming and Goal Programming (MOPGP'06).

Barrico e Antunes (2006-c)

C. Barrico e C.H. Antunes (2006). "Robustness analysis in multi-objective optimization using a degree of robustness concept". Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC'06). IEEE Press, 1887-1892.

Barrico e Antunes (2006-d)

C. Barrico e C.H. Antunes (2006). "Robustness analysis in evolutionary multi-objective optimization". Research Report N.3/2006, INESC – Coimbra (URL: http:// www.inescc.pt/documentos/RR_032006.pdf).

Barrico e Antunes (2007)

C. Barrico e C.H. Antunes (2007). "An evolutionary approach for Assessing the degree of robustness of solutions to multi-objective models". Evolutionary Computation in Dynamic and Uncertain Environments, Computational Intelligence 51. Shengxiang Yang, Yew-Soon Ong e Yaochu Jin (Eds.). Springer, 565-582.

Battiti e Tecchiolli (1994)

R. Battiti e G. Tecchiolli (1994). "The reactive tabu search". ORSA Journal on Computing 6 (2), 126-140.

Baum et al. (1995)

E. Baum, D. Boneh e C. Garret (1995). "On genetic algorithms". Proceedings of Eighth Annual Conference Learning Theory. Springer, 230-239.

Bellman e Zadeh (1970)

R. Bellman e L.A. Zadeh (1970). "Decision making in a fuzzy environment", Management Science 17 (4). Informs, 141–164.

Ben-Tal e Nemirovski (1998)

A. Ben-Tal e A. Nemirovski (1998). "Robust convex optimization". Mathematical & Operations Research 23 (4). Informs, 769-805.

Ben-Tal e Nemirovski (1999)

A. Ben-Tal e A. Nemirovski (1999). "Robust solutions to uncertain linear programs". Operations Research Letters 25 (1). Elsevier, 1-13.

Ben-Tal e Nemirovski (2000)

A. Ben-Tal e A. Nemirovski (2000). "Robust solutions of linear programming problems contaminated with uncertain data". Mathematical Programming 88 (3). Springer, 411-424.

Ben-Tal e Nemirovski (2002)

A. Ben-Tal e A. Nemirovski (2002). "Robust optimization — Methodoly and applications". Mathematical Programming 92 (3). Springer, 453-480.

Bertsimas e Sim (2003)

D. Bertsimas e M. Sim (2003). "Robust discrete optimization and network flows". Mathematical Programming 98 (1-3). Springer, 49-71.

Bertsimas e Sim (2004)

D. Bertsimas e M. Sim (2004). "The price of robustness". Operations Research 52 (1). Informs, 35-53.

Beyer (2000)

H.-G. Beyer (2000). "Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice". Computer Methods in Applied Mechanics and Engineering 186 (2-4). Elsevier, 239-267.

Bierwirth e Mattfeld (1999)

C. Bierwirth e D.C. Mattfeld (1999). "Production scheduling and rescheduling with genetic algorithms". Evolutionary Computation 7 (1), 1-17.

Borges (2005)

A.R.P. Borges (2005). "Abordagens interactivas para tratamento da incerteza em modelos de optimização multiobjectivo para apoio à decisão". Tese de Doutoramento. Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Borges e Antunes (2002)

A.R. Borges e C.H. Antunes (2002). "A weight space-based approach to fuzzy multiple-objective linear programming". Decision Support Systems 34, 427-443.

Branke (1998)

J. Branke (1998). "Creating robust solutions by means of an evolutionary algorithm". Parallel Problem Solving from Nature PPSN V, Lecture Notes in Computer Science 1498. A.E. Eiben, T. Bäck, M. Schoenauer e H.P. Schwefel (Eds). Springer, 119-128.

Branke (1999)

J. Branke (1999). "Memory enhanced evolutionary algorithms for changing optimization problems". Proceedings of the Congress on Evolutionary Computation (CEC'99) 3. P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao e A. Zalzala (Eds.). IEEE Press, 1875-1882.

Branke (2000)

J. Branke (2000). "Efficient evolutionary algorithms for searching robust solutions". Proceedings of Adaptive Computing in Design and Manufacture (ACDM 2000). I.C. Parmee (Ed.). Springer, 275-286.

Branke (2001-a)

J. Branke (2001). "Evolutionary optimization in dynamic environments". Kluwer Academic Publishers.

Branke (2001-b)

J. Branke (2001). "Reducing the sampling variance when searching fir robust solutions". Proceedings of Genetic and Evolutionary Computation Conference (GECCO'01). L. Spector, E.D. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M.-H. Garzon e E. Burke (Eds). Morgan Kaufmann Publishers, 235-242.

Branke e Schmidh (2003)

J. Branke e C. Schmidh (2003). "Selection in the presence of noise". Genetic and Evolutionary Computation Conference (GECCO'03) Part 1, Lectures Notes in Computer Science 2723. E. Cantú-Paz, J.A. Foster, K. Deb, L. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R.K. Standish, G. Kendall, S.W. Wilson, M. Harman, J. Wegener, D. Dasgupta, M.A. Potter, A.C. Schultz, K.A. Dowsland, N. Jonoska e J.F. Miller (Eds.). Springer, 766-777.

Branke e Schmidh (2004)

J. Branke e C. Schmidh (2004). "Sequential sampling in noisy environment". Parallel Problem Solving from Nature PPSN VIII, Lectures Notes in Computer Science 3242. Springer, 202-211.

Branke e Schmidh (2005)

J. Branke e C. Schmidh (2005). "Faster convergence by means of fitness estimation". Soft Computing 9 (1), 13-20.

Branke e Wang (2003)

J. Branke e W. Wang (2003). "Theoretical analysis of simple evolution strategies in quickly changing environments". Genetic and Evolutionary Computation Conference (GECCO'03), Lecture Notes in Computer Science 2723. Springer, 537-548.

Branke et al. (2001)

J. Branke, C. Schmidh e H. Schmeck (2001). "Efficient fitness estimation in noise environment". Proceedings of Genetic and Evolutionary Computation Conference (GECCO'01). L. Spector, E.D. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M.-H. Garzon e E. Burke (Eds). Morgan Kaufmann Publishers, 243-250.

Bui et al. (2005)

L.T. Bui, J. Branke e H.A.Abbass (2005). "Multiobjective optimization for dynamic environments". Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'05) 3. IEEE Press, 2349-2356.

Cantú-Paz (2004)

E. Cantú-Paz (2004). "Adaptive sampling for noisy problems". Genetic and Evolutionary Computation Conference (GECCO'04), Lecture Notes in Computer Science 3102. K. Deb, R. Poli, W. Banzhaf, H.-G. Beyer, E.K. Burke, P.J. Darwen, D. Dasgupta, D. Floreano, J.A. Foster, M. Harman, O. Holland, P.L. Lanzi, L. Spector, A. Tettamanzi, D. Thierens e A.M. Tyr (Eds.). Springer, 947-958.

Cavicchio (1970)

D.J. Cavicchio (1970). "Adaptive search using simulated evolution". Ph.D. Thesis. Ann Arbor, University of Michigan Press.

Chafekar et al. (2004)

D. Chafekar, L. Shi, K. Rasheed e J. Xuan (2004). "Multi-objective GA optimization using reduced models". IEEE Transactions on Systems, Man, and Cybernetics: Part C 9 (2). IEEE Press, 261-265.

Chung e Alonso (2004)

H.-S. Chung e J.J. Alonso (2004). "Multi-objective optimization using approximation model-based genetic algorithms". Technical Report AIAA 2004-4325 (URL: http://aero-comlab.stanford.edu/Papers/chung.aiaa.04-4325.pdf).

Clímaco et al. (2003)

J.N. Clímaco, C.H. Antunes e M.J. Alves (2003). "Programação linear multiobjectivo. Do modelo de programação linear clássico à consideração explícita de várias funções objectivo". Imprensa da Universidade de Coimbra.

Coello (1999)

C. Coello (1999). "A comprehensive survey of evolutionary-based multiobjective optimization techniques". Knowledge and Information Systems 1 (3), 129-156.

Coello (2000)

C.A.C. Coello (2000). "An updated survey of GA-based multiobjective optimization techniques". ACM Computing Surveys 32 (2). ACM, 109-143.

Coello et al. (2002)

C. Coello, D. Veldhuizen e G. Lamont (2002). "Evolutionary algorithms for solving multi-objective problems". Kluwer Academic Publishers.

Cohon (1978)

J.L. Cohon (1978). "Multiobjective programming and planning". Academic Press.

Costa (2003)

L. Costa (2003). "Algoritmos evolucionários em optimização uni e multi-objectivo". Tese de Doutoramento. Universidade do Minho.

Costa e Oliveira (2002)

L. Costa e P. Oliveira (2002). "An evolution strategy for multiobjective optimization". Proceedings of the 2002 Congress on Evolutionary Computation (CEC '02) 1. IEEE Press, 97-102.

Dantzig (1955)

G.B. Dantzig (1955). "Linear programming under uncertainty". Management Science 1 (3-4). Informs, 197-206.

Das (2000)

I. Das (2000). "Robustness optimization for constrained nonlinear programming problems". Engineering Optimization 32 (5). Taylor & Francis, 585-618.

