

Andreia Amador Timóteo

Estimativa de Parâmetros Através do Método de Otimização MCSFilter

Dissertação de Mestrado Integrado em Engenharia Química orientada por
Professor Doutor Lino de Oliveira Santos e por
Professora Doutora Florbela Alexandra Pires Fernandes, apresentada ao Departamento de
Engenharia Química da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Departamento de Engenharia Química, Faculdade de Ciências e Tecnologia da
Universidade de Coimbra

Coimbra
2016



Aos meus avôs: Artur e João,
Aos meus pais.

“A good archer is known not by his arrows but by his aim.”

Thomas Fuller

Agradecimentos

A realização desta dissertação não teria sido possível sem a ajuda de um grande número de pessoas, a quem estarei eternamente agradecida.

Em primeiro lugar, aos meus orientadores, Professor Doutor Lino Santos e Professora Doutora Florbela Fernandes por todo o conhecimento transmitido e por mostrarem tamanha paciência e total disponibilidade para o esclarecimento de dúvidas infundáveis. Este trabalho também vos pertence.

Ao Doutor Andrey Romanenko e ao Engenheiro João Leal por todo o conhecimento transmitido acerca da linguagem de programação Java e pelas horas disponibilizadas para o mesmo fim.

Aos meus amigos com quem partilho a minha vida desde sempre (Diana, André, João Miguel, João André, Melina, Mariana, Joana, Ana Paula), quero agradecer-vos por todas as horas de conversa, por todas as palavras de carinho e também pela constante aprendizagem que transmitimos uns aos outros. Que esta amizade se mantenha sempre tão saudável como começou e que nos ajude a manter o espírito jovem!

Aos meus amigos com quem tive o prazer de iniciar e partilhar esta jornada que agora finda (Diana, Marisa, Cris, Seça, Célia, Máxi, Vareda, Adriana, Sara, Inês, Nuno, Diogo). Agradeço-vos profundamente pelos risos, pelos choros, por todos momentos partilhados. Desejo que esta amizade perdure por muitos e longos anos. Estendo também os meus agradecimentos a todas as pessoas com quem tive o prazer de me cruzar e travar amizade ao longo deste percurso e que me proporcionaram momentos que levo comigo para a vida. Sem vos ter a meu lado, esta vitória que agora alcanço não teria o mesmo sabor.

A toda a minha família, em especial aos meus pais por todos os sacrifícios feitos para que eu pudesse ter todas as condições para poder construir as bases para o meu futuro. Pelo espírito de luta e de sacrifício, contra as adversidades da vida, que sempre me demonstraram, são uma grande referência de sucesso na minha vida. Espero conseguir sempre deixar-vos orgulhosos. Um especial agradecimento à minha tia João, ao meu tio Vitor e aos meus primos, Ricardo e Sofia, por me terem recebido e por me terem feito sentir sempre como se estivesse na minha própria casa.

A todos, e também àqueles cujo nome não foi mencionado mas que direta ou indiretamente contribuíram para o meu sucesso pessoal e académico, o meu mais sincero Obrigada!

Resumo

O principal objetivo deste trabalho é a aplicação de um método de iniciação múltipla (*multistart*) sem derivadas para otimização de problemas de engenharia, desenvolvido por Fernandes (2014).

Numa fase inicial do trabalho, foram implementadas duas versões mais simples do método na linguagem de programação Java. Na primeira versão, MB, a procura local é efetuada para todos os pontos gerados. A segunda versão do método, MRA, permite definir as regiões de atração associadas a um minimizante que já tenha sido encontrado. Esta característica permite evitar a execução de uma procura local quando o ponto gerado aleatoriamente já se encontra dentro de uma região de atração definida. Por fim, o método MCSFilter é extensão do método MRA, sendo acoplado com um cálculo probabilístico.

De forma a testar as três implementações em Java, MB, MRA e MCSFilter, foram realizados testes com 10 problemas de otimização cujas soluções são bem conhecidas e estão publicadas na literatura. Na avaliação do desempenho das três variantes do método, considerou-se o tempo médio de execução, o número médio de procuras locais efetuadas, o número médio de avaliações da função, o valor médio do mínimo global encontrado e o número médio de minimizantes encontrados para cada problema. Concluiu-se que o método MCSFilter apresenta um comportamento melhor.

Numa segunda fase do trabalho, a aplicação do método MCSFilter é demonstrada na resolução de dois exemplos de problemas de otimização em engenharia: 1) a estimativa de parâmetros de um modelo de natureza descontínua que caracteriza o comportamento de uma válvula de controlo que exhibe o fenómeno de *stiction*; e 2) a estimativa de parâmetros do modelo cinético da isomerização do α -pineno, definido por um sistema de equações diferenciais ordinárias.

Concluiu-se que o método MCSFilter converge sempre para o ótimo global, demonstrando ser um método de otimização robusto. Para problemas de otimização com um maior número de variáveis de decisão e modelos matemáticos mais complexos, o tempo de execução não é competitivo com o de outros métodos, tal como o SSm ou FA. Permite obter melhores resultados para modelos descontínuos ou que apresentem dificuldades em obter a informação da derivada quando comparado com um método SQP acoplado de funções de suavização do modelo reportado num trabalho anterior.

Abstract

The main objective of this work is the application of a derivative-free multistart method to solve optimization problems in the engineering field. The applied method is the so called MCSFilter, developed by Fernandes (2014).

First of all, two simpler variants of the method MCSFilter were implemented in the Java programming language. The first variant, MB, performs the local search procedure to all points generated by the multistart method. In the second variant, MRA, attraction regions associated to a minimizer are calculated. If a point is generated inside an attraction region already defined, the local search procedure is not executed. The final variant, MCSFilter, couples the MRA variant with a probabilistic calculation to determine whether the local search procedure is executed or not.

In order to test the implementation in Java of the three variants of the method, tests were performed on 10 problems well studied and well documented in the literature. The results comprise average runtime, average number of local searches, average number of function evaluations, average value of the global minimum, and the average number of distinct minimizers found for each problem.

The second phase of the present work was the application of the MCSFilter method to two problems in the chemical engineering field: 1) parameter estimation of a discontinuous model that describes the behavior of a sticky valve after a set of four distinct patterns; and 2) parameter estimation of the kinetic model associated to the α -pinene isomeration, defined by a set of ODE's.

It was concluded that the MCSFilter optimization method can always achieve the optimum value (global minimum), showing that it is a robust method. When applied to mathematical models with a higher number of decision variables, the runtime is not competitive with other published methods, such as SSm and FA methods, even though the optimum value is always achieved. It is more suitable for mathematical models that have a lower number of decision variables, in order to maintain the runtime within an acceptable range of values. It allows to achieve better results when applied to simpler discontinuous models when compared to the optimal solution obtained using an SQP method and function smoothing approach reported in a previous work.

Conteúdo

Agradecimentos	i
Resumo	iii
Abstract	v
Lista de Figuras	ix
Lista de Tabelas	xi
Nomenclatura	xvi
1 Introdução	1
1.1 Sinopse	1
1.2 Motivação e Objetivos	2
1.3 Conceitos Fundamentais de Otimização	3
1.4 Exemplos de aplicação	5
1.4.1 Problemas Teste	5
1.4.2 Determinação dos parâmetros do modelo de sistemas em Engenharia Química	6
2 Método MCSFilter	11
2.1 Método <i>Multistart</i>	11
2.2 Problemas Teste e Discussão de Resultados	17
2.2.1 Problemas com restrições de limite simples	17
2.2.2 Problemas com restrições de igualdade e de desigualdade	23
3 <i>Stiction</i> em Válvulas de Controle	29
3.1 Modelo	30
3.1.1 Problema de Otimização	32
3.2 Resultados e Discussão	33

4	Isomerização do α-pineno	37
4.1	Modelo Cinético	38
4.1.1	Problema de Otimização	38
4.2	Resultados e Discussão	39
5	Conclusões e Trabalho Futuro	45
5.1	Conclusão	45
5.2	Trabalho Futuro	46
	Bibliografia	49
	Anexos	I
A	Problemas Estudados	III
A.1	Problemas com Restrições de Limite Simples	III
A.2	Problemas com Restrições de Igualdade e de Desigualdade	VI
B	Resultados <i>Multistart</i>	VII
B.1	Problemas com Restrições de Limites Simples	VII
B.2	Problemas com Restrições de Igualdade e de Desigualdade	IX

Lista de Figuras

1.1	Ilustração do sistema <i>Hammerstein</i>	7
2.1	Ilustração do método MCSFilter	12
2.2	Resultados do problema A1	18
2.3	Resultados do problema A2	19
2.4	Resultados do problema A3	20
2.5	Resultados do problema A4	21
2.6	Resultados do problema A5	21
2.7	Resultados do problema A6	22
2.8	Resultados do problema A7	23
2.9	Resultados do problema A8	24
2.10	Resultados do problema A9	25
2.11	Resultados do problema A10	26
3.1	Modelo de Chen utilizado para descrever o comportamento de uma válvula de controlo com <i>stiction</i>	30
3.2	Perfis obtidos com o modelo de Chen modificado	34
3.3	Ampliação dos perfis obtidos pelo modelo de Chen modificado.	35
4.1	Representação esquemática do sistema reacional da isomerização do α -pineno.	37
4.2	Perfis das concentrações previstos pelo modelo e dados experimentais.	42

Lista de Tabelas

3.1	Resultados do método MCSFilter ao problema da determinação dos parâmetros do modelo que descreve uma válvula de controlo com <i>stiction</i>	33
4.1	Resultados obtidos com o método MCSFilter ao problema de estimativa de parâmetros do modelo cinético da isomerização do α -pineno.	40
4.2	Resultados obtidos por Egea et al. (2007, 2008), Larrosa (2008) e Rocha et al. (2016) na resolução do problema de otimização de estimativa de parâmetros da isomerização do α -pineno, utilizando os vários métodos de otimização.	41
B.1	Valores reportados na literatura e usados como referência para a discussão dos resultados dos problemas A1 a A7.	VII
B.2	Resultados de MB para os problemas A1 - A7.	VIII
B.3	Resultados de MRA para os problemas A1 - A7.	VIII
B.4	Resultados de MCSFilter-Java para os problemas A1 - A7.	VIII
B.5	Resultados de MCSFilter-Matlab para os problemas A1 - A7.	IX
B.6	Valores da literatura usados como referência para a discussão dos resultados dos problemas A8 a A10.	IX
B.7	Resultados de MB para os problemas A8 - A10.	X
B.8	Resultados de MRA para os problemas A8 - A10.	X
B.9	Resultados de MCSFilter-Java para os problemas A8 - A10.	X
B.10	Resultados de MCSFilter-Matlab para os problemas A8 - A10.	X

Nomenclatura

Letras Romanas

AF	Número de Avaliações da Função
A_i	Região de Atração de um Minimizante
a	Matriz
\mathcal{B}	Vizinhaça
c	Matriz
$c(k)$	Força Acumulada
$d(k)$	Direção de Deslocamento da Haste da Válvula
d_i	Direção do Passo da Procura Local
E_{rel}	Erro Relativo
$f(x^*)$	Mínimo Global
f_D	Atrito Dinâmico
f_J	Atrito Correspondente ao <i>Slip-Jump</i>
f_S	Atrito Estático
J	Função Objetivo
k	Instante de tempo
l	Limite Inferior
m	Número de Restrições
N	Número de Minimizantes Distintos
N_{LS}	Número de Procuras Locais
n	Número de Variáveis de Decisão
p	Vetor de Parâmetros a Estimar
p_1	Velocidade da Reação 1
p_2	Velocidade da Reação 2

p_3	Velocidade da Reação 3
p_4	Velocidade da Reação 4
p_5	Velocidade da Reação 5
Q	Matriz Ponderal
$s(k)$	Posição da Válvula
t	Tempo
$\mathbf{u}(k)$	Sinal de Entrada Dado à Válvula
u	Limite Superior
x	Vetor de Variáveis de Decisão
$x(k)$	Ordem Dada Considerando a Quantidade de <i>Stiction</i>
x^*	Minimizante Global
$x_{central}$	Ponto inicial da Procura Local
x_i^*	Minimizante Associado a um Procedimento de Procura Local Genérico
x_{novo}	Ponto Obtido Através da Procura Pelas Direções Coordenadas
\tilde{y}_{qk}	Valor do Perfil Experimental
y_1'	Derivada em Ordem ao Tempo da Reação 1
y_2'	Derivada em Ordem ao Tempo da Reação 2
y_3'	Derivada em Ordem ao Tempo da Reação 3
y_4'	Derivada em Ordem ao Tempo da Reação 4
y_5'	Derivada em Ordem ao Tempo da Reação 5
y_1	α -pineno
y_2	Dipenteno
y_3	Allo Ocimeno
y_4	α e β -pironeno
y_5	Dímero
$y_q(p, t_k)$	Valor Previsto pelo Modelo para as Concentrações da Reação do α -pineno
z	Vetor de Valores Previstos pelo Modelo de Chen
\mathbf{z}_{exp}	Vetor de Valores Experimentais de um Válvula com <i>Stiction</i>
$z(k)$	Vetor Composto Pelos Sinais de Saída da Válvula
L	Definição Genérica de Procura Local

P(N) Condição de Paragem de MCSFilter

Letras Gregas

α Comprimento do Passo da Procura Local
 ε Raio da Vizinhança
 γ Contador de Velocidades da Reação de α -pineno
 ι Dimensão do Horizonte de Tempo
 Ω Conjunto de todos os pontos admissíveis
 σ Desvio Padrão
 θ Função Violação das Restrições

Acrónimos

CSFilter Procura Local baseada nas Direções Coordenadas e Filtros
EDO Equações Diferenciais Ordinárias
MB *Multistart* Básico
MCSFilter *Multistart* com Procura Local baseada nas Direções Coordenadas e Filtros
MRA *Multistart* com Regiões de Atração
OP Variável de Saída do Controlador
PL Programação Linear
PNL Programação Não Linear
PNLIM Programação Não Linear Inteira Mista
PV Variável de Saída do Processo

Abreviaturas

min Minimizar
prob Probabilidade
s. a. Sujeito a

Índices Inferiores

B *Multistart* Básico (Capítulo 2)
C *Multistart* Complexo (Capítulo 2)
eq Restrições de Igualdade
exp Valor Experimental
glob Mínimo Global

i	Contador de Restrições de Igualdade
$ineq$	Restrições de Desigualdade
I	<i>Multistart</i> Intermédio (Capítulo 2)
j	Contador de Restrições de Desigualdade
K	Contador de Variáveis de Decisão
L	Limite Inferior
med	Valor Médio
min	Valor Mínimo
O	<i>Multistart</i> Original (Capítulo 2)
q	Contador de Compostos na Reação de Isomerização do α -pineno
U	Limite Superior

1. Introdução

1.1 Sinopse

O presente trabalho tem como principal objetivo o estudo da aplicação de um método de otimização estocástico sem uso de derivadas em problemas da natureza de estimativa de parâmetros na área da engenharia de processos.

No **Capítulo 1** é apresentada uma breve introdução que compreende a motivação e os objetivos pretendidos com a realização do trabalho, bem como uma definição de conceitos fundamentais relacionados com a temática de otimização global. São, também, apresentados alguns modelos matemáticos da literatura para o problema de *stiction* e alguns métodos de otimização aplicados ao problema de isomerização do α -pineno publicados na literatura.

No **Capítulo 2** é apresentado o método de otimização global utilizado com a descrição dos três algoritmos estudados — MB, MRA e MCSFilter. São apresentados os resultados dos testes realizados com problemas da literatura cujas soluções são bem conhecidas. Tratam-se de problemas de programação não linear (PNL) com restrições de limite simples e/ou restrições de igualdade e de desigualdade. Finalmente, são apresentadas as principais conclusões relativamente ao desempenho dos três algoritmos estudados.

No **Capítulo 3** o método MCSFilter é aplicado ao problema da determinação dos parâmetros de *stiction* numa válvula de controlo. O objetivo é estimar os parâmetros do modelo, que definem a severidade do fenómeno de *stiction*, com base em dados experimentais. No final do capítulo é apresentada a discussão de resultados e a sua comparação com valores da literatura.

Segue-se o **Capítulo 4** em que é demonstrada a aplicação do método MCSFilter ao problema de estimativa de parâmetros do modelo cinético da isomerização do α -pineno. Trata-se de um sistema reacional complexo e o objetivo é a determinação das constantes de velocidade das reações. Os resultados são apresentados e comparados com os valores da literatura.

Finalmente, no **Capítulo 5**, são apresentadas as conclusões finais e propostas de trabalho futuro.

1.2 Motivação e Objetivos

A otimização é uma ferramenta que permite obter a melhor e mais rentável solução possível para um problema (Edgar et al., 1989). As decisões a tomar com base nos resultados da otimização podem estar associadas a lucros ou prejuízos, estimativa de parâmetros físicos, controlo de processos ou à quantificação de riscos associados a investimentos (Kolda et al., 2003; Luenberger e Ye, 2008). Sendo extremamente raros os casos em que é possível dispôr de uma representação matemática fidedigna do sistema real, é necessário interpretar os resultados de um processo de otimização como sendo a melhor aproximação possível e não o valor ótimo (Luenberger e Ye, 2008).

Nos casos em que é possível representar matematicamente o sistema real, os modelos podem apresentar não linearidades ou descontinuidades (Egea et al., 2007). Esta última característica dificulta a aplicação de métodos de otimização que requerem o cálculo das derivadas da função objetivo e das restrições do problema de otimização não linear. Os métodos de otimização global em que não é necessário calcular derivadas constituem uma ferramenta apelativa para a resolução de problemas de otimização (Kolda et al., 2003).

Dos métodos de otimização global, destacam-se os métodos de procura direta, no qual se insere o método de procura local pelas direções coordenadas (Kolda et al., 2003). Os métodos de procura direta são de fácil implementação e apresentam uma convergência rápida em direção ao ótimo na resolução de problemas simples. No entanto, a sua convergência torna-se mais lenta quando o problema é mais complexo — maior número de variáveis de decisão ou quando o modelo é definido por um sistema de equações diferenciais (Kolda et al., 2003). Neste trabalho, pretende-se, então, aplicar um método global que usa a procura pelas direções coordenadas desenvolvido por Fernandes (2014) — MCSFilter — a problemas de natureza de estimativa de parâmetros do modelo no contexto de ciclos de controlo e de um sistema reacional complexo. Este método consiste num método estocástico baseado no algoritmo de *multistart* com uma procura local baseada nas direções coordenadas e filtros. Foi originalmente implementado em Matlab[®] e, recentemente, na linguagem de programação Java.

Os objetivos do presente trabalho compreendem, numa fase inicial, a implementação em Java de duas variantes mais simples do método MCSFilter — MB e MRA. A primeira variante, MB (*Multistart* Básico), é uma versão básica que consiste em executar o método de procura pelas direções coordenadas para um determinado número de pontos. A segunda variante, MRA (*Multistart* com Regiões de Atração), é uma extensão da versão MB com regiões de atração. As três variantes do método (MB, MRA e MCSFilter) foram testadas com 10 problemas de otimização cujas soluções são conhecidas na literatura. Estes testes têm como objetivo verificar a reprodutibilidade dos resultados relativamente à implementação original em Matlab[®] e também identificar as vantagens/desvantagens de cada versão do método. O objetivo final é poder concluir sobre qual a variante do método

que mais se adequa à resolução dos problemas de engenharia analisados neste trabalho.

A variante do método com melhores resultados é aplicada aos problemas de otimização relacionados com a estimativa de parâmetros dos modelos matemáticos que descrevem os problemas físicos. O objetivo é também comparar a solução obtida com os resultados reportados na literatura e concluir sobre o desempenho e aplicabilidade do método.

1.3 Conceitos Fundamentais de Otimização

A otimização de uma função genérica é feita maximizando ou minimizando-a, tendo em conta as restrições aplicadas às variáveis de decisão (Nocedal e Wright, 1999). Os conceitos de otimização que se seguem são relativos a problemas de minimização já que é possível escrever qualquer problema na forma genérica, como é apresentado na Definição 1.

Definição 1. (Nocedal e Wright, 1999; Luenberger e Ye, 2008) *Seja um problema de minimização genérico representado por:*

$$\begin{aligned} \min \quad & f(x) \\ \text{s. a.} \quad & h_i(x) = 0, \quad i = 1, 2, \dots, m_{eq} \\ & g_j(x) \leq 0, \quad j = 1, 2, \dots, m_{ineq} \\ & \text{para } x \in \mathbb{R}^n : \\ & l_K \leq x_K \leq u_K, \quad K = 1, 2, \dots, n \end{aligned}$$

em que f é a função objetivo, h é o vetor das restrições de igualdade, g é o vetor das restrições de desigualdade, x é o vetor das variáveis de decisão, m_{eq} é o número de restrições de igualdade, m_{ineq} é o número de restrições de desigualdade e n é o número de variáveis de decisão.

Um outro conceito fundamental em otimização é a definição de região de admissibilidade, apresentada na Definição 2

Definição 2. (Nocedal e Wright, 1999) *Um ponto que satisfaz as restrições do problema de otimização diz-se ponto admissível. O conjunto de todos os pontos admissíveis, Ω , é designado por região admissível ou domínio admissível e escreve-se na forma de:*

$$\Omega = \{x \in \mathbb{R}^n : h_i(x) = 0, \quad i = 1, \dots, m_{eq}, \quad g_j(x) \leq 0, \quad j = 1, \dots, m_{ineq}\}$$

Se $\Omega = \emptyset$ diz-se que o problema não tem solução.

Definida a região de admissibilidade, é necessário definir matematicamente os conceitos de mínimo e minimizante. Este conceitos são apresentados nas Definições 3, 4 e 5.

