

Volume 41, 2012

Editores

Cassio Machiaveli Oishi

Universidade Estadual Paulista - UNESP
Presidente Prudente, SP, Brasil

Fernando Rodrigo Rafaeli

Universidade Estadual Paulista - UNESP
São José do Rio Preto, SP, Brasil

Rosana Sueli da Motta Jafelice (Editor Chefe)

Universidade Federal de Uberlândia - UFU
Uberlândia, MG, Brasil

Rubens de Figueiredo Camargo

Universidade Estadual Paulista - UNESP
Bauru, SP, Brasil

Sezimária de Fátima P. Saramago

Universidade Federal de Uberlândia - UFU
Uberlândia, MG, Brasil

Vanessa Avansini Botta Pirani (Editor Adjunto)

Universidade Estadual Paulista - UNESP
Presidente Prudente, SP, Brasil



A Sociedade Brasileira de Matemática Aplicada e Computacional - SBMAC publica, desde as primeiras edições do evento, monografias dos cursos que são ministrados nos CNMAC.

Para a comemoração dos 25 anos da SBMAC, que ocorreu durante o XXVI CNMAC em 2003, foi criada a série **Notas em Matemática Aplicada** para publicar as monografias dos minicursos ministrados nos CNMAC, o que permaneceu até o XXXIII CNMAC em 2010.

A partir de 2011, a série passa a publicar, também, livros nas áreas de interesse da SBMAC. Os autores que submeterem textos à série Notas em Matemática Aplicada devem estar cientes de que poderão ser convidados a ministrarem minicursos nos eventos patrocinados pela SBMAC, em especial nos CNMAC, sobre assunto a que se refere o texto.

O livro deve ser preparado em **Latex (compatível com o Miktex versão 2.7)**, as figuras em **eps** e deve ter entre **80 e 150 páginas**. O texto deve ser redigido de forma clara, acompanhado de uma excelente revisão bibliográfica e de **exercícios de verificação de aprendizagem** ao final de cada capítulo.

Veja todos os títulos publicados nesta série na página
<http://www.sbmac.org.br/notas.php>

**TÉCNICAS DE INTELIGÊNCIA
COMPUTACIONAL INSPIRADAS NA
NATUREZA: APLICAÇÃO EM PROBLEMAS
INVERSOS EM TRANSFERÊNCIA
RADIATIVA**

2ª edição

Antônio José da Silva Neto
ajsneto@iprj.uerj.br

Instituto Politécnico
Universidade do Estado do Rio de Janeiro

José Carlos Becceneri
becce@lac.inpe.br

Laboratório Associado de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais



Sociedade Brasileira de Matemática Aplicada e Computacional

São Carlos - SP, Brasil
2012

Coordenação Editorial: Elbert Einstein Nehrer Macau

Coordenação Editorial da Série: Rosana Sueli da Motta Jafelice

Editora: SBMAC

Capa: Matheus Botossi Trindade

Patrocínio: SBMAC

Copyright ©2012 by Antônio José da Silva Neto e José Carlos Becce-
neri.

Direitos reservados, 2012 pela SBMAC. A publicação nesta série não
impede o autor de publicar parte ou a totalidade da obra por outra
editora, em qualquer meio, desde que faça citação à edição original.

Catálogo elaborado pela Biblioteca do IBILCE/UNESP
Bibliotecária: Maria Luiza Fernandes Jardim Froner

Silva Neto, Antônio J.

Técnicas de Inteligência Computacional Inspiradas na Natureza:
Aplicações em Problemas Inversos em Transferência Radiativa
- São Carlos, SP : SBMAC, 2012, 148 p., 20.5 cm - (Notas em
Matemática Aplicada; v. 41) - 2ª edição

e-ISBN 978-85-8215-002-3

1. Inteligência Computacional 2. Bio-inspirado 3. Problemas
Inversos

I. Silva Neto, Antônio J. II. Becceneri, José C. III. Luz, Eduardo
F. P. IV. Chalhoub, Ezzat S. V. de Sousa, Fabiano L. VI. Soeiro,
Francisco J. C. P. VII. Campos Velho, Haroldo F. VIII. Biondi Neto,
Luiz IX. Stephany, Stephan X. Sacco, Wagner F. XI. Silva, José D. S.
XII. Lobato, Fran S. XIII. Steffen Jr., Valder XIV. Título.
XV. Série.

CDD - 51

Esta edição em formato e-book é uma edição revisada do livro original
do mesmo título publicado em 2009 nesta mesma série pela SBMAC.

Para Gilsineida, Lucas e Luísa
Antônio José da Silva Neto

Para Rossana, Leandro e Amanda
José Carlos Becceneri

Agradecimentos

Os autores deste livro agradecem o empenho do Dr. Eduardo P. F. Luz, recém doutorado pelo INPE - Instituto Nacional de Pesquisas Espaciais, que em um período relativamente curto, mas com um investimento concentrado de tempo, foi capaz de preparar a versão final de todo o texto em LaTeX, assim como a colaboração da Dra. Nancy Isabel Alvarez Acevedo, do Instituto Politécnico da Universidade do Estado do Rio de Janeiro, na digitação dos Capítulos 2 e 3.

Agradecem também a Comissão Científica do XXXII Congresso Nacional de Matemática Aplicada e Computacional pela aprovação do mini-curso homônimo, que levou à preparação deste material, e a Profa. Rosana Jafelice da Universidade Federal de Uberlândia que conduziu o processo que levou à publicação desta 2a. edição do livro.

Finalizando, agradecem o apoio financeiro dos seguintes órgãos de fomento que têm permitido o desenvolvimento das atividades de pesquisa e formação de recursos humanos nos níveis de graduação e de pós-graduação envolvendo três instituições de pesquisa e ensino do país (UERJ - Universidade do Estado do Rio de Janeiro, INPE - Instituto Nacional de Pesquisas Espaciais, e UFU- Universidade Federal de Uberlândia):

CAPES Fundação Coordenação de Aperfeiçoamento de Pessoal de
Nível Superior

CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico

FAPERJ Fundação Carlos Chagas Filho de Amparo à Pesquisa do
Estado do Rio de Janeiro

FAPESP Fundação de Amparo à Pesquisa do Estado de São Paulo

Conteúdo

| | |
|--|-----------|
| Prefácio | 13 |
| 1 Introdução | 15 |
| 2 Transferência Radiativa | 19 |
| 2.1 Introdução | 19 |
| 2.2 Formulação Matemática do Problema de Transferência Radiativa | 20 |
| 2.2.1 Meio participante homogêneo unidimensional . . | 20 |
| 2.2.2 Meio participante heterogêneo unidimensional em duas camadas | 24 |
| 2.3 Considerações Finais | 26 |
| 3 Problemas Inversos em Transferência Radiativa - Uma Formulação Implícita | 27 |
| 3.1 O Que é um Problema Inverso? | 27 |
| 3.2 Uma Formulação Implícita para o Problema Inverso . . | 29 |
| 3.3 Dados Experimentais | 33 |
| 3.4 Solução do Problema Inverso com o Método de Levenberg- Marquardt (LM) | 34 |
| 3.5 Considerações Finais | 37 |
| 4 Fundamentos de Otimização e Inteligência Artificial | 39 |
| 4.1 Conceitos Básicos na Área de Otimização | 39 |
| 4.1.1 Heurísticas e Meta-heurísticas | 39 |
| 4.2 Inteligência Artificial | 42 |

| | | |
|----------|--|-----------|
| 4.2.1 | Conexionista | 43 |
| 4.2.2 | Simbólica | 45 |
| 4.3 | Considerações Finais | 47 |
| 5 | Recozimento Simulado (<i>Simulated Annealing</i>) | 49 |
| 5.1 | Motivação e Histórico do Método | 49 |
| 5.2 | Descrição do Algoritmo | 51 |
| 5.3 | Aplicação ao Problema Inverso de Transferência Radi- ativa | 55 |
| 5.4 | Considerações Finais | 58 |
| 6 | Algoritmos Genéticos (<i>Genetic Algorithms</i>) | 59 |
| 6.1 | Motivação e Histórico do Método | 59 |
| 6.2 | Descrição do Algoritmo | 60 |
| 6.2.1 | Seleção | 62 |
| 6.2.2 | Cruzamento | 63 |
| 6.2.3 | Mutação | 63 |
| 6.2.4 | Determinação do tamanho da cadeia binária . . | 64 |
| 6.2.5 | Algoritmo | 64 |
| 6.3 | Aplicação ao Problema Inverso de Transferência Radi- ativa | 65 |
| 6.4 | Considerações finais | 67 |
| 7 | Otimização por Colônia de Formigas (<i>Ant Colony Op- timization</i>) | 69 |
| 7.1 | Motivação e Histórico do Método | 69 |
| 7.2 | Descrição do Algoritmo | 71 |
| 7.3 | Aplicação ao Problema Inverso de Transferência Radi- ativa | 73 |
| 7.3.1 | Problema inverso adotado como exemplo | 73 |
| 7.3.2 | Aplicação do ACO ao problema inverso escolhido | 74 |
| 7.3.3 | Resultados numéricos da aplicação do ACO ao exemplo de problema inverso | 79 |
| 7.4 | Considerações Finais | 82 |

| | | |
|-----------|---|------------|
| 8 | Otimização Extrema Generalizada (<i>Generalized Extremal Optimization</i>) | 83 |
| 8.1 | Motivação e Histórico do Método | 83 |
| 8.2 | Descrição do Algoritmo | 85 |
| 8.2.1 | O GEO canônico | 85 |
| 8.2.2 | Algumas Considerações Sobre Novas Versões do GEO | 87 |
| 8.3 | Aplicação ao Problema Inverso de Transferência Radiativa | 88 |
| 8.4 | Considerações Finais | 91 |
| 9 | Algoritmo de Colisão de Partículas (<i>Particle Collision Algorithm</i>) | 93 |
| 9.1 | Motivação e Histórico do Método | 93 |
| 9.2 | Descrição do Algoritmo | 95 |
| 9.2.1 | A versão canônica | 95 |
| 9.2.2 | Algoritmo de colisão de múltiplas partículas | 99 |
| 9.3 | Aplicação ao Problema Inverso de Transferência Radiativa | 100 |
| 9.3.1 | Meio homogêneo unidimensional | 101 |
| 9.3.2 | Meio heterogêneo unidimensional em duas camadas | 103 |
| 9.4 | Considerações finais | 105 |
| 10 | Evolução Diferencial (<i>Differential Evolution</i>) | 107 |
| 10.1 | Motivação e Histórico do Método | 107 |
| 10.2 | Descrição do Algoritmo | 109 |
| 10.2.1 | A Inicialização do Algoritmo de DE | 109 |
| 10.2.2 | O Operador de Mutação | 110 |
| 10.2.3 | O Operador de Cruzamento | 111 |
| 10.2.4 | O Operador de Seleção | 111 |
| 10.2.5 | O Critério de Parada | 112 |
| 10.3 | Aplicação ao Problema Inverso de Transferência Radiativa | 113 |
| 10.3.1 | Meio homogêneo unidimensional | 114 |

| | |
|---|------------|
| 10.3.2 Meio heterogêneo unidimensional em duas ca- madas | 116 |
| 10.4 Considerações Finais | 117 |
| 11 Projetos de Aplicação | 119 |
| 12 Considerações Finais | 125 |
| Bibliografia | 127 |

Prefácio

Já há algum tempo os autores dos capítulos deste livro vêm trabalhando em conjunto buscando a aplicação de métodos estocásticos, em particular heurísticas bio-inspiradas, na solução de problemas inversos, principalmente aqueles envolvendo o fenômeno de transferência radiativa.

Devido à relevância das aplicações destes problemas, por exemplo em sensoriamento remoto, modelos climáticos e ambientais, e ótica hidrológica, bem como à atualidade das técnicas de Inteligência Computacional inspiradas na natureza, decidiram, com base em suas experiências e nos trabalhos por eles publicados em diversos eventos científicos e periódicos nacionais e internacionais, preparar este material. Este cumpre alguns propósitos centrais: apresentar uma descrição sucinta do problema de transferência radiativa e das heurísticas bio-inspiradas, inclusive com a apresentação dos algoritmos destas, e também apresentar os resultados obtidos com a utilização destes métodos, bem como de suas hibridizações com métodos estocásticos e métodos determinísticos.

Os problemas inversos podem ser formulados e resolvidos de diversas maneiras, sendo muito frequente a utilização de problemas de otimização nos quais se busca, por exemplo, a minimização do funcional construído com a soma dos resíduos quadrados entre os valores calculados e os valores medidos para uma grandeza observável. No caso dos problemas envolvendo transferência radiativa esta grandeza pode ser a intensidade da radiação com dependência angular e/ou espectral.

Obviamente existe um conjunto muito maior de métodos publica-

dos na vasta literatura recente sobre o assunto, mas os seis métodos apresentados neste livro foram implementados pelos autores, e sempre considerando como problemas modelo a transferência radiativa em meios homogêneos unidimensionais e meios heterogêneos unidimensionais multi-camadas.

O nível exigido de cálculo e computação é acessível tanto para alunos que estejam em estágios avançados de seus cursos de graduação em Matemática, Computação ou Engenharia, bem como para alunos de pós-graduação nestas áreas. Pesquisadores que também queiram ter um primeiro contato com os assuntos mencionados acima poderão encontrar um material acessível e que contém um bom número de referências a outros trabalhos que podem apresentar um maior nível de detalhe em cada um dos assuntos tratados.

Antônio J. Silva Neto
Nova Friburgo, maio de 2012.

José Carlos Becceneri
São José dos Campos, maio de 2012.

Capítulo 1

Introdução

Antônio José da Silva Neto
José Carlos Becceneri

Devido às relevantes aplicações em diversas áreas da atividade humana, um número crescente de pesquisadores tem se dedicado à análise e solução de problemas inversos. Na área da computação, cada vez mais os pesquisadores voltam um olhar para a natureza na tentativa de apreender como solucionar os problemas reais, entre os quais os problemas inversos.

Esse olhar para a natureza se justifica pela simples razão que ela tem evoluído através de milhões de anos, dando soluções otimizadas a todos os problemas naturais que ocorrem. Então, por que não aprender com ela? É justamente isso que uma parte da Inteligência Computacional, a que desenvolve os algoritmos inspirados na natureza, os chamados algoritmos bio-inspirados, tenta fazer: desenvolver técnicas computacionais cujos modelos tentam imitar comportamentos de seres encontrados na natureza, tais como, formigas, pássaros, abelhas e vaga-lumes, entre outros.

Os autores deste livro têm apresentado e publicado diversos trabalhos em congressos e periódicos nacionais e internacionais, e têm tido a oportunidade de acompanhar o significativo crescimento do interesse nestas áreas de pesquisa.

Neste livro é dado enfoque na formulação e na solução de problemas

inversos empregando técnicas de inteligência computacional inspiradas na natureza. Serão então usados como exemplos didáticos tanto na formulação quanto na solução dos problemas inversos, problemas associados à transferência radiativa em meios participantes, i.e. meios absorvedores, espalhadores e emissores. Estes problemas estão relacionados a diversas aplicações relevantes tais como tomografia, óptica hidrológica, e modelos climáticos.

Os principais objetivos deste livro são:

(i) fornecer de forma sucinta e clara os fundamentos matemáticos envolvidos na formulação e na solução de problemas diretos e inversos, tendo por base o fenômeno de transferência radiativa, i.e. interação de radiação com meios participantes (emissores, absorvedores e espalhadores); (ii) descrever algumas técnicas de inteligência computacional inspiradas na natureza; e (iii) apresentar a aplicação das técnicas de inteligência computacional para a solução de problemas inversos.

A seguir, apresenta-se a estrutura deste livro.

No Capítulo 2, apresenta-se o problema de transferência radiativa que será utilizado em todos os capítulos de 5 a 10, bem como a sua importância em diversos problemas atuais.

No Capítulo 3, explica-se o que são problemas inversos e porque são tão estudados atualmente.

O Capítulo 4 discute os conceitos de Inteligência Computacional e Inteligência Artificial, apresenta brevemente cada uma das técnicas que serão estudadas nos demais capítulos e apresenta os conceitos mais comumente utilizados na área de otimização.

Nos capítulos de 5 a 10 são apresentadas as metaheurísticas que serão estudadas neste livro. Todos esses possuem 4 seções: uma explicação da motivação do método, a explicação do algoritmo computacional, sua aplicação ao problema de transferência radiativa e considerações finais.

O Capítulo 5 apresenta a metaheurística Recozimento Simulado, o Capítulo 6 os Algoritmos Genéticos, o Capítulo 7 a Colônia de Formigas, o Capítulo 8 a Otimização Extrema Generalizada, o Capítulo 9 a Colisão de Partículas, e o Capítulo 10 o Algoritmo de Evolução Diferencial.

O Capítulo 11 apresenta uma série de projetos para serem resolvidos como exercícios.

O Capítulo 12 apresenta as considerações finais dos editores deste livro, baseado na experiência que obtiveram durante a edição do mesmo.