Das et al. (1994)

D. Das, H.S. Nagi e D.P. Kothari (1994). "Novel method for solving radial distribution networks". IEE Proceedings - Generation, Transmission and Distribution 141 (4). IEEE Press, 291-298.

Deb (1989)

K. Deb (1989) "Genetic algorithms in multi-modal function optimization". Master Thesis. Department of Engineering Mechanics, University of Alabama.

Deb (1999)

K. Deb (1999). "Evolutionary algorithms for multi-criterion optimization in engineering design". Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming, and Industrial Applications. K. Miettinen, M. M. Makela, P. Neittaanmaki e J. Periaux (Eds.). John Wiley & Sons.

Deb (2001)

K. Deb (2001) "Multi-objective optimization using evolutionary algorithms". John Wiley & Sons.

Deb e Gupta (2004)

K. Deb e H. Gupta (2004) "Introducing robustness in multiple-objective optimization". KanGAL Report Number 2004016. Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur (URL: http://www.iitk.ac.in/kangal/papers/k2004016.pdf).

Deb e Gupta (2005)

K. Deb e H. Gupta (2005). "Searching for robust pareto-optimal solutions in multi-objective optimization". Evolutionary Multi-Criterion Optimization, Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005, Proceedings, Lecture Notes in Computer Science 3410. C.A.C. Coello, A.H. Aguirre e E. Zitzler (Eds.) Springer, 150-164.

Deb et al. (2000)

K. Deb, A. Pratap e T. Meyarivan (2000). "A fast and elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II". Parallel Problem Solving from Nature PPSN VI, Lecture Notes in Computer Science 1917. M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo e H.-P. Schwefel (Eds.). Springer, 849-858.

Deb et al. (2002)

K. Deb, A. Pratap, S. Agarwal e T. Meyarivan (2002). "A fast and elitist multi-objective genetic algorithm: NSGA-II". IEEE Transactions on Evolutionary Computation 6 (2). IEEE Press, 182-197.

De Jong (1975)

K.A. De Jong (1975). "An analysis of the behavior of a class of genetic adaptive systems". Ph.D. Thesis. University of Michigan Press.

De Jong e Sarma (1992)

K.A. De Jong e J. Sarma (1992). "Generation gap revisited". Foundations of Genetic Algorithms 2. D. Whitley (Ed.). Morgan Kaufmann Publishers, 19-28.

Delfanti et al. (2000)

M. Delfanti, G. Granelli, P. Marannino e M. Montagna (2000). "Optimal capacitor placement using deterministic and genetic algorithms". IEEE Transactions on Power Systems 15 (3). IEEE Press, 1041-1046.

Dias e Clímaco (1999)

L. Dias e J. Clímaco (1999). "On computing ELECTRE's credibility indices under partial information". Journal of Multi-Criteria Decision Analysis 8. John Wiley & Sons, 74-92.

Eiben e Smith (2003)

A.E. Eiben e J.E. Smith (2003). "Introduction to evolutionary computing". Natural Computing Series. Springer.

El-Beltagy (1999)

M.A. El-Beltagy, P.B. Nair e A.J. Keane (1999). "Metamodeling techniques for evolutionary optimization of computationally expensive problems: promises and limitations". Proceedings of Genetic and Evolutionary Computation Conference (GECCO'99). L. Spector, E.D. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M.-H. Garzon e E. Burke (Eds.). Morgan Kaufmann Publishers, 196-203.

El-Ghaoui e Lebret (1997)

L. El-Ghaoui e H. Lebret (1997). "Robust solutions to least-square problems to uncertain data matrices". SIAM Journal on Matrix Analysis and Applications 18. SIAM, 1035-1064.

El-Ghaoui et al. (1998)

L. El-Ghaoui, F. Oustry e H. Lebret (1998). "Robust solutions to uncertain semidefinite programs". SIAM Journal on Optimization 9 (1). SIAM, 33-52.

Emmerich et al. (2002)

M. Emmerich, A. Giotis, M. Özdenir, T. Bäck e K. Giannakoglou (2002). "Metamodel-assisted evolution strategies". Parallel Problem Solving from Nature PPSN VII, Lecture Notes in Computer Science 2439. J.J. Merelo-Guervos, P. Adamidis, H.-G. Beyer, J.L. Fernandez-Villacanas e H.-P. Schwefel (Eds.). Springer, 371-380.

Erickson et al. (2001)

M. Erickson, A. Mayer e J. Horn (2001). "The Niched Pareto Genetic Algorithm 2 applied to the design of groundwater remediation systems". Evolutionary Multi-Criterion Optimization, First International Conference, EMO 2001, Zurich, Switzerland, March 7-9, 2001, Proceedings, Lecture Notes in Computer Science 1993. E. Zitzler, K. Deb, L. Thiele, C.A.C. Coello e D. Corne (Eds.). Springer, 681–695.

Farina (2000)

M. Farina (2000). "A minimal cost hybrid strategy for Pareto optimal front approximation". Evolutionary Optimization 3 (1), 41-52.

Farina (2002)

M. Farina (2002). "A neural network based generalized response surface multiobjective evolutionary algorithms". Proceedings of the IEEE Congress on Evolutionary Computation (CEC'02). IEEE Press, 956-961.

Farina et al. (2004)

M. Farina, K. Deb e P. Amato (2004). "Dynamic multiobjective optimization problems: Test cases, approximations, and applications". IEEE Transactions on Evolutionary Computation 8 (5). IEEE Press, 425-442.

Fitzpatrick e Grefenstette (1988)

J.M. Fitzpatrick e J.J. Grefenstette (1988). "Genetic algorithms in noisy environments". Machine Language 3 (2-3). Kluwer Academic Publishers, 101-120.

Fleurent e Ferland (1994)

C. Fleurent e J.A. Ferland (1994). "Genetics hybrids for the quadratic assignment problem". Quadratic Assignment and Related Problems 16, DIMACS Series in Discrete Mathematics and Theoretical Computer Science. P.M. Pardalos e H. Wolkowicz (Eds.). American Mathematical Society, 173-188.

Fogel et al. (1966)

L.J. Fogel, A.J. Owens e M.J. Walsh (1966). "Artificial intelligence through simulated evolution". John Wiley & Sons.

Fonseca e Fleming (1993)

C.M. Fonseca e P.J. Fleming (1993). "Genetic algorithms for multi-objective optimization: formulation, discussion and generalization". Proceedings of the Fifth Internacional Conference on Genetic Algorithms. S. Forrest (Ed.). Morgan Kauffman Publishers, 416–423.

Fonseca e Fleming (1995)

C.M. Fonseca e P.J. Fleming (1995). "An overview of evolutionary algorithms in multiobjective optimization". Evolutionary Computation 3 (1), 1-16.

Fonseca e Fleming (1998-a)

C.M. Fonseca e P.J. Fleming (1998). "Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: A unified formulation". IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 28 (1). IEEE Press, 26 – 37.

Fonseca e Fleming (1998-b)

C.M. Fonseca e P.J. Fleming (1998). "Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part II: A application example". IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 28 (1). IEEE Press, 38 - 47.

Glover (1986)

F. Glover (1986). "Future paths for integer programming and links to artificial intelligence". Computers and Operations Research 13 (5). Elsevier, 533-549.

Goh e Tan (2006)

C.K. Goh e K.C. Tan (2006). "Noise handling in evolutionary multi-objective optimization". Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC'06). IEEE Press, 1354-1361.

Goldberg (1989)

D.E. Goldberg (1989). "Genetic algorithms in search, optimization and machine learning". Addison-Wesley.

Goldberg e Richardson (1987)

D.E. Goldberg e J. Richardson (1987). "Genetic algorithms with sharing for multimodal function optimization". Proceedings of the Second International Conference on Genetic Algorithms and Their Applications. J.J. Grefenstette (Ed.). Lawrence Erlbaum Associates, 41–49.

Goldberg et al. (1992)

D.E. Goldberg, K. Deb e J.H. Clark (1992). "Genetic algorithms, noise, and the sizing of populations". Complex Systems 6. Complex Systems Publications, 333-362.

Gomes et al. (2004)

A. Gomes, C.H. Antunes e A. Martins (2004). "A multiple objective evolutionary approach for the design and selection of load control strategies". IEEE Transactions on Power Systems 19 (2). IEEE Press, 1173-1180.

Greiner (1996)

H. Greiner (1996). "Robust optical coating design with evolution strategies". Applied Optics 35 (28). OSA, 5477-5483.

Guntsch e Middendorf (2000)

M. Guntsch e M. Middendorf (2002). "Applying population-based ACO to dynamic optimization problems". Proceedings of ANTS'2000, Lecture Notes in Computer Science 2463. M. Dorigo, G. Di Caro e M. Sampels (Eds.). Springer, 111-122.

Guntsch et al. (2000)

M. Guntsch, J. Branke, M. Middendorf e H. Schmeck (2000). "ACO strategies for dynamic TSP". Proceedings of ANTS'2000. M. Dorigo, M. Middendorf e T. Stützle (Eds.), 59-62.

Guntsch et al. (2001)

M. Guntsch, M. Middendorf e H. Schmeck (2001). "An ant colony optimization approach to dynamic TSP". Proceedings of Genetic and Evolutionary Computation Conference (GECCO'01). L. Spector, E.D. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M.H. Garzon e E. Burke (Eds.). Morgan Kaufmann Publishers, 860-867.