Definição 3. (Nocedal e Wright, 1999) *O ponto $x^* \in \Omega$ é um minimizante global de f se a função f atinge o seu valor mais baixo, $f(x^*)$, em Ω :*

$$f(x^*) \leq f(x), \text{ para todo } x \in \Omega.$$

Segundo a Definição 3, tem-se x^* como sendo o minimizante global de f , o que significa que $f(x^*)$ é definido como o mínimo global da função, ou seja, o valor mais baixo que f atinge, quando respeitadas as restrições h_i e g_j .

Outros conceitos que importa definir são minimizante local e mínimo local (Definição 5), que dependem do conceito de vizinhança (Definição 4).

Definição 4. (Nocedal e Wright, 1999) *Designa-se por vizinhança \mathcal{B} de \tilde{x} , de raio $\varepsilon > 0$ o conjunto de pontos definido por*

$$\mathcal{B}(\tilde{x}, \varepsilon) = \{x \in \mathbb{R}^n : \|x - \tilde{x}\| \leq \varepsilon\}.$$

Definição 5. (Nocedal e Wright, 1999) *Define-se minimizante local de f como x^* se existir uma vizinhança $\mathcal{B}(x^*, \varepsilon)$, tal que f atinge o seu valor mais baixo em $\Omega \cap \mathcal{B}(x^*, \varepsilon)$, isto é,*

$$f(x^*) \leq f(x) \text{ para todo } x \in \Omega \cap \mathcal{B}(x^*, \varepsilon),$$

sendo $f^* = f(x^*)$ o mínimo local.

Segundo a Definição 4, tem-se que para uma determinada vizinhança, um minimizante local, x^* , corresponde ao menor valor da função, $f(x^*)$, que se designa de mínimo local.

Os métodos de otimização podem ser classificados de várias formas. Podem, por exemplo, ser agrupados em métodos de otimização local — caso se pretenda determinar apenas um mínimo local — ou podem ser agrupados em otimização global/multimodal — caso se pretenda determinar o mínimo global (Belegundu e Chandrupatla, 2011; Edgar et al., 1989). Também podem ser classificados consoante a linearidade das funções associadas ao problema, nomeadamente, programação linear (PL) ou programação não linear (PNL). A natureza das funções do problema condicionam a escolha do método de otimização a ser utilizado (Fernandes, 2014). Por exemplo, se uma das funções f , g ou h é não linear, então trata-se de um problema de PNL. No caso de todas as funções serem lineares, trata-se de um problema de PL. Os problemas de programação linear são mais simples de resolver, enquanto que os problemas de programação não linear são muito mais complexos. Esta complexidade advém da não linearidade das funções envolvidas, que podem apresentar vários minimizantes locais e globais.

Outra classificação usual para os métodos de otimização passa por considerar métodos determinísticos e métodos estocásticos (Fernandes, 2014).

Os métodos estocásticos são uma opção a ter em conta como demonstrado pela resolução de diversos problemas de otimização em Edgar et al. (1989) e Belegundu e Chandrupatla (2011). Exemplos bem conhecidos deste tipo de métodos são o *multistart* e o algoritmo genético (Edgar et al., 1989).

O método MCSFilter, usado para resolver problemas PNL, foi usado também para resolver problemas não lineares inteiros mistos (PNLIM) como foi demonstrado em (Fernandes et al., 2014). A versão de *multistart* usada na presente dissertação foi ainda reajustada com o método de procura local *Hooke and Jeeves* acoplado com a metodologia dos filtros por forma a localizar o melhor valor da função objetivo, em problemas PNLIM (Fernandes et al., 2013a; Costa et al., 2014; Fernandes et al., 2016). Foi também utilizada uma procura local *Branch and Bound* para tratar as variáveis inteiras.

1.4 Exemplos de aplicação

Nesta secção é apresentado um problema-teste, a título de exemplo, ao qual foi aplicado as três variantes do método *multistart* (MB, MRA e MCSFilter). São brevemente descritos os restantes problemas-teste, bem como o que é estudado em cada um. Na segunda parte, são expostos alguns modelos matemáticos e métodos de otimização reportados na literatura que foram utilizados na resolução do problema da deteção e quantificação de *stiction* presente nas válvulas de controlo. São também apresentados alguns métodos de otimização reportados na literatura utilizados para a estimativa de parâmetros do modelo cinético que descreve a isomerização do α -pineno.

1.4.1 Problemas Teste

Os problemas em que vão ser aplicadas as três variantes do método *multistart* são problemas cujas soluções se encontram documentadas na literatura. São problemas-tipo, comumente usados para testar novos algoritmos de otimização. A formulação dos problemas-tipo pode ter restrições de limites simples ou restrições de igualdade e/ou desigualdade. A título de exemplo, apresenta-se em seguida, o problema 1.1 que é o primeiro problema que é utilizado neste trabalho para testar as três variantes do método de otimização.

$$\begin{aligned} \min f(x) &\equiv \left(x_1 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6\right)^2 + 10 \left(1 - \frac{1}{8\pi}\right) \cos(x_1) + 10 \\ \text{s.a.} \quad &-5 \leq x_1 \leq 10 \\ &0 \leq x_2 \leq 15 \end{aligned} \tag{1.1}$$

Este é um problema com restrições de limites simples, ou seja, limites superior e inferior nas variáveis de decisão. A região de admissibilidade assim definida contém três minimizantes globais, uma vez que obtém para todos eles o mesmo valor de mínimo.

São testados, além do problema acima exposto, mais 6 problemas com restrições de limite simples. Alguns destes apresentam a mesma função genérica entre si, diferindo no valor das suas constantes ou no número de variáveis de decisão. O método é, também, testado em 3 problemas com restrições de igualdade e/ou desigualdade, havendo 2 com uma restrição de igualdade e 1 com uma restrição de desigualdade. Verifica-se que os problemas com restrições de igualdade são mais complexos do que os restantes devido à dificuldade que existe em ser cumprida a restrição aquando da procura local.

Foram estudados o número médio de minimizantes distintos encontrados, N_{med} , o número médio de avaliações da função, AF_{med} , o tempo médio de execução, t_{med} , e o número médio de procuras locais efetuadas, $N_{LS,med}$, para 10 execuções. Os resultados e principais conclusões são apresentados no Capítulo 2.

1.4.2 Determinação dos parâmetros do modelo de sistemas em Engenharia Química

Neste trabalho, a aplicação do MCSFilter é também demonstrada na determinação dos parâmetros do modelo de dois sistemas típicos em Engenharia Química. O primeiro problema é o da determinação dos parâmetros do modelo de uma válvula de controlo com *stiction*. O segundo problema diz respeito à determinação dos parâmetros cinéticos da isomerização do α -pineno.

Fenómeno de *Stiction* em válvulas de controlo

Num processo industrial, existem centenas de válvulas de controlo necessárias ao bom funcionamento da fábrica e a sua manutenção requer enormes recursos quer a nível técnico quer a nível humano. O fenómeno de *stiction* presente nas válvulas de controlo de um processo industrial pode afetar o processo ao nível de consumo energético, da qualidade final dos produtos e/ou perda de lucros (Choudhury et al., 2004, 2008b; Brásio et al., 2014b; Chen et al., 2008). Por conseguinte, é imperativo que a sua deteção e quantificação seja rápida e eficaz.

A deteção do fenómeno de *stiction* numa válvula de controlo pode ser feita através de um modelo que descreve o comportamento da válvula, análise dos dados do sinal de saída do controlador (Choudhury et al., 2004) ou através de um programa de reconhecimento de padrões típicos associados à presença de *stiction* (Srinivasan et al., 2005a).

Choudhury et al. (2004) propõem um método que recorre à análise dos perfis do sinal de saída do processo para detetar a presença e quantificar o fenómeno de *stiction* nas válvulas de controlo do processo. A presença do fenómeno de *stiction* é confirmada quando o perfil da variável de saída do processo em função do sinal de saída do controlador (PV vs OP) apresenta um forma elíptica (Choudhury et al., 2008a). Por vezes, esta deteção pode

ser difícil devido à presença de ruído estático ou às alterações que ocorrem na dinâmica do processo. Srinivasan et al. (2005a) propõem que a detecção de *stiction* é feita com base nos padrões das oscilações no sinal de saída do controlador e do processo uma vez que, na presença de *stiction*, as oscilações podem apresentar-se com padrões de forma triangular, quadrada ou dentada. A quantificação do fenômeno de *stiction* é estabelecida a partir da distância entre os picos. Quando não existe presença de *stiction*, o comportamento do sinal de saída do processo e do controlador assemelha-se ao de uma onda sinusoidal.

A detecção do fenômeno de *stiction* numa válvula de controlo pode também ser feita com base no modelo, segundo o sistema de Hammerstein, ilustrado na Figura 1.1. Este sistema é constituído por uma parte linear (parte dinâmica do processo juntamente com a válvula de controlo) e uma parte não linear (correspondente ao fenômeno de *stiction*).

Para além disto, são vários os modelos matemáticos adotados para descrever o fenômeno de *stiction*. Os mais comumente utilizados são modelos *data-driven*, ou seja, que apenas se baseiam na utilização dos valores do sinal de saída do controlador. Os modelos utilizados por Srinivasan et al. (2005b) e Srinivasan e Rengaswamy (2008) (que seguem o modelo desenvolvido por Choudhury et al. (2005)) contêm apenas um parâmetro enquanto que os modelos utilizados por Kano et al. (2004), Romano e Garcia (2011), Choudhury et al. (2008a), Jelali (2008) e Chen et al. (2008) são condicionados por dois parâmetros. Foi demonstrado que os modelos que apenas dependem de um parâmetro não são realistas uma vez que não têm em conta a dinâmica da haste, descrevendo apenas a posição da válvula quando esta já não está em movimento (Srinivasan et al., 2005b).

Em alguns estudos realizados sobre a detecção e quantificação do fenômeno de *stiction* em válvulas de controlo utilizando modelos matemáticos, há a aplicação de um método de otimização para estimativa dos parâmetros do modelo, sendo muito comum a minimização de uma função objetivo desenvolvida com base no critério dos mínimos quadrados (Srinivasan et al., 2005b; Choudhury et al., 2008a; Jelali, 2008; Romano e Garcia, 2011; Brásio et al., 2014b). Ding et al. (2011) apresenta um estudo acerca da aplicação de métodos

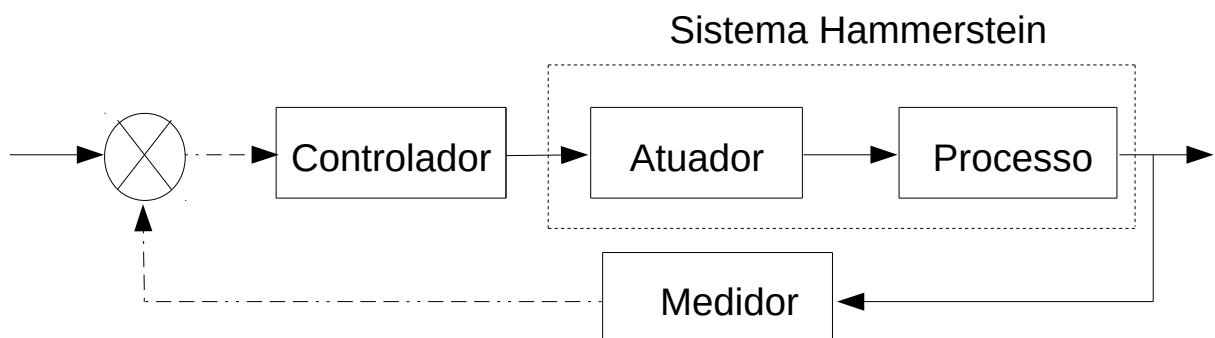


Figura 1.1: Ilustração do sistema de *Hammerstein* utilizado em modelos que descrevem o comportamento de válvulas em que está presente o fenômeno de *stiction*.

estocásticos e Newtonianos para estimativa de parâmetros num sistema de Hammerstein. Concluiu-se que os métodos estocásticos exigem mais tempo de computação do que os Newtonianos para chegar à solução.

Lee et al. (2008) e Farenzena e Trierweiler (2012) apresentam propostas para estimar os parâmetros de *stiction*, como já haviam sido feitas por outros autores. Lee et al. (2008) recorre a um modelo (Kano et al., 2004) baseado na medição do tempo de resposta da válvula associada à presença de *stiction* de forma a detetar a sua presença e quantificar a sua severidade. A otimização de parâmetros do modelo é feita através de uma técnica de otimização global em que é aplicado um algoritmo *multistart* para percorrer todo o espaço de procura. Farenzena e Trierweiler (2012) utiliza dois modelos *data-driven* de descrição do comportamento de *stiction* — um modelo com apenas um parâmetro descrito, por Choudhury et al. (2006), e outro proposto por Jelali (2008) com dois parâmetros. Concluiu-se que o método DIRECT é o melhor, dos métodos estudados, para a estimativa de parâmetros dos modelos.

Rossi e Scali (2005) fizeram um estudo comparativo de técnicas já publicadas para a detecção de *stiction*, utilizando diferentes métodos e modelos. Concluíram que nenhuma das técnicas pode ser considerada completamente válida, uma vez que a presença de *stiction* num ciclo de controlo altera bastante a dinâmica do processo e, consequentemente, o padrão dos resultados no sinal de saída do controlador. Um excelente trabalho de revisão, referente aos modelos e técnicas de otimização aplicadas ao problema de deteção e quantificação do fenómeno de *stiction*, pode ser consultado em Brásio et al. (2014a).

Modelo cinético da isomerização do α -pineno

O modelo cinético da isomerização do α -pineno foi desenvolvido por Box e Draper (1965), no âmbito de um estudo para estimar as 5 velocidades de reação, a partir dos valores experimentais registados em 1947 por Fugitt e Hawkins. Box et al. (1973) verificaram que poderia existir alguma dependência linear entre os dados experimentais das concentrações dos compostos envolvidos na isomerização do α -pineno, bem como entre os erros associados às leituras experimentais. Box e Draper (1965) concluíram que as concentrações de α e β -pironeno teriam sido determinadas, assumindo-se que corresponderiam a 3% da conversão total de α -pineno. Estas dependências lineares podem estar na origem da dificuldade que alguns métodos de otimização apresentam em encontrar o valor ótimo da função objetivo.

Ames (1962) definiu que existem 3 relações lineares entre os dados experimentais, sendo a primeira resultante de um balanço de massa, a segunda devido ao facto de a isomerização ser irreversível e a terceira que a soma de todas as percentagens de concentrações teria que ser igual a 100. Seguindo o seu trabalho Box et al. (1973) resolveram o problema de otimização aplicando um método para a minimização de uma função objetivo definida

segundo o critério dos mínimos quadrados de forma a estimar as velocidades de reação. Segundo Tjoa e Biegler (1991), é necessário ter-se o conhecimento das velocidades de reação correspondentes à isomerização do α -pineno de forma a poder-se determinar quais as condições de operação mais favoráveis à produção de determinado composto. Com a determinação destes valores é também mais fácil integrar parâmetros como o calor de reação e a conversão global de forma a maximizar a produção. No caso de Tjoa e Biegler (1991), a determinação dos parâmetros ótimos foi realizada através da minimização de uma função objetivo desenvolvida segundo o método dos mínimos quadrados. A avaliação da função foi feita através de integração numérica do sistema de EDO's. Os resultados de Tjoa e Biegler (1991) foram muito semelhantes aos obtidos por Box et al. (1973) e os valores iniciais dos parâmetros são bastante próximos do ótimo, o que facilita a convergência do método.

Para a resolução do problema de otimização na estimativa de parâmetros referentes à isomerização do α -pineno, Dolan et al. (2004) segue os métodos de Averick et al. (1992) e de Tjoa e Biegler (1991). Dolan et al. (2004) utiliza como integrador numérico um método de colocação com polinômios de grau 3, que será equivalente ao método de Runge-Kutta de 3^a ordem, de forma a calcular os valores das concentrações dos compostos associados à isomerização do α -pineno previstos pelo modelo. À semelhança do que é usado no presente trabalho, Dolan et al. (2004) também utilizou um método baseado em filtros sendo que a solução ótima obtida é semelhante às soluções obtidas por Box et al. (1973) e Tjoa e Biegler (1991).

O método de otimização utilizado por Egea et al. (2007) e por Egea et al. (2008), *SSm*, é o mesmo que foi utilizado por Larrosa (2008) e por Rodriguez-Fernandez et al. (2006). O método apresentado por Egea et al. (2007) é o método *SSm* — *Scatter Search method* — e trata-se de um método populacional com uma parte estocástica, de procura dispersa metaheurística, acoplado a uma componente que recorre à informação da derivada da função, ou seja, trata-se de um método de otimização híbrido.

Também implementado por Egea et al. (2007), o método desenvolvido por Csendes (1988), denominado por *Global*, é um método híbrido para otimização global que realiza uma procura local aleatória. Csendes (1988) baseou-se no pressuposto de que o problema é unimodal e implementou dois algoritmos do método *Global* em que a diferença é apenas no método utilizado para a procura local, sendo o primeiro um método de quasi-Newton e o segundo um método de procura direta com uma parte estocástica. Um método com características semelhantes ao *Global* foi desenvolvido por Storn e Price (1997), denominando-se por DE (*Direct Method*). É um método meta-heurístico de procura direta.

O método utilizado por Runarsson e Yao (2000) — SRES — é um método que utiliza funções de penalidade para transformar o problema que contém restrições num problema sem restrições. Nos métodos em que tradicionalmente se usam funções de penalidade, é muito comum que exista sobrepenalização ou subpenalização — que advêm de um

desequilíbrio entre a comparação das funções de penalidade e objetivo — o que faz com que a exclusão de pontos possa ser excessiva. Neste método, além das funções de penalidade, é introduzida uma parte estocástica que permite balancear esta comparação de forma a encontrar uma população viável para minimizar a função objetivo, recorrendo ao método SRES.

Jones (2001) desenvolveu um algoritmo para um método determinístico de otimização global denominado de DIRECT — *D*ividing *R*ECTangles — em que a procura local é realizada através da divisão do espaço de procura em retângulos mais pequenos e a avaliação da função é feita no centro de cada retângulo formado. Este método é indicado para problemas cujas funções ou restrições apresentem descontinuidades ou que sejam altamente não lineares.

O método *OptQuest*, desenvolvido por Laguna e Martí (2003) segundo Ugray et al. (2005) é um método meta-heurístico de procura dispersa. Permite separar o ciclo de avaliação do modelo, do cálculo da função objetivo — geralmente os sistemas que se apresentam com este formato, como a isomerização do α -pineno, são denominados como sistemas de *black box*. Resumidamente, o problema de otimização é resolvido à parte da componente de avaliação da função.

Todos estes métodos — *Global*, DE, SRES, DIRECT e *OptQuest* — foram implementados por Egea et al. (2007) e aplicados na determinação dos parâmetros do modelo cinético referente à isomerização do α -pineno.

Um dos estudos mais recentes referente à isomerização do α -pineno é o FA — *Firefly Algorithm* (Rocha et al., 2016) — que se trata de um método estocástico para problemas de otimização global. Tal como o nome indica, FA é uma analogia para o comportamento dos pirilampos, seguindo as regras do mundo animal de que todos os pirilampos são unissexo e que a atração é proporcional à luz, ou seja, os pirilampos são atraídos para os que têm a luz mais brilhante. Neste caso, indica que qualquer ponto pode ser atraído ou atrair e que quanto mais baixo for o valor da função em determinado ponto mais vai atrair outros pontos para si.

Os resultados obtidos na literatura para os métodos apresentados acima são utilizados como valores de referência para comparação com os resultados obtidos na aplicação do método MCSFilter.

2. Método MCSFilter

Neste capítulo apresenta-se o método de otimização global — MCSFilter. Trata-se de um método de iniciação múltipla com uma procura local baseada nas direções coordenadas e filtros desenvolvido por Fernandes (2014) — o método MCSFilter. Na Secção 2.1 é feita uma descrição do método e apresentado o algoritmo. A implementação computacional original do MCSFilter foi realizada na linguagem de programação Matlab[®], no entanto, neste trabalho foi utilizada uma nova versão, implementada em Java. Na Secção 2.2 é apresentado um estudo da nova versão em Java. Este estudo realizou-se com recurso a problemas com soluções bem conhecidas na literatura e tem como objetivo confirmar a correta implementação computacional do algoritmo em Java. São apresentados e discutidos os resultados dos problemas com restrições de limite simples e problemas com restrições de igualdade e/ou desigualdade. Finalmente, são apresentadas as principais conclusões acerca do desempenho do método.

2.1 Método *Multistart*

O método *multistart* tem duas fases: a fase global e a fase local. Na fase global são gerados aleatoriamente pontos em que é avaliada a função objetivo. Na segunda fase é realizada uma procura local a partir de cada um dos pontos gerados. Assim, o método *multistart* consiste em gerar aleatoriamente pontos do espaço de procura e, a partir destes, executar a procura local de modo a obter um minimizante local. Este processo é repetido até que as condições de paragem sejam satisfeitas (Kolda et al., 2003; Fernandes, 2014).

Nesta estratégia *multistart* é muito provável que um ou mais minimizantes sejam encontrados mais do que uma vez, o que significa um esforço computacional adicional, que se traduz num maior tempo de execução do método. Para evitar estas características indesejáveis, é introduzido o conceito de região de atração de modo a impedir que algumas procuras locais sejam realizadas. Na prática, o cálculo destas regiões de atração é bastante difícil, pelo que, como alternativa, pode calcular-se uma probabilidade *prob* que determina a probabilidade de o ponto gerado aleatoriamente pertencer ou não à região de atração mais próxima dele. Este cálculo está descrito no Algoritmo 3 (Fernandes et al., 2013b).