Finaliza-se o livro com a Bibliografia e um pequeno índice remissivo.

Capítulo 2

Transferência Radiativa

Antônio José da Silva Neto
Ezzat Selim Chalhoub
Haroldo Fraga de Campos Velho

2.1 Introdução

A formulação e a solução de problemas diretos e inversos em transferência radiativa estão diretamente relacionadas a um grande número de áreas de elevado interesse científico e tecnológico, tais como tomografia [27], sensoriamento remoto e ciências ambientais [171], e a estimativa de propriedades radiativas [86, 61, 163], entre muitas outras.

Recentemente tem sido observado um interesse crescente nos métodos estocásticos de Monte Carlo para a solução do problema direto [12], bem como em variantes do Método de Ordenadas Discretas [31, 32], que foi inicialmente proposto por Wick [178] e Chandrasekhar [35].

Alguns pesquisadores têm também realizado comparações entre diferentes estratégias visando identificar métodos precisos e rápidos, do ponto de vista computacional, para serem usados tanto nos problemas diretos quanto nos problemas inversos em transferência radiativa [33, 34].

2.2 Formulação Matemática do Problema de Transferência Radiativa

Neste livro é usado o problema inverso de transferência radiativa em meios participantes, i.e. meios absorvedores, espalhadores e emissores, como o problema base para o emprego de técnicas de inteligência computacional para a estimativa de propriedades radiativas, com o problema inverso sendo formulado como um problema de otimização.

Neste capítulo é apresentado o problema direto de transferência radiativa em um meio homogêneo plano-paralelo, e também em um meio heterogêneo formado por duas camadas.

2.2.1 Meio participante homogêneo unidimensional

Na Figura 2.1 é representado um meio participante, isto é absorvedor, espalhador e emissor, unidimensional de espessura L , com paredes refletoras difusas, sujeito à incidência de radiação externa com intensidades f_1 e f_2 .

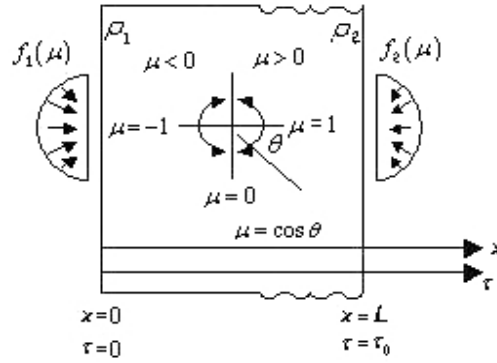


Figura 2.1: Representação esquemática de um meio participante unidimensional com superfícies internas refletoras difusas e submetido à incidência de radiação externa de intensidades f_1 e f_2 .

A formulação matemática do problema de transferência radiativa no meio, i.e. interação da radiação com o meio participante, em regime permanente, para um dado comprimento de onda (a dependência

é omitida para simplificar a notação), é homogêneo, ou seja as propriedades radiativas são constantes, o espalhamento é isotrópico, e com simetria azimutal, é dada pela versão linear da equação de Boltzmann [128],

$$\mu \frac{\partial I(x, \mu)}{\partial x} + \beta I(x, \mu) = \frac{\sigma_s}{2} \int_{-1}^1 I(x, \mu') d\mu' + S(x, \mu);$$

$$0 < x < L, -1 \leq \mu \leq 1 \quad (2.2.1a)$$

$$I(0, \mu) = f_1(\mu) + 2\rho_1 \int_0^1 I(0, -\mu') \mu' d\mu'; \quad \mu > 0 \quad (2.2.1b)$$

$$I(L, \mu) = f_2(\mu) + 2\rho_2 \int_0^1 I(L, \mu') \mu' d\mu'; \quad \mu < 0 \quad (2.2.1c)$$

onde I representa a intensidade da radiação, x é a coordenada espacial, μ o cosseno do ângulo polar θ (vide Figura 2.1), σ_s o coeficiente de espalhamento, S uma fonte interna de radiação, e ρ_1 e ρ_2 as refletividades difusas nas superfícies internas do meio, respectivamente em $x = 0$ e $x = L$, e β é o coeficiente de extinção total

$$\beta = \kappa_a + \sigma_s \quad (2.2.2)$$

onde κ_a é o coeficiente de absorção.

Apenas para citar um exemplo, na Tabela 2.1 são fornecidos os valores dos coeficientes de absorção e de espalhamento para a água do mar.

| Tipo | $\kappa_a(m^{-1})$ | $\sigma_s(m^{-1})$ |
|---|--------------------|--------------------|
| água do mar pura (pure sea water) | 0,0405 | 0,0025 |
| oceano longe da costa (clear ocean) | 0,114 | 0,037 |
| oceano junto à costa (coastal ocean) | 0,179 | 0,219 |
| água turva (turbid water) | 0,366 | 1,824 |

Tabela 2.1: Coeficientes de absorção κ_a e de espalhamento σ_s para a água do mar, [120].

Quando a geometria, as propriedades radiativas e as condições de contorno são conhecidas, o problema de transferência radiativa formulado com as equações (2.2.1) pode ser resolvido fornecendo os valores calculados da intensidade da radiação, $I(x, \mu)$, em todo o domínio espacial, $0 \leq x \leq L$, e angular, $-1 \leq \mu \leq 1$. Este é o problema direto de transferência radiativa em um meio homogêneo unidimensional.

Em problemas envolvendo a transferência radiativa é usual o emprego da variável ótica, τ , definida como

$$d\tau = \beta dx \Rightarrow \tau = \int_0^x \beta dx' \quad (2.2.3)$$

Considerando a integração em todo o domínio espacial obtém-se a espessura ótica do meio,

$$\tau_0 = \int_0^L \beta dx' \quad (2.2.4)$$

que para um meio homogêneo corresponde a

$$\tau_0 = \beta L \quad (2.2.5)$$

Dividindo a equação (2.2.1a) pelo coeficiente de extinção total β definido na equação (2.2.2) e empregando as definições da variável ótica τ e da espessura ótica τ_0 , nas equações (2.2.3) e (2.2.4), respectivamente, bem como desprezando o termo de fonte interna $S(x, \mu)$, o problema de transferência radiativa em um meio participante absorvedor e espalhador é formulado matematicamente por [128],

$$\mu \frac{\partial I(\tau, \mu)}{\partial \tau} + I(\tau, \mu) = \frac{\omega}{2} \int_{-1}^1 I(\tau, \mu') d\mu'; \quad 0 < \tau < \tau_0, -1 \leq \mu \leq 1 \quad (2.2.6a)$$

$$I(0, \mu) = f_1(\mu) + 2\rho_1 \int_0^1 I(0, -\mu') \mu' d\mu'; \quad \mu > 0 \quad (2.2.6b)$$

$$I(\tau_0, \mu) = f_2(\mu) + 2\rho_2 \int_0^1 I(\tau_0, \mu') \mu' d\mu'; \quad \mu < 0 \quad (2.2.6c)$$

onde ω é o albedo de espalhamento simples, que corresponde à razão entre o coeficiente de espalhamento e o coeficiente de extinção total

$$\omega = \frac{\sigma_s}{\beta} = \frac{\sigma_s}{\kappa_a + \sigma_s} \quad (2.2.7)$$

e os outros símbolos nas equações (2.2.6) já foram definidos anteriormente.

Apenas para citar um exemplo, na Tabela 2.2 são fornecidos os valores do coeficiente de extinção total, da espessura ótica e do albedo de espalhamento simples considerando uma lâmina de água do mar com 5 m de espessura.

| Tipo | $\beta = \kappa_a + \sigma_s (m^{-1})$ | $\tau_0 = \beta L$ | $\omega = \frac{\sigma_s}{\beta} = \frac{\sigma_s}{\kappa_a + \sigma_s}$ |
|---|--|--------------------|--|
| água do mar pura (pure sea water) | 0,043 | 0,215 | 0,058 |
| oceano longe da costa (clear ocean) | 0,151 | 0,755 | 0,245 |
| oceano junto à costa (coastal ocean) | 0,398 | 1,990 | 0,550 |
| água turva (turbid water) | 2,190 | 10,95 | 0,833 |

Tabela 2.2: Coeficiente de extinção total β , espessura ótica τ_0 e albedo de espalhamento simples ω para a água do mar. Valores calculados a partir da Tabela 2.1 para uma lâmina de água com $L = 5$ m de espessura.

Para citar um segundo exemplo, na Tabela 2.3 são fornecidos os valores do coeficiente de extinção β e da espessura ótica para nuvens cirrus. Os valores apresentados foram obtidos por Das et al. [47] na Universidade Central Nacional de Taiwan.

Nos problemas tratados neste livro, considera-se que as fontes externas de radiação são isotrópicas, e, portanto,

$$f_1(\mu) = A_1 = \text{constante}, \mu > 0 \quad (2.2.8a)$$

$$f_2(\mu) = A_2 = \text{constante}, \mu < 0 \quad (2.2.8b)$$

| Tipo de nuvem cirrus | Propriedade | Média anual |
|---|------------------------------|--|
| Sub-visível ($\tau_0 < 0,03$) (<i>sub-visible</i>) | $\beta(km^{-1})$ τ_0 | $0,013 \pm 0,008$ $0,012 \pm 0,008$ |
| Fina ($0,03 < \tau_0 < 0,3$) (<i>thin</i>) | $\beta(km^{-1})$ τ_0 | $0,118 \pm 0,15$ $0,114 \pm 0,071$ |
| Espessa ($\tau_0 > 0,3$) (<i>dense</i>) | $\beta(km^{-1})$ τ_0 | $0,42 \pm 0,33$ $0,67 \pm 0,4$ |

Tabela 2.3: Propriedades médias de nuvens cirrus obtidas na Universidade Central Nacional de Taiwan no período de 1999 a 2006 [47]. β é o coeficiente de extinção e τ_0 a espessura ótica (*optical depth*)

2.2.2 Meio participante heterogêneo unidimensional em duas camadas

Há um elevado interesse no tratamento do problema de transferência radiativa em meios multi-camadas [156], devido a relevantes aplicações em diferentes áreas tais como modelos climáticos regionais e globais [71], pesquisa sobre o Sistema Solar [76], sensoriamento remoto da Terra [171] e estudos de nuvens [18].

Na Figura 2.2 é representado um meio participante, isto é absorvedor, espalhador e emissor, unidimensional, composto por duas camadas de espessura L_1 e L_2 , com paredes internas refletoras difusas, sujeito à incidência de radiação externa com intensidades f_1 e f_2 .

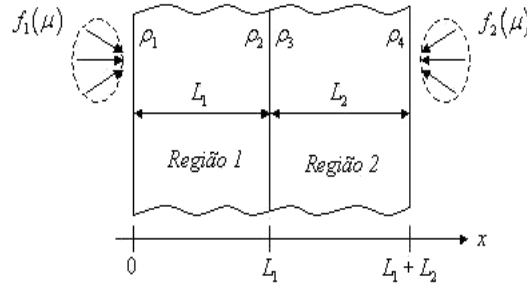


Figura 2.2: Representação esquemática de um meio participante heterogêneo unidimensional composto por duas camadas com superfícies internas refletoras difusas e submetido à incidência de radiação externa de intensidades f_1 e f_2 .

A formulação matemática do problema de transferência radiativa em regime permanente no meio heterogêneo unidimensional composto por duas camadas cinzas, ou seja as propriedades radiativas não dependem do comprimento de onda, homogêneas, ou seja as propriedades radiativas são constantes em cada camada, com espalhamento isotrópico, e simetria azimutal, é dada pela versão linear da equação de Boltzmann [156],

Região 1

$$\mu \frac{\partial I_1(x, \mu)}{\partial x} + \beta_1 I_1(x, \mu) = \frac{\sigma_{s1}}{2} \int_{-1}^1 I_1(x, \mu') d\mu';$$

$$0 < x < L_1, -1 \leq \mu \leq 1 \quad (2.2.9a)$$

$$I_1(0, \mu) = f_1(\mu) + 2\rho_1 \int_0^1 I_1(0, -\mu') \mu' d\mu'; \mu > 0 \quad (2.2.9b)$$

$$I_1(L_1, \mu) = 2(1 - \rho_3) \int_0^1 I_2(L_1, -\mu') \mu' d\mu' + 2\rho_2 \int_0^1 I_1(L_1, \mu') \mu' d\mu';$$

$$\mu < 0 \quad (2.2.9c)$$

Região 2

$$\mu \frac{\partial I_2(x, \mu)}{\partial x} + \beta_2 I_2(x, \mu) = \frac{\sigma_{s2}}{2} \int_{-1}^1 I_2(x, \mu') d\mu';$$

$$L_1 < x < L_1 + L_2, -1 \leq \mu \leq 1 \quad (2.2.10a)$$

$$I_2(L_1, \mu) = 2(1 - \rho_2) \int_0^1 I_1(L_1, \mu') \mu' d\mu' + 2\rho_3 \int_0^1 I_2(L_1, -\mu') \mu' d\mu';$$

$$\mu > 0 \quad (2.2.10b)$$

$$I_2(L_1 + L_2, \mu) = f_2(\mu) + 2\rho_4 \int_0^1 I_2(L_1 + L_2, \mu') \mu' d\mu'; \mu < 0 \quad (2.2.10c)$$

onde I_i representa a intensidade da radiação na camada i , com $i = 1$ ou 2 , β_i é o coeficiente de extinção total,

$$\beta_i = \kappa_{a_i} + \sigma_{s_i} \quad (2.2.11)$$

κ_{a_i} é o coeficiente de absorção, σ_{s_i} é o coeficiente de espalhamento, e

ρ_j são as refletividades difusas, com $j = 1, \dots, 4$. Os outros símbolos matemáticos foram definidos na subseção anterior.

Nas equações (2.2.9c) e (2.2.10b) foi considerado que tanto a reflexão quanto a transmissão são difusas na interface entre as duas camadas.

Quando a geometria, as propriedades radiativas e as condições de contorno são conhecidas, o problema de transferência radiativa formulado com as equações (2.2.9) e (2.2.10) pode ser resolvido fornecendo os valores calculados da intensidade da radiação $I_1(x, \mu)$ para $0 \leq x \leq L_1$, e $I_2(x, \mu)$ para $L_1 \leq x \leq L_1 + L_2$, em todo o domínio angular, $-1 \leq \mu \leq 1$. Este é o problema direto de transferência radiativa em um meio heterogêneo unidimensional formado por duas camadas homogêneas.

2.3 Considerações Finais

O albedo de espalhamento simples ω e as refletividades difusas ρ_j , com $j = 1$ ou 2 para o meio homogêneo unidimensional, e $j = 1, \dots, 4$ para o meio heterogêneo unidimensional em duas camadas, são grandezas cujos valores estão no intervalo $[0, 1]$. A espessura ótica é uma grandeza positiva, não possuindo um limite superior. Porém, para a solução dos problemas inversos com os métodos descritos neste livro é necessário definir uma região de busca para os valores das incógnitas, sendo então escolhido um valor máximo $\bar{\tau}_0$ usualmente entre 2 e 10, exceto quando indicado de outra forma. Em resumo, tem-se

$$0 \leq \omega \leq 1 \quad (2.3.12)$$

$$0 \leq \rho_j \leq 1 \quad j = 1 \text{ ou } 2, \quad \text{ou } j = 1, \dots, 4 \quad (2.3.13)$$

$$0 \leq \tau_0 \leq \bar{\tau}_0 \quad (2.3.14)$$

onde $\bar{\tau}_0$ é um limite superior artificial escolhido de acordo com o problema que se deseja resolver. Este mesmo procedimento empregado para a espessura ótica é adotado para as intensidades das fontes externas, A_1 e A_2 , bem como para os coeficientes de absorção e de espalhamento, κ_a e σ_s , no meio heterogêneo unidimensional de duas camadas.

Capítulo 3

Problemas Inversos em Transferência Radiativa - Uma Formulação Implícita

*Antônio José da Silva Neto
Haroldo Fraga de Campos Velho*

3.1 O Que é um Problema Inverso?

Em primeiro lugar vamos definir o que é um problema direto. Considere o problema de transferência radiativa descrito no Capítulo 2, representado esquematicamente na Figura 2.1 para um meio participante homogêneo unidimensional e na Figura 2.2 para um meio participante heterogêneo unidimensional, composto por duas camadas.

A radiação externa incidente nas superfícies de contorno entra no meio e interage com o mesmo. Para este processo, que envolve os fenômenos de absorção e de espalhamento, dispõe-se de um modelo que é descrito matematicamente pela versão linear da equação de Boltzmann. Nos dois casos aqui considerados, i.e. meio homogêneo unidimensional e meio heterogêneo unidimensional em duas camadas, tem-se as formulações matemáticas descritas respectivamente pelas equações (2.2.1) ou (2.2.6), e pelas equações (2.2.9, 2.2.10).

Quando a geometria, as propriedades radiativas e as condições de contorno são conhecidas, o problema de transferência radiativa pode ser resolvido fornecendo os valores calculados da intensidade da radiação. Este é o Problema Direto (PD).