Gupta e Deb (2005)

H. Gupta e K. Deb (2005). "Handling constraints in robust multi-objective optimization". Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'05) 1. IEEE Press, 25-32.

Gupta e Rosenhead (1968)

S.K. Gupta e J. Rosenhead (1968). "Robustness in sequential investment decisions". Management Science 15 (2). Informs, 18-29.

Hajela e Lin (1992)

P. Hajela e C.-Y. Lin (1992). "Genetic search strategies in multicriterion optimal design". Structural and Multidisciplinary Optimization 4 (2). Springer, 99–107.

Haubelt et al. (2005)

C. Haubet, J. Gamenik e J. Teich (2005). "Initial population construction for convergence improvement of MOEAs". Evolutionary Multi-Criterion Optimization, Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005, Proceeding, Lecture Notes in Computer Science 3410. C.A.C. Coello, A.H. Aguirre e E. Zitzler (Eds.). Springer, 191-205.

Haubelt e Teick (2003)

C. Haubet e J. Teich (2003). "Accelerating design space exploration using pareto-front arithmetics". Proceedings of Asia and South Pacific Design, Automation and Conference. IEEE Press, 525-531.

Hill e Hiremath (2005)

R. Hill e C. Hiremath (2005). "Improving genetic algorithm convergence using problem structure and domain knowledge in multidimensional knapsack problems". International Journal of Operational Research 1 (1-2). Inderscience Publisers, 145-159.

Hites (2002)

R. Hites (2002). "A new method for solving combinatorial problems with uncertainty". Technical Report 2002/12. Service de Mathématiques de la Gestion. Université Libré de Bruxelles (URL: http://code.ulb.ac.be/dbfiles/media682.pdf).

Holland (1975)

J.H. Holland (1975). "Adaptation in natural and artificial systems". Ann Arbor, University of Michigan Press.

Horn et al. (1994)

J. Horn, N. Nafploitis e D. Goldberg (1994). "A niched pareto genetic algorithm for multi-objective optimization". Proceedings of the First IEEE Conference on Evolutionary Computation (CEC'94) 1. IEEE Press, 82–87.

Hughes (2001)

E.J. Hughes (2001). "Evolutionary multi-objective ranking with uncertainty and noise". Evolutionary Multi-Criterion Optimization, First International Conference, EMO 2001, Zurich, Switzerland, March 7-9, 2001, Proceedings, Lecture Notes in Computer Science 1993. E. Zitzler, K. Deb, L. Thiele, C.A.C Coello e D. Corne (Eds.). Springer, 329-343.

Iba (1994)

K. Iba (1994). "Reactive power optimization by genetic algorithm". IEEE Transactions on Power Systems 9 (2). IEEE Press, 685-692.

Inuiguchi e Kume (1994)

M. Inuiguchi e Y. Kume (1994). "Minimax regret in linear programming problems with an interval objective function". Multicriteria Decision Making. G.H. Tzeng, H.F. Wang, U.P. Wen e P.L. Yu (Eds.). Springer, 65–74.

Inuiguchi e Sakawa (1995)

M. Inuiguchi e M. Sakawa (1995). "Minimax regret solution to linear programming problems with an interval objective function". European Journal of Operational Research 86 (3). Elsevier, 526–536.

Jensen (2003)

M.T. Jensen (2003). "Generating robust and flexible job shop schedules using genetic algorithms". IEEE Transactions on Evolutionary Computation 7 (3). IEEE Press, 275-288.

Jin (2005)

Y. Jin (2005). "A compreensive survey of fitness approximation in evolutionary computation". Soft Computation 9 (1), 3-12.

Jin e Branke (2005)

Y. Jin e J. Branke (2005). "Evolutionary optimization in uncertain environments - A survey". IEEE Transactions on Evolutionary Computation 9 (3). IEEE Press, 1-15.

Jin e Sendhoff (2002)

Y. Jin e B. Sendhoff (2002). "Fitness approximation in evolutionary computation — — A survey". Proceedings of Genetic and Evolutionary Computation Conference (GECCO'02). W.B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M.A. Potter, A.C. Schultz, J.G. Miller, E. Burke e N. Jonoska (Eds). Morgan Kaufmann Publishers, 1105-1112.

Jin e Sendhoff (2003)

Y. Jin e B. Sendhoff (2003). "Trade-off between performance and robustness: An evolutionary multiobjective approach". Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, April 8-11, 2003, Proceedings, Lecture Notes in Computer Science 2632. C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, e L. Thiele (Eds.). Springer, 237-251.

Jin e Sendhoff (2004)

Y. Jin e B. Sendhoff (2004). "Reducing fitness evaluations using clustering techniques and neural network ensembles". Genetic and Evolutionary Computation. Springer, 688-699.

Jin et al. (2000)

Y. Jin, M. Olhofer e B. Sendhoff (2000). "On evolutionary optimization with approximate fitness functions". Proceedings of Genetic and Evolutionary Computation Conference (GECCO'00). L.D. Whitley, D.E. Goldberg, E. Cantú-Paz, L. Spector, I.C. Parmee e H.-G. Beyer (Eds.). Morgan Kaufmann Publisers, 786-792.

Jin et al. (2001-a)

Y. Jin, M. Olhofer e B. Sendhoff (2001). "Dynamic weighted aggregation for evolutionary multi-objective optimization: why does it work and how?". Proceedings of Genetic and Evolutionary Computation Conference (GECCO'01). L. Spector, E.D. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M.-H. Garzon e E. Burke (Eds). Morgan Kaufmann Publishers, 1042-1049.

Jin et al. (2001-b)

Y. Jin, M. Olhofer e B. Sendhoff (2001). "Managing approximate models in evolutionary aerodynamic design optimization". Proceedings of IEEE Congress on Evolutionary Computation (CEC'01) 1. IEEE Press, 592 – 599.

Jin et al. (2002)

Y. Jin, M. Olhofer e B. Sendhoff (2002). "A framework for evolutionary optimization with approximate fitness functions". IEEE Transactions on Evolutionary Computation 6 (5). IEEE Press, 481-494.

Kennedy e Eberhart (2001)

J. Kennedy e R. Eberhart (2001). "Swarm intelligence". Morgan Kaufmann Publishers.

Kicinger et al. (2005)

R. Kicinger, T. Arciszewski e K.A. De Jong (2005). "Evolutionary computation and structural design: A survey of the state of the art". Computers & Structures, 83 (23-24) : 1943-1978.

Kim e You (1999)

K.-H. Kim, S.-K. You (1999). "Voltage profile improvement by capacitor placement and control in unbalanced distribution systems using GA". Proceedings of IEEE Power Engineering Society Summer Meeting 2. IEEE Press, 800-805.

Knowles e Corne (2000)

J.D. Knowles e D.W. Corne (2000). "Approximating the nondominated front using the pareto archived evolution strategy". Evolutionary Computation 8 (2), 149-172.

Kouvelis e Yu (1997)

P. Kouvelis e G. Yu (1997). "Robust discrete optimization and its applications". Kluwer Academic Publishers.

Lee et al. (1995)

K.Y. Lee, X. Bai e Y.-M. Park (1995). "Optimization method for reactive power planning by using a modified simple genetic algorithm". IEEE Transactions on Power Systems 10 (4). IEEE Press, 1843-1850.

Leon et al. (1994)

V.J. Leon, S.D. Wu e R.H. Storer (1994). "Robustness measures and robust scheduling for job shops". IIE Transactions 26 (5), 32-43.

Levitin et al. (2000)

G. Levitin, A. Kalyuhny, A. Shenkman e M. Chertkov (2000). "Optimal capacitor allocation in distribution systems using a genetic algorithm and a fast energy loss computation technique". IEEE Transactions on Power Delivery 15 (2). IEEE Press, 623-628.

Li et al. (2005)

M. Li, S. Azarm e V. Aute (2005). "A multi-objective genetic algorithm for robust design optimization". Proceedings of Genetic and Evolutionary Computation Conference (GECCO'05). H.-G. Beyer e U.-M. O'Reilly (Eds.). ACM, 771-778.

Lim et al. (2005)

D. Lim, Y.S. Ong e B.S. Lee (2005). "Inverse multi-objective robust evolutionary design optimization in the presence of uncertainty". Workshops Proceedings of Genetic and Evolutionary Computation Conference (GECCO WORKSHOPS 2005). F. Rothlauf (Ed.). ACM, 55-62.

Lin et al. (1997)

S.-C. Lin, E.D. Goodman e W.F. Punch (1997). "A genetic algorithm approach to dynamic job shop scheduling problems". Proceedings of the 7th International Conference of Genetic Algorithms. T. Bäck (Ed.). Morgan Kaufmann Publishers, 481-488.

Loughlin e Ranjithan (1999)

D.H. Loughlin e S.R. Ranjithan (1999). "Chance-constrained genetic algorithms". Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99). L. Spector, E.D. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M.-H. Garzon e E. Burke (Eds). Morgan Kaufmann Publishers, 369-376.

Limbourg e Aponte (2005)

P. Limbourg e D.S. Aponte (2005). "An optimization algorithm for imprecise multi-objective problem functions". Proceedings of IEEE Congress on Evolutionary Computation (CEC'05). IEEE Press, 459-466.