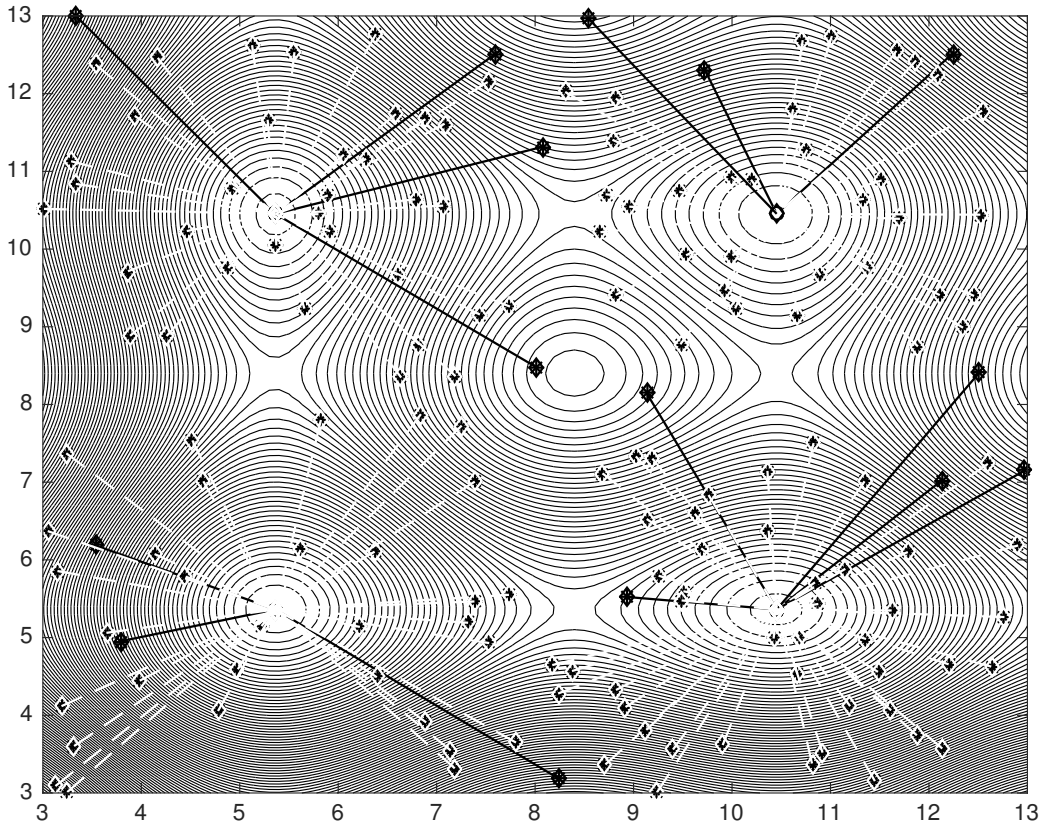


Figura 2.1: Ilustração do método MCSFilter (Amador et al., 2016).

A Figura 2.1 ilustra o funcionamento de um método baseado no *multistart*. Os traços a preto e a cheio, representam a procura local que foi realizada desde a geração aleatória do ponto até à convergência para um minimizante. A branco e a tracejado, são os pontos que foram gerados dentro de uma região de atração já estabelecida e que foram descartados. Isto é, a procura local não foi aplicada nestes pontos e é assumido que convergiriam para o minimizante pertencente a essa região de atração (Fernandes et al., 2013b; Fernandes, 2014).

Neste trabalho, testaram-se três algoritmos que usam diferentes variantes do método *multistart*: o *Multistart* Básico (MB), o *Multistart* com Regiões de Atração (MRA) e o *Multistart* com Procura Local Baseada nas Direções Coordenadas e Filtros, *Multistart based on Coordinate Search Filter* (MCSFilter) (Fernandes, 2014). O método de procura local que é utilizado nas três variantes do método *multistart* está apresentado no Algoritmo 4, documentado em Fernandes et al. (2013b) e Fernandes (2014).

Os algoritmos para os 3 tipos de *multistart* (Algoritmos 1, 2 e 3) encontram-se descritos em seguida. O Algoritmo 1 diz respeito à variante mais básica do *multistart*. Gera aleatoriamente um ponto do espaço de procura, x , e a partir deste encontra um minimizante, x^* , usando uma procura local. Os valores dos minimizantes são sempre

Algoritmo 1 *Multistart* Básico

```

1:  $k = 1$ ;
2: gerar aleatoriamente  $x \in \Omega$ ;
3: determinar  $x_1^* = \mathbf{CSFilter}(x)$ ;
4: enquanto a condição de paragem não for satisfeita fazer
5:   gerar aleatoriamente  $x \in \Omega$ ;
6:   determinar  $x^* = \mathbf{CSFilter}(x)$ ;
7:   se  $x^* \notin \{x_i^*, i = 1, \dots, k\}$  então
8:      $k = k + 1$ ;
9:      $x_k^* = x^*$ ;
10:  fim se
11: fim enquanto

```

guardados. O processo repete-se até que as condições de paragem sejam verificadas. Para uma melhor compreensão do Algoritmo 2 é necessário introduzir a definição de região de atração (Definição 6).

Definição 6. (Fernandes, 2014) *Seja $x \in \Omega$, com Ω o espaço de procura. Seja \mathbf{L} o procedimento de procura local. A região de atração de um minimizante, x_i^* , associado ao procedimento de procura local \mathbf{L} é definida por*

$$A_i \equiv \{x \in \Omega, \mathbf{L}(x) = x_i^*\},$$

onde x_i^* é o minimizante quando o procedimento de procura local \mathbf{L} é iniciado no ponto x .

Em torno de cada minimizante, local ou global, existe uma região de atração cujo raio é atualizado a cada iteração. Se um dado ponto gerado aleatoriamente se encontrar dentro de uma região de atração, admite-se que este tenderá sempre para o minimizante associado àquela região de atração. As regiões de atração têm a vantagem de diminuir o número de avaliações da função e o tempo de execução, uma vez que um ponto gerado aleatoriamente que já esteja inserido numa região de atração será descartado e a procura local não será iniciada.

Algoritmo 2 *Multistart* com Regiões de Atração

```

1:  $k = 1$ ;
2: gerar aleatoriamente  $x \in \Omega$ ;
3: determinar  $x_1^* = \mathbf{CSfilter}(x)$  e a região de atração  $A_1$ ;
4: enquanto a condição de paragem não for satisfeita fazer
5:   gerar aleatoriamente  $x \in \Omega$ ;
6:   se  $x \notin \cup_{i=1}^k A_i$  então
7:     determinar  $x^* = \mathbf{CSFilter}(x)$ ;
8:      $k = k + 1$ ;
9:      $x_k^* = x^*$  e determinar a correspondente região de atração  $A_k$ ;
10:  fim se
11: fim enquanto

```

O Algoritmo 2, do método *Multistart* com Regiões de Atração, é uma extensão do Algoritmo 1 com uma contabilização de regiões de atração, A_i . Assim, se um determinado ponto inicial gerado aleatoriamente se encontrar dentro de uma das regiões de atração, este ponto será descartado e a procura local não será realizada. Assume-se, neste caso, que a procura local iniciada convergiria para o mesmo minimizante já obtido. No caso de se gerar aleatoriamente um ponto que não pertença às regiões de atração a procura local é iniciada. Se se obter um minimizante conhecido, atualiza-se a região de atração associada a este minimizante. A região de atração será dada pela distância máxima entre todos os pontos iniciais para os quais se obteve o mesmo minimizante e ele próprio.

A variante MCSFilter diferencia-se da MRA na forma como os pontos gerados aleatoriamente são descartados. Repare-se que no Algoritmo 2 os pontos iniciais são descartados sempre que pertençam a alguma região de atração já definida. No algoritmo 3 — MCSFilter — o uso do ponto gerado aleatoriamente para iniciar a procura local depende, também, de uma determinada probabilidade *prob*, que depende do número de vezes que a procura

Algoritmo 3 MCSFilter

necessita parâmetros de entrada; $M^* = \emptyset$, $k = 1$;

- 1: gerar aleatoriamente $x \in [l, u]$;
 calcular α_{min} ;
 - 2: determinar $x_1^* = \mathbf{CSFilter}(x)$; $R_1 = \|x - x_1^*\|$, $r_1 = 1$;
 $M^* = M^* \cup \{x_1^*\}$;
 - 3: **repetir**
 - 4: gerar aleatoriamente $x \in [l, u]$;
 - 5: identificar o índice o : $dist_o = \min_{j=1, \dots, k} \{dist_j \equiv \|x - x_j^*\|\}$;
 - 6: **se** $dist_o < R_o$ **então**
 - 7: **se** a direção x para x_o^* é ascendente **então**
 - 8: $prob = 1$;
 - 9: **senão**
 - 10: calcular $prob$;
 - 11: **fim se**
 - 12: **senão**
 - 13: $prob = 1$;
 - 14: **fim se**
 - 15: **se** $\zeta < prob$ **então**
 - 16: Calcular $x^* = \mathbf{CSFilter}(x)$
 - 17: **se** x^* é um novo minimizante **então**
 - 18: $k = k + 1$, $x_k^* = x^*$, $r_k = 1$;
 $M^* = M^* \cup \{x_k^*\}$ e $R_k = \|x - x_k^*\|$;
 - 19: **senão**
 - 20: $R_w = \max\{R_w, \|x - x_w^*\|\}$; $r_w = r_w + 1$;
 - 21: **fim se**
 - 22: **senão**
 - 23: $R_o = \max\{R_o, \|x - x_o^*\|\}$; $r_o = r_o + 1$;
 - 24: **fim se**
 - 25: **até** as condições de paragem serem satisfeitas
-

local foi iniciada e do número de minimizantes distintos encontrados. No seguimento do exposto, é fácil verificar que neste algoritmo pode acontecer que o ponto inicial pertença a uma região de atração mas, mesmo assim, a procura local é iniciada. Note-se ainda que, à semelhança do Algoritmo 2, sempre que o ponto inicial não pertence a nenhuma região de atração a procura local é iniciada.

Relativamente às condições de paragem, nos métodos *multistart*, tem que existir um compromisso entre a fiabilidade das soluções obtidas e o custo implícito à sua obtenção. Em Fernandes (2014) foi usada uma estimativa para a fração/parte de espaço de procura não explorado dada por

$$P(N) = \frac{N(N+1)}{N_{LS}(N_{LS}-1)}, \quad (2.1)$$

em que N é o número de minimizantes distintos encontrados e N_{LS} é o número de procuras locais (*Local Search*) realizadas. A condição de paragem é dada por:

$$P(N) \leq 0,1. \quad (2.2)$$

Este critério mostra que o espaço de procura inexplorado é pequeno (Fernandes, 2014).

O Algoritmo 3 é apresentado de forma resumida e aconselha-se a consulta de Fernandes et al. (2013b) e Fernandes (2014) para mais detalhes sobre o método.

A procura local usada nas três versões apresentadas é a procura local CSFilter, que será apresentada de seguida.

Método Local CSFilter

O método CSFilter consiste numa procura através dos eixos coordenados acoplado à metodologia dos filtros para tratamento das restrições do problema. Tomando como exemplo um problema genérico com duas variáveis de decisão e sem restrições: a partir de um ponto inicial, $x_{central}$, avaliam-se os quatro novos pontos obtidos a partir das quatro direções coordenadas (norte, sul, este e oeste), em que x_{novo} é definido por

$$x_{novo} = x_{central} + \alpha d_i, \quad i = 1, \dots, 4, \quad \alpha \in (0, 1]. \quad (2.3)$$

Se para algum destes quatro novos pontos se obtiver um valor inferior do valor da função, o correspondente novo ponto passa a ser o ponto central na iteração seguinte e repete-se a procura, novamente, segundo as direções coordenadas. Se não se obtiver qualquer redução no valor da função, reduz-se o valor do passo — α — nas quatro direções e repete-se o cálculo com o valor de α atualizado e com o $x_{central}$ da iteração anterior. Este processo converge para uma solução local.

Quando são introduzidas restrições no problema de otimização, o processo de resolução complica-se. Uma vez que não se pretende utilizar funções de penalidade, a alternativa é

usar a metodologia dos filtros para tratar as restrições. Assim, o CSFilter é um método que permite resolver problemas com restrições, utilizando para tal a técnica dos filtros. Dado que não se usa qualquer informação da derivada, este método pode ser usado para determinar mínimos locais de funções descontínuas ou não diferenciáveis. Nesta procura local — usada neste trabalho como uma caixa-negra — o problema de otimização é interpretado e reescrito como um problema bi-objetivo onde se pretende minimizar a função objetivo, f , e a função violação das restrições, θ .

Num problema genérico com n variáveis, tem-se $2n$ direções coordenadas e, portanto, $2n$ novos pontos para testar (designados por pontos de sondagem). É nesta parte relativa aos $2n$ novos pontos que são introduzidos os filtros; para testar os novos pontos avalia-se a função objetivo, a função violação das restrições e as condições de aceitabilidade relativamente ao filtro. O filtro é um conjunto de pontos que define uma região proibida e funciona como um mecanismo de aceitação de pontos candidatos a possíveis soluções dos problemas. Isto é, um ponto candidato só será aceite como possível solução se melhorar o valor da violação das restrições ou o valor da função objetivo.

Se o ponto candidato for aceite pelo filtro passa a ser o ponto $x_{central}$ da iteração seguinte; se nenhum ponto for aceite pelo filtro, o valor do passo, α , é reduzido para

Algoritmo 4 CSFilter

necessita x (obtido através da procura de iniciação múltipla) e parâmetros de entrada;

$$x_{\mathcal{F}}^{best} = x; k = 1;$$

1: iniciar o filtro;

2: calcular α_1 ;

3: **repetir**

4: determinar $P_k = \{x_{trial}^i = x + \alpha_k d_i, d_i \in \mathcal{D}_{\oplus}, \alpha_k > 0\}$;

5: **repetir**

6: verificar a aceitabilidade dos pontos de sondagem;

7: **se** existir algum x_{trial}^i aceite pelo filtro **então**

8: atualizar o filtro;

9: selecionar x_{trial}^{best} ;

10: seja $x = x_{trial}^{best}$; atualizar $x_{\mathcal{F}}^{best}$ se apropriado;

11: **senão**

12: determinar $P_k = \{x_{trial}^i = x_{\mathcal{F}}^{best} + \alpha_k d_i, d_i \in \mathcal{D}_{\oplus}, \alpha_k > 0\}$;

13: verificar a aceitabilidade dos pontos candidatos x_{trial}^i ;

14: **se** existir algum x_{trial}^i aceite pelo filtro **então**

15: atualizar o filtro;

16: escolher x_{trial}^{best} ;

17: seja $x = x_{trial}^{best}$; atualizar $x_{\mathcal{F}}^{best}$ se apropriado;

18: **senão**

$$19: \alpha_{k+1} = \frac{\alpha_k}{2};$$

20: **fim se**

21: **fim se**

22: **até** novo x_{trial}^{best} ser aceite

23: **até** as condições de paragem serem verificadas

metade e verificam-se novamente as condições a partir do mesmo ponto central, com α atualizado.

A condição de paragem deste método está relacionada com o valor do passo usado, isto é, o processo repete-se até

$$\alpha \leq \alpha_{min} \quad (2.4)$$

com um valor de α_{min} pequeno.

A procura local CSFilter resumida é apresentada no Algoritmo 4.

Todos os Algoritmos apresentados para as diferentes variantes do método *multistart* estão numa versão simplificada. Para maior detalhe consultar Fernandes (2014).

2.2 Problemas Teste e Discussão de Resultados

Para testar a implementação em Java do algoritmo MCSFilter usam-se problemas bem conhecidos da literatura. São problemas de programação não linear (PNL), não convexos e multimodais; um deles apresenta, ainda, descontinuidade na função objetivo. Estas características tornam estes problemas de difícil resolução. No entanto, permitem concluir acerca do comportamento do método perante diferentes tipos de problemas e da qualidade da solução encontrada.

Os problemas testados foram agrupados em problemas com restrições de limite simples e problemas com restrições de igualdade e/ou desigualdade, de acordo com o Anexo A.1 e o Anexo A.2, respetivamente.

Para a realização dos testes foi utilizado um computador portátil de 2,6 GHz Core i7, com 8 GB de RAM e sistema operativo MacOS X El Capitan. Foi definido $\alpha_{min} = 1 \times 10^{-5}$ na Equação 2.4 e considerou-se 100 pontos iniciais gerados para MB e MRA. Relativamente às condições de paragem de MCSFilter (quer na versão em Matlab[®], quer na versão em Java) definiu-se também $\alpha_{min} = 1 \times 10^{-5}$ para a condição de paragem do CSFilter segundo a Equação 2.4 e $P \leq 0,1$ para a condição de paragem relativa ao *multistart*, de acordo com Fernandes (2014).

O cálculo das médias referentes aos parâmetros analisados foi feito com base em 10 execuções do método.

2.2.1 Problemas com restrições de limite simples

Nas figuras seguintes apresentam-se os resultados para as três variantes do *multistart*. Analisar-se-á o número médio de minimizantes distintos encontrados, N_{med} , o número médio de avaliações da função, AF_{med} , o tempo médio de execução, t_{med} , e o número médio de procuras locais usadas, $N_{LS,med}$. Lembra-se que a procura local CSFilter é usada nas três variantes de *multistart*.

Para simplificar a notação nas figuras a seguir apresentadas denotar-se-á por:

- **B** - a variante de *multistart* mais Básica (algoritmo MB) e tem-se os resultados em estudo N_{med-B} , AF_{med-B} , t_{med-B} e $N_{LS,med-B}$, respetivamente.
- **I** - a variante de *multistart* Intermédia (algoritmo MRA) e tem-se, desta forma, N_{med-I} , AF_{med-I} , t_{med-I} e $N_{LS,med-I}$.
- **C** - a variante mais Complexa do *multistart* (algoritmo MCSFilter-Java). Tem-se, neste caso, N_{med-C} , AF_{med-C} , t_{med-C} e $N_{LS,med-C}$.
- **O** - a variante Original do algoritmo MCSFilter (implementada em Matlab[®]). Tem-se, neste caso, N_{med-O} , AF_{med-O} , t_{med-O} e $N_{LS,med-O}$.

Relativamente aos resultados médios obtidos com o algoritmo MCSFilter-Matlab, estes foram cordialmente cedidos por Florbela Fernandes uma vez que os resultados publicados em Fernandes (2014) dizem respeito a uma versão anterior do MCSFilter. Os valores referentes à versão implementada em Matlab[®], foram obtidos num computador portátil de 2,8 GHz Core i7 com 16 GB de RAM, sistema operativo MacOS X El Capitan. Os problemas A1 até A7 encontram-se definidos no Anexo A.1, representam problemas com restrições de limites simples nas variáveis de decisão e serão analisados em seguida.

Problema A1

A Figura 2.2 diz respeito ao problema A1. Este problema tem 3 minimizantes globais sendo todos eles obtidos com as três variantes do método *multistart*, bem como na versão original. O tempo médio de execução diminui à medida que a complexidade do algoritmo

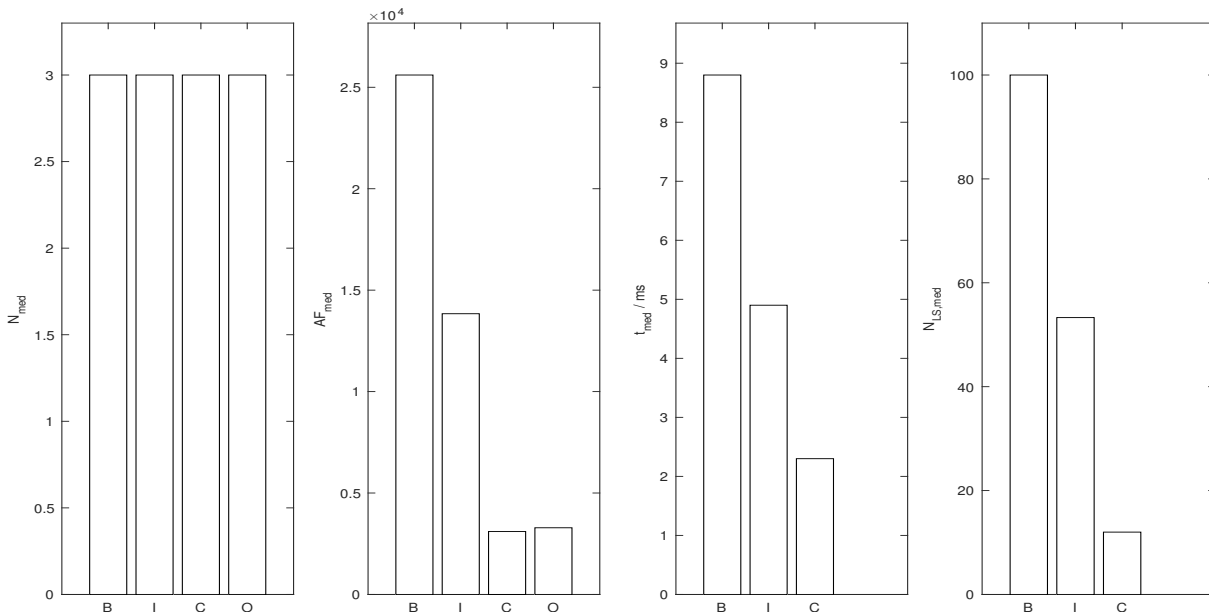


Figura 2.2: Resultados para N_{med} , AF_{med} , t_{med} e $N_{LS,med}$ referentes ao problema A1, definido no Anexo A, secção A.1.

aumenta (de MB para MRA e deste para MCSFilter-Java). Pela Figura 2.2, tem-se $t_{med-B} \leq t_{med-I} \leq t_{med-C}$.