Conforme a representação feita na Figura 3.1(a), a partir do conhecimento dos fenômenos envolvidos em um determinado processo, sendo aqui considerado o problema de transferência radiativa, e do modelo matemático usado para descrevê-lo, incluindo a geometria do meio sob análise, e de uma causa conhecida, por exemplo a intensidade da radiação incidente nas superfícies de contorno no meio participante, pode ser calculado o efeito, ou seja a intensidade da radiação (radiância) em qualquer ponto do meio e em qualquer direção angular. A incógnita do problema direto de transferência radiativa, i.e. o efeito, é, portanto, a intensidade da radiação.

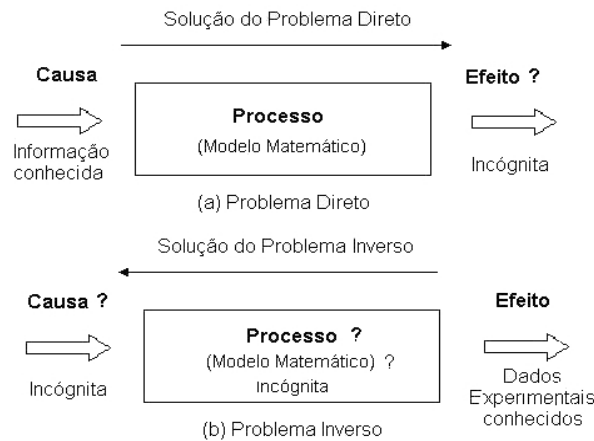


Figura 3.1: Representação esquemática do Problema Direto (a) e do Problema Inverso (b) [145].

Considere agora que a condição de contorno, i.e. a intensidade de uma ou das duas fontes externas de radiação, e/ou as propriedades radiativas do meio, i.e. os coeficientes de absorção e de espalhamento e as refletividades difusas, não são conhecidos. Porém, estão disponíveis valores experimentais da intensidade da radiação, \vec{Y} , medidos exter-

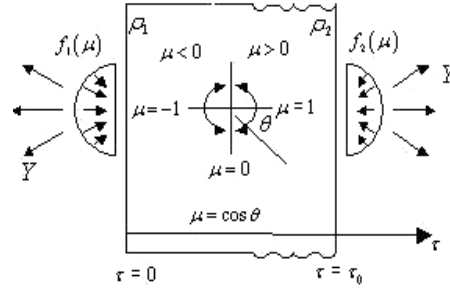


Figura 3.2: Representação esquemática de um meio participante homogêneo unidimensional e da radiação que emerge do mesmo com intensidade \bar{Y} e que é medida experimentalmente.

namente ao meio conforme representado na Figura 3.2.

É possível então determinar as incógnitas do problema a partir destes dados experimentais? Fazendo esta pergunta de outra forma, é possível determinar a causa, i.e. as intensidades da radiação externa incidente, e/ou o que se desconhece do processo que está sendo investigado, i.e. as propriedades radiativas, a partir dos valores medidos experimentalmente da radiação que emerge do meio? Este é o Problema Inverso (PI), representado esquematicamente na Figura 3.1 (b).

Diversas questões relevantes com relação à solução do problema inverso surgem, como por exemplo a existência e a unicidade da solução, o condicionamento do mesmo e a possível amplificação dos ruídos, e a sensibilidade dos dados experimentais (e do modelo) às incógnitas que se deseja determinar. Não é o objetivo deste livro tratar cada um destes temas, e são sugeridas então as Refs. [26, 38, 148, 145] para o leitor interessado.

3.2 Uma Formulação Implícita para o Problema Inverso

Existem diversas formas de formular e de resolver os problemas inversos, e conseqüentemente existem diversas classificações para os mes-

mos [26]. Beck divide os problema inversos em dois grupos: estimativa de parâmetros e estimativa de funções [17]. Esta classificação é baseada na dimensão do objeto matemático a ser estimado, i.e. uma constante (dimensão finita), ou uma função (dimensão infinita). Silva Neto e Moura Neto [148] propuseram uma classificação que também leva em conta a dimensão do modelo matemático usado na representação do processo estudado. Desta forma um modelo de dimensão infinita (representado por equações diferenciais parciais, ou equações integro-diferenciais, por exemplo) pode estar relacionado à estimativa de objetos matemáticos tanto de dimensão finita quanto de dimensão infinita.

Do ponto de vista dos métodos desenvolvidos para a formulação e solução de problemas inversos, Dulikravich et al. [53] indicam que estes podem ser agrupados basicamente de acordo com duas abordagens: métodos inversos puros e métodos baseados em otimização. Neste livro fazemos referência às formulações explícitas e implícitas para os problemas inversos [146, 150], e esta classificação parece estar em acordo com aquela de Dulikravich e colaboradores.

Na formulação explícita as equações usadas na modelagem do problema direto são manipuladas, obtendo-se uma expressão onde as incógnitas a serem determinadas aparecem explicitamente. Roberty, Silva Neto e colaboradores propuseram e implementaram o método fonte-detetor [83], que requer a solução do problema direto, e McCormick e Silva Neto [147] propuseram e implementaram um método analítico baseado nos momentos da radiação emergente do meio, que não requer a solução do problema direto. Na Ref. [150] Roberty, Silva Neto e colaboradores apresentam também uma formulação explícita matricial.

Com relação às formulações implícitas, uma abordagem muito usual empregada na solução de problemas inversos em transferência radiativa consiste na minimização de uma função objetivo, também denominada função custo, como por exemplo aquela dada pela soma dos resíduos quadrados entre os valores calculados com o modelo e os valores medidos experimentalmente para a intensidade da radiação. O ponto focal da busca da solução do problema inverso passa então a

ser o emprego de métodos determinísticos, métodos estocásticos e/ou métodos híbridos para a minimização da função objetivo.

Considere que no problema inverso relativo ao meio participante homogêneo unidimensional descrito na subseção 2.2.1 deseja-se determinar a espessura ótica τ_0 , o albedo de espalhamento simples, ω , e as refletividades difusas ρ_1 e ρ_2 . Tem-se então o vetor de incógnitas

$$\vec{Z} = \{\tau_0, \omega, \rho_1, \rho_2\}^T \quad (3.2.1)$$

onde o sobrescrito T indica transposta. Os Capítulos 5 a 10 deste livro tratam deste problema. No Capítulo 10 é considerado um problema inverso um pouco diferente no qual as intensidades das fontes externas isotrópicas de radiação são consideradas desconhecidas, i.e. A_1 e A_2 , além do albedo de espalhamento simples ω e da espessura ótica τ_0 . O vetor de incógnitas corresponde então a

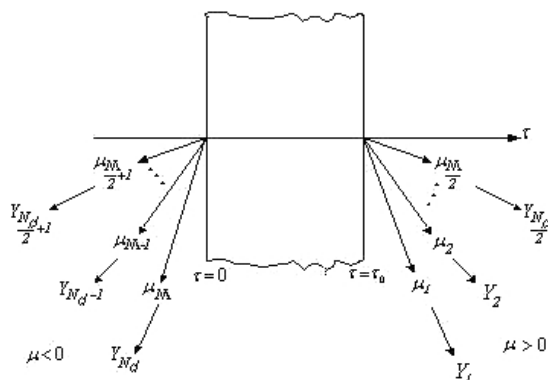
$$\vec{Z} = \{\omega, \tau_0, A_1, A_2\}^T \quad (3.2.2)$$

No problema inverso relativo ao meio participante heterogêneo unidimensional em duas camadas descrito na Seção 2.2.2 deseja-se determinar os coeficientes de absorção κ_a e de espalhamento, σ_s , nas duas camadas. Tem-se então o vetor de incógnitas

$$\vec{Z} = \{\sigma_{s1}, \kappa_{a1}, \sigma_{s2}, \kappa_{a2}\}^T \quad (3.2.3)$$

Considere agora que dados experimentais da radiação que deixa o meio, i.e. Y_i , com $i = 1, 2, \dots, N_d$, onde N_d é o número total de dados experimentais, estão disponíveis. Na Figura 3.3 é representada a situação na qual os dados experimentais são obtidos nas superfícies de contorno do meio e, por conveniência, nas posições angulares correspondentes à discretização do domínio angular descrita na Seção 2.3.1 (vide Figura 2.3). Nesta situação tem-se então $N_d = M$.

Para o problema inverso relacionado ao meio participante heterogêneo unidimensional em duas camadas também são utilizados dados experimentais medidos na interface entre as duas camadas, mas também por conveniência, estes são adquiridos nas posições angulares correspondentes à discretização do domínio angular representada na Figura 2.3.



Como também são usados os dados experimentais obtidos nas superfícies de contorno, tem-se neste caso um total de $N_d = 2M$ dados experimentais.

Para a abordagem de interesse neste livro, o problema inverso de transferência radiativa é formulado como um problema de otimização no qual se busca minimizar a função objetivo dada pela soma dos resíduos quadrados,

onde I_i e Y_i correspondem respectivamente ao valor calculado e ao valor medido experimentalmente da intensidade da radiação para uma mesma posição τ e para uma mesma direção do ângulo polar θ representado por $\mu = \cos\theta$, e os elementos do vetor de resíduos \vec{F} corres-

pondem a

$$F_i = I_i(\vec{Z}) - Y_i; \quad i = 1, 2, \dots, N_d \quad (3.2.5)$$

Esta formulação para o problema inverso é denominada de formulação implícita porque as incógnitas do problema, representadas pelo vetor \vec{Z} , não aparecem explicitamente na formulação. Elas estão embutidas na formulação dos problemas diretos correspondentes e os seus efeitos para os problemas inversos são percebidos quando os problemas diretos são resolvidos e participam então da composição da função objetivo dada pela equação (3.2.4).

3.3 Dados Experimentais

Para a solução de problemas inversos deve-se sempre buscar o uso de dados experimentais reais porque desta forma pode ser verificado se o modelo matemático empregado representa adequadamente os fenômenos envolvidos no processo sob análise. Porém, nem sempre se tem acesso a estes dados, e além disso uma etapa importante no desenvolvimento de uma metodologia para a formulação e solução de problemas inversos consiste na avaliação do desempenho do método desenvolvido. É uma prática usual, portanto, a geração de dados experimentais sintéticos através da adição de ruídos pseudo-aleatórios, aos valores calculados da intensidade da radiação empregando os valores exatos das incógnitas que se deseja determinar, ou seja

$$Y_i = I_i(\vec{Z}_{exato}) + \ell_i \sigma \quad (3.3.6)$$

onde ℓ_i é um número aleatório gerado no intervalo $[-1, 1]$ e σ simula o desvio padrão dos erros experimentais.

Esta abordagem que emprega o mesmo modelo matemático para gerar os dados experimentais sintéticos e depois para resolver o problema inverso é denominada crime inverso (*inverse crime*). Porém, apesar de não permitir a avaliação do modelo matemático do problema direto quanto à representação dos fenômenos envolvidos no processo sob análise, esta abordagem permite uma boa avaliação do método

desenvolvido para o problema inverso porque tem-se o conhecimento prévio da solução que se deseja obter, \vec{Z}_{exato} , e é também exercido o controle do nível de ruído presente no dado experimental.

Uma maneira de atenuar o crime inverso, como um estágio anterior ao uso de dados experimentais reais, consiste no uso de um método para a solução do problema direto e geração do dado experimental sintético, e o uso de outro método na solução do problema inverso.

3.4 Solução do Problema Inverso com o Método de Levenberg-Marquardt (LM)

O método de Levenberg-Marquardt (LM) [110] é um método determinístico baseado no gradiente, de busca local, muito empregado na solução de problemas inversos formulados implicitamente como problemas de otimização [148].

Buscando a minimização da função objetivo dada pela equação (3.2.4) é escrita a equação do ponto crítico

$$\frac{\partial Q(\vec{Z})}{\partial \vec{Z}} = 0, \text{ i.e. } \frac{\partial Q(\vec{Z})}{\partial Z_j} = 0 \text{ com } j = 1, \dots, N_z \quad (3.4.7)$$

onde N_z representa o número de incógnitas, que leva então à equação

$$J^T \vec{F}(\vec{Z}) = 0 \quad (3.4.8)$$

onde o sobrescrito T indica transposta e J representa a matriz Jacobiana cujos elementos são calculados usando

$$J_{ij} = \frac{\partial I_i}{\partial Z_j} \quad (3.4.9)$$

com $i = 1, 2, \dots, N_d$ e $j = 1, 2, \dots, N_z$, onde N_d representa o número total de dados experimentais.

Fazendo uma expansão de Taylor e retendo apenas os termos até a primeira ordem, resulta

$$\vec{F}(\vec{Z}^{n+1}) = \vec{F}(\vec{Z}^n) + J \Delta \vec{Z}^n \quad (3.4.10)$$

onde

$$\Delta \vec{Z}^n = \vec{Z}^{n+1} - \vec{Z}^n \longrightarrow \vec{Z}^{n+1} = \vec{Z}^n + \Delta \vec{Z}^n \quad (3.4.11)$$

Das equações (3.4.8) a (3.4.11) obtém-se

$$(J^n)^T J^n \Delta \vec{Z}^n = - (J^n)^T \vec{F}^n \quad (3.4.12)$$

onde $\vec{F}^n = \vec{F}(\vec{Z}^n)$.

Visando melhorar o desempenho do algoritmo na busca do valor \vec{Z}^* que minimiza a função objetivo (3.2.4), Marquardt [110] propôs que um termo fosse adicionado à diagonal da matriz $J^T J$. Um dos revisores do trabalho de Marquardt indicou que, em 1944, Levenberg havia feito uma proposta semelhante. O método passou então a ser conhecido como o método de Levenberg-Marquardt. A equação (3.4.12) passa então a ser escrita como

$$\left[(J^n)^T J^n + \lambda^n I \right] \Delta \vec{Z}^n = - (J^n)^T \vec{F}^n \quad (3.4.13)$$

onde λ^n é um fator de amortecimento, I representa a matriz identidade, e n é um contador de iterações.

O fator de amortecimento é alterado durante o processo iterativo do algoritmo visando auxiliar na convergência do método, e ao final do procedimento iterativo ele deve estar próximo a zero [110, 148]. É apresentado a seguir um algoritmo para o método de Levenberg-Marquardt.

Passo 0 - Obtenha os dados experimentais Y_i , $i = 1, 2, \dots, N_d$.

Passo 1 - Faça $n = 0$ e escolha \vec{Z}^0 e λ^0 .

Passo 2 - Calcule $J^n e (J^n)^T$ a partir da equação (3.4.9).

Passo 3 - Calcule $I(\vec{Z}^n)$ com o problema direto, o vetor de resíduos \vec{F}^n com a equação (3.2.5), e o valor da função custo $Q(\vec{Z}^n)$ com a equação (3.2.4).

Passo 4 - Calcule $\Delta \vec{Z}^n$ com a equação (3.4.13).

Passo 5 - Calcule \vec{Z}^{n+1} com a equação (3.4.11).

Passo 6 - Verifique se o critério de parada estabelecido a priori foi atendido. Em caso afirmativo pare.

Passo 7 - Faça $n = n + 1$.

Calcule $Q(\vec{Z}^n)$.

Se $Q(\vec{Z}^n) \leq Q(\vec{Z}^{n-1})/a$, onde $a > 1$, faça $\lambda^n = \lambda^{n-1}/b$, onde $b > 1$, e volte para o Passo 2. Caso contrário, faça $\lambda^n = c\lambda^{n-1}$, com $c > 1$, faça $\vec{Z}^n = \vec{Z}^{n-1}$, e volte para o Passo 4.

A estratégia para alteração do valor do fator de amortecimento λ^n descrita no Passo 7 foi proposta por Marquardt [110]. Silva Neto e Özisik [149] usaram os seguintes valores: $a = b = c = 2$ em um problema inverso em transferência radiativa.

Em muitas aplicações, inclusive naqueles descritas nos Capítulos 5 a 10 deste livro, quando aplicável, o cálculo do vetor de correções $\Delta \vec{Z}^n$ foi feito empregando diretamente a equação (3.4.12), e o vetor com a nova estimativa para as incógnitas, \vec{Z}^{n+1} foi obtido então usando

$$\vec{Z}^{n+1} = \vec{Z}^n + \gamma \Delta \vec{Z}^n \quad (3.4.14)$$

onde $0 < \gamma < 1$.

O objetivo central deste livro consiste na apresentação de um conjunto de métodos estocásticos, de busca global, para a solução de problemas inversos de transferência radiativa formulados como problemas de otimização, i.e. com uma formulação implícita. Porém, em muitos casos mostrados nos Capítulos 5 a 10 são empregadas hibridizações onde o método estocástico é usado para gerar uma estimativa inicial para o vetor de incógnitas, \vec{Z}^0 , que é então usado no método de Levenberg-Marquardt.