Mahfoud e Goldberg (1995)

S.W. Mahfoud e D.E. Goldberg (1995). "Parallel recombinative simulated annealing: A genetic algorithm". Parallel Computing 21 (1), 1-28.

Markon et al. (2001)

S. Markon, D.V. Arnold, T. Bäck, T. Beielstein e H.-G. Beyer (2001). "Thresholding - a selection operator for noisy ES". Proceedings of the IEEE Congress on Evolutionary Computation (CEC'01). IEEE Press, 465-472.

Matos (2007)

M.A. Matos (2007). "Decision under risk as a multicriteria problem". European Journal of Operational Research 181. Elsevier, 1516–1529.

Mendes et al. (2004)

R. Mendes, J. Kennedy e J. Neves (2004). "The fully informed particle swarm: Simpler, maybe better". IEEE Transactions on Evolucionary Computation 8 (3). IEEE Press, 204-210.

Michalewicz (1996)

Z. Michalewicz (1996). "Genetic algorithms + Data structures = Evolution programs". Springer.

Miettinen (1999)

K.M. Miettinen (1999). "Nonlinear multiobjective optimization". Kluwer Academic Publishers.

Miller (1997)

B.L. Miller (1997) "Noise, sampling, and efficient genetic algorithms". Ph.D. Thesis. University of Illinois at Urbana-Champaign, Urbana.

Miller e Goldberg (1996)

B.L. Miller e D.E. Goldberg (1996). "Genetic algorithms, selection schemes, and the varying effects of noise". Evolutionary Computation 4 (2), 113-131.

Mitchell (1997)

M. Mitchell (1997). "An introduction to genetic algorithms". Cambridge, Massachusetts, London. MIT Press.

Morrison (2004)

R.W. Morrison (2004). "Designing evolutionary algorithms for dynamic environments". Natural Computing Series. Springer.

Mulvey et al. (1995)

J.M. Mulvey, M.J. Verderbel e S.A. Zenios (1995). "Robust optimisation of large scale systems". Operations Research 43 (2). Informs, 264-281.

Mutoh et al. (2003)

A. Mutoh, S. Kato, T. Nakamura e H. Itoh (2003). "Reducing execution time on genetic algorithms in real-world applications using fitness prediction". Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2003) 1. IEEE Press, 552-559.

Nair e Keane (1998)

P. Nair e A. Keane (1998). "Combining approximation concepts with genetic algorithm-based structural optimization procedures". Proceedings of the 39th AIAA/ ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, CA, AIAA-98-1912. AIAA, 1741-1751.

Oberkampf et al. (2004)

W.L. Oberkampf, J.C. Helton, C.A. Joslyn, S.F. Wojtkiewicz e S. Ferson (2004). "Challenge problems: Uncertainty in system response given uncertain parameters". Reliability Engineering & System Safety 85. Elsevier, 11-19.

Oliveira e Antunes (2007)

C. Oliveira e C.H. Antunes (2007). "Multiple objective linear programming models with interval coefficients – An illustrated overview". European Journal of Operational Research 181. Elsevier, 1434–1463.

Ong et al. (2006)

Y.S. Ong, P.B. Nair e K.Y. Lum (2006). "Max-min surrogate-assisted evolutionary algorithm for robust design". IEEE Transactions on Evolutionary Computation 10 (4). IEEE Press, 392-404.

Osyczka (2002)

A. Osyczka (2002). "Evolutionary algorithms for single and multicriteria design optimization". Structural and Multidisciplinary Optimization 24 (1). Springer, 88-89.

Paenke et al. (2006)

I. Paenke, J. Branke e Y. Jin (2006). "Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation". IEEE Transactions on Evolutionary Computation 10 (4). IEEE Press, 405-420.

Pires (2001)

D.F. Pires (2001). "Modelo multiobjectivo para apoio à localização de condensadores em redes radiais de distribuição". Tese Mestrado em Sistemas e Automação. Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Pires et al. (2005)

D.F. Pires, A.G. Martins e C.H. Antunes (2005). "A multiobjective model for VAR planning in radial distribution networks based on Tabu Search". IEEE Transactions on Power Systems 20 (2). IEEE Press, 1089-1094.

Poles et al. (2006)

S. Poles, Y. Fu e E. Rigoni (2006). "The effect of initial population sampling on the convergence of multi-objective genetic algorithms". Proceedings of the 7th International Conference on Multi-Objective Programming and Goal Programming (MOPGP'06).

Prins (2004)

C. Prins (2004). "A simple and effective evolutionary algorithm for the vehicle routing problem". Computers & Operations Research 31 (12), 1985-2002.

Rasheed (2000)

K. Rasheed (2000). "Informed operators: speeding up genetic-algorithm-based design optimization using reduced models". Proceedings of genetic and Evolutionary Computation Conference (GECCO'00). L.D. Whitley, D.E. Goldberg, E. Cantú-Paz, L. Spector, I.C. Parmee e H.-G. Beyer (Eds.). Morgan Kaufmann Publishers, 628-635.

Rattray e Shapiro (1997)

L.M. Rattray e J. Shapiro (1997). "Noisy Fitness Evaluation in Genetic Algorithms and the Dynamics of Learning". Foundations of Genetic Algorithms. R.K. Belew e M.D. Vose (Eds.). Morgan Kaufmann Publishers, 117-139.

Ray (2002)

T. Ray (2002). "Constrained robust optimal design using a multi-objective evolutionary algorithm". Proceedings of the IEEE Congress on Evolutionary Computation (CEC'02). IEEE Press, 419-424.

Ray e Tsai (2004)

T. Ray e H.M. Tsai (2004). "A parallel hybrid optimization algorithm for robust airfoil design". 42nd AIAA Aerospace Sciences Meeting and Exhibit. AIAA, 11474–11482.

Rechenberg (1964)

I. Rechenberg (1964). "Cybernetic solution path of an experimental problem". Royal Aircraft Establishment. Farnborough, England. Library Translation Nº 1122.

Rechenberg (1973)

I. Rechenberg (1973). "Evolutionsstratgie: optimierung technischer systeme nach prinzipien der biologischen evolution". Frommann-Holzboog Verlag.

Reeves (1995)

C.R. Reeves (1995), "Modern heuristic techniques for combinatorial problems". McGraw-Hill International.

Rosenblatt e Lee (1987)

M.J. Rosenblatt e H.L. Lee (1987). "A robustness approach to facilities design". International Journal of Production Research 25, 479-486.

Rosenhead (1989)

M.J. Rosenhead (1989). "Rational analysis for a problematic world". John Wiley & Sons.

Rosenhead et al. (1972)

M.J. Rosenhead, M. Elton e S.K. Gupta (1972). "Robustness and optimality as criteria for strategic decisions". Operational Research Quarterly 23 (4). OSR, 413-430.

Roy (1998)

B. Roy (1998). "A missing link in operational research decision aiding: robustness analysis". Foundations of Computing and Decision Sciences, 23 (3) : 141-160.

Rudolph (1999)

G. Rudolph (1999). "Evolutionary search under partially ordered sets". Technical Report CI-67/99. Department of Computer Science/LS11, University of Dortmund.

Sano e Kita (2000)

Y. Sano e H. Kita (2000). "Optimization of noisy fitness functions by means of genetic algorithms using history of search". Parallel Problem Solving from Nature PPSN VI, Lecture Notes in Computer Science 1917. M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo e H.-P. Schwefel (Eds). Springer : 571-580.

Sano e Kita (2002)

Y. Sano e H. Kita (2002). "Optimization of noisy fitness functions by means of genetic algorithms using history of search with test of estimation". Proceedings of the IEEE Congress on Evolutionary Computation (CEC'02). IEEE Press, 360-365.

Sano et al. (2000)

Y. Sano, H. Kita, I. Kamihira e M. Yamaguchi (2000). "Online optimization of an engine controller by means of a genetic algorithm using history of search". Proceedings of the Asia-Pacific Conference on Simulated Evolution and Learning. Springer, 2929-2934.

Schaffer (1984)

J.D. Schaffer (1984) "Some experiments in machine learning using vector evaluated genetic algorithms". Ph.D. Thesis. Vanderbilt University, Nashville, TN.

Schaffer (1985)

J.D. Schaffer (1985). "Multiple objective optimization with vector evaluated genetic algorithms". Proceedings of the First International Conference on Genetic Algorithms and Their Applications. J.J. Grefensttete (Ed.). Lawrence Erlbaum Associates, 93-100.

Schwefel (1981)

H.-P. Schwefel (1981). "Numerical optimization of computer models". John Wiley & Sons.

Schwefel (1995)

H.-P. Schwefel (1995). "Evolution and optimum seeking". John Wiley & Sons.

Sebald e Fogel (1992)

A.V. Sebald e D.B. Fogel (1992). "Design of fault tolerant neural networks for pattern classification". Proceedings of the First Annual Conference of Evolutionary Programming. D.B. Fogel e W. Atmar (Eds.). Evolutionary Programming Society, 90-99.

Sevaux e Sörensen (2004)

Sevaux e K. Sörensen (2004). "Robustness analysis: optimization". European Working Group "Multiple Criteria Decision Aiding" Series 3 Nº10.

Silverman (1986)

B.W. Silverman. "Density estimation for statistics and data analysis". Monographs on Statistics and Applied Probability. Chapman and Hall.