O mesmo comportamento é visível no que diz respeito às procuras locais. O número médio de avaliações da função é muito elevado no algoritmo MB e diminui para o algoritmo MRA bem como deste para o algoritmo MCSFilter-Java. É possível evidenciar que o facto de os algoritmos MRA e MCSFilter usarem regiões de atração leva a que não sejam iniciadas várias procuras locais — o que explica a diminuição do número médio de procuras locais de MB para MRA e deste para os dois MCSFilter. Explica, também, a diminuição drástica que pode ser observada no número médio de avaliações da função.

Comparando agora, em particular, os resultados para os dois algoritmos MCSFilter (versões C e O nas figuras) observa-se que para um número médio semelhante de avaliações da função têm-se tempos médios de execução distintos. Obteve-se um tempo médio de execução menor com o MCSFilter-Java (ver secção B.1 do Anexo B), situação que se verifica para todos os problemas estudados. Atendendo a que as características do computador onde foi executado o MCSFilter-Matlab são ligeiramente melhores, pode inferir-se que a linguagem de programação é determinante para se obter um tempo de execução mais baixo para o MCSFilter-Java.

Problema A2

O Problema A2, tem a particularidade de ter seis minimizantes, dois globais e quatro locais. Este problema é muito usado para testar algoritmos dada a dificuldade em encontrar alguns dos seus minimizantes locais. Os resultados obtidos encontram-se na Figura 2.3.

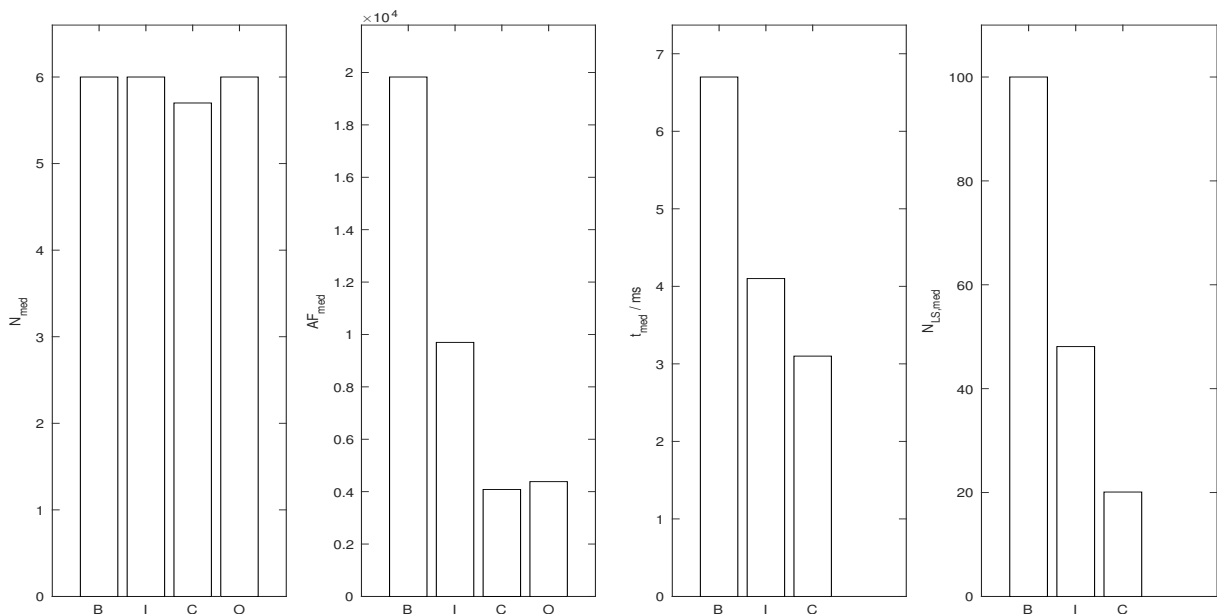


Figura 2.3: Resultados para N_{med} , AF_{med} , t_{med} e $N_{LS,med}$ referentes ao problema A2, definido no Anexo A, secção A.1.

Analisando a Figura 2.3, é fácil observar que a análise feita relativamente ao problema A1 é análoga para o problema A2.

Neste caso, refira-se ainda que as duas versões do MCSFilter encontram sempre o minimizante global. O facto de MCSFilter-Java apresentar um número médio de minimizantes encontrados inferior a 6 (5, 7) revela que em 3 das 10 execuções não foram encontrados os 6 minimizantes distintos em causa.

Problemas A3, A4 e A5

Os problemas A3, A4 e A5 são definidos pela mesma função genérica, distinguindo-se pelas matrizes a e c usadas em cada um dos problemas. Esta distinção leva a que o problema A3 tenha 5 minimizantes, o problema A4 possua 7 minimizantes e o problema A5 tenha 10 minimizantes. Os resultados para o problema A3 estão apresentados na Figura 2.4.

Da Figura 2.4 observa-se que o número médio de minimizantes distintos encontrados apenas corresponde ao valor da literatura no caso do algoritmo MB. Como acontecia para o problema A2, o mínimo global é sempre encontrado em todos os problemas, embora nos casos I e C não tenham sido encontrados todos os minimizantes distintos em todas as execuções.

O número de avaliações da funções é semelhante, quando comparados os algoritmos MCSFilter (Java e Matlab[®]), sendo que os tempos de execução são muito mais reduzidos para a implementação feita em Java (consultar Anexo B, Secção B.1).

Analisando a Figura 2.5, correspondente aos resultados obtidos para o problema A4, é

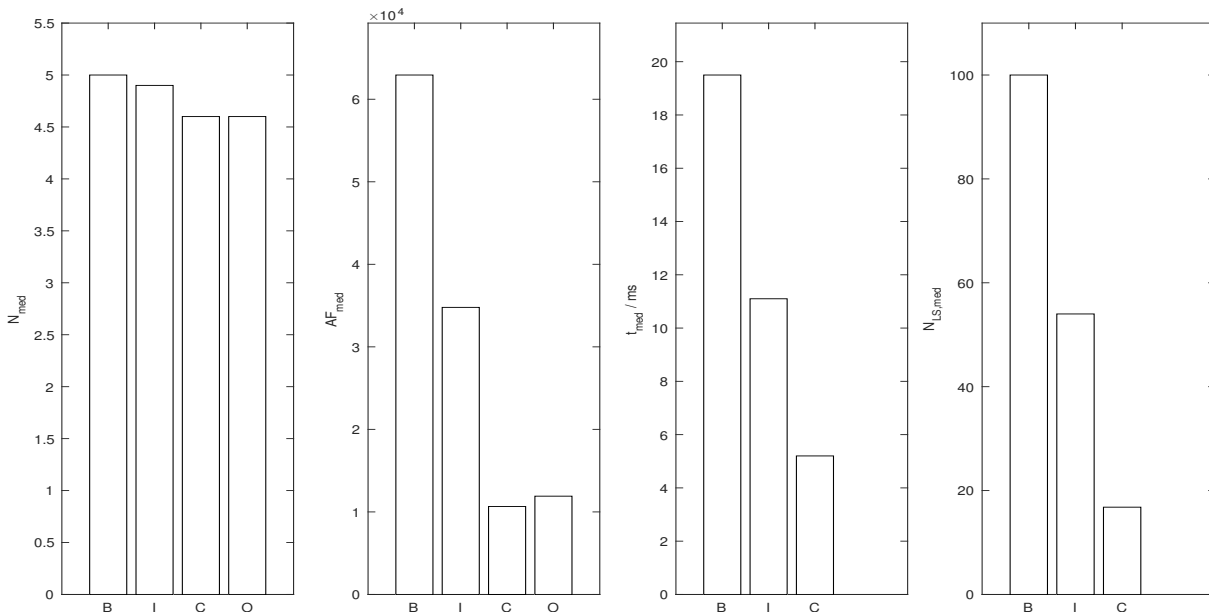


Figura 2.4: Resultados para N_{med} , AF_{med} , t_{med} e $N_{LS,med}$ referentes ao problema A3, definido no Anexo A, secção A.1.

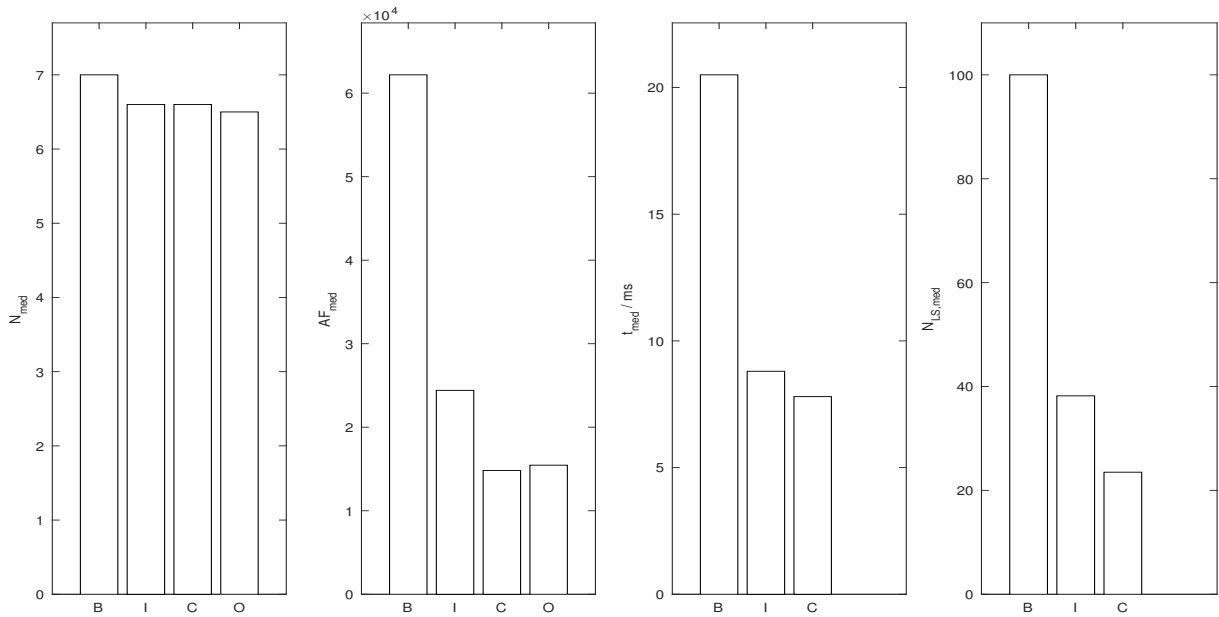


Figura 2.5: Resultados para N_{med} , AF_{med} , t_{med} e $N_{LS,med}$ referentes ao problema A4, definido no Anexo A, secção A.1.

perceptível que os algoritmos MCSFilter (em ambas as linguagens de programação) não atingem um valor de N_{med} correspondente ao valor reportado na literatura. Lembra-se que o cálculo de N_{med} é feito considerando as 10 execuções do método e, como tal, o valor médio apresentado para N_{med-C} e N_{med-O} significam que não foram encontrados todos os minimizantes distintos em todas as execuções. É importante ressaltar que o minimizante global é sempre atingido. A justificação para que não se tenham encontrado todos os minimizantes pode estar relacionada com a geração aleatória dos pontos iniciais e

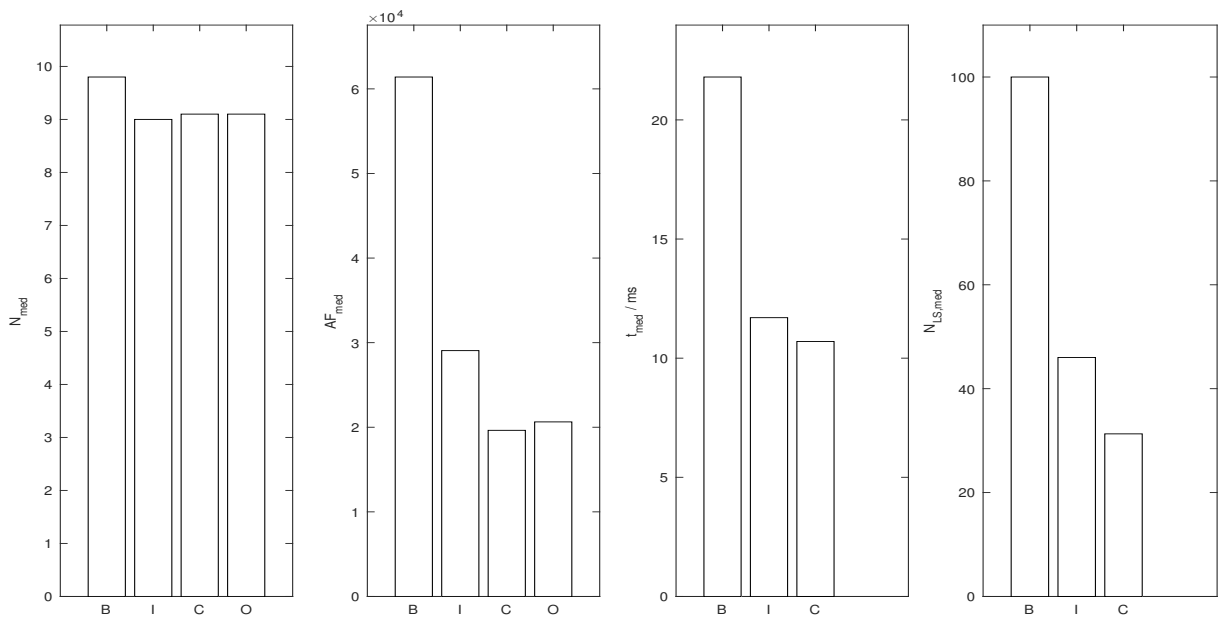


Figura 2.6: Resultados para N_{med} , AF_{med} , t_{med} e $N_{LS,med}$ referentes ao problema A5, definido no Anexo A, secção A.1.

a respetiva convergência do procedimento local para minimizantes já encontrados. Pode estar também relacionada com a condição de paragem do método ($P \leq 0,1$), em que os 10% do espaço de procura que não foram analisados incidiram na região onde existia um ou mais minimizantes por encontrar.

Os resultados para o problema A5 são apresentados na Figura 2.6.

Quando comparados os três problemas — A3, A4 e A5 (atendendo a que a função objetivo é semelhante) — é visível que aumentam gradualmente AF_{med} , t_{med} e $N_{LS,med}$, nas três variantes do algoritmo, o que é expectável já que o número de minimizantes é, respetivamente, 5, 7 e 10 (o que leva a que sejam necessárias mais procuras locais, ou seja, um maior número de avaliações da função).

Em relação aos tempos médios de execução e ao número médio de procuras locais, é visível que no caso do algoritmo MB, estes valores são significativamente superiores aos resultados dos algoritmos MRA e MCSFilter (Java e Matlab[®]), que se deve à introdução das regiões de atração nestes dois últimos algoritmos (MRA e MCSFilter-Java).

Problemas A6 e A7

Nas Figuras 2.7 e 2.8 são apresentados os resultados para os problemas A6 e A7, respetivamente. Estes dois problemas são definidos genericamente da mesma forma, apenas variando o número de variáveis de decisão de ambos. Uma vez que o problema A7 apresenta mais variáveis de decisão, é expectável que tanto o número médio de avaliações da função necessárias, bem como o tempo médio de execução sejam superiores neste problema.

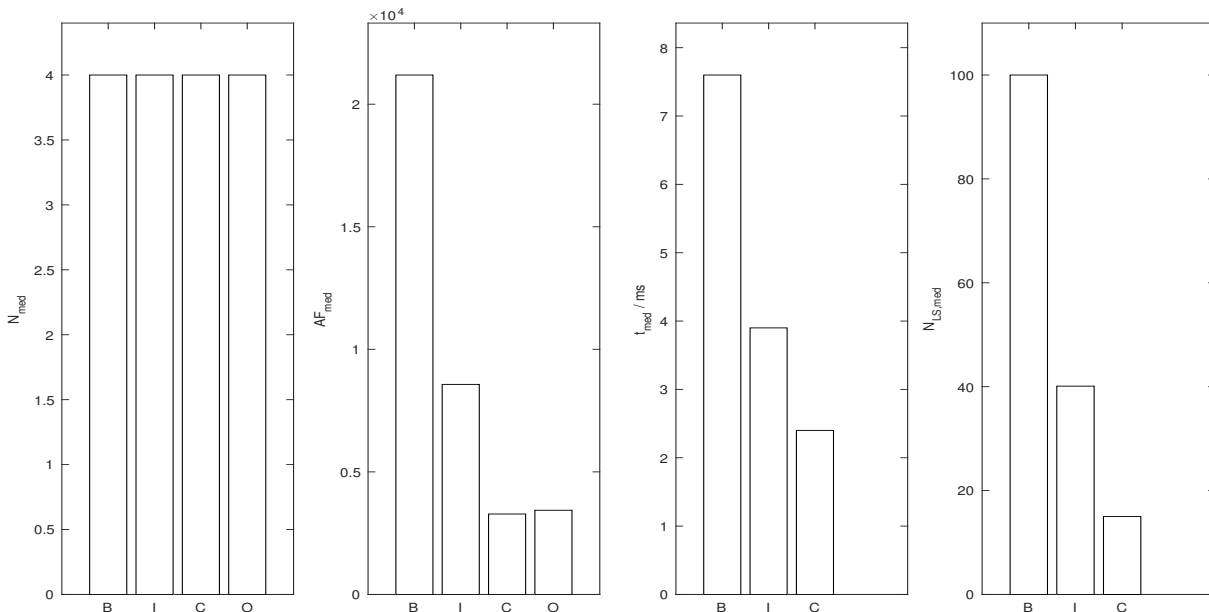


Figura 2.7: Resultados para N_{med} , AF_{med} , t_{med} e $N_{LS,med}$ referentes ao problema A6, definido no Anexo A, secção A.1.

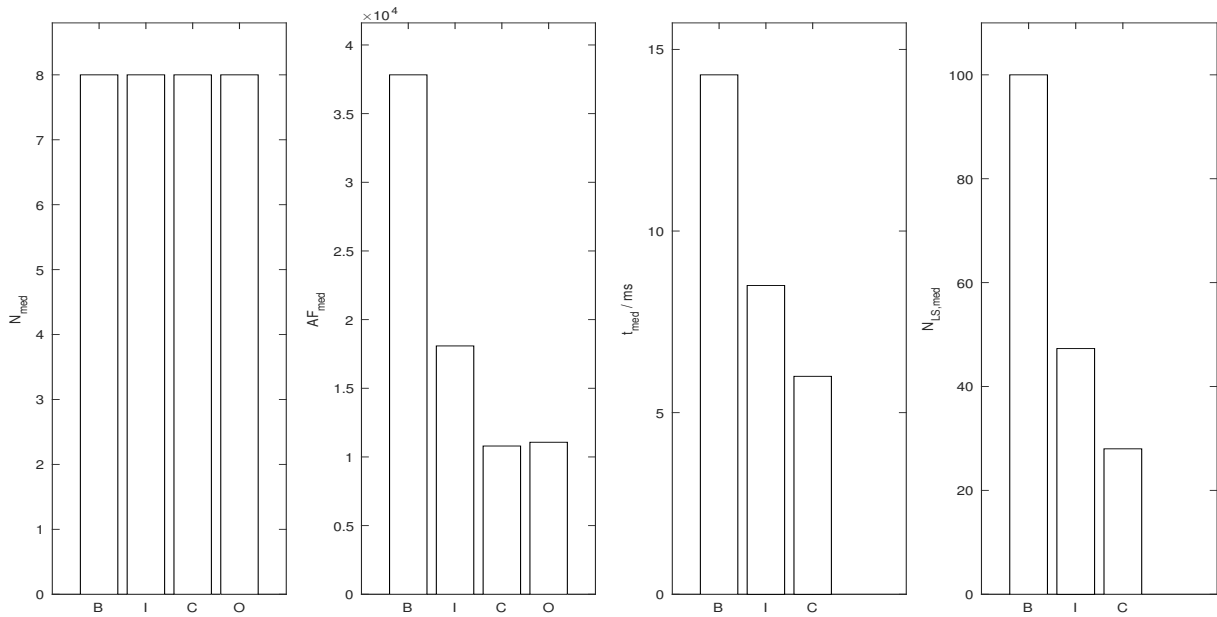


Figura 2.8: Resultados para N_{med} , AF_{med} , t_{med} e $N_{LS,med}$ referentes ao problema A7, definido no Anexo A, secção A.1.

Como se pode observar na Figura 2.7 e na Figura 2.8, todos os minimizantes distintos são encontrados em todas as execuções. Os resultados obtidos relativamente a AF_{med} , t_{med} e $N_{LS,med}$ seguem a mesma tendência dos problemas anteriores, uma vez que estes valores diminuem à medida que se aumenta a complexidade do algoritmo. Esta redução de esforço computacional deve-se à introdução das regiões de atração nos algoritmos MRA e MCSFilter, sendo os valores referentes ao algoritmo MB sempre muito superiores devido à condição de paragem do método (100 pontos a partir dos quais é necessário iniciar a procura local).

Relativamente aos dois problemas acima referidos, apesar de serem definidos genericamente através da mesma função, o problema A6 (possui 2 variáveis de decisão) tem 4 minimizantes distintos, enquanto que o problema A7 (possui 3 variáveis de decisão) tem 8. Esta diferença leva a um aumento, já esperado, de AF_{med} , t_{med} e de $N_{LS,med}$.