3.5 Considerações Finais

Na Figura 3.4 é apresentado um resumo de alguns dos conceitos discutidos neste capítulo. Apenas para exemplificar são listados alguns métodos correspondentes às formulações explícitas e implícitas.

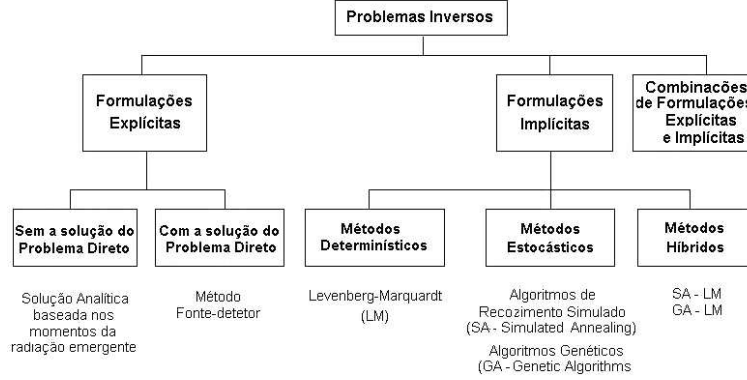


Figura 3.4: Exemplos de formulações explícitas e implícitas para a solução do problema inverso em transferência radiativa.

Em todos os problemas inversos de transferência radiativa tratados neste livro estaremos buscando determinar estimativas para os parâmetros apresentados nas equações (3.2.1) a (3.2.3), que são variáveis reais. Conforme descrito na Seção 2.4, o albedo de espalhamento simples ω e as refletividades difusas ρ_1 e ρ_2 estão no intervalo $[0, 1]$. Para a espessura ótica τ_0 é atribuído um valor superior $\bar{\tau}_0$ artificial apenas para a definição de uma região de busca no intervalo $[0, \bar{\tau}_0]$. O mesmo procedimento é aplicado para as intensidades da radiação externa e para os coeficientes de espalhamento e absorção, σ_s e κ_a , quando aplicável.

Na maior parte das situações tratadas neste livro são considerados os seguintes valores para as intensidades das fontes de radiação externa:

$$f_1(\mu) = A_1 = 1, \quad \text{para } \mu > 0 \quad (3.5.15a)$$

$$f_2(\mu) = A_2 = 0, \quad \text{para } \mu < 0 \quad (3.5.15b)$$

Este é um caso difícil, intencionalmente escolhido, para o tratamento do problema inverso. Conforme pode ser observado na Figura 3.2, o efeito da refletividade difusa ρ_1 somente será percebido no dado experimental \vec{Y} em $\tau = 0$, para $\mu < 0$, após a radiação externa entrar no meio, interagir com o mesmo através de interações de absorção e espalhamento, ser refletida parcialmente de volta para o interior do meio na superfície de contorno em $\tau = \tau_0$, interagir novamente com o meio e após atingir a superfície de contorno em $\tau = 0$ uma fração da radiação dada por $(1 - \rho_1)I(0, \mu)$, com $\mu < 0$, deixa o meio e é captada pelos detetores localizados externamente à superfície em $\tau = 0$.

Uma análise de sensibilidade permite identificar que a estimativa do parâmetro ρ_1 é, na situação aqui descrita, de qualidade inferior à estimativa dos outros parâmetros de interesse [85].

Capítulo 4

Fundamentos de Otimização e Inteligência Artificial

Luiz Biondi Neto
José Carlos Becceneri
José Demisio Simões da Silva
Eduardo Fávero Pacheco da Luz
Antônio José da Silva Neto

4.1 Conceitos Básicos na Área de Otimização

O problema de achar o valor máximo ou mínimo de uma função objetivo, possivelmente sujeito a um conjunto de restrições, é conhecido como problema de otimização.

Faz-se a ressalva de que os problemas inversos tratados nos Capítulos 5 a 10, formulados como problemas de otimização, tem por objetivo a determinação de variáveis reais.

4.1.1 Heurísticas e Meta-heurísticas

O nome heurística é derivado da palavra grega *heuriskein*, que significa descobrir. Hoje esse termo é usado para descrever um método “que, baseado na experiência ou julgamento, parece conduzir a uma boa

solução de um problema, mas que não garante produzir uma solução ótima”. Esse termo pode ser considerado como associado a um conhecimento circunstancial, não verificável matematicamente.

Já o termo “meta-heurística” deriva da composição de duas palavras gregas: “heurística”, já explicada no parágrafo anterior, e o prefixo “meta”, que significa “após”, indicando um nível superior de descoberta.

Uma meta-heurística é uma estratégia de busca, não específica para um determinado problema, que tenta explorar eficientemente o espaço das soluções viáveis desse problema. São algoritmos aproximados que incorporam mecanismos para evitar confinamento em mínimos ou máximos locais. Conhecimentos específicos do problema podem ser utilizados na forma de heurística para auxiliar no processo de busca (por exemplo, na busca de um possível bom vizinho de um determinado ponto).

Resumindo, pode-se dizer que meta-heurísticas são mecanismos de alto nível para explorar espaços de busca, cada uma usando um determinado tipo de estratégia.

De grande importância na aplicabilidade de uma meta-heurística é o balanço dinâmico entre *diversificação* e *intensificação*, fazendo uma distinção entre os termos ingleses *exploration* e *exploitation*. O primeiro pode-se traduzir por diversificação, exploração diversificada, busca em largura ou simplesmente exploração; o segundo por exploração focada, busca em profundidade ou intensificação. Um dos desafios na aplicação de uma meta-heurística é encontrar o equilíbrio ideal entre diversificação e intensificação.

A estratégia de busca de uma meta-heurística depende da filosofia por ela empregada, e seu objetivo é escapar dos mínimos locais a fim de proceder a exploração do espaço de busca por soluções ainda melhores. Na Figura 4.1, os pontos *b* e *c* são considerados mínimos locais e o ponto *a*, mínimo global entre os pontos *x* e *y*.

Classificação das Meta-heurísticas

Talvez o modo mais intuitivo de se classificar meta-heurísticas seja com base nas origens do algoritmo. Nesse sentido, pode-se diferenciar

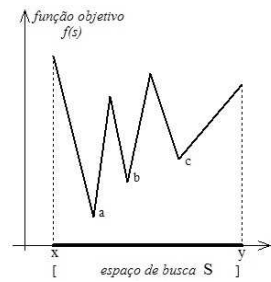


Figura 4.1: Mínimos locais e mínimo global.

os algoritmos inspirados na natureza, como os Algoritmos Genéticos [78] e os Algoritmos das Formigas [22], daqueles não-inspirados na natureza, como a meta-heurística *Iterated Local Search* [134]. Essa classificação pode não ser muito significativa, visando mais a oferecer uma abordagem didática. Por exemplo, o uso de memória na Busca Tabu [65] é ou não inspirada em um processo natural?

Diz-se que uma meta-heurística é bio-inspirada quando suas regras de busca tentam simular alguns aspectos do comportamento de seres vivos como, por exemplo, o já citado Algoritmo das Formigas [22, 51], o algoritmo dos Pássaros [54], o algoritmo das Abelhas [2] ou dos Vermes Luminosos [90].

No método construtivo ou de trajetória, parte-se de um conjunto solução vazio e segue-se acrescentando elementos a esse conjunto até obter uma solução viável para o problema. Métodos populacionais partem de um conjunto de soluções iniciais (a população inicial) e tentam encontrar uma solução melhor alterando-se elementos dessa população.

Um exemplo de modelo construtivo é o Algoritmo das Formigas, pois este descreve uma trajetória no espaço de busca durante a sua execução.

Meta-heurísticas baseadas em população, ao contrário, executam um processo de busca modificando a população inicial, como os Algoritmos Genéticos (*Genetic Algorithms*, GA) e o Recozimento Simulado (*Simulated Annealing*, SA).

4.2 Inteligência Artificial

Apesar da pesquisa da Inteligência Artificial (IA) nos transportar à antiguidade, nos remetendo até mesmo à era Aristotélica, Alan Mathison Turing, matemático inglês que desenvolveu a lógica computacional, é considerado precursor da IA. Em 1950 escreveu um artigo, publicado no Jornal filosófico *Mind*, “*Computing Machinery and Intelligence*” onde descrevia o famoso “Teste de Turing” cuja proposta era verificar se uma máquina realmente pensava [168]. Posteriormente Turing desenvolveu a “Máquina de Turing”, um aparelho que deveria ler um conjunto de instruções de uma fita executando-as sequencialmente [169].

Entretanto, pode-se dizer que o nascimento oficial da IA ocorreu, de fato, logo após a conferência de verão em Dartmouth College, New Hampshire, USA, em 1956, justificando assim sua fama de representar uma das áreas da Ciência da Computação mais recentes. Nessa conferência foi proposto por um grupo de jovens cientistas, John McCarthy (Dartmouth) [111], Marvin Minsky (Harvard) [118], Nathaniel Rochester (IBM) [113] e Claude Elwood Shannon (Bell Laboratories) [144] um estudo sobre o tema IA, expressão ainda pouco conhecida e utilizada para designar um tipo de inteligência, construída pelo homem, para dotar máquinas de comportamentos inteligentes. Constitui-se dessa forma, um ramo da computação que procura representar o conhecimento através de símbolos diferentemente da representação feita exclusivamente por números, tradicionalmente usada na computação. Para tal, faz uso de heurísticas e regras aproximadas para adquirir, representar e processar adequadamente o conhecimento.

Desde o início o termo IA causou muita controvérsia. Na realidade para se criar uma definição precisa de IA seria necessário, antes, definir, claramente, a própria inteligência o que sempre foi muito difícil. IA dispõe de diversas interpretações que podem parecer até mesmo conflitantes. Assim, surgiram diversas definições operacionais, algumas apócrifas, outras nominadas, algumas mais gerais, outras mais específicas, e dentre elas julgamos importante mencionar as seguintes:

- IA é a capacidade de o computador processar símbolos, automa-

tizando faculdades mentais até desconhecidas das máquinas;

- IA é um ramo da ciência que elabora programas para computadores de maneira simbólica, no lugar da forma numérica frequentemente usada;
- IA é o estudo das faculdades mentais através do uso de modelos computacionais (Eugene Charniak e Drew McDermott) [36];
- IA é o estudo de como fazer os computadores realizarem tarefas que, normalmente são feitas melhor pelas pessoas (Elaine Rich) [136];
- IA é o estudo das computações que tornam possível “perceber”, “raciocinar” e “agir” (Patrick Henry Winston) [181];
- IA é o estudo e a programação de agentes inteligentes, isto é, tudo que é capaz de perceber, corretamente, o ambiente por meio de sensores e agir, corretamente, sobre o ambiente por meio de atuadores (David Poole) [131].

O objetivo geral da IA é a concepção e implementação de sistemas inteligentes computacionais baseados em teorias e modelos criados para expressar a capacidade cognitiva do ser humano, imitando aspectos do comportamento, tais como: raciocínio, percepção, aprendizado e adaptação, dentre outros.

Foram criados, quase que intuitivamente, dois ramos para a construção de sistemas inteligentes: a linha conexionista e a linha simbólica. Atualmente essa divisão não parece suficiente para abrigar todos os tipos de sistemas inteligentes disponíveis, mas ela será adotada, por simplicidade, nas Subseções 4.2.1 e 4.2.2.

4.2.1 Conexionista

A linha conexionista visa à modelagem da inteligência humana imitando a estrutura neuro-física e a funcionalidade do cérebro, de seus neurônios, e de suas interligações. Esta proposta foi formalizada quando o neurocientista Warren Sturgis McCulloch e o lógico matemático

Walter Pitts propuseram um primeiro modelo matemático simples para um neurônio [114], que modelava o funcionamento de operações da lógica binária, através de um sistema de inequações lineares de primeira ordem para representar o conhecimento relacionado. A especificação automática dos parâmetros de um neurônio ou de um conjunto de neurônios (a aprendizagem) começou a ser buscada com o trabalho do neurofisiologista Donald Hebb [75] que propôs a primeira regra de aprendizado para atualizar as conexões entre células neuronais com funcionamento descrito pelo modelo de McCulloch e Pitts.

O primeiro modelo de Rede Neural Artificial (*Artificial Neural Network*, ANN) foi proposto por Frank Rosenblatt e consistia de um conjunto de neurônios dispostos em uma única camada e denominado de Perceptron [137]. Este modelo foi criticado por Marvin Minsky e Seymour Papert [119] que mostraram os pontos negativos investigando as propriedades matemáticas da rede Perceptron, mostrando que as mesmas só se aplicavam em problemas linearmente separáveis, impedindo assim a rede de, por exemplo, aprender corretamente o mapeamento de um problema não-linear simples como o OR exclusivo [74].

Por esse motivo, durante um longo período essa linha de pesquisa ficou prejudicada e, de certa forma praticamente paralisada. Paul John Werbos [177], em sua tese de doutorado, criou um algoritmo que permitia o treinamento de redes de múltiplas camadas introduzindo o conceito de retro-propagação de erro e transpondo o grande obstáculo levantado por Minsky e Papert. Na época não foi dada a devida relevância a essa descoberta, não sendo devidamente noticiada na comunidade científica. Posteriormente, D. E. Rumelhart, G. E. Hinton e R. J. William através da publicação de dois volumes sobre Processamento Paralelo e Distribuído, editada por David Rumelhart e James McClelland [138], divulgaram essas novas regras de treinamento idealizadas por Werbos, dando um novo impulso a essa linha de pesquisa. A partir dessa publicação o algoritmo de retro-propagação de erro desenvolvido por Werbos tem sido amplamente divulgado e usado.

Nesse contexto, não poderiam deixar de serem citados alguns pesquisadores, igualmente responsáveis pelo desenvolvimento da área co-

nexionista, dentre eles os seguintes: Bernard Widrow e Marcian Hoff [179] que propuseram uma poderosa regra de aprendizado, produzindo o algoritmo do mínimo quadrado médio; Nils Nilsson [125] introduziu o conceito de máquina de aprendizado; Jack D. Cowan criou um neurônio artificial de disparo suave, baseado na função logística [43]; Stephen Grossberg criou o modelo aditivo do neurônio envolvendo equações diferenciais não lineares [69]; C. von der Malsburg demonstrou o mapeamento auto organizável [174]; W. Little e G. Shaw [94] propuseram um modelo probabilístico de um neurônio; Willshaw e von der Malsburg [180] publicaram o primeiro trabalho sobre mapas auto-organizáveis; Tuevo Kohonen iniciou pesquisa envolvendo memórias associativas [88]; Gail Carpenter e Stephen Grossberg desenvolveram a teoria da ressonância adaptativa (*Adaptive Resonance Theory*, ART) [28]; Tuevo Kohonen desenvolveu o algoritmo não supervisionado com aprendizagem competitiva [89]; John Hopfield [80] introduziu a arquitetura de redes recorrentes, trabalhando também na área de memórias associativas; Dave S. Broomhead e David G. Lowe descreveram os procedimentos para utilização de redes com funções de base radial (RBF) [24]; Vladimir Vapnik desenvolveu a Máquina de Vetor de Suporte [170].

Com a chegada dos novos microprocessadores com elevado poder computacional e baixo custo surgiu a nova geração de computadores com grande capacidade de armazenamento e alta velocidade de processamento, que aliada ao desenvolvimento de algoritmos específicos, inicialmente baseados nas pesquisas de Werbos (redes de múltiplas camadas), possibilitou o aprendizado de problemas complexos e altamente não lineares, desde que os mesmos fossem rigorosamente mapeados. Assim, foi dado novo impulso às pesquisas em ANNs com forte dependência computacional (*hardware* e *software*) [176].

4.2.2 Simbólica

A linha simbólica baseia-se na lógica tradicional e tem como pioneiros os pesquisadores Allen Newell, John Clifford Shaw e Herbert Simon que desenvolveram a primeira linguagem de IA o IPL-11 [124]. Posteriormente, os mesmos pesquisadores criaram o solucionador de proble-

mas gerais (GPS). John McCarthy [112] cria no MIT (*Massachusetts Institute of Technology*) o LISP (*LISt Processing*), que é uma linguagem de programação funcional e que foi muito usada em IA, e até hoje os trabalhos nesse campo são realizados usando-se algum dialeto do LISP [48].

Programas inteligentes, baseados no conhecimento, procuram reproduzir o raciocínio inferencial humano na busca da solução para os problemas. Esses programas inteligentes são constituídos basicamente de: máquina de inferência que manipula o conhecimento representado para desenvolver uma solução do problema, descrito nas informações contidas na base de conhecimento; e a base de conhecimento que trabalha associada a uma base de dados contendo o conhecimento específico do problema.

Entretanto, o conhecimento pode ser encontrado de inúmeras formas diferentes. Um simples fato, um conjunto de fatos ou o relacionamento entre eles são exemplos típicos de como o conhecimento pode se apresentar. Por outro lado, um conhecimento específico pode ser representado apenas sob a forma algorítmica, enquanto outros, heurísticamente, através de regras de manuseio ou estratégias que são usadas para melhorar a eficiência do sistema, na busca da solução de um problema complexo. Para que essas diferentes espécies de conhecimento possam ser usadas pelo computador na solução de um problema, é necessário que eles sejam adequadamente representados dentro de um programa que possa interpretá-los.