Smith e Vavak (1999)

J.E. Smith e F. Vavak (1999). "Replacement strategies in steady state genetic algorithms: Dynamics environments". Journal of Computing & Information Technology, Special Issue on Evolutionary Computing 7 (1), 49-60.

Sörensen (2001)

K. Sörensen (2001). "Tabu searching for robust solutions". Proceedings of the 4th Metaheuristics International Conference (MIC'01), 707-712.

Sörensen (2003)

K. Sörensen (2003) "A framework for robust and flexible optimization using metaheuristics". Ph.D. Thesis. University of Antwerp.

Soyster (1973)

A.L. Soyster (1973). "Convex programming with set-inclusive constraints and applications to inexact linear programming". Operations Research 21 (5). Informs, 1154-1157.

Srinivas e Deb (1994)

N. Srinivas e K. Deb (1994). "Multiobjective optimization using nondominated sorting in genetic algorithms". Evolutionary Computation 2 (3), 221-248.

Stanhope e Daida (1999)

S.A. Stanhope e J.M. Daida (1999). "Genetic algorithm fitness dynamics in a changing environment". Proceedings of the IEEE Congress on Evolutionary Computation (CEC'99) 3. IEEE Press, 1851-1858.

Steuer (1986)

R. Steuer (1986). "Multiple criteria optimization : Theory, computation and application". John Wiley & Sons.

Taillard (1990)

E. Taillard (1990). "Robust taboo search for the quadratic assignment problem". Parallel Computing 17 (4-5). Elsevier, 443-455.

Teich (2001)

J. Teich (2001). "Pareto-front exploration with uncertain objectives". Evolutionary Multi-Criterion Optimization, First International Conference, EMO 2001, Zurich, Switzerland, March 7-9, 2001, Proceedings, Lecture Notes in Computer Science 1993. E. Zitzler, K. Deb, L. Thiele, C.A.C. Coello e D. Corne (Eds.). Springer, 314-328.

Thompson (1996)

A. Thompson. "Evolutionary techniques for fault tolerance". Proceedings of UKACC International Conference on Control, 693–698.

Thompson (1998)

A. Thompson (1998). "On the automatic design of robust electronics through artificial evolution". Proceedings of the Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98). Springer, 13-24.

Tsutsui e Ghosh (1997)

S. Tsutsui e A. Ghosh (1997). "Genetic algorithm with a robust solution searching scheme". IEEE Transactions on Evolutionary Computation 1 (3). IEEE Press, 201-219.

Tsutsui et al. (1996)

S. Tsutsui, A. Ghosh e Y. Fujimoto (1996). "A robust solution searching scheme in genetic search". Parallel Problem Solving from Nature PPSN IV, Lecture Notes in Computer Science. H.-M. Voigt, W. Ebeling, I. Rechenberg e H.-P. Schwefel (Eds.). Springer, 543-552.

Veldhuizen (1999)

D.A.V. Veldhuizen (1999). "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations". Ph.D. Thesis. Graduate School of Engineering of the Air Force Institute of Technology, Air University.

Veldhuizen e Lamont (1998)

D.A.V. Veldhuizen e G.B. Lamont (1998). "Multiobjective evolutionary algorithm research: A history and analysis". Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Veldhuizen e Lamont (2000)

D.A. Veldhuizen e G.B. Lamont (2000). "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art". Evolutionary Computation 8 (2), 125-147.

Vincke (1999-a)

P. Vincke (1999). "Robust solutions and methods in decision-aid". Journal of Multicriteria Decision Analysis 8. John Wiley & Sons, 181-187.

Vincke (1999-b)

P. Vincke (1999). "Robust and neural methods for aggregating preferences into an outranking relation". European Journal of Operational Research 112. Elsevier, 405-412.

Vincke (2003)

P. Vincke (2003). "About robustness analysis". European Working Group "Multiple Criteria Decision Aiding" Newsletter 3 (8).

Weicker (2003)

K. Weicker (2003). "Evolutionary algorithms and dynamic optimization problems". Der Andere Verlag.

Whitley (1989)

D. Whitley (1989). "The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best". Proceedings of the Third International Conference on Genetic Algorithms. G.J.E. Rawlings (Ed.). Morgan Kaufmann Publishers, 116-121.

Wiesmann et al. (1998)

D. Wiesmann, U. Hammel e T. Bäck (1998). "Robust design of multilayer optical coatings by means of evolutionary algorithms". IEEE Transactions on Evolutionary Computation 2 (4). IEEE Press, 162-167.

Yamaguchi e Arima (2002)

Y. Yamaguchi e T. Arima (2002). "Aerodynamic optimization for the transonic compressor stator blade". Optimization in Industry. I.C. Parmee e P. Hajela (Eds.). Springer, 163–172.

Zadeh (1965)

L.A. Zadeh (1965). "Fuzzy sets". Information and Control 8, 338-353.

Zimmermann (1998)

H.J. Zimmermann (1998). "A fresh perspective on uncertainty modeling: Uncertainty vs. uncertainty modeling". Uncertainty Analysis in Engineering and Sciences: Fuzzy Logic, Statistics, and Neural Network Approach. International Series in Intelligent Technologies 11. B.M. Ayyub e M.M. Gupta (Eds.). Kluwer Academic Publishers, 353-364.

Zitzler (1999)

E. Zitzler (1999). "Evolutionary algorithms for multiobjective optimization: Methods and applications". Ph.D. Thesis. Swiss Federal Institute of Technology Zurich.

Zitzler e Thiele (1998)

E. Zitzler e L. Thiele (1998). "An evolutionary algorithm for multiobjective optimization: The strength pareto approach". Technical Report 43, Computer Engineering and Networks Laboratory (TIK). Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich (URL: http://www.tik.ee.ethz.ch/sop/publicationListFiles/zt1998a.pdf).

Zitzler e Thiele (1999)

E. Zitzler e L. Thiele (1999). "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach". IEEE Transactions on Evolutionary Computation 3 (4). IEEE Press, 257–271.

Zitzler et al. (2000)

E. Zitzler, K. Deb e L. Thiele (2000). "Comparison of multiobjective evolutionary algorithms: Empirical results". Evolutionary Computation 8 (2), 173–195.

Zitzler et al. (2001)

E. Zitzler, M. Laumanns e L. Thiele (2001). "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization". Proceedings of the EUROGEN 2001 — Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. K.C. Giannakoglou et al. (Eds.), 95–100.

ANEXO

Problema da compensação de energia reactiva em redes de distribuição

1. Modelo matemático

O problema da compensação de energia reactiva consiste em determinar o número, a localização e a dimensão dos condensadores *shunt* a instalar, neste caso numa rede radial, de forma a reduzir as perdas do sistema e melhorar o perfil de tensões. No entanto, é necessário um equilíbrio entre os custos (associados à instalação de novos condensadores) e os aspectos de avaliação técnica e de qualidade de serviço.

O modelo matemático usado neste trabalho, que foi desenvolvido por Pires et al. (2005), assume explicitamente a natureza multi-objectivo do problema, considerando duas funções objectivo: minimização das perdas (resistivas) e minimização dos custos de instalação de novas fontes de potência reactiva. Os requisitos de qualidade de serviço de um perfil de tensões aceitável nos barramentos são incluídos como restrições que resultam da legislação.

A terminologia usada na definição do modelo é a seguinte:

SE	sub-estação
m	índice de barramento (nó) de ramo lateral
n	índice de alimentador (ramo) lateral
k	índice de barramento do alimentador principal
L	índice de variável do tipo carga

C índice de variável do tipo compensação

F índice de variável de um tipo de condensador padrão

c_j custo do condensador de capacidade Q_{F_j}

$P^k_{L_{nm}}$ carga (potência real) no barramento *m* (do lateral *n*, derivado do barramento *k* do alimentador principal)

$Q^k_{L_{nm}}$ carga (potência reactiva) no barramento *m* (do lateral *n*, derivado do barramento *k* do alimentador principal)

P^k_{nm} potência real que flui no ramo *m*; isto é, passa do barramento *m* para o *m+1* (lateral *n*, derivado do barramento *k* do alimentador principal)

Q^k_{nm} potência reactiva que flui no ramo *m*; isto é, passa do barramento *m* para o *m+1* (lateral *n*, derivado do barramento *k* do alimentador principal)

$Q^k_{C_{nm}}$ potência reactiva injectada desde o condensador Q_C localizado no barramento *m* (lateral *n*, derivado do barramento *k* do alimentador principal)

B^k_{nm} barramento *m* do lateral *n* (lateral *n* derivado do barramento *k* do alimentador principal)

V^k_{nm} tensão no barramento *m* (lateral *n* derivado do barramento *k* do alimentador principal)

T^k_{nm} ramo *m* do lateral *n* (lateral *n* derivado do barramento *k* do alimentador principal); este ramo liga o barramento B^k_{nm} ao $B^k_{n(m+1)}$

Q_{F_j} capacidade do condensador padrão j

M^k_n número de barramentos do lateral *n* (lateral *n* derivado do barramento *k* do alimentador principal)

N^k número de laterais que derivam do barramento *k* do alimentador principal

K número de barramentos do alimentador principal

Y número de condensadores considerados para instalação

r^k_{nm} resistência do troço *m* do lateral *n* (lateral *n* derivado do barramento *k* do alimentador principal)

x^k_{nm} indutância do troço *m* do lateral *n* (lateral *n* derivado do barramento *k* do alimentador principal)

$$b^k_{nm} = \begin{cases} 1, & \text{se é possível instalar um condensador em } B^k_{nm} \\ 0, & \text{caso contrário} \end{cases}$$

O conjunto de equações recursivas (1-5) descreve os requisitos físicos associados ao fluxo de potência em cada ramo num sistema de distribuição radial.