Para todos os problemas com restrições de limite simples em estudo, observa-se um esforço computacional mais elevado, isto é, mais avaliações da função, mais tempo de execução e mais procuras locais para o algoritmo mais simples (MB). Estes valores tendem a diminuir à medida que o algoritmo se torna mais complexo (MRA e MCSFilter que utilizam regiões de atração).

De uma forma geral, o método MCSFilter-Java é, dos métodos em estudo, o que apresenta um comportamento melhor no conjunto de problemas usado.

2.2.2 Problemas com restrições de igualdade e de desigualdade

Os problemas que se seguem dizem respeito a funções com restrições de igualdade e/ou desigualdade. A notação referida na secção anterior relativa às figuras mantém-se na

presente secção.

Os problemas têm restrições genéricas (igualdade ou desigualdade) o que faz com que sejam mais complicados de resolver do que os problemas até aqui estudados. Desta forma, nos resultados obtidos, após as 10 execuções, podem existir alguns valores que requeiram especial atenção. Nos problemas com restrições de igualdade a região admissível, sendo uma igualdade, pode representar dificuldades acrescidas à resolução do problema. Assim, aquando da procura local (relembra-se que a procura local usada é o CSFilter) pode acontecer que o valor do passo α (Equação 2.4) comece a ser reduzido numa fase precoce do método, o que pode levar a que a condição de paragem seja verificada antes de se encontrar a solução ótima. Esta redução do passo α pode levar, ainda, a que o método seja mais lento (necessita de mais iterações) e necessite de muitas mais avaliações da função.

Problema A8

A Figura 2.9 diz respeito ao problema A8. No que diz respeito ao método MB, o comportamento acima descrito pode acontecer para os 100 pontos gerados aleatoriamente, o que justifica que o método MB possa ser mais lento quando comparado com as outras versões do *multistart* implementadas em Java que usam as regiões de atração. Este problema tem apenas um minimizante, no entanto, para o MB é visível que são encontrados em média cerca de 2,5 minimizantes. Este valor significa que a condição de paragem do CSFilter foi verificada prematuramente, levando à paragem da procura local sem, no entanto, ser encontrada a solução ótima pretendida. Uma forma de ultrapassar este problema introduzido pela restrição de igualdade é relaxar mais as restrições de igualdade e/ou

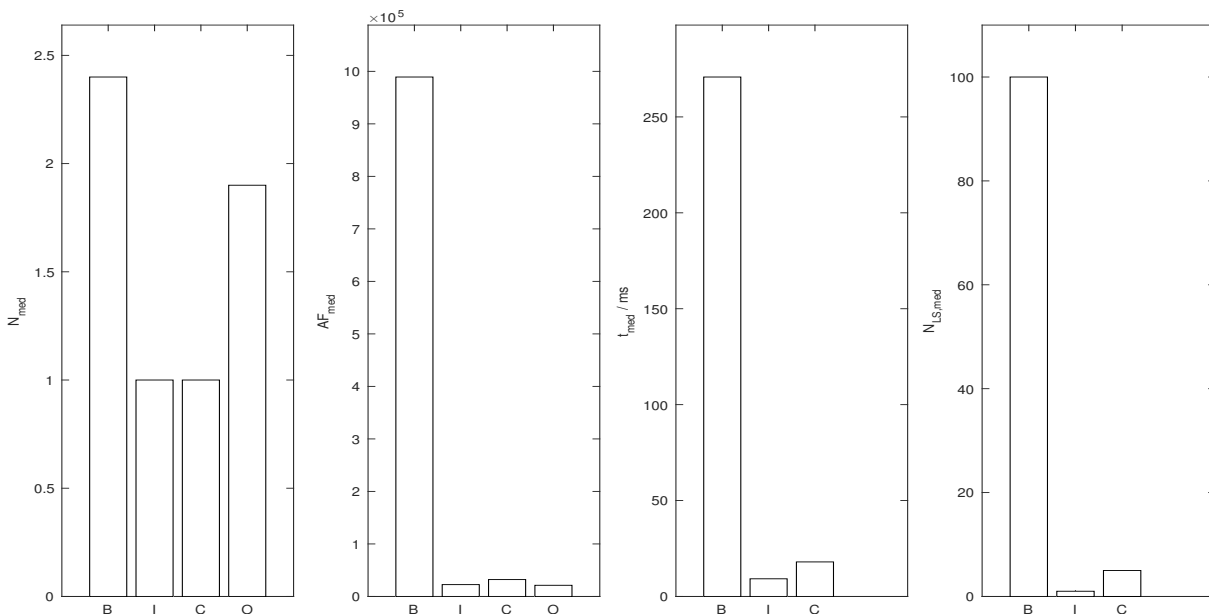


Figura 2.9: Resultados para N_{med} , AF_{med} , t_{med} e $N_{LS,med}$ referentes ao problema A8, definido no Anexo A, secção A.2.

diminuir o valor do parâmetro α_{min} na condição de paragem.

Comparando as versões MCSFilter (Java e Matlab[®]) é possível observar que ambas apresentam valores muito próximos para o número de avaliações da função. Embora uma comparação direta entre o tempo médio de execução não seja justa, MCSFilter-Java apresentou tempos muito menores do que os verificados para MCSFilter-Matlab (ver Anexo B, secção B.2). Relembrando que as características do computador onde foi executado o método MCSFilter-Matlab são ligeiramente melhores que as do computador onde foi executado o método MCSFilter-Java, pode inferir-se que a linguagem de programação na qual é implementado o método influencia significativamente os resultados obtidos.

Problema A9

A Figura 2.10 diz respeito ao Problema A9, sendo este um problema com uma restrição de igualdade e de desigualdade.

O número de minimizantes conhecidos na literatura para este problema é superior a 2 (Koziel e Michalewicz, 1999), e no trabalho de Pereira et al. (2013) os autores fazem referência a 5 minimizantes distintos encontrados com um método baseado no *simulated annealing*. É importante referir que a versão MCSFilter-Matlab apenas encontra 1 minimizante. Com MB, e com os 100 pontos obrigatórios de início da procura local, foram encontrados em média cerca de 4 minimizantes. Quando se utiliza o algoritmo MCSFilter-Java, em algumas das 10 execuções realizadas foram encontrados 5 minimizantes, sendo que alguns deles podem corresponder a paragens prematuras do método.

Quando comparadas as duas versões de MCSFilter — Matlab[®] e Java — a versão em

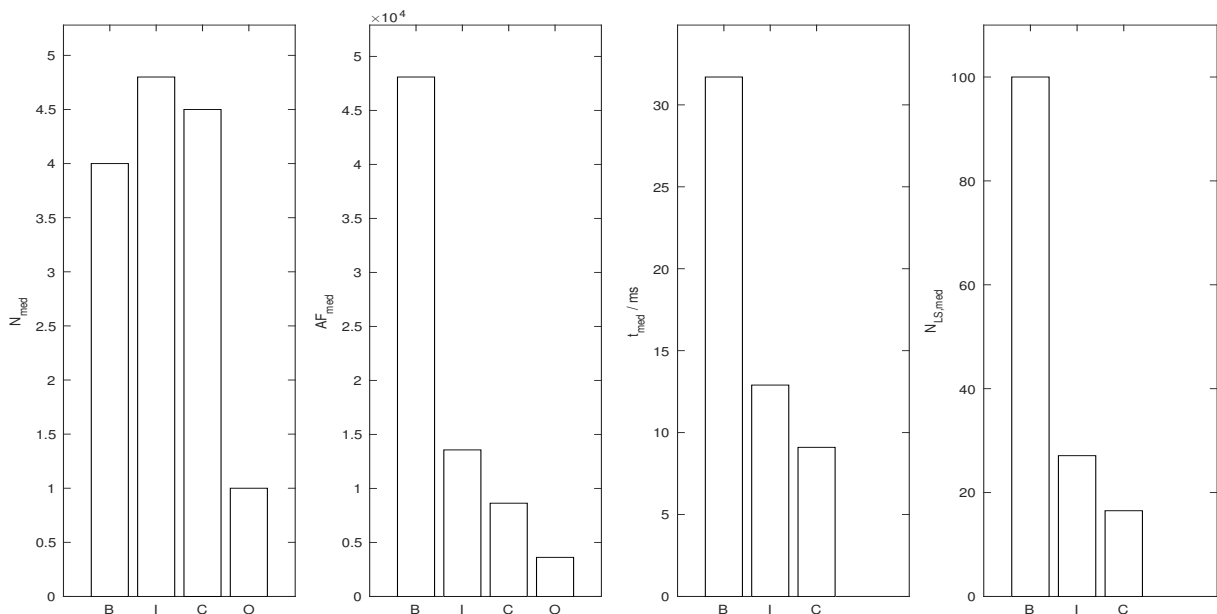


Figura 2.10: Resultados para N_{med} , AF_{med} , t_{med} e $N_{LS,med}$ referentes ao problema A9, definido no Anexo A, secção A.2.

Java apresenta mais avaliações da função, justificadas pelo maior número de eventuais minimizantes distintos encontrados pelo método.

Problema A10

A Figura 2.11 é referente ao problema A10 que, tal como o problema A8, é um problema com uma restrição de igualdade. O número médio de minimizantes obtidos remete para a mesma discussão relativa ao Problema A8 em que a condição de paragem é satisfeita prematuramente. Salienta-se mais uma vez que, apesar desta dificuldade, o mínimo global é sempre obtido, nas duas implementações do MCSFilter (Java e Matlab[®]).

Em relação ao número médio de avaliações da função, tempo médio de execução e número médio de procuras locais, o comportamento nos problemas com restrições de igualdade e/ou desigualdade é análogo ao comportamento verificado nos problemas com restrições de limite simples. Estes valores diminuem à medida que se aumenta a complexidade do algoritmo. Obtêm-se valores superiores nos problemas com maior complexidade, como seria expectável.

A dimensão do problema é também um fator a ter em conta: quanto maior a sua dimensão, mais avaliações da função necessita de fazer o procedimento local em cada iteração (uma vez que tem que avaliar $2n$ pontos por iteração).

O tratamento estatístico dos resultados obtidos, nas 10 execuções, nas três implementações em estudo (MB, MRA e MCSFilter-Java) para todos os problemas englobou também um cálculo do erro relativo, E_{rel} , entre o valor do mínimo global encontrado e o valor do mínimo global reportado na literatura. Os valores de E_{rel} , para todos os casos, são

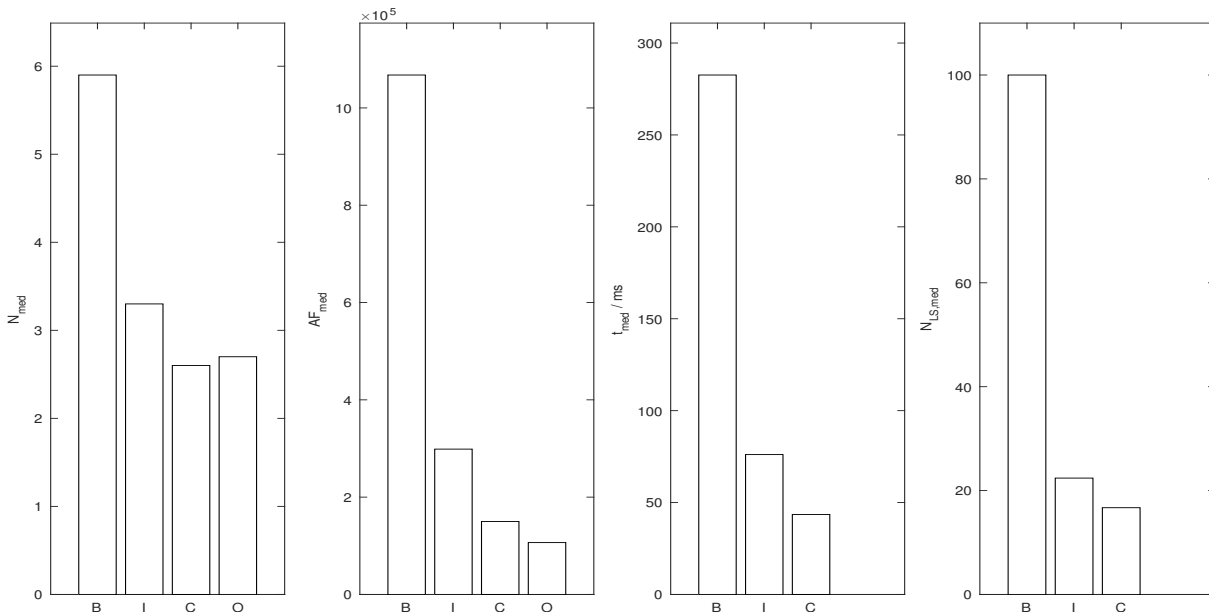


Figura 2.11: Resultados para N_{med} , AF_{med} , t_{med} e $N_{LS,med}$ referentes ao problema A10, definido no Anexo A, secção A.2.

aproximadamente zero, o que significa que o valor do mínimo global obtido está muito próximo do valor reportado na literatura. Calculou-se também o desvio padrão, σ , entre as soluções obtidas nas 10 execuções. Os valores de σ são, para todos os problemas, de 0,00, o que demonstra a boa qualidade da solução obtida. Todos estes valores, podem ser consultados no Anexo B.

Através da discussão dos resultados na Secção 2.2, podem formar-se algumas conclusões primárias acerca do desempenho dos três algoritmos, MB, MRA e MCSFilter (Matlab[®] e Java). Estas conclusões são apresentadas em seguida.

O mínimo global é sempre atingido; nem sempre são encontrados todos os minimizantes em todas as execuções. No entanto, pelos resultados obtidos, conclui-se que se obtém um número de minimizantes distintos igual ou muito próximo do resultado pretendido; as regiões de atração reduzem significativamente o esforço exigido pelo método e, consequentemente, o esforço computacional necessário. Esta redução traduz-se numa diminuição do tempo de execução; o aumento da complexidade entre cada versão do método *multistart* resulta num esforço computacional reduzido em relação ao anterior; a implementação do método na linguagem Java permite obter melhores tempos de execução, mantendo o número de avaliações da função semelhante ao da versão original — MCSFilter-Matlab; a geração aleatória de pontos pode influenciar os resultados finais no que diz respeito ao número de avaliações da função objetivo e de procuras locais; os resultados obtidos na literatura conseguem ser replicados e até melhorados em determinadas situações, o que evidencia a boa qualidade das soluções encontradas; dos três métodos de *multistart* testados, de uma forma geral, o que obtém melhor resultado é o MCSFilter-Java.

Nos capítulos seguintes, é demonstrada a aplicação do método MCSFilter na resolução do problema de estimativa de parâmetros de modelos de dois problemas em Engenharia Química: a estimativa dos parâmetros do modelo de uma válvula de controlo com *stiction* (Capítulo 3) e a estimativa de parâmetros da isomerização de α -pineno (Capítulo 4).

3. *Stiction* em Válvulas de Controlo

Neste capítulo é realizada uma breve introdução onde constam conceitos relativos ao fenómeno do *stiction*. Na Secção 3.1 é apresentado o modelo de Chen modificado que foi utilizado para prever o comportamento da válvula de controlo com *stiction*. Dentro desta secção é definido o problema de otimização ao qual foi aplicado o método MCSFilter. Na Secção 3.2 são apresentados e discutidos os resultados obtidos na resolução do problema. Estes são também comparados com os resultados reportados por Brásio et al. (2014b). Por último, é apresentada uma representação do perfil do comportamento da válvula previsto pelo modelo quando aplicadas a melhor e a pior soluções obtidas, sendo estas comparadas com o perfil real (experimental) e com a ordem dada à válvula no instante de tempo correspondente.

As válvulas de controlo são um dos elementos essenciais num ciclo de controlo de um processo químico. Em geral, são o único atuador mecânico presente no ciclo de controlo. O seu bom funcionamento é determinante para garantir níveis de produtividade do processo e a qualidade do produto final, minimizar o consumo energético e o desgaste dos equipamentos (Chen et al., 2008; Choudhury et al., 2008b; Brásio et al., 2014b).

Um dos maiores problemas associados ao funcionamento das válvulas é a presença de atrito estático que, em excesso, se designa por *stiction* (Choudhury et al., 2008b). Este fenómeno tem origem no movimento de dois objetos justapostos. No caso de uma válvula de controlo, o fenómeno de *stiction* resulta do movimento da haste em contacto com a zona de empacotamento da válvula (Choudhury et al., 2008b). As forças de atrito estático intensificam-se com o desgaste destas peças (Choudhury et al., 2004, 2005, 2008a,b).

Dado o papel fundamental que as válvulas de controlo têm no desempenho da operação num processo industrial, é necessário detetar com maior brevidade possível os casos em que as válvulas exibem este comportamento anómalo de forma a serem tomadas medidas que atenuem o seu impacto no desempenho do processo. De igual forma, é essencial a existência de métodos que permitam quantificar a severidade deste fenómeno de modo a permitir tomar uma decisão quanto à melhor forma de solucionar o problema (Choudhury et al., 2005; Srinivasan et al., 2005b; Choudhury et al., 2008a).

Para detetar e quantificar o problema de *stiction* existem diversas abordagens, como por exemplo, a análise do perfil de PV (*process variable*) em função de OP (*output variable*). A existência de *stiction* dá origem a um perfil com forma elíptica (Choudhury et al., 2004,

2005, 2006; Lee et al., 2008; Chen et al., 2008; Farenzena e Trierweiler, 2012). Todavia, este método não é muito fiável na medida em que existem outros fatores que podem gerar um perfil de PV elíptico que não está relacionado com a presença de *stiction*. Srinivasan et al. (2005a) propõem a utilização de um programa de reconhecimento de padrões que analisa o sinal de resposta, PV, e determina se o comportamento oscilatório exibido se deve à presença de *stiction* na válvula ou se tem outra origem: quando a causa é a dinâmica do processo, a resposta exibe um comportamento sinusoidal; quando se trata de ruído estático, o sinal é mais aleatório; e quando se trata de *stiction* o perfil do sinal pode exibir picos triangulares, quadrados ou dentados. No entanto, também devido ao facto de a análise do gráfico PV *vs* OP não ser fiável como meio de deteção e quantificação do fenómeno de *stiction*, a maioria dos trabalhos reportados na literatura apresentam estratégias baseadas na utilização de um modelo matemático que descreve o comportamento da variável de controlo com *stiction* (Rossi e Scali, 2005; Srinivasan et al., 2005b; Chen et al., 2008; Jelali, 2008; Romano e Garcia, 2011; Brásio et al., 2014a,b).

3.1 Modelo

O comportamento das válvulas que exibem *stiction* pode ser previsto por modelos baseados apenas nos valores da variável de saída do controlador (podendo estes estar condicionados por apenas um parâmetro (Srinivasan et al., 2005b; Srinivasan e Rengaswamy, 2008) ou por dois parâmetros (Choudhury et al., 2008a; Jelali, 2008; Brásio et al., 2014a)). Neste trabalho é utilizado um modelo de Chen modificado (Brásio et al., 2014b). Trata-se de um modelo descontínuo *data-driven*, ou seja, que apenas requer informação da variável de saída

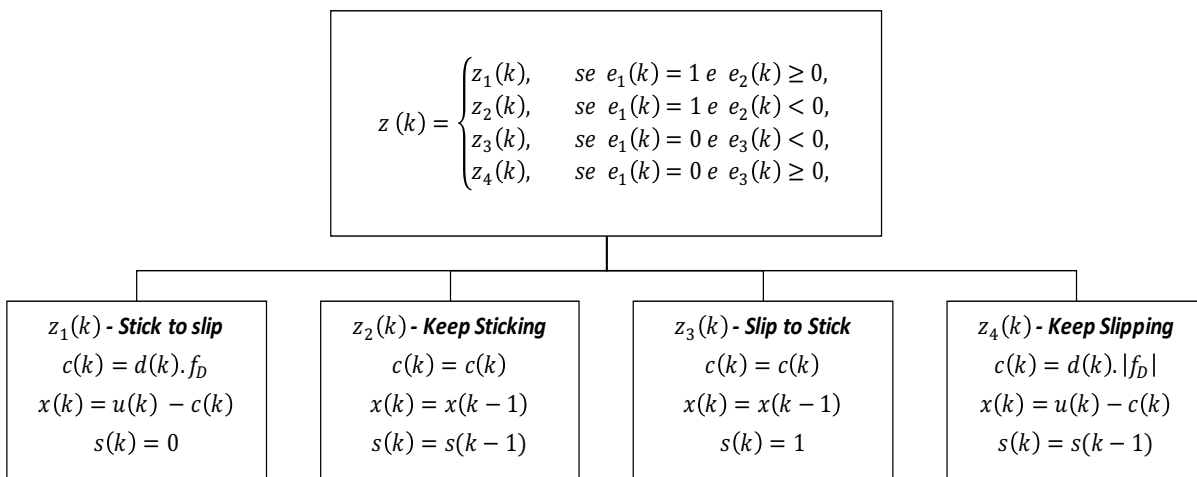


Figura 3.1: Modelo de Chen utilizado para descrever o comportamento de uma válvula de controlo com *stiction*. Adaptado de Brásio et al. (2014b).

do controlador (Chen et al., 2008; Brásio et al., 2014b). Estão representados na Figura 3.1 os quatro possíveis padrões de comportamento, em que $z_i^\top = \begin{bmatrix} c(k) & x(k) & s(k) \end{bmatrix}$ é o vetor composto pelos sinais de saída da válvula para cada instante de tempo, k . $c(k)$ representa a força acumulada que é compensada pelo atrito e é definida por $c(k) = c(k-1) + u(k) - u(k-1)$, enquanto $s(k)$ representa a posição de abertura da válvula.