Todo sistema computadorizado, que utiliza o conhecimento de um domínio específico para obter a solução de um problema, é considerado um Sistema Especialista (SE), desde que a solução obtida pelo SE seja essencialmente a mesma, da encontrada por um especialista humano, confrontado com o mesmo problema [129]. Devido ao fato de trabalharem com o conhecimento, que é uma entidade simbólica, tais sistemas também são conhecidos como Sistemas Simbólicos.

Um SE [81] é um programa inteligente de computador que usa “raciocínio inferencial” para solucionar problemas de um campo específico do conhecimento humano. Sua característica principal é a capacidade de explicação, semelhante à encontrada nos seres humanos, dotados

de alto grau de especialização. Percebe-se, nitidamente, que um SE se enquadra plenamente na maioria das definições conceituais e operacionais de IA mostradas inicialmente.

Para facilitar o processamento do conhecimento representado pela lógica, Alain Colmerauer e colaboradores [40] desenvolveu, com base na linguagem ALGOL, uma linguagem denominada PROLOG (*PROgrammation en LOGique*). É uma linguagem orientada para o processamento simbólico, permitindo a resolução de problemas que envolvem a representação simbólica de objetos e seus relacionamentos. O PROLOG tem a facilidade de permitir declarar fatos, especificar regras e fazer perguntas utilizando a lógica, e seu interpretador é também um provador de teoremas [39].

Merece destaque o fato de que não se pode separar a história da IA da evolução dos computadores. Certamente, sem a evolução da computação as pesquisas em ANNs, por exemplo, não teriam avançado tanto como pode ser verificado nas pesquisas desenvolvidas atualmente [19].

4.3 Considerações Finais

O termo IA refere-se a programas ou sistemas computacionais destinados a solucionar problemas realizando processamento sobre o conhecimento especialista representado.

O termo IC refere-se a um conjunto de técnicas computacionais inspiradas na natureza tendo como objetivo resolver os mais diversos tipos de problemas.

Para clarificar o discernimento, neste capítulo foram apresentados alguns conceitos importantes nas áreas de otimização, IA e IC. Para clarificar o discernimento entre essas áreas, apresenta-se a Figura 4.2, mostrando um diagrama de Venn, onde cada área é tratada como um conjunto de competências com as respectivas intersecções. Conclui-se que a IC é um subconjunto da IA, no sentido que todo problema de IC pode ser entendido como um problema de IA, mas nem todo problema de IA é visto como um problema de IC. Também se vê que há problemas de otimização que não são vistos como problemas de IC.

Por exemplo, pode-se ter um método exato do tipo *branch-and-bound* que não se encaixa nas áreas de IA ou de IC.

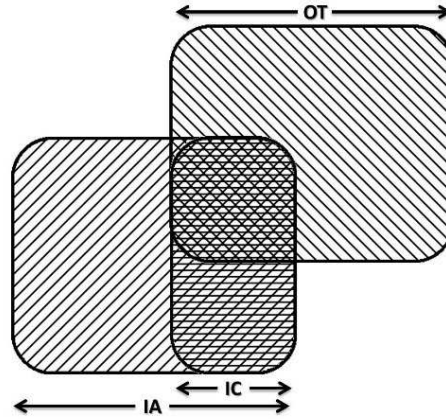


Figura 4.2: Diagrama de Venn para as áreas de otimização (OT), IA e IC

Este capítulo pode ser visto como uma introdução aos capítulos que se seguem. Na primeira parte, os conceitos básicos encontrados na área de otimização foram definidos e exemplificados. Na segunda parte, discutimos os conceitos de IA, IC e as ideias básicas de algumas metaheurísticas que são empregadas nos Capítulos 5 a 10. Nestes são tratadas, portanto, as metaheurísticas selecionadas como casos de estudos, unindo as áreas de otimização e problemas inversos em transferência radiativa.

O método de Evolução Diferencial (DE) apesar de ter por origem a busca da melhora do algoritmo de Recozimento Simulado, ele não é inspirado na natureza conforme os outros métodos considerados neste livro. Porém, por ser um método que está atraindo a atenção de muitos pesquisadores neste momento, e ter sido aplicado nos problemas de transferência radiativa que são usados como os problemas modelo neste livro, ele foi incluído no Capítulo 10.

Capítulo 5

Recozimento Simulado (*Simulated Annealing*)

Francisco José da Cunha Pires Soeiro
José Carlos Becceneri
Antônio José da Silva Neto

5.1 Motivação e Histórico do Método

O Recozimento Simulado (*Simulated Annealing*, SA) tem sua origem na analogia entre o processo físico do resfriamento de um metal em estado de fusão e o problema de otimização. Baseado em ideias da mecânica estatística e no algoritmo de simulação proposto por Metropolis *et al.* [117] o SA foi apresentado inicialmente como uma técnica de otimização combinatória por Kirkpatrick *et al.* [84] que o utilizaram no projeto de sistemas eletrônicos. A mecânica estatística é a disciplina central da física da matéria condensada que estuda as propriedades agregadas de um conjunto elevado de átomos existentes em um líquido ou sólido. Como o número de átomos é da ordem de 10^{23} por centímetro cúbico, somente as características mais prováveis do sistema em equilíbrio térmico serão observadas em experimentos. Isto é caracterizado pelas propriedades médias e pequenas flutuações em

torno das mesmas. Cada configuração é definida por um conjunto de posições atômicas $\{r_i\}$ e afetadas pela probabilidade de Boltzmann $\exp[-E\{r_i\}/k_B T]$, onde $E\{r_i\}$ é a energia da configuração, k_B é a constante de Boltzmann e T a temperatura.

Uma questão fundamental na mecânica estatística é saber o que acontece ao sistema em temperaturas baixas, se o mesmo permanece em estado líquido ou se solidifica e, neste caso, se atinge uma configuração cristalina ou se transforma em vidro. Isto está relacionado diretamente com a energia interna do material. Baixas temperaturas não são condições suficientes para garantir que baixos níveis de energia interna foram atingidos. Experimentos que levam a baixos níveis de energia interna do material são baseados em um recozimento cuidadoso, ou seja, numa redução bem lenta da temperatura desde o estado de fusão do material, que vai resultar numa estrutura cristalina pura. Se não for feito assim os cristais resultantes terão muitos defeitos ou a substância pode se transformar em um vidro, que é uma estrutura apenas ótima localmente. Este processo de se achar um estado de baixa energia é simulado para a solução de um problema de otimização. A função objetivo, ou função custo, corresponderá ao nível de energia do sistema, que nas duas situações, física ou simulada, se deseja minimizar. A temperatura do sistema físico não tem equivalente no problema de otimização. Esta será apenas um parâmetro de controle. No processo iterativo se apenas configurações que levam a uma redução da energia forem aceitas haverá uma convergência rápida de uma temperatura elevada para $T = 0$, o que fisicamente significa uma têmpera ou uma solução metaestável do ponto de vista matemático. No procedimento de Metropolis [117] foi incorporado um aproveitamento cuidadoso de passos de subida, ou seja, passos que resultam em um aumento da função objetivo, visando facilitar a obtenção do ótimo global. E foi esta probabilidade de se aceitar passos que levam a um valor pior da função objetivo que levou a um algoritmo que foge dos mínimos locais possibilitando a convergência para o ótimo global. Metropolis introduziu um algoritmo simples para simular o sistema de átomos em equilíbrio numa determinada temperatura. Em cada passo do algoritmo são dados deslocamentos aleatórios em cada átomo e a

variação de energia é calculada ΔE ($\Delta E = E_{i+1} - E_i$). Se $\Delta E < 0$ o deslocamento é aceito e a nova configuração passa a ser o ponto de partida para o próximo passo. No caso de $\Delta E > 0$, a nova configuração pode ser aceita de acordo com a probabilidade

$$P(\Delta E) = \exp(-\Delta E/k_B T) \quad (5.1.1)$$

Um número aleatório uniformemente distribuído no intervalo $[0, 1]$ é calculado e comparado com $P(\Delta E)$. O critério de Metropolis estabelece que a configuração será aceita se o número aleatório for menor que $P(\Delta E)$, caso contrário é rejeitado e a configuração anterior é utilizada como ponto de partida para o próximo passo. Repetindo-se este procedimento básico diversas vezes o movimento dos átomos de um material na temperatura T é simulado. Utilizando-se a função objetivo no lugar da energia e definindo-se as configurações atômicas como conjuntos de variáveis de projeto $\{x_i\}$, o procedimento de Metropolis gera um conjunto de configurações de um problema de otimização a uma certa temperatura. Esta temperatura é apenas um parâmetro de controle. A constante de Boltzmann também passa a ser um simples fator de escala normalmente igualado à unidade. O SA consiste em primeiro “fundir” o sistema a ser otimizado a uma temperatura elevada e depois em reduzir a temperatura até que o sistema “congele” e não ocorra nenhuma melhora no valor da função objetivo. Em cada temperatura a simulação deve ser executada num número tal de vezes que o estado de equilíbrio seja atingido. A sequência de temperaturas e o número de rearranjos $\{x_i\}$ tentados em cada temperatura para o equilíbrio representam o esquema de recozimento do SA.

Atualmente o SA continua sendo bastante utilizado, muitas vezes em conjunto com outras técnicas de otimização combinatória, tais como algoritmos genéticos [133].

5.2 Descrição do Algoritmo

Um algoritmo básico do SA é apresentado a seguir. A nomenclatura utilizada é a seguinte:

- X é a solução gerada na iteração corrente;
- X^* é a melhor solução encontrada;
- f é a função objetivo;
- T_0 é a temperatura inicial;
- T é a temperatura corrente;
- p é um número real, entre 0 e 1, gerado aleatoriamente.

Passo 1. Atribuir a X uma solução inicial

Passo 2. Fazer $X^* = X$

Passo 3. Definir uma temperatura inicial T_0

Passo 4. Verificar se as condições de parada foram encontradas

Passo 5. Escolher um ponto X' vizinho de X

Passo 6. Calcular $\Delta E = f(X') - f(X)$

Passo 7. Verificar se $\Delta E < 0$

Passo 8. Se Passo-7 for verdadeiro: fazer $X = X'$. Se $(f(X') < f(X^*))$ fazer $X^* = X'$

Passo 9. Se Passo-7 não for verdadeiro: gerar um número aleatório p' . Se $p' < \exp(-\Delta E/T)$ fazer $X = X'$

Passo 10. Verificar se o número de pontos determinados numa dada temperatura é menor ou igual ao número máximo definido *a priori*. Se for verdadeiro, retornar ao Passo 5

Passo 11. Atualizar T

Passo 12. Retornar ao Passo 4

Alguns comentários sobre os passos do algoritmo do SA são feitos a seguir.

No Passo 1, obtém-se uma solução inicial, o que pode ser feito de forma aleatória, normalmente com base na experiência.

No Passo 2, atribui-se a X^* o valor da solução inicial X , por ser a melhor solução conhecida até este passo.

No Passo 3, atribui-se a T o valor da temperatura inicial (T_0). O parâmetro T_0 deve ser suficientemente grande para que todas as

transições sejam inicialmente aceitas. Evidentemente, esse valor depende do tipo do problema e da instância analisada. Na literatura, existem muitas propostas para o cálculo da temperatura inicial [1]. Cita-se aqui dois exemplos: $T_0 = \ln f(X_0)$, onde $f(X_0)$ é o valor da função objetivo da solução inicial ou $T_0 = \Delta E_{max}$, onde ΔE_{max} é a diferença máxima de custo entre duas soluções vizinhas. No último caso, porém, o cálculo pode consumir muito tempo computacional. Por isso, frequentemente, usamos uma estimativa para T_0 .

No Passo 4, é estabelecido um critério de parada. No código utilizado neste capítulo além do número máximo de avaliações da função objetivo, o algoritmo termina pela comparação dos últimos N_ϵ pontos de mínimo encontrados ao fim de cada temperatura com o mais recente mínimo e o melhor de todos encontrados ao longo de todo o processo. Se a diferença entre todos esses valores for menor que ϵ o algoritmo termina. Este critério ajuda a assegurar que o ótimo global foi encontrado.

No Passo 5, escolhe-se um vizinho da solução corrente X . Essa função de escolha é essencial ao bom desempenho do algoritmo, pois, se analisarmos muitos vizinhos, podemos comprometer o tempo de processamento. No presente trabalho um novo valor de X, X' , é determinado variando-se o elemento i do vetor X , da seguinte maneira:

$$x_i = x_i + rv_i \quad (5.2.2)$$

onde r é um número aleatório uniformemente distribuído no intervalo $[-1, 1]$ e v_i o i -ésimo elemento de V (vetor com os comprimentos dos passos). Depois de N_s passos sobre todas as variáveis de projeto (elementos de X), o vetor V com os comprimentos dos passos correspondentes a cada variável é ajustado de modo que cerca de 50% de todos os movimentos sejam aceitos. A finalidade é amostrar amplamente o espaço de projeto. Se um número elevado de pontos é aceito para X , então o passo correspondente é aumentado. Para uma dada temperatura isto aumenta o número de rejeições e diminui a percentagem de aceitações.

No Passo 6, calcula-se a diferença entre os valores da solução corrente e do novo ponto encontrado na vizinhança (X').

No Passo 8, se o valor da função no ponto X' for menor do que em X , então X' passa a ser a solução corrente. Da mesma forma verifica-se se o valor corrente é menor que o valor armazenado em X^* . Em caso afirmativo, X^* recebe o valor de X' .

No Passo 9, caso o valor de $f(X')$ seja maior que o valor de $f(X)$, gera-se um número aleatório p' entre 0 e 1, indicando que uma solução pior foi encontrada em X' . Se f' é maior ou igual a f , o critério de Metropolis já mencionado decide se o ponto será aceito ou não. O valor

$$p = e^{-|f'-f|/T} \quad (5.2.3)$$

é computado e comparado com p' . Se $p > p'$, o novo ponto é aceito, X é atualizado e o algoritmo se move numa direção de subida. Se $p < p'$, então X' é rejeitado. Dois fatores diminuem a probabilidade de um movimento ascendente: baixas temperaturas e grandes diferenças nos valores das funções calculadas. Essa é uma tentativa de se escapar de mínimos locais.

No passo 10, o número máximo de pontos (soluções) determinados em uma dada temperatura é dado pelo produto do número de variáveis pelo número de perturbações de cada variável em cada passo (N_s) e pelo número de mudanças de passo em cada temperatura (N_t). Por exemplo, com quatro variáveis de projeto tem-se $4 \times N_s \times N_t$.

No Passo 11, determina-se que a temperatura seja atualizada. Após N_t vezes nos ciclos (*loops*) acima a temperatura é reduzida. A nova temperatura é dada por

$$T' = r_T \times T \quad (5.2.4)$$

onde r_T é um número entre 0 e 1. Uma temperatura baixa diminui a probabilidade de movimentos de subida, produzindo um número elevado de pontos rejeitados e portanto diminuindo os comprimentos dos passos. Além disso, o primeiro ponto a ser testado em uma nova temperatura é o ótimo atual. Passos pequenos e início no ótimo atual significam que a área do espaço de projeto mais promissora é mais explorada.

Todos os parâmetros mencionados são definidos pelo usuário. Note-se que no início o SA tem uma estimativa grosseira do espaço de projeto movendo-se com passos maiores. À medida que a temperatura cai o método vai lentamente focalizando a área onde o mínimo global deve estar localizado.

O código do SA utilizado neste trabalho é baseado no programa desenvolvido em Fortran por Goffe *et al.* [66] que usaram um algoritmo implementado por Corana *et al.* [41]. Os parâmetros utilizados para a solução do problema de transferência radiativa apresentado na Seção 5.3 a seguir foram:

- $T_0 = 5,0$ - temperatura inicial;
- $r_T = 0,75$ - coeficiente de redução de temperatura;
- $[V] = [1; 1; 1; 1]$ - vetor com o passo inicial (quatro variáveis de projeto);
- $N_s = 20$ - número de perturbações de cada variável em cada passo;
- $N_t = 5$ - número de mudanças de passo em cada temperatura;
- $N_\epsilon = 4$ - número de temperaturas consecutivas onde o critério de convergência deve ser satisfeito;
- $\epsilon < 10^{-6}$ - critério de convergência;
- $N_{max} = 100.000$ - número máximo de avaliações da função objetivo.

Esses parâmetros foram obtidos empiricamente e produziram a necessária robustez para a solução do problema.

5.3 Aplicação ao Problema Inverso de Transferência Radiativa

O método descrito foi aplicado ao problema inverso de transferência radiativa em um meio homogêneo unidimensional, descrito na Subseção

2.2.1 e no Capítulo 3 [151, 152]. Os parâmetros a se determinar correspondem ao vetor indicado na equação (3.2.1) as seguintes incógnitas:

- τ_0 - espessura ótica;
- ω - albedo de espalhamento simples;
- ρ_1 - refletividade na superfície interna esquerda do meio participante homogêneo unidimensional;
- ρ_2 - refletividade na superfície interna direita do meio participante homogêneo unidimensional.