O cálculo do fluxo de carga impõe uma significativa carga computacional na avaliação do mérito de cada solução. O procedimento usado para este fim (equações recursivas 1-5) foi adaptado às redes radiais, tal como as usadas no sistema de distribuição eléctrica portuguesa (ver Baran e Wu (1989-a), Baran e Wu (1989-b), Das et al. (1994) e Pires et al. (2005)).

$$P^k_{n(i+1)} = P^k_{ni} - r^k_{ni}\frac{\left(P^k_{ni}\right)^2+\left(Q^k_{ni}\right)^2}{\left(V^k_{ni}\right)^2} - P^k_{L_{n(i+1)}}, \quad \forall n\neq 0 \text{ e } k\neq 0 \tag{1}$$

$$P^0_{0(i+1)} = P^0_{0i} - r^0_{0i}\frac{\left(P^0_{0i}\right)^2+\left(Q^0_{0i}\right)^2}{\left(V^0_{0i}\right)^2} - P^0_{L_{0(i+1)}} - \sum_{n=1}^{N^{i+1}} P^{(i+1)}_{n0} \tag{2}$$

$$Q^k_{n(i+1)} = Q^k_{ni} - x^k_{ni}\frac{\left(P^k_{ni}\right)^2+\left(Q^k_{ni}\right)^2}{\left(V^k_{ni}\right)^2} - Q^k_{L_{n(i+1)}} + Q^k_{C_{n(i+1)}}, \quad \forall n\neq 0 \text{ e } k\neq 0 \tag{3}$$

$$Q^0_{(i+1)} = Q^0_{0i} - x^0_{0i}\frac{\left(P^0_{0i}\right)^2+\left(Q^0_{0i}\right)^2}{\left(V^0_{0i}\right)^2} - Q^0_{L_{0(i+1)}} - \sum_{n=1}^{N^{i+1}} Q^{(i+1)}_{n0} + Q^0_{C_{0(i+1)}} \tag{4}$$

$$\left(V^k_{n(i+1)}\right)^2 = \left(V^k_{ni}\right)^2 - 2\left(r^k_{ni}\,P^k_{ni} + x^k_{ni}\,Q^k_{ni}\right) + \left(\left(r^k_{ni}\right)^2+\left(x^k_{ni}\right)^2\right)\frac{\left(P^k_{ni}\right)^2+\left(Q^k_{ni}\right)^2}{\left(V^k_{ni}\right)^2} \tag{5}$$

O alimentador principal tem índice o n = 0, isto é, é considerado como lateral 0, e k = 0, isto é, começa na subestação (SE). Além das equações de fluxo de potência, existem outras condições a serem satisfeitas por cada lateral (incluindo o alimentador). A partir do último barramento (nó) de cada ramo, não existe potência (real ou reactiva) a fluir para outros ramos:

$$P^k_{nm} = Q^k_{nm} = 0 \tag{6}$$

Consideram-se duas funções objectivo, relacionadas com a minimização das perdas resistivas da rede (7) e com a minimização dos custos associados com a instalação dos condensadores (8).

$$\text{Min} \sum_{k=0}^{K}\sum_{n=0}^{N^k}\sum_{m=0}^{M^k_n} r^k_{ni}\frac{\left(P^k_{ni}\right)^2+\left(Q^k_{ni}\right)^2}{\left(V^k_{ni}\right)^2} \tag{7}$$

$$\text{Min} \sum_{k=0}^{K}\sum_{n=0}^{N^k}\sum_{m=0}^{M^k_n}\sum_{j=1}^{Y}\left({}^{j}a^k_{nm}\,c_j\right) \tag{8}$$

Os novos condensadores caracterizam-se pelas suas capacidades e pelo seu custo de instalação. Consideram-se unidades padrão, pois são as que geralmente são usadas nos sistemas de distribuição. c_j é o custo do condensador de capacidade Q_{F_j} (j = 1,...,Y) e,

$$ {}^{j}a_{nm}^{k} = \begin{cases} 1, & \text{se é instalado um condensador de capacidade } Q_{F_j} \text{ em } B_{nm}^{k} \\ 0, & \text{caso contrário} \end{cases} \tag{9} $$

$$ Q_{C_{nm}}^{k} = b_{nm}^{k} \sum_{j=1}^{Y} \left({}^{j}a_{nm}^{k} Q_{F_j} \right), \quad \forall\, m,n,k \tag{10} $$

As restrições (1-6), equações de fluxo de carga, são de natureza física. As restrições (11) impõem que pode ser colocado, no máximo, um condensador em B_{nm}^{k}.

$$ \sum_{j=1}^{Y} \left({}^{j}a_{nm}^{k} \right), \quad \forall\, m,n,k \tag{11} $$

As restrições (12) estão relacionadas com a qualidade de serviço, e correspondem aos limites inferiores e superiores da intensidade da tensão nos nós da rede.

$$ \left(V_{nm}^{k} \right)_{min} \leqslant V_{nm}^{k} \leqslant \left(V_{nm}^{k} \right)_{max} \tag{12} $$

O problema multi-objectivo aqui formulado é não linear e envolve variáveis discretas e contínuas.

2. Redes radiais

Uma rede de distribuição radial pode ser representada pela estrutura apresentada na Figura 1. O início da rede encontra-se na subestação (SE), a qual é também o início do alimentador principal. Ao longo deste alimentador há pontos de derivação (nós ou barramentos), nos quais se podem encontrar cargas activas e reactivas alimentadas directamente, ramos laterais, postos de transformação e elementos de controlo da potência reactiva. Estes últimos caracterizam-se por fornecerem energia reactiva à rede (através de condensadores, por exemplo) ou por consumirem energia reactiva da rede (caso das indutâncias). Os ramos laterais têm a mesma estrutura do alimentador principal, mas com algumas alterações devido à sua menor dimensão.

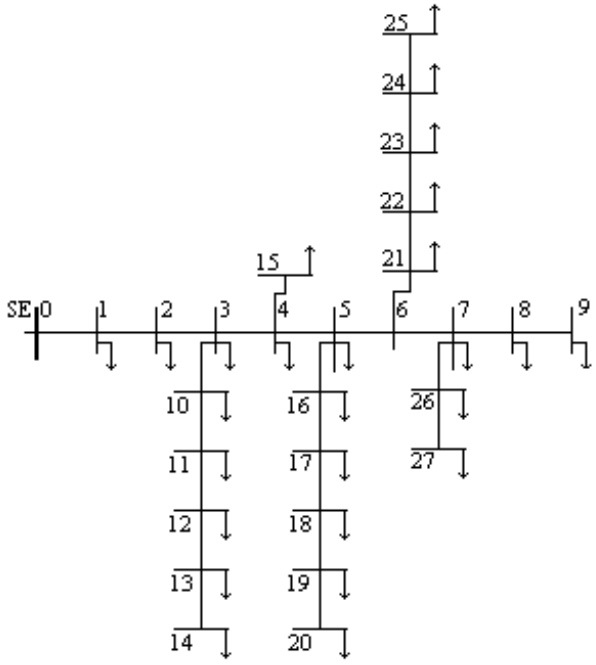

Figura 1: Exemplo de um sistema de distribuição radial.

3. Definição do cenário de carga

Para se definir a compensação de energia reactiva para uma rede de distribuição, é necessário determinar o cenário de carga para o qual vão ser dimensionadas e colocadas as baterias de condensadores. Para a rede portuguesa, não se consideraram como cenários de estudo, nem a ocorrência simultânea de pontas, nem situações intermédias arbitrariamente caracterizadas por uma fracção da ponta igual em todos os nós de carga. Procurou-se, sim, estabelecer para os diversos nós da rede valores de fracção da ponta diferentes uns dos outros, na proximidade das condições de carga mais severas da rede. Não sendo possível obter o diagrama de carga para esta rede, devido à dificuldade de dispor de diagramas de carga típicos de todos os nós e fazer uma análise repetida ao longo de um dia, com um intervalo de tempo de, por exemplo 1 hora, estabeleceu-se o seguinte processo (Pires (2001)):

1) considerou-se um intervalo de tempo, entre as 9 horas e as 12 horas, durante o qual se supõe que todas as pontas ocorrem;
2) considerou-se uma curva aproximativa da zona da ponta de um diagrama de carga, com uma duração superior ao intervalo considerado;
3) considerou-se uma distribuição aleatória dos instantes de ocorrência das pontas (numa determinada hora, em alguns postos de transformação estará a ocorrer a ponta, outros terão apenas uma percentagem da carga máxima);

4) com os dados obtidos traçou-se a ponta de um diagrama de carga, entre as 9 horas e as 12 horas, a partir do qual é possível obter vários cenários de carga da rede.

Para realizar o ponto 3, considerou-se uma função que distribui aleatoriamente as pontas dos 94 postos de transformação da rede eléctrica de EDP-Distribuição, ao longo do intervalo 9-12 horas.