O modelo de Chen é também denominado por “*Two-layer binary tree data-driven model*”, ou seja, é um modelo composto por uma árvore binária de dois níveis que compreende 4 resultados possíveis para $z_i(k)$. O resultado final obtido é definido consoante o valor das variáveis $e_i(k)$, $i = 1, 2, 3$ definidas por

$$e_1(k) = s(k-1), \quad e_2(k) = c(k) - f_S, \quad e_3(k) = \begin{cases} 1, & \text{se } f_D > 0 \text{ e } |c(t)| > f_D, \\ 1, & \text{se } f_D = 0 \text{ e } d(t) \cdot d(t-1) > 0, \\ 1, & \text{se } f_D < 0 \text{ e } |c(t)| < -f_D, \\ 0, & \text{caso contrário,} \end{cases} \quad (3.1)$$

em que $d(k)$ é a direção em que a haste da válvula se está a deslocar (Brásio et al., 2014b).

Este modelo depende de dois parâmetros: o atrito dinâmico, f_D , e o atrito estático, f_S . Quando a válvula se está a mover, o atrito estático é praticamente nulo. Quando o atrito dinâmico é inferior ao atrito estático a válvula move-se (Choudhury et al., 2008b).

Estes parâmetros são linearmente dependentes, tendo-se que (Chen et al., 2008; Brásio et al., 2014b)

$$f_S = f_D + f_J \quad (3.2)$$

em que o parâmetro f_J representa um valor de atrito correspondente a um comportamento de *slip-jump*. A abertura da válvula exhibe uma variação abrupta quando f_J atinge um valor superior a f_S (Chen et al., 2008; Choudhury et al., 2008b; Brásio et al., 2014b). Sendo f_S e f_D linearmente dependentes, é definido o vetor de parâmetros do modelo, por $p^\top = \begin{bmatrix} f_D & f_J \end{bmatrix}$. Este é, portanto, o conjunto de parâmetros que é necessário estimar com base nos dados experimentais de modo a caracterizar o comportamento da válvula com *stiction* (na Secção 3.1.1).

No primeiro caso ilustrado na Figura 3.1, para $z_1(k)$ tem-se uma situação em que a válvula está completamente presa, sob efeito da ordem dada pelo controlador para se mover. Ocorre um movimento abrupto, que irá resultar na deslocação da haste da válvula para um ponto de abertura que não cumpre o valor de *set-point* estabelecido, onde se prende novamente (Chen et al., 2008). No segundo caso, $z_2(k)$, a válvula não responde de todo ao controlador. Mantém-se na posição inicial, o que significa que o sinal do controlador não é suficiente para ultrapassar o atrito estático (Chen et al., 2008). Esta

condição corresponde a uma situação em que a válvula prende por completo. Para que esta condição seja ultrapassada, o *set-point* tem que ser inflacionado (Choudhury et al., 2005, 2006, 2008a). Desta forma, a válvula vai soltar-se e escorregar até atingir um ponto que não corresponde à abertura desejada e prender novamente, $z_3(k)$. No caso de $z_4(k)$, a válvula está em constante movimento, o que pode ser causado por mudanças na direção da deslocação da válvula.

3.1.1 Problema de Otimização

O problema da determinação dos parâmetros do modelo da válvula de controlo com base em valores experimentais pode ser formulado como um problema de otimização. É de salientar que o modelo aqui utilizado para descrever o movimento de uma válvula com *stiction* é de natureza descontínua. Assim, não é viável a aplicação de um método de otimização que requeira a informação sobre a derivada da função objetivo, a não ser que se introduzam funções de suavização das discontinuidades, tal como demonstrado em Brásio et al. (2014b). O método MCSFilter é, portanto, um bom candidato à resolução do problema, como será demonstrado mais à frente.

Matematicamente, o problema da determinação dos parâmetros do modelo da válvula é baseado na minimização de uma função objetivo definida pelo método dos mínimos quadrados sujeita a restrições de limite simples, tal como em Jelali (2008), Romano e Garcia (2011) e Brásio et al. (2014b). As variáveis de decisão do problema de otimização são os parâmetros contidos em p , sendo o problema definido por:

$$\underset{p}{\text{minimizar}} \quad J = (\mathbf{z}_{\text{exp}} - \mathbf{z})^T \mathbf{Q} (\mathbf{z}_{\text{exp}} - \mathbf{z}), \quad (3.3a)$$

$$\text{sujeito a} \quad \mathbf{z}_L \leq \mathbf{z} \leq \mathbf{z}_U, \quad (3.3b)$$

$$\mathbf{u}_L \leq \mathbf{u} \leq \mathbf{u}_U, \quad (3.3c)$$

$$p_L \leq p \leq p_U, \quad (3.3d)$$

em que J é a função objetivo que se pretende minimizar e \mathbf{Q} é uma matriz que contém as penalizações atribuídas a cada elemento do vetor. O vetor \mathbf{z} representa as posições da válvula correspondentes a um horizonte de tempo previstos pelo modelo da válvula e \mathbf{z}_{exp} representa os valores experimentais correspondentes. O vetor \mathbf{u} é o perfil do sinal de entrada enviado à válvula pelo controlador e p é o vetor de parâmetros do modelo — f_D e f_J — que se pretendem estimar. Os subscritos L e U representam os limites superior e inferior, respetivamente (Brásio et al., 2014b).

3.2 Resultados e Discussão

Neste exemplo são utilizados como perfis experimentais, \mathbf{z}_{exp} , e o vetor do sinal de entrada \mathbf{u} , os valores de Brásio et al. (2014b), obtidos por simulação de uma válvula de controle com *stiction* em que $f_D = 0,90$ e $f_J = 1,90$. Estes perfis são de dimensão $\iota = 98$.

As restrições do problema Equação 3.3 foram definidas de acordo com: $0 \leq f_D \leq 5$, $0 \leq f_J \leq 5$, $0 \leq x(k) \leq 1$ e $0 \leq u(k) \leq 1$, $k = 1, \dots, \iota$. Definiu-se a matriz \mathbf{Q} como sendo a matriz identidade. Realizou-se uma simulação Monte Carlo com 1000 execuções do método de otimização num computador portátil de 2,6 GHz Core i7, com 8 GB de RAM e sistema operativo MacOS El Capitan. A estimativa inicial em cada execução foi gerada aleatoriamente pelo método MCSFilter, exceto na primeira procura local onde se considerou $f_D = f_J = 0.20$. O tempo médio de cada execução foi de 7×10^{-2} s.

Os resultados da simulação Monte Carlo encontram-se apresentados na Tabela 3.1, onde J é o valor da função objetivo, f_J é a percentagem de *slip-jump* presente, f_D é a percentagem de atrito dinâmico presente, t é o valor do tempo de execução e AF é o número de avaliações da função. Os perfis previstos pelo modelo resultantes da melhor e pior estimativa obtidas pelo método e o perfil experimental são apresentados na Figura 3.2.

Tabela 3.1: Resultados do método MCSFilter ao problema da determinação dos parâmetros do modelo que descreve uma válvula de controle com *stiction*.

	Melhor	Pior	Médio	Brásio et al. (2014b)
J_{glob}	$6,19 \times 10^{-11}$	$1,55 \times 10^{-2}$	$4,18 \times 10^{-5}$	1×10^{-3}
$f_J/\%$	1,868	1,700	-	1,732
$f_D/\%$	0,899	0,913	-	0,898
t/ms	69	68	70,14	-
AF	6901	6382	6795	-

O melhor resultado da Tabela 3.1 é 8 ordens de grandeza inferior ao resultado apresentado por Brásio et al. (2014b). O pior valor é ligeiramente superior ao valor na mesma referência, enquanto que o valor médio se mantém 2 ordens de grandeza abaixo do valor de referência (última coluna da Tabela 3.1). Considerando o melhor e o pior resultado, o parâmetro f_J tem uma diferença de +7,28% e de +1,88% relativamente aos valores apresentados por Brásio et al. (2014b), respetivamente. Em relação ao valor “real” de $f_J = 1,90$, a melhor e a pior soluções conseguidas pelo método MCSFilter apresentam diferenças de -1,71% e de -11,76%, respetivamente. Quanto ao valor “real” $f_D = 0,90$, as diferenças são de -0,11% no caso da melhor estimativa e de +1,42% para a pior estimativa. Em relação aos resultados de Brásio et al. (2014b), estes valores são de +0,11%

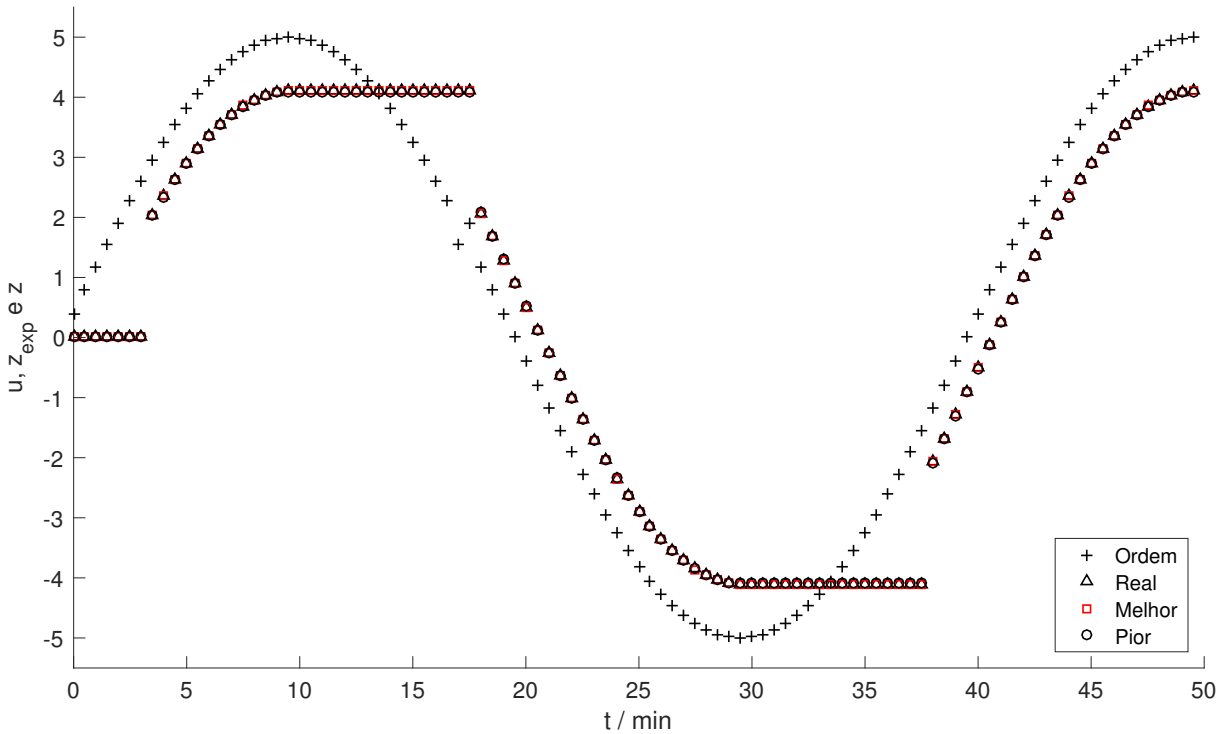


Figura 3.2: Perfis de z obtidos com o modelo de Chen utilizando a melhor e a pior estimativa dos parâmetros dada pelo método MCSFilter (ver Tabela 3.2), perfil “experimental”, z_{exp} e o perfil do sinal de comando dado à válvula.

e de +1,67%, respectivamente. Analisando as diferenças relativas calculadas, conclui-se que, no caso do parâmetro f_J , a melhor estimativa obtida com MCSFilter está mais próxima do valor “real” do que a estimativa de Brásio et al. (2014b). Quanto ao parâmetro f_D , as diferenças relativas não mostram diferenças significativas entre a melhor e a pior solução. A média dos valores obtidos para a função objetivo, J_{glob} , revela a boa qualidade da solução obtida. Da observação da Tabela 3.1 verifica-se que tanto o tempo da melhor e como da pior soluções estão abaixo da média.

Na Figura 3.2 são apresentados os perfis previstos pelo modelo com a melhor e a pior solução, bem como o perfil “experimental”.

A comparação visual entre os perfis obtidos com a melhor e a pior solução de parâmetros permite concluir que não há diferenças significativas. Esta “sobreposição” dos dois perfis era expectável uma vez que o valor da função objetivo em ambos os casos é bastante reduzido.

Para melhor visualizar as diferenças entre o perfil da posição da válvula “experimental” e o perfil previsto pelo método, na Figura 3.3, é apresentada uma ampliação dos perfis da Figura 3.2 entre $t = 12,4$ min e $t = 14,2$ min.

Tal como se pode observar, mesmo a pior solução obtida com o método MCSFilter se aproxima razoavelmente bem do perfil “real” ou “experimental”.

É conveniente realçar que à escala industrial, as alternativas mais comuns para a resolução em casos reais do problema criado pelo fenómeno de *stiction* passam, essencialmente,

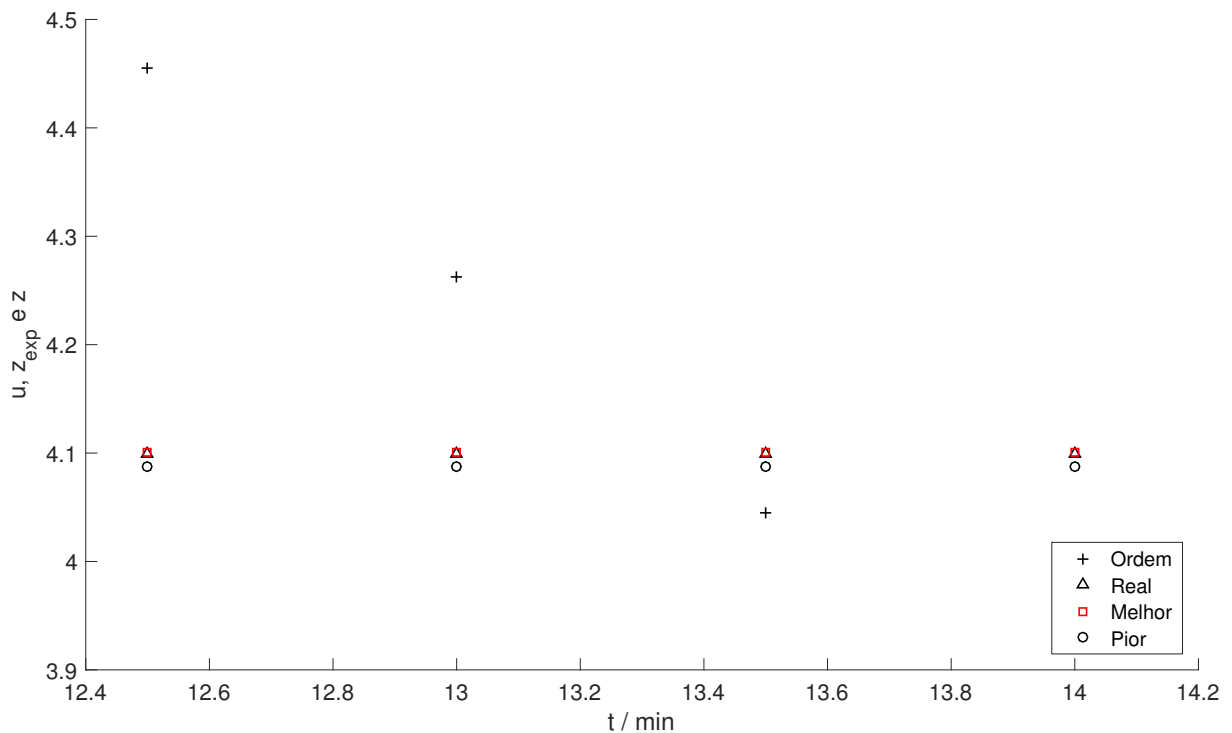


Figura 3.3: Ampliação dos perfis obtidos pelo modelo de Chen modificado (Chen et al., 2008; Brásio et al., 2014b).

pela manutenção física do equipamento ou pela nova estimativa de parâmetros de modelo que descreve o comportamento da válvula. Esta nova estimativa tem em conta o perfil “experimental” mais recente (Srinivasan et al., 2005a) (uma vez que o fenômeno de *stiction* piora com o desgaste dos materiais e o perfil “experimental” será diferente). Descorar o fenômeno de *stiction* e a sua necessidade de resolução pode provocar alterações na dinâmica do processo significativas. Estas podem estar relacionadas com o aumento do consumo energético, baixando os lucros, e pode também influenciar a qualidade do produto final (no caso de ser exigida uma temperatura reacional exata e a válvula com *stiction* estar a controlar a entrada de fluido de aquecimento/arrefecimento no reator, por exemplo). Na tomada de decisões, o tempo de execução do método é um fator determinante no caso de ser necessária uma nova estimativa de parâmetros. Pode concluir-se que o método MCSFilter se apresenta como um método promissor e capaz de competir, para este tipo de modelos, com outros métodos já estudados.

4. Isomerização do α -pineno

Neste capítulo, é ilustrada a aplicação do método MCSFilter na determinação dos parâmetros do modelo cinético da isomerização do α -pineno. Na Secção 4.1 é apresentado o modelo cinético constituído por um sistema de 5 equações diferenciais ordinárias. Nesta secção, é também definido o problema de otimização em causa. Na Secção 4.2 são apresentados e discutidos os resultados da resolução do problema de otimização com o método MCSFilter. Estes são comparados com os resultados reportados na literatura. Por fim, são representados graficamente os perfis das concentrações dos compostos presentes na isomerização do α -pineno previstos pelo modelo com a melhor solução obtida na resolução do problema de otimização.

O primeiro estudo reportado na literatura sobre a isomerização do α -pineno remonta a 1947, desenvolvido por Fuguitt e Hawkins (Box et al., 1973). A isomerização do α -pineno dá origem à formação de dipenteno e de allo ocinemo que, por sua vez, origina α e β -pironeno irreversivelmente e um dímero (reação reversível). As reações encontram-se esquematizadas na Figura 4.1, em que y_1 é o α -pineno, y_2 é o dipenteno, y_3 é o allo ocinemo, y_4 é o α e β -pironeno e y_5 é o dímero. p_γ representa a velocidade de cada reação, $\gamma = 1, \dots, 5$.

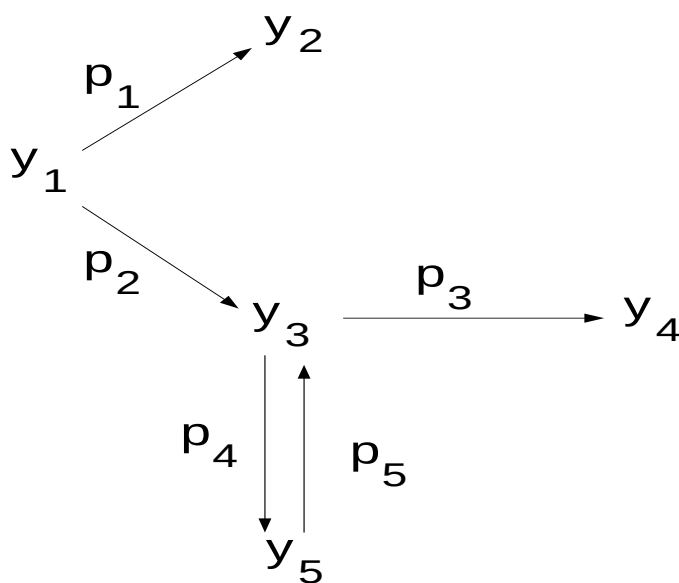


Figura 4.1: Representação esquemática do sistema reacional da isomerização do α -pineno.

4.1 Modelo Cinético

Quando se trata da modelação de sistemas reais, muitas vezes existem algumas incertezas quanto ao funcionamento do sistema ou complexidades que não são possíveis de transpor para um modelo matemático. Na indústria, a dinâmica inerente ao processo pode ser representada recorrendo um sistema de equações diferenciais (Egea et al., 2007). Estes modelos, que podem ser lineares ou não lineares, podem apresentar descontinuidades, o que dificulta a obtenção da informação da função derivada. Como tal, é importante testar métodos de otimização baseados numa procura direta que não recorram à informação da derivada da função objetivo. Por conseguinte, é necessário estudar a aplicabilidade destes métodos e comparar os resultados obtidos com outros já publicados (Egea et al., 2007).

Box et al. (1973) definiram que o modelo cinético poderia ser desenvolvido através da derivação das concentrações dos produtos em ordem ao tempo. Com base nesse pressuposto, e utilizando as concentrações experimentais medidas por Fuguitt e Hawkins em 1947 para 8 intervalos de tempo, (Box e Draper, 1965) propuseram o seguinte modelo cinético:

$$y_1' = -(p_1 + p_2) y_1, \quad (4.1a)$$

$$y_2' = p_1 y_1, \quad (4.1b)$$

$$y_3' = p_2 y_1 - (p_3 + p_4) y_3 + p_5 y_5, \quad (4.1c)$$

$$y_4' = p_3 y_3, \quad (4.1d)$$

$$y_5' = p_4 y_3 - p_5 y_5, \quad (4.1e)$$

em que y_q' é a derivada da concentração do componente q em ordem ao tempo, $q = 1, \dots, 5$, e p_γ , $\gamma = 1, \dots, 5$, são parâmetros do modelo.

Dada a sua natureza complexa, este modelo foi utilizado por vários autores no âmbito da integração e teste de algoritmos (Csendes, 1988; Tjoa e Biegler, 1991; Storn e Price, 1997; Runarsson e Yao, 2000; Jones, 2001; Dolan et al., 2004; Ugray et al., 2005; Egea et al., 2008; Rocha et al., 2016). Na Secção 4.2, são comparados os resultados reportados na literatura com os resultados obtidos aplicando o método MCSFilter.