Não temos dados experimentais disponíveis para a solução do problema inverso. Foram utilizados dados experimentais sintéticos Y_i adicionando ruído pseudo-aleatório r aos valores da intensidade da radiação térmica I_i calculados com os valores exatos das incógnitas que se deseja determinar, conforme a expressão abaixo:

$$Y_i = I_i + r\sigma \quad (5.3.5)$$

onde σ representa o desvio padrão dos erros experimentais. Esta expressão é equivalente à equação (3.3.6). Inicialmente consideramos apenas o caso sem ruído para avaliar a potencialidade do método. Introduzimos o ruído na solução híbrida da qual será descrita mais adiante. Vamos analisar dois casos apresentados na Tabela 5.1.

| Propriedade radiativa | Caso 1 | Caso 2 |
|-------------------------------|--------|--------|
| Espessura ótica τ_0 | 1,0 | 1,0 |
| Albedo ω | 0,5 | 0,5 |
| Refletividade difusa ρ_1 | 0,1 | 0,95 |
| Refletividade difusa ρ_2 | 0,95 | 0,5 |

Tabela 5.1: Valores das propriedades radiativas a serem determinadas.

Como solução inicial foram escolhidos pontos afastados da solução exata para testar a capacidade do método de enfrentar as dificuldades no espaço de projeto (região de busca) e evitar os mínimos locais.

A solução para os dois casos é apresentada na Tabela 5.3. Verifica-se a precisão dos resultados embora com um elevado esforço computacional.

| Propriedade radiativa | Caso 1 | Caso 2 |
|-------------------------------|--------|--------|
| Espessura ótica τ_0 | 5,0 | 5,0 |
| Albedo ω | 0,95 | 0,95 |
| Refletividade difusa ρ_1 | 0,95 | 0,5 |
| Refletividade difusa ρ_2 | 0,1 | 0,1 |

Tabela 5.2: Valores iniciais das variáveis.

| Caso | τ_0 | ω | ρ_1 | ρ_2 | Valor final da função objetivo eq. (3.2.4) | Número de avaliações da função objetivo |
|------|----------|----------|----------|----------|--|---|
| 1 | 0,9999 | 0,5002 | 0,0968 | 0,9499 | $2,8E-13$ | 36000 |
| 2 | 0,9998 | 0,5001 | 0,9497 | 0,5000 | $2,1E-10$ | 36400 |

Tabela 5.3: Solução obtida com o Recozimento Simulado (SA).

Uma alternativa ao SA puro para a redução do custo computacional é a utilização de um método híbrido onde inicialmente o SA é utilizado por alguns ciclos e depois um método local baseado no gradiente como o método de Levenberg-Marquardt (LM) [110] é utilizado para a obtenção de uma solução mais rápida e precisa. A ideia é aproveitar as qualidades dos dois tipos de métodos. Usa-se o método global para se chegar a um ponto localizado na região de convergência do método local que é então utilizado. Ilustramos esta alternativa com o Caso 1 acima. A Tabela 5.4 mostra a solução do LM sozinho partindo do mesmo ponto utilizado pelo SA (Tabela 5.2). Neste caso foi introduzido um erro de 5% nos dados experimentais sintéticos.

| Iteração | τ_0 | ω | ρ_1 | ρ_2 | Função objetivo eq. (3.2.4) |
|---------------|----------|-----------|----------|----------|-----------------------------|
| 0 | 5,0 | 0,95 | 0,95 | 0,1 | 10,0369 |
| 1 | 5,7856 | $9,63E-1$ | $6,6E-2$ | $1,0E-4$ | 1,7664 |
| 2 | 7,0822 | $9,97E-1$ | $1,0E-4$ | $1,0E-4$ | 2,5778 |
| \vdots | \vdots | \vdots | \vdots | \vdots | \vdots |
| 20 | 9,2521 | 1,0064 | $1,0E-4$ | $1,0E-4$ | 2,4646 |
| Solução exata | 1,0 | 0,5 | 0,1 | 0,95 | 0,0 |

Tabela 5.4: Resultados para o Caso 1 e erro de 5% nos dados experimentais sintéticos usando o LM.

Verifica-se que o LM não conseguiu convergir partindo do ponto especificado. Utilizemos então o SA para iniciar o problema porém rodando apenas por dois ciclos (duas temperaturas) o que perfaz um total de 800 avaliações da função objetivo.

| Caso | τ_0 | ω | ρ_1 | ρ_2 | Valor final da função objetivo eq. (3.2.4) | Número de avaliações da função objetivo |
|------|----------|----------|----------|----------|--|---|
| 1 | 0,9530 | 0,5310 | 0,0001 | 0,9444 | $8,047E-3$ | 800 |

Tabela 5.5: Solução do SA para o Caso 1 com 2 ciclos.

A partir do resultado obtido na Tabela 5.5 roda-se o LM chegando-se à resposta correta em um número pequeno de iterações (Tabela 5.6).

| Iteração | τ_0 | ω | ρ_1 | ρ_2 | Função objetivo eq. (3.2.4) |
|----------|----------|----------|----------|----------|--------------------------------|
| 0 | 0,953 | 0,531 | 0,0001 | 0,944 | $8,047E-3$ |
| 1 | 0,998 | 0,506 | 0,035 | 0,949 | $4,833E-3$ |
| 2 | 1,001 | 0,503 | 0,070 | 0,949 | $5,07E-6$ |
| 3 | 1,001 | 0,503 | 0,099 | 0,950 | $2,27E-6$ |

Tabela 5.6: Solução para o Caso 1 utilizando o LM após 2 ciclos de SA (800 avaliações da função objetivo).

5.4 Considerações Finais

Pelos resultados apresentados verifica-se o bom desempenho do SA. A desvantagem é o elevado esforço computacional, característica comum a todos os métodos de otimização global. O uso de soluções híbridas com a utilização de um método baseado em gradientes (método local) após alguns ciclos do SA mostrou ser uma boa alternativa para se diminuir o esforço computacional e refinar a solução.

Capítulo 6

Algoritmos Genéticos (*Genetic Algorithms*)

Francisco José da Cunha Pires Soeiro

Luiz Biondi Neto

Antônio José da Silva Neto

6.1 Motivação e Histórico do Método

Os Algoritmos Genéticos (GA) foram desenvolvidos com base nas leis de evolução natural das espécies. Origina-se dos trabalhos de Holland e seus colaboradores [78]. Estes algoritmos têm sua base filosófica na Teoria de Darwin da sobrevivência dos mais bem adaptados ao meio ambiente [46]. Pertencem à categoria dos métodos de busca estocástica com a vantagem de que esta busca ocorre nas regiões do espaço de projeto onde há maior potencial de se encontrar o ótimo global. De maneira análoga ao processo natural onde a população de uma dada espécie se adapta ao seu habitat natural, uma população de projetos (candidatos à solução do problema de otimização) é criada e então permitida se adaptar ao espaço de projeto. As diversas alternativas de projetos que representam uma população numa determinada geração são submetidas a cruzamentos entre eles, tendo o melhor projeto maior probabilidade de ser utilizado. Combinações das melhores

características dos membros no processo de cruzamento resultam em uma população de filhos melhor que a dos pais. Se a medida que indica a adaptabilidade de uma população é correspondente à função objetivo em um problema de otimização, vide equação (3.2.4), sucessivas gerações produzirão melhores valores para a função objetivo. A terminologia utilizada neste trabalho pode ser encontrada em Goldberg [67] e aplicações à engenharia em [70, 93].

No trabalho original de Holland [78], o GA era caracterizado por cadeias de bits representando possíveis soluções de um dado problema e por transformações utilizadas para variar estas soluções. No processo natural, por analogia, as informações genéticas são armazenadas em cadeias de cromossomos que são alteradas através de gerações para a adaptação ao meio ambiente. Esta estrutura cromossômica representa a memória da geração e é alterada pela reprodução dos indivíduos. Além da reprodução, mutações ocasionais da informação genética podem alterar a constituição dos cromossomos. O GA, de maneira similar à natureza, utiliza cromossomos para representar as possíveis soluções do problema no processo de procura da melhor solução.

6.2 Descrição do Algoritmo

O GA é baseado em um processo elitista de seleção dos indivíduos para reprodução. Os membros da população mais adaptados terão maiores oportunidades de serem selecionados contribuindo mais para a melhoria da constituição dos cromossomos na geração seguinte. Este processo é facilitado se for definida uma função de adaptabilidade que será a medida de como um determinado indivíduo é bom em relação aos outros numa determinada geração durante o processo de evolução. Para problemas sem restrições a própria função objetivo servirá como função de adaptabilidade. No caso de ser um problema de minimização utilizam-se os artifícios usuais para se transformar um problema de minimização em um problema de maximização de uma função. No presente trabalho onde se pretende minimizar uma função erro utiliza-se esta função com o sinal negativo como função de adaptabilidade, ou seja,

$$F_{adap} = F_{max} = -F_{min} \quad (6.2.1)$$

Para um problema com restrições a formulação com função penalidade exterior pode ser utilizada para transformá-lo em um problema sem restrições.

Como transformar os diferentes projetos (soluções candidatas para o problema de minimização) em cadeias de cromossomos é visto a seguir. Seja a função objetivo abaixo com três variáveis reais:

$$\text{Minimizar } F = 100 (X_1 + X_2 + X_3) \quad (6.2.2)$$

Como é um problema de minimização de função, a função de adaptabilidade será:

$$F_{adap} = -100 (X_1 + X_2 + X_3) \quad (6.2.3)$$

Cada variável será convertida em seu equivalente binário sendo mapeada em uma cadeia de comprimento fixo de 0's e 1's. A precisão numérica de cada variável (solução) determinará o comprimento desta cadeia. Se considerarmos uma cadeia de 10 dígitos binários os valores máximos e mínimos de cada variável serão os seguintes:

$$(X_i)_{min} = 0000000000 \quad (6.2.4)$$

$$(X_i)_{max} = 1111111111 \quad (6.2.5)$$

Um mapeamento linear converteria valores intermediários de números binários em valores reais de variáveis de projeto da seguinte forma:

$$X = X^{(l)} + \frac{X^{(u)} - X^{(l)}}{2^q - 1} X_{bin} \quad (6.2.6)$$

onde:

- $X^{(u)}$ - valor real máximo da variável de projeto;
- $X^{(l)}$ - valor real mínimo da variável de projeto;

- q - comprimento da cadeia de binários;
- X_{bin} - valor real correspondente ao binário atual, calculado como

$$X_{bin} = \sum_{k=0}^{q-1} 2^k b_k, \text{ onde } b_k = 0 \text{ ou } 1 \quad (6.2.7)$$

Três números binários correspondentes a cada variável de projeto são justapostos formando uma cadeia de 30 dígitos 0's e 1's, isto é, 10 dígitos para cada uma das três variáveis reais X_1 , X_2 e X_3 . Esta cadeia de 30 dígitos representa uma possível solução do problema. Uma sequência de tais cadeias pode ser construída para representar uma população de soluções, cada indivíduo estando associado a uma função de adaptabilidade. A Tabela 6.1 mostra uma população de cinco projetos. Nesta tabela estão os valores das variáveis de projeto, suas representações binárias e o valor da função de adaptabilidade para cada projeto. Os valores máximos e mínimos permitidos para cada variável são $X_i^{(u)} = 10,0$ e $X_i^{(l)} = 1,0$. Uma vez que as representações em binário da população estão disponíveis, simulações da evolução genética podem ser efetuadas. O mecanismo artificial de transformação (evolução) dos genes compreende três operações: seleção, cruzamento e mutação.

| Membro da população | Variáveis de projeto | | | Representação binária | Função de adaptabilidade |
|------------------------|----------------------|-------|-------|--------------------------------|-----------------------------|
| | X_1 | X_2 | X_3 | | |
| 1 | 1,783 | 3,367 | 3,367 | 000101100101000011010100001101 | -851,70 |
| 2 | 1,730 | 5,689 | 7,616 | 000101001110000101011011110000 | -1503,50 |
| 3 | 3,111 | 9,604 | 5,751 | 001111000011110100101000011100 | -1846,60 |
| 4 | 3,921 | 5,786 | 2,865 | 010100110010001000000011010100 | -1257,20 |
| 5 | 4,889 | 3,850 | 8,179 | 011011101001010001001100110000 | -1691,80 |

Tabela 6.1: Exemplo de uma população em números binários.

6.2.1 Seleção

A simulação da evolução genética é aqui limitada a uma dimensão de população que permanece constante ao longo de todo o processo. Dois pais selecionados criam apenas dois descendentes. Variações destas ideias podem ser encontradas em diferentes simulações e em diferentes

trabalhos. O processo de seleção é tendencioso no sentido de produzir membros mais bem adaptados e eliminar os menos adaptados. Dentre as diversas maneiras existentes, a mais simples para se selecionar os membros para cruzamento, é designar uma probabilidade a cada membro baseada em sua função de adaptabilidade. Se f_i é a medida de adaptabilidade do i -ésimo membro pode-se associar a este membro a probabilidade $f_i/\sum_m f_i$, onde m é o tamanho da população. Uma nova população do mesmo tamanho da anterior será criada mas com uma média de adaptabilidade maior. Já que a seleção é elitista, uma ideia bastante utilizada é se levar o indivíduo da população mais bem adaptado (maior valor da função de adaptabilidade) diretamente para a geração seguinte, além de, claro, fazê-lo reproduzir com uma maior probabilidade que os demais.

6.2.2 Cruzamento

O processo de cruzamento permite que as características de diversos projetos sejam intercambiadas criando assim uma geração seguinte mais adaptada. Um tipo usual de cruzamento pode ser feito selecionando-se dois pais, baseado nas respectivas probabilidades de serem escolhidos, escolhendo-se aleatoriamente um ou dois pontos na cadeia genética (binária) e trocando os dígitos 0's e 1's entre os dois pais. Isto está ilustrado abaixo onde as cadeias binárias foram quebradas em dois pontos.

Pai1 = 11001001001001101001 Pai2 = 01011100011100100010

Os pontos de cruzamento foram sublinhados. Há então a troca de dígitos formando os filhos:

Filho1 = 11001100011001101001 Filho2 = 01011001001100100010

Uma probabilidade p_c é definida para determinar se o cruzamento será executado. Na literatura outros tipos de cruzamento podem ser encontrados em [67, 135].

6.2.3 Mutação

A mutação protege a busca genética de uma perda prematura de bom material genético durante a seleção e cruzamento. O processo

de mutação simplesmente é feito escolhendo-se alguns poucos membros da população e de acordo com uma probabilidade p_m troca-se 0 por 1 e vice-versa num ponto escolhido aleatoriamente em suas cadeias binárias (outras formas de mutação são descritos em [135]).

6.2.4 Determinação do tamanho da cadeia binária

Um número binário com q dígitos permite a representação de 2^q valores de uma variável contínua. Se a variável contínua tiver uma precisão A_c , então o número de dígitos na cadeia binária será estimada da seguinte maneira:

$$2^q \geq \frac{(X^{(u)} - X^{(l)})}{A_c} + 1 \quad (6.2.8)$$

onde A_c representa a precisão das variáveis (resolução) e os demais símbolos já foram definidos na equação (6.2.6).

Uma vantagem dos algoritmos genéticos é poder lidar facilmente com variáveis discretas e inteiras. Neste caso faz-se na equação (6.2.8) o valor de A_c igual a 1. Para discussão das aplicações de variáveis inteiras e discretas em projetos de engenharia ver [70, 93].

6.2.5 Algoritmo

O algoritmo de um GA simples é apresentado a seguir.

Passo 1. Gerar uma população inicial;

Passo 2. Avaliar a função de adaptabilidade para cada indivíduo;

Passo 3. Verificar as condições de parada. Caso sejam satisfeitas - terminar. Caso contrário - ir para o Passo 4;

Passo 4. Selecionar indivíduos para cruzamento;

Passo 5. Criar uma nova geração através do processo de cruzamento e mutação.

Passo 6. Avaliar a adaptabilidade dos novos indivíduos;

Passo 7. Substituir a geração anterior pela nova. Ir para o Passo 3.

Alguns comentários sobre os passos acima são apresentados a seguir.

No Passo 1 a população é gerada. Esta geração é feita em binários e é muito simples. Conhecendo-se o comprimento da cadeia binária

basta em cada posição gerar 0's ou 1's para se ter a população. O tamanho da população é um dos parâmetros a serem definidos. A literatura através da experiência de diversos pesquisadores apresenta sugestões sobre os valores bem sucedidos. Uma população pequena tem pouco material genético e conseqüentemente não faz uma boa amostragem do espaço de projeto. Já uma população muito grande vai aumentar bastante o esforço computacional tornando o método pouco eficiente.