Com este procedimento obteve-se um conjunto de situações que se aproxima mais das condições de ponta da rede, sem ter que se considerar a ocorrência simultânea de pontas em todos os postos de transformação (situação pouco provável e que facilmente poderia conduzir a solução para uma situação de sobrecompensação). Desta forma, adoptou-se a situação de carga correspondente às 11 horas para fazer a colocação das baterias na rede, já que foi quando se obteve a maior percentagem de carga no período 9-12 horas (ver Pires (2001)).

4. Características das redes usadas

Neste anexo apresentam-se os dados associados às redes eléctricas que serviram para testar as metodologias propostas neste trabalho.

Inicialmente apresentam-se as tabelas com os dados de cada uma das redes usadas para testar a metodologia apresentada e dos condensadores que poderão ser instalados, a qual também se encontra em Pires (2001).

Depois, apresentam-se os dados referentes às perdas activas e reactivas nos troços, e o perfil de tensões em cada nó, quando as redes são consideradas sem compensação (isto é, sem qualquer condensador instalado), os quais foram determinados com as ferramentas construídas como suporte da metodologia proposta neste trabalho.

4.1. Rede portuguesa

Nesta secção apresentam-se os dados referentes às características das entidades associadas à rede (troços e nós) portuguesa, assim como os resultados obtidos com o cálculo do trânsito de potência nas redes sem compensação. Nesta rede, a tensão da subestação (nó 0) é de 15.75 MVAr. De referir ainda que é possível instalar condensadores em todos os nós da rede, com excepção do nó 0 (subestação).

Troço	Origem	Destino	Resistência (Ohm)	Reactância (Ohm)
0	0	1	0.112	0.1873
1	1	2	0.0763	0.1274
2	2	3	0.1891	0.3161
3	3	4	0.2243	0.3749
4	4	5	0.2571	0.4297
5	5	6	0.134	0.2239
6	6	7	0.2986	0.4991
7	7	8	0.1953	0.3265
8	8	9	0.5097	0.8519
9	9	10	1.5303	1.5101
10	10	11	0.1889	0.1864
11	11	12	0.1816	0.1793
12	12	13	0.0661	0.0653
13	13	14	0.4115	0.4061
14	14	15	0.2584	0.255
15	15	16	0.2033	0.2006
16	16	17	0.7243	0.7148
17	17	18	0.2162	0.2134
18	18	19	0.35	0.3454
19	19	20	1.4775	0.3891
20	20	21	0.45	0.1185
21	21	22	0.771	0.203
22	22	23	0.885	0.2331
23	23	24	0.9915	0.2611
24	24	25	0.384	0.1011
25	25	26	0.7245	0.1908
26	26	27	1.185	0.3121
27	27	28	1.2353	0.6899
28	28	29	0.3557	0.1987
29	29	30	0.9494	0.3406
30	30	31	0.6899	0.3853
31	31	32	1.5707	0.8773
32	4	33	1.2655	0.454
33	4	34	0.1688	0.0943
34	34	35	0.2741	0.1531
35	35	36	0.2552	0.1425
36	5	37	0.4165	0.2326
37	5	38	1.4835	0.3907
76	76	77	0.2113	0.118
77	77	78	1.1249	0.4035
78	78	79	1.1738	0.6556
79	79	80	0.619	0.3457
80	80	81	0.5684	0.3174
81	19	82	0.8393	0.3011
82	82	83	0.2133	0.1191
83	83	84	0.3645	0.2036
84	84	85	0.3206	0.1791

Troço	Origem	Destino	Resistência (Ohm)	Reactância (Ohm)
38	38	39	1.8	0.474
39	39	40	0.5177	0.2892
40	40	41	0.7148	0.3992
41	7	42	1.0575	0.2785
42	42	43	0.5198	0.2903
43	43	44	0.3341	0.1866
44	8	45	0.349	0.1949
45	9	46	0.5771	0.3223
46	46	47	0.3598	0.2009
47	47	48	0.7688	0.4294
48	48	49	0.2599	0.1451
49	49	50	0.8654	0.4833
50	9	51	0.5248	0.5179
51	51	52	0.1737	0.1714
52	52	53	0.6148	0.6068
53	53	54	0.198	0.1954
54	54	55	0.198	0.1954
55	55	56	0.285	0.2813
56	56	57	0.1429	0.141
57	57	58	0.3409	0.1904
58	58	59	0.3679	0.2055
59	59	60	0.3591	0.2006
60	60	61	0.3503	0.1957
61	61	62	0.4219	0.2356
62	62	63	1.538	0.5517
63	63	64	0.9788	0.3511
64	64	65	1.4911	0.5349
65	10	66	0.969	0.2552
66	66	67	0.6705	0.1766
67	11	68	0.4354	0.2432
68	12	69	0.4631	0.2586
69	69	70	0.2707	0.1512
70	14	71	0.6683	0.3732
71	71	72	0.8525	0.4762
72	15	73	0.3314	0.1851
73	17	74	0.405	0.2262
74	18	75	0.4367	0.2439
75	18	76	0.3416	0.1908
85	21	86	0.7675	0.4286
86	23	87	1.5914	0.5709
87	24	88	0.702	0.3921
88	24	89	2.0743	0.7441
89	89	90	0.678	0.2432
90	90	91	0.5738	0.3205
91	26	92	0.5913	0.3303
92	27	93	1.1865	0.3124

Tabela 3: Características dos troços da rede portuguesa.

Nó	Potência activa (MW)	Potência reactiva (MVAr)	Nó	Potência activa (MW)	Potência reactiva (MVAr)
0	0	0	47	0.0369	0.0179
1	0.0225	0.0109	48	0.0639	0.031
2	0.2403	0.1164	49	0.0684	0.0331
3	0.0243	0.0118	50	0.0279	0.0135
4	0	0	51	0.081	0.0392
5	0	0	52	0.0693	0.0336
6	0.0288	0.014	53	0.0621	0.0301
7	0	0	54	0.0351	0.017
8	0	0	55	0.2052	0.0994
9	0	0	56	0.0315	0.0153
10	0	0	57	0.5211	0.2524
11	0	0	58	0.2124	0.1029
12	0	0	59	0.0396	0.0192
13	0.0576	0.0279	60	0.045	0.0218
14	0	0	61	0.0171	0.0083
15	0	0	62	0.0216	0.0105
16	0.0189	0.0092	63	0.0351	0.017
17	0	0	64	0.0702	0.034
18	0	0	65	0.0342	0.0166
19	0.0558	0.027	66	0.0225	0.0109
20	0.0405	0.0196	67	0.0459	0.0222
21	0	0	68	0.0333	0.0161
22	0.054	0.0262	69	0.0369	0.0179
23	0	0	70	0.045	0.0218
24	0	0	71	0.0756	0.0366
25	0.0468	0.0227	72	0.0675	0.0327
26	0	0	73	0.0279	0.0135
27	0	0	74	0.0387	0.0187
28	0.0135	0.0065	75	0.0531	0.0257
29	0.0036	0.0017	76	0.0657	0.0318
30	0.018	0.0087	77	0.063	0.0305
31	0.0216	0.0105	78	0.0675	0.0327
32	0.009	0.0044	79	0.045	0.0218
33	0.0648	0.0314	80	0.009	0.0044
34	0.0657	0.0318	81	0.0162	0.0078
35	0.0594	0.0288	82	0.0675	0.0327
36	0.0135	0.0065	83	0.2961	0.1434
37	0.1611	0.078	84	0.072	0.0349
38	0.0261	0.0126	85	0.0765	0.0371
39	0.1341	0.065	86	0.0909	0.044
40	0.0855	0.0414	87	0.072	0.0349
41	0.0414	0.0201	88	0.063	0.0305
42	0.0414	0.0201	89	0.0216	0.0105
43	0.0414	0.0201	90	0.0369	0.0179
44	0.0216	0.0105	91	0.0207	0.01
45	0.0252	0.0122	92	0.0171	0.0083
46	0.0459	0.0222	93	0.09	0.0436

Tabela 4: Potências tomadas, activa e reactiva, nos nós da rede portuguesa.

Alimentador	Nó fonte	Nó a seguir ao fonte	Último nó
0	0	1	32
1	4	33	33
2	4	34	36
3	5	37	37
4	5	38	41
5	7	42	44
6	8	45	45
7	9	46	50
8	9	51	65
9	10	66	67
10	11	68	68
11	12	69	70
12	14	71	72
13	15	73	73
14	17	74	74
15	18	75	75
16	18	76	81
17	19	82	85
18	21	86	86
19	23	87	87
20	24	88	88
21	24	89	91
22	26	92	92
23	27	93	93

Tabela 5: Caracterização dos alimentadores da rede portuguesa.

Capacidade (MVAr)	Custo
0.030	168
0.060	245
0.075	289

Tabela 6: Caracterização dos condensadores a instalar na rede portuguesa.