4.1.1 Problema de Otimização

Este sistema reacional é um sistema complexo, uma vez que é composto por 5 reações, uma delas reversível, onde existem reações simultâneas e em cadeia. Numa produção industrial, as condições de operação irão determinar a produtividade do processo que se traduz em lucros ou prejuízos. Estas são um fator decisivo na produção dos composto de interesse num sistema como o que está a ser estudado e, como tal, é importante otimizar o processo de forma a determiná-las. Por sua vez, um dos fatores que influencia a alteração das condições de operação são as velocidades de reação. São estes os parâmetros que se

pretende estimar com o problema de otimização.

Matematicamente, o problema de otimização pode ser formulado pelo método dos mínimos quadrados com restrições de limite simples (Tjoa e Biegler, 1991). Sendo o modelo contituído por equações diferenciais, as avaliações da função objetivo efetuadas nas procuras locais são executadas recorrendo a um integrador numérico (Tjoa e Biegler, 1991). Os parâmetros do modelo são as variáveis de decisão do problema de otimização (Rocha et al., 2016). O problema da determinação dos parâmetros do modelo cinético que melhor ajustam os perfis de concentração previstos pelo modelo aos valores experimentais pode ser formulado da seguinte forma (Larrosa, 2008):

$$\underset{p}{\text{minimizar}} \quad J(p) = \sum_{q=1}^5 \sum_{k=1}^8 (y_q(p, t_k) - \tilde{y}_{qk})^2, \quad (4.2a)$$

$$\text{sujeito a} \quad p_L \leq p \leq p_U, \quad (4.2b)$$

em que $J(p)$ é a função objetivo que se pretende minimizar, $y_q(p, t_k)$ é o valor de concentração previsto pelo modelo para o composto q , para $q = 1, \dots, 5$, no instante de tempo k , com $k = 1, \dots, 8$, e \tilde{y}_{qk} é o valor do perfil experimental do composto q , no instante de tempo k . Os subscritos U e L dizem respeito aos limites superiores e inferiores, respetivamente.

O integrador numérico utilizado para obter os valores previstos pelo modelo é baseado no método Adams Bash Forth, disponível numa versão implementada em Java. Este método permite que os erros associados aos valores previstos pelo modelo sejam minimizados, pois permite selecionar tolerâncias mais apertadas na determinação de $y_q(p, t_k)$, sem comprometer demasiado o tempo de execução do método de otimização (Durrant, 1990). No presente estudo, as tolerâncias absoluta e relativa utilizadas no integrador são iguais, com um valor de 1×10^{-8} .

Quando aplicado o método MCSFilter para resolução do problema de otimização da determinação dos parâmetros do modelo cinético, é expectável que os tempos de execução sejam elevados uma vez que se trata de um processo iterativo em que em cada avaliação da função é necessário integrar o modelo cinético.

4.2 Resultados e Discussão

Os limites superiores e inferiores estabelecidos no problema de otimização foram, respetivamente:

$$p_U^\top = \left[1 \times 10^{-4} \quad 1 \times 10^{-4} \quad 1 \times 10^{-4} \quad 1 \times 10^{-3} \quad 1 \times 10^{-4} \right],$$

$$p_L^\top = \left[1 \times 10^{-6} \quad 1 \times 10^{-6} \quad 1 \times 10^{-6} \quad 1 \times 10^{-5} \quad 1 \times 10^{-6} \right].$$

A estimativa inicial dos parâmetros foi:

$$p_0^\top = \left[1 \times 10^{-5} \quad 1 \times 10^{-5} \quad 1 \times 10^{-5} \quad 1 \times 10^{-4} \quad 1 \times 10^{-5} \right].$$

O tempo médio de cada execução foi cerca de 4 h. São apresentados valores médios do tempo de execução, t , do valor do mínimo global encontrado, J_{glob} , e do número de avaliações da função, AF, com base em 10 execuções do método MCSFilter. Usou-se, para tal, um computador portátil de 2,6 GHz Core i7, com 8 GB de RAM e sistema operativo MacOS El Capitan. Os resultados estão apresentados na Tabela 4.1.

Tabela 4.1: Resultados obtidos com o método MCSFilter ao problema de estimativa de parâmetros do modelo cinético da isomerização do α -pineno.

	Melhor	Pior	Médio
J_{glob}	19,9074	20,1241	19,9798
$p_1 \times 10^5$	5,936	5,904	-
$p_2 \times 10^5$	2,965	2,969	-
$p_3 \times 10^5$	2,067	1,978	-
$p_4 \times 10^4$	2,776	2,835	-
$p_5 \times 10^5$	4,090	4,448	-
t / s	9895	14510	14479
AF	34241	49997	41308

Na Tabela 4.2 apresentam-se os valores da melhor solução encontrada, J_{glob} , a média das soluções encontradas (quando existente), $J_{glob,med}$, o tempo de execução, t , e o número de avaliações da função, AF, da melhor solução encontrada. Os valores médios reportados por Egea et al. (2007) e Rocha et al. (2016) são referentes a 10 execuções do método, tal como no estudo aqui apresentado com o método MCSFilter.

Segundo Averick et al. (1992) a solução deste problema é encontrada facilmente quando o ponto inicial é próximo da solução ótima conhecida. Todavia, sendo um sistema de natureza complexa, verifica-se que à medida que os pontos iniciais se afastam da solução ótima o tempo de execução aumenta e o método, muitas vezes, apenas consegue convergir para soluções locais, não convergindo para a solução global. É necessário ter especial atenção à tolerância do integrador numérico e tentar encontrar a um compromisso entre o tempo de execução e o erro nos valores previstos pelo modelo.

Num estágio inicial do estudo deste problema de otimização, foram considerados pontos iniciais muito próximos do ótimo. O método MCSFilter convergiu para o mínimo global. O valor correspondente da função $J(p)$ é 19.9074, com uma diferença relativa de +0,17%

Tabela 4.2: Resultados obtidos por Egea et al. (2007, 2008), Larrosa (2008) e Rocha et al. (2016) na resolução do problema de otimização de estimativa de parâmetros da isomerização do α -pineno, utilizando os vários métodos de otimização.

<i>Solver</i>	J_{glob}	$J_{glob,med}$	t / s	AF
<i>SSm</i> (Larrosa, 2008)	19,872	19,872	122	9518
<i>SSm</i> (Egea et al., 2007)	19,872	24,747	41	1163
<i>SSm</i> (Rodriguez-Fernandez et al., 2006)	19,87	-	-	-
<i>Global</i> $\times 10^{-4}$ (Egea et al., 2008)	3,1638	3,5225	45	1277
<i>DE</i> $\times 10^{-4}$ (Egea et al., 2008)	3,1951	3,2051	46	1250
<i>SRES</i> $\times 10^{-4}$ (Egea et al., 2008)	3,3858	3,8398	47	1300
<i>DIRECT</i> $\times 10^{-4}$ (Egea et al., 2008)	3,6421	3,6421	45	1053
<i>OQNLP</i> $\times 10^{-4}$ (Egea et al., 2008)	3,1252	3,1252	51	1565
FA (Rocha et al., 2016)	19,8772	25,6777	-	5860

face ao valor obtido por Larrosa (2008) (Tabela 4.2). O valor médio não divergiu muito do melhor tendo-se obtido uma diferença relativa média de +0,54% em relação ao valor apresentado por Larrosa (2008).

Quanto aos valores obtidos para os parâmetros, as diferenças relativas em relação ao valor de Larrosa (2008) foram de +0,10%, +0,17%, +0,83%, +0,95% e de +2,25% para p_1 , p_2 , p_3 , p_4 e p_5 , respetivamente. Estas diferenças devem-se a factores como a aleatoriedade inerente ao método MCSFilter, responsável por gerar o vetor p com que se inicia a procura local, mas também à condição de paragem da procura local, CSFilter.

Foram realizados testes com uma redução do comprimento do passo α (Equação 2.4) para $\alpha_{min} = 1 \times 10^{-7}$. No entanto, o aumento de esforço computacional e de tempo de execução verificados não justifica a melhoria de valores que se poderia obter. Estas execuções foram interrompidas devido ao elevado tempo de execução verificado.

Como foi referido na Secção 1.4, a estratégia de otimização de Egea et al. (2007), *SSm*, envolve um método de procura dispersa acoplado com um método que recorre à informação da derivada da função, o que permite uma convergência mais rápida do método da procura local quando se aproxima do ótimo. No caso de MCSFilter, a convergência em direção ao ótimo é rápida numa fase inicial, tornando-se mais lenta à medida que o comprimento do passo α diminui. A diminuição da condição de paragem da procura local, CSFilter, leva a um aumento do número de avaliações da função objetivo, que se traduzirá inevitavelmente num aumento do tempo de execução.

Comparando os tempos de execução apresentados nas tabelas 4.1 e 4.2, confirma-se que o método MCSFilter apresenta um tempo de execução significativamente superior,

relativamente aos restantes métodos. O valor de AF para a melhor execução do método MCSFilter (Tabela 4.1), comparado com os valores reportados na literatura (última coluna da Tabela 4.2), confirma que o valor correspondente ao método MCSFilter é significativamente superior.

É importante analisar, também, os valores médios da função objetivo calculados, $J_{glob,med}$. Em relação ao ótimo, o método MCSFilter apresenta uma diferença relativa média de +0,54%, enquanto que para o método SSm aplicado por Egea et al. (2007), este valor é de +24,5%. Relativamente ao método FA, aplicado por Rocha et al. (2016), a diferença relativa entre o valor médio da função objetivo reportado e o valor ótimo é de +29,2%. As diferenças relativas médias apresentadas demonstram que a qualidade da solução obtida para os métodos SSm e FA é significativamente inferior quando comparada com a solução apresentada com o método MCSFilter. Observando o valor da pior solução obtida com o método MCSFilter (terceira coluna da Tabela 4.1), pode inferir-se que, apesar do elevado tempo de execução, há convergência em todas as execuções realizadas para valores muito próximos do valor ótimo da função objetivo. Os restantes métodos apresentados na Tabela 4.2 não serão comparados com o MCSFilter, uma vez que o valor da função objetivo reportado está muito longe do valor ótimo.

Foi ainda realizado um único teste, com uma estimativa inicial aleatória

$$p^T = \left[6,05 \times 10^{-5} \quad 3,60 \times 10^{-5} \quad 4,18 \times 10^{-5} \quad 4,79 \times 10^{-4} \quad 9,86 \times 10^{-5} \right],$$

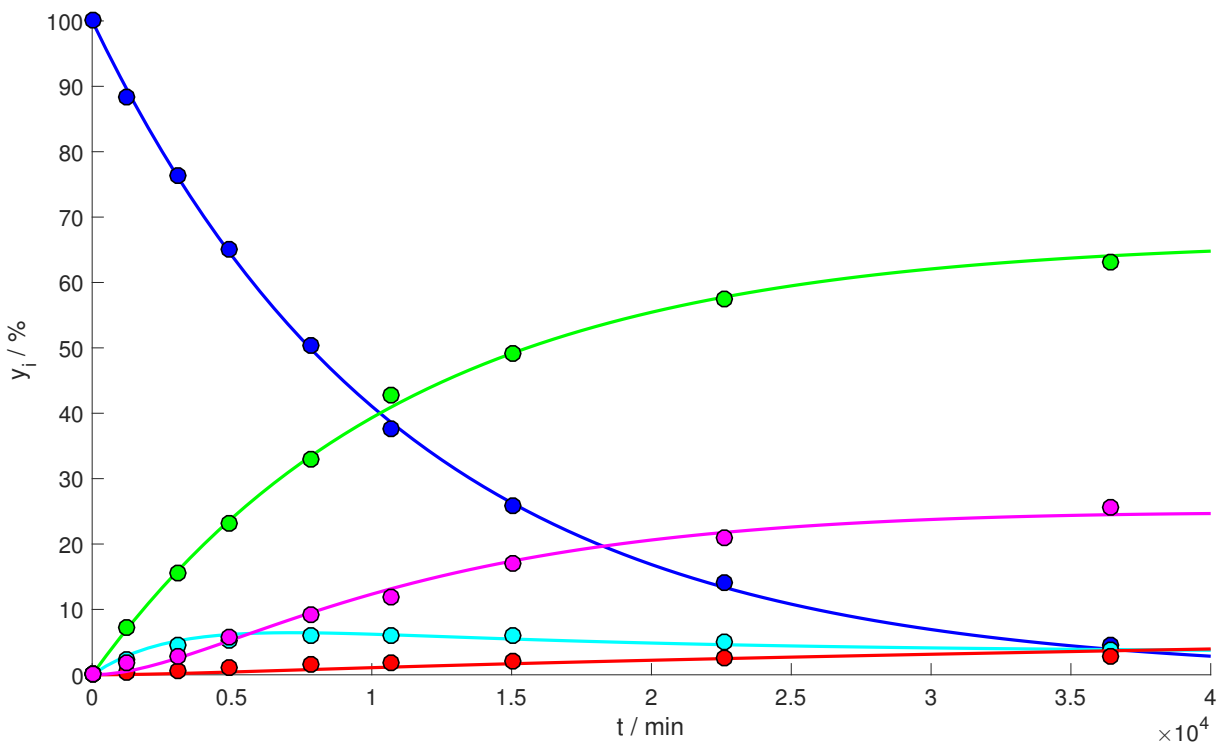


Figura 4.2: Ajuste dos perfis previstos pelo modelo cinético da isomerização do α -pineno aos dados experimentais obtidos por Fuguitt e Hawkins em 1947 (Box e Draper, 1965), utilizando os melhores parâmetros estimados pelo método MCSFilter.

em que se obteve uma solução da função objetivo de $J = 19,88$. Este resultado apresenta uma diferença relativa de $+0,04\%$ em relação ao valor de Larrosa (2008). O tempo de execução foi de cerca de 3,3 h. Esta simulação-teste permite confirmar que uma procura local iniciada com um vetor de variáveis de decisão mais próximos do ótimo atinge mais facilmente a solução (Averick et al., 1992).

Atendendo ao melhor valor obtido da Tabela 4.1, integrou-se o modelo cinético com os parâmetros correspondentes para verificar o ajuste do modelo aos pontos experimentais. O resultado está apresentado na Figura 4.2.

Como é possível observar na Figura 4.2, o ajuste do modelo (linhas) aos dados experimentais (pontos) é satisfatório, tal como também tinha sido verificado por Larrosa (2008). Este resultado confirma a boa qualidade da solução alcançada através do método MCSFilter.

5. Conclusões e Trabalho Futuro

No presente capítulo são apresentadas as principais conclusões resultantes da aplicação do método MCSFilter (Secção 5.1). Na Secção 5.2, são apresentadas perspectivas de trabalho futuro no âmbito da formulação e da aplicação do método MCSFilter.

5.1 Conclusão

Um dos objetivos deste trabalho foi a aplicação de um método *multistart* sem derivadas para otimização de problemas de engenharia. Foram considerados, neste trabalho, dois problemas de estimativa de parâmetros. O primeiro problema refere-se à estimativa dos parâmetros do modelo de uma válvula de controlo com *stiction*, sendo o modelo desta de natureza descontínua. O segundo problema é relativo à determinação dos parâmetros do modelo cinético da isomerização do α -pineno, sendo o modelo constituído por 5 EDO's. O método de otimização global utilizado para a resolução destes problemas de otimização é o método MCSFilter. Trata-se de um método baseado no algoritmo *multistart* acoplado com uma procura local baseada nas direções coordenadas e filtros.

Foram implementadas duas variantes do método MCSFilter — MB e MRA. A variante MB é uma versão básica do método MCSFilter em que a procura local é executada para todos os pontos gerados aleatoriamente. Na variante MRA é introduzido o conceito de região de atração, que evita a execução da procura local em pontos gerados que se encontrem dentro de uma região já definida. Neste caso, assume-se que a procura local do ponto convergiria para o minimizante associado a essa região de atração. O método MCSFilter é uma extensão do método MRA, acoplado de um cálculo probabilístico.

As três variantes do método MCSFilter — MB, MRA e MCSFilter — foram testadas com 10 problemas cujas soluções são bem conhecidas na literatura para demonstrar as vantagens e desvantagens de cada uma delas. Avaliou-se o número médio de minimizantes distintos encontrados, o número médio de avaliações da função, o tempo médio de execução e o número médio de procuras locais efetuadas. Além dos testes realizados com as três variantes do método MCSFilter, os resultados também foram comparados com uma versão de MCSFilter implementada em Matlab[®]. Os resultados desta simulação, permitem concluir que a implementação em Java reduz significativamente os tempos de execução, comparativamente à versão implementada em Matlab[®], observando-se um número de

avaliações da função equivalente nos dois casos. Para estes problemas, concluiu-se também que o método MCSFilter consegue sempre atingir o mínimo global, mesmo nas funções mais complexas, e que as regiões de atração têm um papel fundamental na redução do tempo de execução e do esforço computacional exigido pelo método. Concluiu-se ainda que, das três variantes, o método MCSFilter é o que, de uma forma geral, produz melhores resultados. Por conseguinte, foi o método escolhido para resolver os dois exemplos de otimização da área de engenharia.

A resolução do primeiro exemplo refere-se à estimativa de parâmetros do modelo descontínuo que caracteriza o comportamento de uma válvula de controlo que exhibe o fenómeno de *stiction*. Foi possível concluir que o método MCSFilter determina o ótimo global com maior precisão quando comparado com um método que recorre à informação da derivada de um modelo equivalente, com uma função contínua formulada com recurso a técnicas de suavização (Brásio et al., 2014b). O vetor de parâmetros p correspondente permite um excelente ajuste dos valores previstos pelo modelo aos pontos experimentais, uma vez que os valores dos parâmetros estimados são muito próximos dos valores “reais”. Relativamente ao tempo médio de execução, o método MCSFilter demonstrou ser competitivo em relação a outros métodos de otimização.

Quanto ao problema de estimativa de parâmetros do modelo cinético da isomerização do α -pineno, apesar do elevado tempo médio de execução e número médio de avaliações da função, o método MCSFilter determinou a solução ótimo do problema. Quando comparado com resultados da literatura, o valor da melhor solução obtida com o MCSFilter apresenta uma diferença relativa média de +0,54%, em relação ao valor de Larrosa (2008). Da análise dos valores médios obtidos para a função objetivo com os diferentes métodos (MCSFilter, SSm e FA), concluiu-se que a qualidade da solução obtida com MCSFilter nas 10 execuções é muito superior à dos métodos SSm e FA.

Em suma, a estimativa de parâmetros nos dois exemplos acima descritos foi bem sucedida, como se pode comprovar pelo ajuste dos perfis previstos pelos modelos aos correspondentes perfis experimentais (Figuras 3.2 e 4.2).

5.2 Trabalho Futuro

Na secção anterior, concluiu-se que os resultados obtidos na aplicação do método MCSFilter são bastante promissores na área de estimativa de parâmetros de modelos matemáticos. Para problemas desta natureza (estimativa de parâmetros), apresentam-se as seguintes propostas:

- Relativamente ao problema de *stiction* numa válvula de controlo, implementar outros métodos de otimização global, utilizando o mesmo modelo de Chen, e comparar o desempenho desses métodos com os resultados do método MCSFilter;

- Aplicar o método MCSFilter a problemas da mesma natureza com modelos descontínuos mais complexos e com maior número de variáveis de decisão;
- Aplicar o método MCSFilter ao problema de estimativa de parâmetros do modelo que descreve a isomerização do α -pineno com outro integrador numérico, de modo a reduzir o tempo de execução.

Para um estudo mais aprofundado acerca da robustez do método MCSFilter, este pode ser aplicado em problemas reais de natureza e complexidade diversas.

Bibliografia

- Amador, A., Fernandes, F. P., Santos, L. O., e Romanenko, A. (2016). Application of MCSFilter to estimate stiction control valve parameters. Aceite para publicação em *AIP Conference Proceedings*.
- Ames, W. F. (1962). Canonical forms for nonlinear kinetic differential equations. *Industrial & Engineering Chemistry Fundamentals*, 1(3):214–218.
- Averick, B. M., Carter, R. G., Moré, J. J., e Xue, G. (1992). The minpack-2 test problem collection. Relatório técnico, Mathematics and Computer Science Division, Argonne National Laboratory – Preprint MCS-P153-0692.
- Belegundu, A. D. e Chandrupatla, T. R. (2011). *Optimization Concepts and Applications in Engineering*. Cambridge University Press.
- Box, G. E. P. e Draper, N. R. (1965). The bayesian estimation of common parameters from several responses. *Biometrika*, 52(3-4):355–365.
- Box, G. E. P., Hunter, W. G., MacGregor, J. F., e Erjavec, J. (1973). Some problems associated with the analysis of multiresponse data. *Technometrics*, 15(1):33–51.
- Brásio, A. S. R., Romanenko, A., e Fernandes, N. C. P. (2014a). Modeling, detection and quantification, and compensation of stiction in control loops: The state of the art. *Industrial & Engineering Chemistry Research*, 53(39):15020–15040.
- Brásio, A. S. R., Romanenko, A., e Fernandes, N. C. P. (2014b). Stiction detection and quantification as an application of optimization. In Murgante, B., Misra, S., Rocha, A. M. A. C., Torre, C., Rocha, J. G., Falcão, M. I., Taniar, D., Apduhan, B. O., e Gervasi, O., editors, *Computational Science and Its Applications – ICCSA 2014: 14th International Conference, Guimarães, Portugal, June 30 – July 3, 2014, Proceedings, Part II*, LNCS, pages 169–179. Springer International Publishing, Cham.
- Chen, S., Tan, K. K., e Huang, S. (2008). Two-layer binary tree data-driven model for valve stiction. *Industrial & Engineering Chemistry Research*, 47(8):2842–2848.
- Choudhury, M. A. A. S., Jain, M., e Shah, S. L. (2008a). Stiction - definition, modelling, detection and quantification. *Journal of Process Control*, 18:232–243.