No Passo 2 a função de adaptabilidade de cada indivíduo é calculada. Isto é feito no domínio dos números reais (Tabela 6.1). Aqui já se pode designar uma probabilidade de seleção para cada indivíduo.

No Passo 3 estabelecem-se as condições de parada. Normalmente o número máximo de gerações é definido. Pode-se limitar também o tempo de processamento ou o número de gerações passadas sem que haja melhoria na solução do problema.

No Passo 4 é feita a seleção para o cruzamento. A literatura reporta diversas ideias [67]. A mais simples é a roleta da sorte onde as áreas são proporcionais à probabilidade de cada indivíduo de ser selecionado. Os mais bem adaptados terão maior probabilidade de escolha.

No Passo 5 é feito o cruzamento simultaneamente à mutação. Aqui devem ser definidas as probabilidades de cruzamento p_c e mutação p_m . Estes valores são parâmetros arbitrados pelo usuário e, como no caso do tamanho da população, existem valores sugeridos pelos pesquisadores e desenvolvedores de software, baseados em experiências bem sucedidas.

6.3 Aplicação ao Problema Inverso de Transferência Radiativa

O método descrito foi aplicado ao problema inverso de transferência radiativa descrito na Subseção 2.2.1 e no Capítulo 3 [163, 152, 146, 153, 154]. O código do GA utilizado neste trabalho foi desenvolvido em Fortran por D. Carroll

[<http://cuaerospace.com/carroll/ga.html>] . Foram utilizados os parâmetros sugeridos pelo autor que produziram resultados satisfatórios

para o problema em tela.

- $npopsiz = 100$ - tamanho da população;
- $pcross = 0,5$ - probabilidade de cruzamento;
- $pmutate = 0,01$ - probabilidade de mutação.

Os parâmetros do problema de transferência radiativa a se determinar correspondem ao vetor indicado na equação (3.2.1) representando as seguintes incógnitas:

- τ_0 - espessura ótica;
- ω - albedo de espalhamento;
- ρ_1 - refletividade na superfície esquerda do meio participante homogêneo unidimensional;
- ρ_2 - refletividade na superfície direita do meio participante homogêneo unidimensional.

Vamos analisar os casos apresentados na Tabela 6.2.

| Propriedade radiativa | Caso 1 | Caso 2 |
|-------------------------------|--------|--------|
| Espessura ótica τ_0 | 1,0 | 1,0 |
| Albedo ω | 0,5 | 0,5 |
| Refletividade difusa ρ_1 | 0,95 | 0,1 |
| Refletividade difusa ρ_2 | 0,5 | 0,95 |

Tabela 6.2: Valores das propriedades radiativas a serem determinadas.

O Caso 1 foi resolvido utilizando-se apenas o GA. A solução é apresentada na Tabela 6.3.

| Caso | τ_0 | ω | ρ_1 | ρ_2 | Valor final da função objetivo eq. (3.2.4) | Número de avaliações da função objetivo |
|------|----------|----------|----------|----------|--|---|
| 1 | 1,0160 | 0,5665 | 0,8665 | 0,4839 | $6,7E-05$ | 5000 (50 gerações) |

Tabela 6.3: Solução do Algoritmo Genético (GA) para o Caso 1

Verifica-se que após 50 gerações, ou 5000 avaliações da função objetivo, o valor de ρ_1 não foi corretamente determinado. Conforme descrito na Seção 3.5, é uma variável com pequena sensibilidade. Decidiu-se para o Caso 2 utilizar uma estratégia híbrida com o método de Levenberg-Marquardt (LM) [110], para reduzir o esforço computacional e obter um resultado mais preciso. O LM foi rodado após apenas 3 gerações do GA, ou seja, após 300 avaliações da função objetivo. Também neste caso foi introduzido um erro de 2,5% nos dados experimentais sintéticos. O resultado é apresentado na Tabela 6.4.

| Iteração | τ_0 | ω | ρ_1 | ρ_2 | Função Objetivo eq. (3.2.4) |
|----------|----------|----------|----------|----------|--------------------------------|
| 0 | 0,965 | 0,495 | 0,0001 | 0,953 | $3,715E - 3$ |
| 1 | 1,001 | 0,500 | 0,101 | 0,949 | $1,591E - 4$ |
| 2 | 1,001 | 0,499 | 0,109 | 0,949 | $2,387E - 3$ |
| 3 | 1,001 | 0,499 | 0,109 | 0,949 | $1,946E - 6$ |

Tabela 6.4: Resultados para o Caso 2 utilizando o LM após 3 gerações do GA e 2,5% de erro nos dados experimentais.

A solução utilizando o método híbrido é bastante precisa com um esforço computacional muito menor, quando comparado com o GA puro.

6.4 Considerações finais

Pelos resultados apresentados verifica-se o bom desempenho do GA. A desvantagem é o elevado esforço computacional, característica comum a todos os métodos de otimização global, como já foi relatado no Capítulo 5. O uso de soluções híbridas com a utilização de um método baseado em gradientes (método local) após algumas gerações do GA mostrou ser uma boa alternativa para se diminuir o esforço computacional e se ter uma solução mais refinada.

Diversas variantes do GA tem sido propostas. Uma destas variantes é o GA-Epidêmico, no qual, após algumas gerações atuando o GA tradicional, parte da população é dizimada e é substituída por novos indivíduos [115]. A estratégia do GA-Epidêmico foi aplicada com su-

cesso em problemas inversos de condução do calor e produz resultados superiores ao GA padrão [37].

Recentemente Cuco et al. [44] aplicaram o GA-Epidêmico para um problema inverso em transferência de massa e fizeram uma comparação com o GEO (*Generalized Extremal Optimization*).

Melhoramentos do Algoritmo Genético para problemas clássicos, como *Permutation Flowshop Scheduling*, tem sido recentemente apresentados, como pode ser visto em [122].

Capítulo 7

Otimização por Colônia de Formigas (*Ant Colony Optimization*)

José Carlos Becceneri

Stephan Stephany

Haroldo Fraga de Campos Velho

Antônio José da Silva Neto

7.1 Motivação e Histórico do Método

Muitos aspectos do comportamento de um grupo de insetos são estudados pelas teorias de auto-organização, que tentam explicar como indivíduos de comportamento simples podem construir uma sociedade complexa. Por exemplo, sabe-se que em tais sociedades há grupos de trabalhadores especializados e, aparentemente, não há um gerente operacional. Então, como explicar a complexidade das tarefas realizadas? Como a cooperação acontece? Como escrito em [25], “estruturas complexas resultam da interação de comportamentos surpreendentemente simples executados por indivíduos possuindo apenas informação local”.

Para entender como o método de Otimização por Colônia de Formigas (*Ant Colony Optimization*, ACO) foi criado, é necessário se conhecer um pouco sobre o comportamento das formigas. Cada formiga, quando caminha em busca de alimento, deposita no solo uma substância química chamada de feromônio. Na possibilidade de seguir por vários caminhos, uma formiga pode, para decidir entre eles, considerar a quantidade de feromônio - depositado anteriormente por outras formigas - em cada um dos caminhos. Um caminho com muito feromônio é mais atrativo que outro com menos. Com o decorrer do tempo, essa substância química sofre evaporação. O que se nota na natureza é que, após um determinado tempo, a maioria das formigas decide trilhar um caminho mais curto (possivelmente o mais otimizado) entre o seu ninho e a fonte de alimento.

O algoritmo de otimização por formigas foi proposto por Marco Dorigo, em 1992, em sua tese de doutorado [49]. A ideia básica de todo o algoritmo baseado em formigas é o uso de um mecanismo conhecido como reforço positivo [50], baseado na analogia com o comportamento de certas espécies de formigas que, como já mencionado, depositam nos caminhos por elas trilhados uma substância química chamada de feromônio, possibilitando o reforço dos caminhos mais trilhados, os quais são, possivelmente, os melhores. Um feromônio virtual é utilizado para manter as boas soluções na memória do computador. Também há o conceito de reforço negativo, implementado através da analogia com o processo de evaporação que o feromônio sofre na natureza. A combinação do reforço positivo (depósito de feromônio) com o negativo (evaporação), permite que se evite, na maioria dos casos, uma convergência prematura do algoritmo para soluções, possivelmente não ruins, mas talvez, longe da ótima. O comportamento cooperativo é outro conceito importante aqui: algoritmos de colônia de formigas fazem uso da exploração simultânea de diferentes soluções por meio de diferentes formigas. As melhores formigas influenciam a exploração das demais, através das estratégias empregadas para atualizar o feromônio nos caminhos.

7.2 Descrição do Algoritmo

Para permitir uma descrição simples do funcionamento do ACO, considera-se sua aplicação ao Problema do Caixeiro Viajante (PCV).

Esse problema é estabelecido do seguinte modo: Um caixeiro viajante deve visitar n cidades e retornar à cidade de onde partiu, sendo que, com exceção da cidade inicial, cada cidade deve ser visitada apenas uma vez. O objetivo é minimizar o custo dessa viagem. Para detalhes sobre o surgimento desse problema, que data de 1832, e sobre as muitas aplicações que ele pode ter, consulte [91].

Um grafo deve representar o conjunto de cidades, onde cada nó representa uma única cidade do problema e os arcos representam a união entre as cidades, sendo que, cada arco que une duas cidades, é valorado pelo custo da viagem entre essas cidades.

Cada formiga pode ser considerada um agente que:

- ao se mover do nó i para o nó j , pode depositar feromônio no caminho sendo percorrido, dependendo da estratégia de depósito sendo utilizada. Em geral, há duas possibilidades: ou todas as formigas depositam feromônio nos caminhos por elas percorridos, ou apenas a melhor formiga de cada iteração deposita feromônio no caminho percorrido. Para os resultados descritos neste capítulo é utilizada a estratégia de depositar feromônio apenas no caminho percorrido pela melhor formiga de cada iteração.
- possui uma tabela Γ com as cidades que pode visitar. Isso é necessário para evitar que uma formiga vá para uma cidade por onde ela já passou;
- para selecionar um nó a ser visitado, usa uma função de probabilidade. Uma formiga k , estando no nó i , calcula sua probabilidade de ir para o nó j na iteração t através da expressão

$$p_{ij}^k(t) = \frac{[f_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \Gamma_i^k} [f_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad (7.2.1)$$

onde:

- $f_{ij}(t)$ é a quantidade de feromônio entre os nós i e j no instante t ;
- η_{ij} é o inverso da distância entre os nós i e j . Esse parâmetro é chamado de desejabilidade. No caso do PCV, quanto maior a distância entre os nós, menor é o desejo de ir para esse nó; quanto menor essa distância, maior o desejo.
- os parâmetros α e β controlam o peso relativo da quantidade (intensidade) do feromônio e da desejabilidade, respectivamente.

A cada iteração:

- o feromônio depositado em cada arco sofre uma evaporação. Seu valor é reduzido por uma constante de evaporação f_e , com $0 \leq f_e < 1$. Uma possível equação que pode ser utilizada para realizar essa atualização é:

$$f_{ij}(t) \leftarrow (1 - f_e) \cdot f_{ij}(t) + f_e \cdot f_0 \quad (7.2.2)$$

Outras estratégias tanto para realizar o depósito, quanto para realizar a evaporação podem ser definidos. Vide [20] para uma discussão ampla sobre esse assunto.

A seguir os passos para a implementação do ACO para um problema de otimização são apresentados.

Passo 1. Inicialização do sistema:

1. Calcular a desejabilidade de cada nó;
2. Definir a quantidade de feromônio de cada arco;
3. Distribuir as formigas de modo aleatório entre os nós.

Passo 2. Para cada iteração definida:

1. Cada formiga percorre todos os nós do grafo, calculando, em cada nó, sua probabilidade de movimento.

Passo 3. Verificar qual formiga obteve a melhor rota.

Passo 4. Atualização do feromônio:

1. Incrementar a quantidade de feromônio na melhor rota;
2. Reduzir a quantidade de feromônio nas demais rotas.

Essa metaheurística tem sido muito empregada atualmente. Suas ideias conceituais são simples e os resultados encontrados em muitos problemas práticos são muito bons. Veja, por exemplo, as referências [15, 13, 14, 30, 165, 166].

Nas referências [13, 14] é apresentada uma contribuição teórica, que tem mostrado excelentes resultados quando utilizada: é o depósito de feromônio em caminhos adjacentes do melhor caminho trilhado por uma formiga.

7.3 Aplicação ao Problema Inverso de Transferência Radiativa

7.3.1 Problema inverso adotado como exemplo

Exemplifica-se a seguir a aplicação do ACO a um problema inverso de transferência radiativa, conforme [166], no qual deseja-se estimar um conjunto de propriedades radiativas do meio, expressos pelo vetor $\vec{Z} = \{\tau_0, \omega, \rho_1, \rho_2\}^T$ onde as incógnitas correspondem, respectivamente, à espessura ótica, ao albedo de espalhamento simples e às refletividades difusas em cada uma das superfícies que delimitam o meio. É aqui tratado, portanto, o problema inverso de transferência radiativa em um meio homogêneo unidimensional conforme descrito na Subseção 2.2.1 e no Capítulo 3. Adotando-se uma metodologia implícita na resolução deste problema inverso, formula-se o mesmo como um problema de otimização, no qual a função objetivo é dada segundo a equação (7.3.3), sendo calculada a partir da diferença quadrática entre as intensidades de radiação geradas para a solução candidata (\vec{I}^{mod}) e as intensidades experimentais \vec{Y} (ambas expressas por vetores de dimensão N_d , onde N_d representa o número total de dados experimentais). Assim, para uma dada iteração t , tem-se:

$$Q\left(\vec{Z}(t)\right)=\sum_{i=1}^{N_d}\left[I_i^{\text{mod}}\left(\tau_0, \omega, \rho_1, \rho_2\right)-Y_i\right]^2 \quad (7.3.3)$$

A equação (7.3.3) é equivalente à equação (3.2.4).

Como dados experimentais reais não estavam disponíveis, foram gerados dados experimentais sintéticos com a equação (3.3.6).

7.3.2 Aplicação do ACO ao problema inverso escolhido

O ACO foi concebido para problemas similares ao PCV, que diferem de problemas inversos como o acima exposto. Uma primeira dificuldade é que não se pode fazer encontrar nestes problemas inversos algo análogo ao caminho da formiga (correspondente a uma rota possível do PCV). Assim, na abordagem proposta por Becceneri e Zinober [15], associa-se cada formiga a uma possível solução candidata e, da mesma forma que no ACO original, a cada iteração é gerada uma nova população de formigas aleatoriamente e sob influência da matriz de feromônio. O ACO pressupõe que cada caminho possível tem uma quantidade de feromônio, que pode variar com o tempo, sendo estes caminhos correspondentes a rotas possíveis do PCV. No caso destes problemas inversos, atribui-se o feromônio como uma propriedade não do caminho das formigas, mas como uma propriedade dos valores possíveis de cada uma das variáveis que compõem as soluções candidatas. Assim, numa nova iteração/população, cada formiga é gerada aleatoriamente e sob influência de uma matriz de feromônio assim definida.

Entretanto, nos problemas inversos como o exposto, tenta-se estimar um conjunto de variáveis contínuas, enquanto que no PCV as combinações de arcos entre cidades a serem visitadas formam um conjunto discreto de caminhos possíveis, embora seu número possa “explodir combinatoriamente”. Na abordagem proposta, o domínio contínuo de cada variável a ser estimada é então discretizado em n valores possíveis. Por exemplo, no problema em questão, o albedo de espalhamento ω e as refletividades difusas ρ_1 e ρ_2 variam no intervalo entre 0 e 1, e estas são discretizadas em 127 intervalos correspondentes a 128 valores possíveis, isto é $n = 128$, cuja resolução (diferença entre

dois valores) é de 0,007874 ou $1/127$. A espessura ótica não possui um limite superior, mas para a formulação e solução do problema inverso é considerado um valor limite $\bar{\tau}_0$ conforme representado na Figura 7.1. Ao se gerar uma nova população de nf formigas, é preciso então escolher 4 valores discretizados para essas propriedades para cada formiga. O espaço possível de soluções discretizadas é então $n^4 = 128^4$, ou seja, da ordem de centenas de milhões.

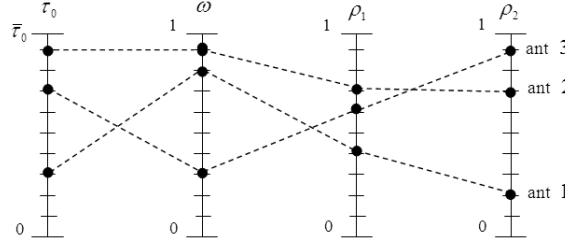


Figura 7.1: Representação esquemática de três formigas no problema inverso de transferência radiativa.