Troço	Potência activa (MW)	Potência reactiva (MVAr)	Troço	Potência activa (MW)	Potência reactiva (MVAr)
0	0.015287	0.025564	47	0.000113	0.000063
1	0.010324	0.017239	48	0.000014	0.000008
2	0.023266	0.038891	49	0.000004	0.000002
3	0.027318	0.045660	50	0.006804	0.006714
4	0.028729	0.048016	51	0.002015	0.001989
5	0.012188	0.020365	52	0.006451	0.006367
6	0.026775	0.044753	53	0.001889	0.001865
7	0.016620	0.027784	54	0.001787	0.001763
8	0.042788	0.071515	55	0.001789	0.001766
9	0.038222	0.037717	56	0.000843	0.000832
10	0.004408	0.004350	57	0.000459	0.000256
11	0.004096	0.004044	58	0.000152	0.000085
12	0.001367	0.001351	59	0.000107	0.000060
13	0.007991	0.007886	60	0.000067	0.000037
14	0.004245	0.004189	61	0.000066	0.000037
15	0.003227	0.003184	62	0.000179	0.000064
16	0.011226	0.011079	63	0.000064	0.000023
17	0.003188	0.003147	64	0.000010	0.000004
18	0.003229	0.003187	65	0.000027	0.000007
19	0.003815	0.001005	66	0.000008	0.000002
20	0.001018	0.000268	67	0.000003	0.000002
21	0.001251	0.000329	68	0.000019	0.000010
22	0.001143	0.000301	69	0.000003	0.000002
23	0.000901	0.000237	70	0.000084	0.000047
24	0.000134	0.000035	71	0.000024	0.000013
25	0.000160	0.000042	72	0.000002	0.000001
26	0.000216	0.000057	73	0.000004	0.000002
27	0.000048	0.000027	74	0.000008	0.000004
28	0.000009	0.000005	75	0.000153	0.000086
29	0.000022	0.000008	76	0.000054	0.000030
30	0.000008	0.000004	77	0.000135	0.000048
31	0.000004	0.000002	78	0.000037	0.000020
32	0.000028	0.000010	79	0.000002	0.000001
33	0.000017	0.000009	80	0.000001	0.000001
34	0.000008	0.000004	81	0.001404	0.000504
35	0.000000	0.000000	82	0.000269	0.000150
36	0.000057	0.000032	83	0.000051	0.000029
37	0.000654	0.000172	84	0.000012	0.000007
38	0.000657	0.000173	85	0.000041	0.000023
39	0.000045	0.000025	86	0.000054	0.000019
40	0.000007	0.000004	87	0.000018	0.000010
41	0.000063	0.000017	88	0.000085	0.000031
42	0.000011	0.000006	89	0.000015	0.000005
43	0.000001	0.000000	90	0.000002	0.000001
44	0.000001	0.000001	91	0.000001	0.000001
45	0.000195	0.000109	92	0.000063	0.000017
46	0.000080	0.000045			

Tabela 7: Perdas activa e resistiva nos troços da rede portuguesa sem compensação.

Perdas totais	Valor
Activas	0.3204378
Reactivas	0.4458570

Tabela 8: Perdas totais na rede portuguesa sem compensação.

Nó	Tensão (p.u.)	Nó	Tensão (p.u.)	Nó	Tensão (p.u.)
0	1.000000	32	0.870410	64	0.911813
1	0.995628	33	0.977110	65	0.911549
2	0.992662	34	0.977384	66	0.913675
3	0.985693	35	0.977280	67	0.913522
4	0.977507	36	0.977262	68	0.911486
5	0.968552	37	0.968197	69	0.909050
6	0.964315	38	0.966548	70	0.908982
7	0.955034	39	0.964338	71	0.903091
8	0.949108	40	0.963989	72	0.902765
9	0.934028	41	0.963832	73	0.900787
10	0.914005	42	0.954509	74	0.891035
11	0.911568	43	0.954333	75	0.888779
12	0.909264	44	0.954294	76	0.888387
13	0.908459	45	0.949061	77	0.888143
14	0.903633	46	0.933259	78	0.887317
15	0.900840	47	0.932869	79	0.886842
16	0.898679	48	0.932193	80	0.886752
17	0.891125	49	0.932055	81	0.886698
18	0.888912	50	0.931923	82	0.883785
19	0.886081	51	0.929003	83	0.883235
20	0.881274	52	0.927424	84	0.882921
21	0.879897	53	0.922133	85	0.882779
22	0.877903	54	0.920502	86	0.879491
23	0.875860	55	0.918916	87	0.875241
24	0.873941	56	0.917012	88	0.873682
25	0.873481	57	0.916086	89	0.873052
26	0.872789	58	0.915179	90	0.872840
27	0.871762	59	0.914637	91	0.872770
28	0.871205	60	0.914187	92	0.872729
29	0.871072	61	0.913837	93	0.871206
30	0.870766	62	0.913455		
31	0.870596	63	0.912343		

Tabela 9: Tensão nos nós da rede portuguesa sem compensação.

4.2. Rede indiana

Nesta secção apresentam-se os dados referentes às características das entidades associadas à rede (troços e nós) indiana, cuja tensão da subestação (nó 0) é de 11.00 MVAr. De referir ainda que é possível instalar condensadores em todos os nós da rede, com excepção no nó 0 (subestação).

Troço	Origem	Destino	Resistência (Ohm)	Reactância (Ohm)	Troço	Origem	Destino	Resistência (Ohm)	Reactância (Ohm)
0	0	1	1.197	0.82	14	4	15	2.55	1.058
1	1	2	1.796	1.231	15	5	16	1.366	0.567
2	2	3	1.306	0.895	16	16	17	0.819	0.34
3	3	4	1.851	1.268	17	17	18	1.548	0.642
4	4	5	1.524	1.044	18	18	19	1.366	0.567
5	5	6	1.905	1.305	19	19	20	3.552	1.474
6	6	7	1.197	0.82	20	6	21	1.548	0.642
7	7	8	0.653	0.447	21	21	22	1.092	0.453
8	8	9	1.143	0.783	22	22	23	0.91	0.378
9	3	10	2.823	1.172	23	23	24	0.455	0.189
10	10	11	1.184	0.491	24	24	25	0.364	0.151
11	11	12	1.002	0.416	25	7	26	0.546	0.226
12	12	13	0.455	0.189	26	26	27	0.273	0.113
13	13	14	0.546	0.227					

Tabela 10: Características dos troços da rede indiana.

Nó	Potência activa (MW)	Potência reactiva (MVAr)	Nó	Potência activa (MW)	Potência reactiva (MVAr)
0	0	0	14	0.03528	0.03528
1	0.03528	0.03528	15	0.03528	0.03528
2	0.014	0.014	16	0.00896	0.00896
3	0.03528	0.03528	17	0.00896	0.00896
4	0.014	0.014	18	0.03528	0.03528
5	0.03528	0.03528	19	0.03528	0.03528
6	0.03528	0.03528	20	0.014	0.014
7	0.03528	0.03528	21	0.03528	0.03528
8	0.014	0.014	22	0.00896	0.00896
9	0.014	0.014	23	0.056	0.056
10	0.056	0.056	24	0.00896	0.00896
11	0.03528	0.03528	25	0.03528	0.03528
12	0.03528	0.03528	26	0.03528	0.03528
13	0.014	0.014	27	0.03528	0.03528

Tabela 11: Potências tomadas, activa e resistiva, nos nós da rede indiana.

Alimentador	Nó fonte	Nó a seguir ao fonte	Último nó
0	0	1	9
1	3	10	14
2	4	15	15
3	5	16	20
4	6	21	25
5	7	26	27

Tabela 12: Caracterização dos alimentadores da rede indiana.

Capacidade (MVAr)	Custo
0.025	77
0.030	168
0.060	245

Tabela 13: Caracterização dos condensadores a instalar na rede indiana.

Troço	Potência activa (MW)	Potência reactiva (MVAr)	Troço	Potência activa (MW)	Potência reactiva (MVAr)
0	0.013212	0.009051	14	0.000060	0.000025
1	0.018113	0.012415	15	0.000277	0.000115
2	0.012682	0.008691	16	0.000138	0.000057
3	0.009075	0.006216	17	0.000214	0.000089
4	0.006100	0.004179	18	0.000064	0.000027
5	0.003716	0.002546	19	0.000013	0.000006
6	0.000433	0.000296	20	0.000637	0.000264
7	0.000011	0.000008	21	0.000257	0.000107
8	0.000006	0.000004	22	0.000181	0.000075
9	0.001613	0.000670	23	0.000018	0.000007
10	0.000315	0.000131	24	0.000009	0.000004
11	0.000133	0.000055	25	0.000053	0.000022
12	0.000020	0.000009	26	0.000007	0.000003
13	0.000013	0.000005			

Tabela 14: Perdas activa e resistiva nos troços da rede indiana sem compensação.

Perdas totais	Valor
Activas	0.067370
Reactivas	0.045075

Tabela 15: Perdas totais na rede indiana sem compensação.

Nó	Tensão (p.u.)	Nó	Tensão (p.u.)	Nó	Tensão (p.u.)
0	1.000000	10	0.947499	20	0.923551
1	0.986537	11	0.945747	21	0.917542
2	0.967353	12	0.944699	22	0.916021
3	0.953598	13	0.944421	23	0.914856
4	0.939753	14	0.944182	24	0.914598
5	0.929419	15	0.938634	25	0.914434
6	0.920390	16	0.927652	26	0.917439
7	0.917930	17	0.926684	27	0.917316
8	0.917632	18	0.925031		
9	0.917355	19	0.924180		

Tabela 16: Tensão nos nós da rede (indiana) sem compensação.