- Choudhury, M. A. A. S., Shah, S. L., e Thornhill, N. F. (2004). Detection and quantification of control valve stiction. In *International Federation of Accountants Symposium on Dynamics and Control of Process Systems*.
- Choudhury, M. A. A. S., Shah, S. L., e Thornhill, N. F. (2008b). *Diagnosis of Process Nonlinearities and Valve Stiction*. Springer-Verlag Berlin Heidelberg.
- Choudhury, M. A. A. S., Shah, S. L., Thornhill, N. F., e Shook, D. S. (2006). Automatic detection and quantification of stiction in control valves. *Control Engineering Practice*, 14(12):1395–1412.
- Choudhury, M. A. A. S., Thornhill, N. F., e Shah, S. L. (2005). Modelling valve stiction. *Control Engineering Practice*, 13(5):641–658.
- Costa, M. F. P., Fernandes, F. P., Fernandes, E. M. G. P., e Rocha, A. M. A. C. (2014). Multiple solutions of mixed variable optimization by multistart hooke and jeeves filter method. *Applied Mathematical Sciences*, 8:2163–2179.
- Csendes, T. (1988). Nonlinear parameter estimation by global optimization - efficiency and reliability. *Acta Cybernetica*, 8(4):361–370.
- Ding, F., Liu, X. P., e Liu, G. (2011). Identification methods for hammerstein nonlinear systems. *Digital Signal Processing*, 21:215–238.
- Dolan, E. D., Moré, J. J., e Munson, T. S. (2004). Benchmarking optimization software with cops 3.0. Relatório técnico, Argonne National Laboratory.
- Durran, D. R. (1990). The third-order adams-bashforth method: An attractive alternative to leapfrog time differencing. *Monthly Weather Review*, 119:702–720.
- Edgar, T. F., Himmelblau, D. M., e Lasdon, L. (1989). *Optimization of chemical processes*. McGraw-Hill Book Company.
- Egea, J. A., Rodríguez-Fernández, M., Banga, J. R., e Martí, R. (2007). Scatter search for chemical and bio-process optimization. *Journal of Global Optimization*, 37(3):481–503.
- Egea, J. A., Schmidt, H., e Banga, J. R. (2008). A new tool for parameter estimation in nonlinear dynamic biological systems using global optimization. In *Poster. 9th International Conference on Systems Biology, ICSB*.
- Farenzena, M. e Trierweiler, J. O. (2012). Valve stiction estimation using global optimization. *Control Engineering Practice*, 20(4):379–385.
- Fernandes, F. A. P. (2014). *Programação Não Linear Inteira Mista e Não Convexa Sem Derivadas*. PhD thesis, Universidade do Minho.

- Fernandes, F. P., Costa, M. F. P., Fernandes, E. M. d. G. P., e Rocha, A. M. A. C. (2013a). Multistart hooke and jeeves filter method for mixed variable optimization. In *Proceedings of the International Conference on Numerical Analysis and Applied Mathematics 2013 (ICNAAM-2013)*, volume 1558, pages 614–617. AIP Publishing.
- Fernandes, F. P., Costa, M. F. P., e Fernandes, E. M. G. P. (2013b). Multilocal programming: A derivative-free filter multistart algorithm. In Murgante, B., Misra, S., Carlini, M., Torre, C. M., Nguyen, H.-Q., Taniar, D., Apduhan, B. O., e Gervasi, O., editors, *Computational Science and Its Applications – ICCSA 2013: 13th International Conference, Ho Chi Minh City, Vietnam, June 24-27, 2013, Proceedings, Part I*, LNCS, pages 333–346. Springer Berlin Heidelberg.
- Fernandes, F. P., Costa, M. F. P., e Fernandes, E. M. G. P. (2014). Branch and bound based coordinate search filter algorithm for nonsmooth nonconvex mixed-integer nonlinear programming problems. In Murgante, B., Misra, S., Rocha, A. M. A. C., Torre, C., Rocha, J. G., Falcão, M. I., Taniar, D., Apduhan, B. O., e Gervasi, O., editors, *Computational Science and Its Applications – ICCSA 2014: 14th International Conference, Guimarães, Portugal, June 30 – July 3, 2014, Proceedings, Part II*, LNCS, pages 140–153. Springer International Publishing, Cham.
- Fernandes, F. P., Costa, M. F. P., Rocha, A. M. A. C., e Fernandes, E. M. G. P. (2016). Improving efficiency of a multistart with interrupted hooke-and-jeeves filter search for solving minlp problems. In Gervasi, O., Murgante, B., Misra, S., Rocha, C. A. M. A., Torre, M. C., Taniar, D., Apduhan, O. B., Stankova, E., e Wang, S., editors, *Computational Science and Its Applications – ICCSA 2016: 16th International Conference, Beijing, China, July 4-7, 2016, Proceedings, Part I*, LNCS, pages 345–358. Springer International Publishing, Cham.
- Jelali, M. (2008). Estimation of valve stiction in control loops using separable least-squares and global search algorithms. *Journal of Process Control*, 18(7):632–642.
- Jones, D. R. (2001). Direct global optimization algorithm. In Floudas, C. A. e Pardalos, P. M., editors, *Encyclopedia of Optimization*, pages 431–440. Springer US, Boston, MA.
- Kano, M., Maruta, H., Kugemoto, H., e Shimizu, K. (2004). Practical model and detection algorithm for valve stiction. In *International Federation of Accountants Symposium on Dynamics and Control of Process Systems*, pages 5–7.
- Kolda, T. G., Lewis, R. M., e Torczon, V. (2003). Optimization by direct search: New perspectives on some classical and modern methods. *Siam Review*, 45(3):385–482.
- Koziel, S. e Michalewicz, Z. (1999). Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evolutionary Computation*, 7(1):19–44.

- Laguna, M. e Martí, R. (2003). *Scatter Search: Methodology and Implementations in C*. Kluwer Academic Publishers, Boston.
- Larrosa, J. A. E. (2008). *New Heuristics for Global Optimization of Complex Bioprocesses*. PhD thesis, University of Vigo.
- Lee, K. H., Ren, Z., e Huang, B. (2008). Novel closed-loop stiction detection and quantification method via system identification. In *2008 International Symposium on Advanced Control of Industrial Processes Conference, Jasper, Canada*, pages 4–7.
- Luenberger, D. G. e Ye, Y. (2008). *Linear and Nonlinear Programming*. Springer, 4th edition.
- Nocedal, J. e Wright, S. J. (1999). *Numerical Optimization*. Springer-Verlag.
- Pereira, A. I., Ferreira, O., Pinho, S. P., e Fernandes, E. M. G. P. (2013). Multilocal programming and applications. In *Handbook of Optimization*, pages 157–186. Springer.
- Rocha, A. M. A., Martins, M. C., Costa, M. F. P., e Fernandes, E. M. (2016). Direct sequential based firefly algorithm for the α -pinene isomerization problem. In Gervasi, O., Murgante, B., Misra, S., Rocha, A. M. A. C., Torre, C. M., Taniar, D., Apduhan, B. O., Stankova, E., e Wang, S., editors, *Computational Science and Its Applications – ICCSA 2016*, volume 9786 of *LNCS*, pages 386–401. Springer Berlin Heidelberg.
- Rodriguez-Fernandez, M., Egea, J. A., e Banga, J. R. (2006). Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. *BioMed Central Bioinformatics*, 7(1):1–18.
- Romano, R. A. e Garcia, C. (2011). Valve friction and nonlinear process model closed-loop identification. *Journal of Process Control*, 21(4):667–677.
- Rossi, M. e Scali, C. (2005). A comparison of techniques for automatic detection of stiction: simulation and application to industrial data. *Journal of Process Control*, 15:505–514.
- Runarsson, T. P. e Yao, X. (2000). Stochastic ranking for constrained evolutionary optimization. *Institute of Electrical and Electronics Engineers Transactions on Evolutionary Computation*, 4(3):284–294.
- Srinivasan, R. e Rengaswamy, R. (2008). Approaches for efficient stiction compensation in process control valves. *Computers & Chemical Engineering*, 32(1):218–229.
- Srinivasan, R., Rengaswamy, R., e Miller, R. (2005a). Control loop performance assessment. 1. a qualitative approach for stiction diagnosis. *Industrial & engineering chemistry research*, 44(17):6708–6718.

- Srinivasan, R., Rengaswamy, R., Narasimhan, S., e Miller, R. (2005b). Control loop performance assessment. 2. hammerstein model approach for stiction diagnosis. *Industrial & engineering chemistry research*, 44(17):6719–6728.
- Storn, R. e Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359.
- Tjoa, I.-B. e Biegler, L. T. (1991). Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. *Industrial & Engineering Chemistry*, 30:376–385.
- Ugray, Z., Lasdon, L., Plummer, J. C., Glover, F., Kelly, J., e Martí, R. (2005). A multistart scatter search heuristic for smooth nlp and minlp problems. In Sharda, R., Voß, S., Rego, C., e Alidaee, B., editors, *Metaheuristic Optimization via Memory and Evolution: Tabu Search and Scatter Search*, pages 25–57. Springer US, Boston, MA.

Anexos

A. Problemas Estudados

Este capítulo está subdividido em duas partes. Na primeira são apresentados os problemas estudados com restrições de limite simples (Secção A.1). Na segunda são apresentados os problemas com restrições de igualdade e de desigualdade (Secção A.2).

A.1 Problemas com Restrições de Limite Simples

Problema A1

(Problema definido no Anexo A de Fernandes (2014), denominado de A.1 - 3 minimizantes)

$$\begin{aligned} \min f(x) &\equiv \left(x_1 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos(x_1) + 10 \\ \text{s.a.} \quad &- 5 \leq x_1 \leq 10 \\ &0 \leq x_2 \leq 15 \end{aligned}$$

Problema A2

(Problema definido no Anexo A de Fernandes (2014), denominado de A.2 - 6 minimizantes)

$$\begin{aligned} \min f(x) &\equiv \left(4 - 2.1 x_1^2 + \frac{x_1^4}{3} \right) x_1^2 + x_1 x_2 - 4 (1 - x_2^2) x_2^2 \\ \text{s.a.} \quad &- 2 \leq x_i \leq 2, \quad i = 1, 2 \end{aligned}$$

Problema A3

(Problema definido no Anexo A de Fernandes (2014), denominado de A.6 - 5 minimizantes)

$$\begin{aligned} \min f(x) &\equiv - \sum_{i=1}^5 \left(\frac{1}{(x - a_i) (x - a_i)' + c_i} \right) \\ \text{s.a.} \quad &0 \leq x_i \leq 10, i = 1, \dots, 4 \\ \text{com} \quad &a = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \end{bmatrix} \quad \text{e} \quad c = \begin{bmatrix} 0,1 \\ 0,2 \\ 0,2 \\ 0,4 \\ 0,4 \end{bmatrix} \end{aligned}$$

Problema A4

(Problema definido no Anexo A de Fernandes (2014), denominado de A.7 - 7 minimizantes)

$$\begin{aligned} \min f(x) &\equiv - \sum_{i=1}^7 \left(\frac{1}{(x - a_i) (x - a_i)' + c_i} \right) \\ \text{s.a.} \quad &0 \leq x_i \leq 10, i = 1, \dots, 4 \\ \text{com} \quad &a = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 3 & 5 & 3 \end{bmatrix} \quad \text{e} \quad c = \begin{bmatrix} 0,1 \\ 0,2 \\ 0,2 \\ 0,4 \\ 0,4 \\ 0,6 \\ 0,3 \end{bmatrix} \end{aligned}$$

Problema A5

(Problema definido no Anexo A de Fernandes (2014), denominado de A.8 - 10 minimizantes)

$$\begin{aligned} \min f(x) &\equiv - \sum_{i=1}^{10} \left(\frac{1}{(x - a_i)(x - a_i)' + c_i} \right) \\ \text{s.a.} \quad &0 \leq x_i \leq 10, \quad i = 1, \dots, 4 \\ \text{com} \quad &a = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 3 & 5 & 3 \\ 8 & 1 & 8 & 1 \\ 6 & 2 & 6 & 2 \\ 7 & 3,6 & 7 & 3,6 \end{bmatrix} \quad \text{e} \quad c = \begin{bmatrix} 0,1 \\ 0,2 \\ 0,2 \\ 0,4 \\ 0,4 \\ 0,6 \\ 0,3 \\ 0,7 \\ 0,5 \\ 0,5 \end{bmatrix} \end{aligned}$$

Problema A6

(Problema definido no Anexo A de Fernandes (2014), denominado de A.12 - 4 minimizantes)

$$\begin{aligned} \min f(x) &\equiv \frac{1}{2} \sum_{i=1}^2 x_i^4 - 16 x_i^2 + 5 x_i \\ \text{s.a.} \quad &-5 \leq x_i \leq 5, \quad i = 1, 2 \end{aligned}$$

Problema A7

(Problema definido no Anexo A de Fernandes (2014), denominado de A.13 - 8 minimizantes)

$$\begin{aligned} \min f(x) &\equiv \frac{1}{2} \sum_{i=1}^2 x_i^4 - 16 x_i^3 + 5 x_i \\ \text{s.a.} \quad &-5 \leq x_i \leq 5, \quad i = 1, \dots, 3 \end{aligned}$$

A.2 Problemas com Restrições de Igualdade e de Desigualdade

Problema A8

(Problema definido no Anexo A de Fernandes (2014), denominado de A.19 - 1 minimizante)

$$\begin{aligned} \min f(x) &\equiv -(\sqrt{n})^n \prod_{i=0}^n x_i \\ \text{s.a.} \quad &\sum_{i=1}^n x_i^2 - 1 = 0 \\ &0 \leq x_i \leq 1, \quad i = 1, 2 \end{aligned}$$

Problema A9

(Problema definido no Anexo A de Fernandes (2014), denominado de A.21 - 2 minimizantes)

$$\begin{aligned} \min f(x) &\equiv -\frac{\sin^3(2\pi x_1) \sin(2\pi x_2)}{x_1^3(x_1 + x_2)} \\ \text{s.a.} \quad &x_1^2 - x_2 + 1 \leq 0 \\ &1 - x_1 + (x_2 - 4)^2 \leq 0 \\ &0 \leq x_i \leq 10, \quad i = 1, 2 \end{aligned}$$

Problema A10

(Problema definido no Anexo A de Fernandes (2014), denominado de A.23 - 2 minimizantes)

$$\begin{aligned} \min f(x) &\equiv x_1^2 + (x_2 - 1)^2 \\ \text{s.a.} \quad &x_2 - x_1^2 = 0 \\ &-1 \leq x_i \leq 1, \quad i = 1, 2 \end{aligned}$$

B. Resultados *Multistart*

Neste capítulo são apresentados os resultados referentes aos problemas estudados no Capítulo 2. Na primeira secção (Secção B.1) encontram-se os resultados referentes aos problemas com restrições de limite simples e na segunda (Secção B.2) encontram-se os resultados relativos aos problemas com restrições de igualdade e de desigualdade.

B.1 Problemas com Restrições de Limites Simples

Os valores da literatura para o valor do ótimo global, f_{lit} , foram retirados de Fernandes (2014) para os problemas de A1 - A5 e de Pereira et al. (2013) para o problema de A6 e A7. Também os valores apresentados na Tabela B.1 para o número de minimizantes distintos reportados na literatura foram consultados em Fernandes (2014).

Tabela B.1: Valores reportados na literatura e usados como referência para a discussão dos resultados dos problemas A1 a A7.

Problema	$f_{glob,lit}$	N_{lit}
A1	0,39789	3
A2	-1,03160	6
A3	-10,1532	5
A4	-10,4029	7
A5	-10,5364	10
A6	-78,3320	4
A7	-117.4985	8

Nas tabelas que se seguem estão apresentados os resultados relativos às execuções dos problemas A1 a A7, com os algoritmos MB, MRA, MCSFilter implementado em Java e MCSFilter implementado em Matlab[®].

Tabela B.2: Resultados de MB para os problemas A1 - A7.

Problema	\bar{f}_{glob}	E_{rel}	$t_{med-B} \times 10^3/s$	$N_{LS,med-B}$	N_{med-B}	AF_{med-B}
A1	0,39789	0,00	8,80	100	3,0	25605
A2	-1,03163	$3,00 \times 10^{-5}$	6,70	100	6,0	19827
A3	-10,1532	0,00	19,50	100	5,0	62937
A4	-10,40292	0,00	20,50	100	7,0	62191
A5	-10,53641	0,00	21,80	100	9,8	61402
A6	-78,33233	0,00	7,60	100	4,0	21191
A7	-117,4985	0,00	14,30	100	8,0	37824

Tabela B.3: Resultados de MRA para os problemas A1 - A7.

Problema	\bar{f}_{glob}	E_{rel}	$t_{med-I} \times 10^3/s$	$N_{LS,med-I}$	N_{med-I}	AF_{med-I}
A1	0,39789	0,00	4,90	53	3,0	13838
A2	-1,03163	$3,00 \times 10^{-5}$	4,10	48	6,0	9698
A3	-10,15320	0,00	11,10	54	4,9	34782
A4	-10,40292	0,00	8,80	38	6,6	24412
A5	-10,53641	0,00	11,70	46	9,0	29060
A6	-78,33233	0,00	3,90	40	4,0	8569
A7	-117,4985	0,00	8,50	47	8,0	18084

Tabela B.4: Resultados de MCSFilter-Java para os problemas A1 - A7.

Problema	\bar{f}_{glob}	E_{rel}	$t_{med-C} \times 10^3/s$	$N_{LS,med-C}$	N_{med-C}	AF_{med-C}
A1	0,39789	0,00	3,30	12	3,0	3100
A2	-1,03163	$3,00 \times 10^{-5}$	3,10	20	5,7	4084
A3	-10,15320	0,00	5,20	17	4,6	10659
A4	-10,40292	0,00	7,80	24	6,6	14823
A5	-10,53641	0,00	10,70	31	9,1	19635
A6	-78,33233	0,00	2,40	15	4,0	3281
A7	-117,4985	0,00	6,00	28	8,0	10795

Além das médias, foram calculados os desvio padrão relativos às 10 execuções realizadas,

para cada uma das variantes do método. Para as três variantes e em todos os problemas, sem exceção, o valor de desvio padrão obtido foi de 0,00.

Tabela B.5: Resultados de MCSFilter-Matlab para os problemas A1 - A7.

Problema	\bar{f}_{glob}	t_{med-O}/s	N_{med-O}	AF_{med-O}
A1	0,39789	0,5065	3,0	3287
A2	-1,03163	0,7743	6,0	4382
A3	-10,15320	1,2876	4,6	11907
A4	-10,40292	1,7674	6,5	15454
A5	-10,53641	2,4527	9,1	20633
A6	-78,33233	0,6205	4,0	3435
A7	-117,49850	1,4858	8,0	11069

B.2 Problemas com Restrições de Igualdade e de Desigualdade

Os valores da literatura para o valor do ótimo, $f_{glob,lit}$, bem como os valores de N_{lit} , foram retirados de Fernandes (2014) para os problemas A8, A9 e A10.

Tabela B.6: Valores da literatura usados como referência para a discussão dos resultados dos problemas A8 a A10.

Problema	f_{lit}	N_{lit}
A8	-1,00000	1
A9	-0,09582	2
A10	0,75000	2

Nas tabelas que se seguem estão apresentados os resultados relativos às execuções dos problemas A8 a A10, com os algoritmos MB, MRA, MCSFilter implementado em Java e MCSFilter implementado em Matlab[®].

Tabela B.7: Resultados de MB para os problemas A8 - A10.

Problema	\bar{f}_{glob}	E_{rel}	$t_{med-B} \times 10^3/s$	$N_{LS,med-B}$	N_{med-B}	AF_{med-B}
A8	-1,00003	$3,00 \times 10^{-5}$	270,80	100	2,4	989215
A9	-0,09583	0,00	31,70	100	4,0	48093
A10	0,74996	$5,00 \times 10^{-5}$	282,60	100	5,9	1067645

Tabela B.8: Resultados de MRA para os problemas A8 - A10.

Problema	\bar{f}_{glob}	E_{rel}	$t_{med-I} \times 10^3/s$	$N_{LS,med-I}$	N_{med-I}	AF_{med-I}
A8	-1,00003	$3,00 \times 10^{-5}$	9,20	1	1,0	22529
A9	-0,09583	0,00	12,90	27	4,8	13567
A10	0,74996	$5,00 \times 10^{-5}$	76,10	22	3,3	298701

Tabela B.9: Resultados de MCSFilter-Java para os problemas A8 - A10.

Problema	\bar{f}_{glob}	E_{rel}	$t_{med-C} * 10^3/s$	$N_{LS,med-C}$	N_{med-C}	AF_{med-C}
A8	-1,00003	$3,00 \times 10^{-5}$	18,00	5	1,0	32136
A9	-0,09583	0,00	9,10	16	4,5	8634
A10	0,74996	$5,00 \times 10^{-5}$	43,50	17	2,6	149910

Além das médias, foram calculados os desvio padrão relativos às 10 execuções realizadas, para cada uma das variantes do método. Para as três variantes e em todos os problemas, sem exceção, o valor de desvio padrão obtido foi de 0,00.

Tabela B.10: Resultados de MCSFilter-Matlab para os problemas A8 - A10.

Problema	\bar{f}_{glob}	t_{med-O}/s	N_{med-O}	AF_{med-O}
A8	-1,00003	7,8133	1,9	21230
A9	-0,09583	2,9925	1,0	3620
A10	0,74996	140,675	2,7	106590