Pode-se então definir a matriz de feromônio como tendo quatro linhas, cada uma correspondente a uma variável e 128 colunas, estas correspondentes aos valores discretizados. Similarmente, adota-se uma estratégia de atualização do feromônio e, na iteração inicial, atribui-se um valor f_0 a todos os 512 elementos desta matriz. Estes elementos sofrem um decréscimo a cada iteração devido à evaporação, segundo uma taxa de decaimento f_e , expressa pela equação (7.2.2). A cada iteração os quatro elementos da matriz correspondentes à solução candidata melhor avaliada são incrementados de f_0 . Então, a equação (7.3.4), expressa a quantidade de feromônio correspondente à variável i , ao valor discretizado j e à iteração t , na qual o termo $\delta_{ij,best}^{t-1}$ é o delta de Kronecker associado, que é nulo exceto para os valores discretizados escolhidos pela melhor formiga na iteração anterior ($t - 1$),

$$f_{ij}^t = (1 - f_e) \cdot f_{ij}^{t-1} + \delta_{ij,best}^{t-1} \cdot f_0 \quad (7.3.4)$$

Na presente abordagem, associa-se i à variável a ser estimada (assumindo valores inteiros no intervalo 1 a 4) e j ao valor discreto de cada variável (assumindo valores inteiros no intervalo 1 a 128).

Neste caso, t_{max} é o número total de iterações, nf é o número de formigas, n a discretização adotada e considerando-se 4 variáveis discretizadas, a complexidade do ACO adaptada para esse problema inverso é dada por $O(t_{max} \cdot n \cdot nf)$.

A geração de cada formiga de uma nova população/iteração é feita aleatoriamente, mas influenciada pela matriz de feromônio, conforme as probabilidades definidas pela equação (7.2.1) na qual os coeficientes α e β ponderavam, respectivamente, a influência da quantidade de feromônio f_{ij} e da desejabilidade η_{ij} . Entretanto, para o problema inverso em questão, f_{ij} representa a quantidade de feromônio associada ao valor discreto j da variável i . Além disso, não é possível definir-se uma desejabilidade η_{ij} , uma vez que, para cada variável a ser estimada, todos os n valores discretos são equivalentes, ou seja, $\eta_{ij} = 1$. Assume-se então $\alpha = 1$ e $\beta = 0$, ou seja, exclui-se a desejabilidade da equação. A nova equação (7.3.5) assim reescrita permite expressar para cada variável i , a probabilidade cumulativa (pc_{ij}) de se escolher o elemento j , calculada com base na quantidade de feromônio de cada elemento 1 a j . Note-se que aqui, optou-se por probabilidades cumulativas e não a probabilidade anteriormente definida (equação 7.2.1),

$$pc_{ij}(t) = \frac{\sum_{l=1}^j f_{il}(t)}{\sum_{l=1}^n f_{il}(t)} \quad (7.3.5)$$

Assim como a matriz de feromônio Λ de dimensão 4×128 (o elemento i da linha j expressa a quantidade de feromônio do valor discreto j da variável i) é conveniente definir aqui a matriz de probabilidades cumulativas PC de dimensão 4×128 (o elemento i da linha j expressa a probabilidade cumulativa do valor discreto j da variável i). Naturalmente, os elementos da última coluna da matriz são unitários, uma vez que a probabilidade cumulativa de cada variável para $j = 128$ é igual a 1.

Nesta abordagem de problema inverso, ambas as matrizes, Λ e PC , são usadas na geração das formigas (na verdade a Λ é usada para calcular a PC) numa dada iteração/geração e são atualizadas a cada

iteração/geração com base na melhor solução candidata obtida, ou seja, na melhor formiga.

Entretanto, foi introduzido um esquema adicional na geração de uma nova formiga, uma nova solução candidata: para escolha dos quatro valores discretos que a compõem, usa-se um esquema de roleta. Define-se um parâmetro q_0 com valor real entre 0 e 1. Para cada variável gera-se um número aleatório e compara-se com esse parâmetro com q_0 . Caso este número seja maior, escolhe-se o valor discreto segundo a matriz PC_{ij} : escolhe-se um j tal que PC_{ij} seja o menor valor maior que o número aleatório gerado. Caso seja menor, toma-se o valor discreto correspondente ao elemento com maior quantidade de feromônio daquela linha da matriz. Consequentemente, o parâmetro q_0 serve para balancear o nível de aleatoriedade na escolha do valor discreto da variável. Se $q_0 = 0$ segue-se sempre a equação, enquanto que se $q_0 = 1$, escolhe-se sempre o elemento com mais feromônio. O valor de q_0 é escolhido empiricamente para o problema inverso. Vale lembrar que a matriz de feromônio é atualizada a cada iteração/geração, bem como a matriz de probabilidades cumulativas PC_{ij} .

Exemplificando, vamos considerar o albedo de espalhamento simples ω . Para simplificar, suponhamos que seu domínio, que é entre 0 e 1, seja discretizado não em 128 valores, mas em 5 valores (0, 0; 0, 25; 0, 50; 0, 75; 1, 00). Ao cabo de uma iteração genérica, vamos assumir que as quantidades de feromônio sejam (no caso correspondente à segunda linha da matriz de feromônio):

$\{0, 05; 0, 10; 0, 20; 0, 05\}$

A somatória das quantidades dessa linha é 0, 40 e a linha correspondente na matriz de probabilidades associada seria:

$\{0, 05/0, 40; 0, 10/0, 40; 0, 20/0, 40; 0, 05/0, 40\} = \{0, 125; 0, 250; 0, 500; 0, 125\}$

e, finalmente, a linha correspondente na matriz de probabilidades cumulativas ficaria:

$\{0, 125; 0, 375; 0, 875; 1, 000\}$

Se, por exemplo, q_0 for 0, 300 e, na geração do valor discreto do albedo de uma nova solução candidata (formiga), sorteia-se o número aleatório 0, 753, sendo este então maior que q_0 , tomar-se-ia simples-

mente o penúltimo valor discreto $(0, 75)$, pois é o que tem maior quantidade de feromônio associado. Ao contrário, caso o número sorteado seja, por exemplo, $0,266$, o qual é menor que q_0 , seleciona-se o valor conforme a equação (7.3.5). Percorrendo-se então a segunda linha da matriz de probabilidade cumulativa, tem-se que o menor valor que é maior que $0,266$ é o segundo $(0, 375)$. Tomar-se-ia então o segundo valor discreto do albedo de espalhamento simples ω $(0, 25)$.

Como nestes problemas inversos não é possível estabelecer uma estimativa inicial segundo uma heurística gulosa [22, 30], adota-se assim um valor qualquer, por exemplo,

$$\vec{Z}^0 = \{\tau_0^0, \omega^0, \rho_1^0, \rho_2^0\} = \{1, 1, 1, 1\}^T \quad (7.3.6)$$

A abordagem acima exposta para aplicação do ACO a problemas inversos de estimação de parâmetros contínuos deu bons resultados, não só nas Refs [15, 166] que serviram para exemplificá-la, mostrando sua competitividade em relação a outros algoritmos de otimização. Em relação ao ACO, essa abordagem é inédita, pois estende sua aplicação ao domínio da otimização contínua.

Como em muitos problemas de otimização estocástica, e mesmo de otimização determinística, a qualidade da solução depende da escolha conveniente de parâmetros de controle do ACO, sendo estes muitas vezes determinados na base da tentativa-e-erro. Estes parâmetros incluem, por exemplo, o número de formigas, as constantes relativas à estratégia de atualização de feromônio, o valor de q_0 usado no esquema de roleta, a discretização adotada para as variáveis contínuas a serem estimadas, etc. É importante ressaltar que o enfoque exposto para uso do ACO em problemas inversos de estimação de variáveis contínuas pode incluir variações relativas à geração de números aleatórios, à estratégia de atualização de feromônio, ao uso da matriz de probabilidades (P ou PC) e esquema de roleta, entre outras.

7.3.3 Resultados numéricos da aplicação do ACO ao exemplo de problema inverso

As Subseções anteriores, 7.2.1 a 7.2.3 mostraram como o ACO foi adaptado para resolução do problema inverso de interesse [166]. Resta agora descrever os resultados numéricos obtidos. Para estimar as quatro propriedades radiativas, todas discretizadas em 128 valores, adotou-se um número limite de 200 iterações, 128 formigas por geração/ iteração e parâmetro $q_0 = 0,0$, ou seja, seleciona-se sempre o valor discretizado de cada variável segundo a probabilidade cumulativa PC dada pela equação (7.3.5). A taxa de decaimento de feromônio é dada por $f_e = 0,03$ e o incremento da melhor solução é igual $f_0 = 0,10$. Cada estimação/reconstrução foi repetida 10 vezes, variando-se apenas a semente usada na geração de números aleatórios para criar novas formigas a cada iteração. Assim, é importante considerar o resultado médio destas repetições, uma vez que há sementes mais ou menos favoráveis, mas não há como determiná-las *a priori*. Foi escolhido um caso considerado difícil, no qual os valores das propriedades radiativas a serem estimadas são:

$$\vec{Z} = \{\tau_0^*, \omega^*, \rho_1^*, \rho_2^*\} = \{1,00; 0,50; 0,10; 0,95\}^T \quad (7.3.7)$$

Considerando-se o problema descrito na Subseção 2.2.1 e no Capítulo 3, tem-se um meio semi-transparente delimitado por duas superfícies infinitas e paralelas. Cada vez que a radiação incide numa dessas superfícies, vinda de fora, ela entra no meio e passa a sofrer interações de absorção e espalhamento. Analogamente a radiação vinda de dentro do meio e que incide nas superfícies do meio é parcialmente refletida para dentro, sendo a parcela não refletida transmitida para fora.

Optou-se também por ter radiação incidente somente do lado esquerdo ($A1 = 1$), sendo, portanto, a radiação incidente no lado direito nula ($A2 = 0$). Em termos do problema inverso de transferência radiativa, isso dificulta a estimação do parâmetro ρ_1 , pois a radiação incide na superfície da esquerda entra, e interage com o meio. Esta por sua vez é refletida parcialmente de volta na superfície direita segundo o valor de ρ_2 (o restante é transmitido para fora do meio), atravessando novamente o meio e sendo refletida mais uma vez pela superfície da

esquerda segundo ρ_1 , sendo que a parcela não refletida é transmitida para fora.

Outro aspecto importante é que, no lugar dos dados experimentais, usaram-se dados sintéticos, ou seja, simulados, expressos pelas radiâncias/ intensidades \vec{Y} e, para se tentar reproduzir a situação experimental, estes dados foram corrompidos com ruído gaussiano de 2% e 5%.

Adotando-se o enfoque descrito anteriormente, a cada iteração é criada uma nova geração de 128 formigas, sendo cada uma delas associada a um conjunto de quatro valores discretos para as propriedades radiativas sendo estimadas, como exemplificado na Figura 7.1.

O processamento requerido é relativamente alto, pois para cada uma das 200 iterações/gerações 128 formigas são geradas e cada uma delas é avaliada pela função objetivo, o que requer a resolução do modelo direto dado pelas equações (2.2.6). Assim, necessita-se executar o modelo direto $128 \times 200 = 25600$ vezes, resultando em alguns minutos de tempo de processamento num microcomputador corrente. É preciso ressaltar que se usou uma malha não muito refinada para o modelo direto. Caso se tivesse optado por uma malha mais fina, i.e., com maior número de pontos, o tempo de processamento chegaria a algumas horas. Então, dependendo do problema, pode ser desejável a paralelização do modelo direto, ou do próprio ACO: cada processador avaliaria uma solução/formiga, executando o correspondente modelo direto.

A Tabela 7.1 ilustra os resultados obtidos para os dados de intensidades/radiâncias \vec{Y} sem ruído, e também com 2% e 5% de ruído gaussiano. Para cada caso, são mostrados os resultados para a média das 10 sementes usadas na geração de números aleatórios, bem como o melhor/pior resultado obtido dentre as 10 sementes. Estas sementes foram também geradas aleatoriamente no intervalo $[0, 1]$, multiplicadas por um fator de 100 e arredondadas para o inteiro mais próximo.

Nesta tabela, para cada caso, aparecem os valores estimados de cada propriedade bem como a iteração na qual se obteve o melhor resultado, pois não necessariamente acha-se a melhor formiga/solução na última iteração. Além disso, aparecem também, para cada caso, o

valor da função objetivo, correspondente à diferença quadrática entre a radiação medida experimentalmente (real) e a radiação calculada para a solução encontrada (modelo). Aparece também a soma das diferenças quadráticas entre o valor estimado e o valor exato dessas propriedades, dada por:

$$d^2 = \sum_{i=1}^4 (Z_i^{exato} - Z_i^{estimado})^2 \quad (7.3.8)$$

| | π_0 | ω | ρ_1 | ρ_2 | $Q(Z)$ eq. (3.2.4) | d^2 eq. (7.3.8) | Iter |
|------------------------|---------|----------|----------|----------|-----------------------|----------------------|-------|
| Dados sem ruído | | | | | | | |
| Melhor semente | 1,0000 | 0,5000 | 0,1016 | 0,9531 | 1,30E-05 | 1,22E-05 | 98 |
| Pior semente | 0,9531 | 0,5547 | 0,2891 | 0,9609 | 6,06E-05 | 4,11E-02 | 122 |
| Média | 0,9813 | 0,5273 | 0,1922 | 0,9547 | 1,03E-05 | 9,62E-03 | 127,3 |
| 2% de ruído | | | | | | | |
| Melhor semente | 1,0000 | 0,5000 | 0,1016 | 0,9531 | 1,49E-05 | 1,22E-05 | 155 |
| Pior semente | 0,9531 | 0,5547 | 0,2891 | 0,9609 | 6,16E-05 | 4,11E-02 | 122 |
| Média | 0,9781 | 0,5289 | 0,1992 | 0,9547 | 1,19E-05 | 1,12E-02 | 132,4 |
| 5% de ruído | | | | | | | |
| Melhor semente | 1,0000 | 0,4922 | 0,0781 | 0,9453 | 2,34E-05 | 5,62E-04 | 149 |
| Pior semente | 0,9531 | 0,5547 | 0,2891 | 0,9609 | 6,69E-05 | 4,11E-02 | 122 |
| Média | 0,9813 | 0,5258 | 0,1867 | 0,9531 | 1,10E-05 | 8,55E-03 | 134,6 |
| Exato | 1,00 | 0,50 | 0,10 | 0,95 | 0,00 | | |

Tabela 7.1: Resultados obtidos com o método de Otimização por Colônia de Formigas (ACO) para diferentes sementes de geração de números aleatórios e usando dados sem ruído, com ruído de 2% e 5%

Nota-se que, conforme esperado, o valor mais difícil de estimar é o de ρ_1 , pois o valor médio da estimativa das 10 sementes não está próximo do valor exato, ao contrário do que ocorre com as outras três propriedades radiativas.

Finalmente, a Figura 7.2 mostra a evolução da função objetivo e da soma das diferenças quadráticas, função objetivo, para um dos casos ilustrados na Tabela 7.1. Pode-se ver que ambos sofrem oscilações mas diminuem gradativamente até atingir a melhor solução e depois ficam estagnados até o número máximo de iterações definido *a priori*.

Assim, conclui-se que o ACO pode ser aplicado com sucesso a problemas inversos de estimação de parâmetros contínuos como demonstrado nesse exemplo e como pode ser encontrado em outras referências

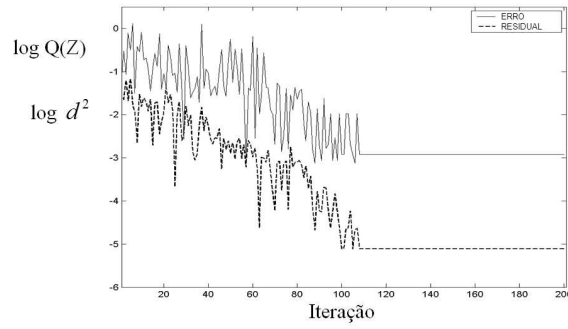


Figura 7.2: Evolução do valor da função objetivo $Q(\vec{Z}(t))$ e das somas das diferenças quadráticas entre os valores estimados e os exatos das propriedades radiativas (d^2) com o número de iterações

dos autores.

7.4 Considerações Finais

O ACO demonstrou ser um otimizador robusto para solução de problemas inversos de estimação de parâmetros contínuos, como visto nas seções anteriores e em trabalhos encontrados na literatura. Uma vez que o ACO é um otimizador estocástico, é adequado para uma busca global no espaço de soluções, embora às custas de um número grande de iterações e, logo, de avaliações da função objetivo, que requerem a execução do modelo direto.

Capítulo 8

Otimização Extrema Generalizada (*Generalized Extremal Optimization*)

Fabiano Luis de Sousa
Francisco José da Cunha Pires Soeiro
Antônio José da Silva Neto

8.1 Motivação e Histórico do Método

O método da Otimização Extrema Generalizada (*Generalized Extremal Optimization*, GEO) [160, 158, 161, 162] foi desenvolvido para ser aplicado a problemas de otimização que apresentem um espaço de projeto complexo, onde métodos tradicionais, como os do tipo gradiente, sejam de difícil aplicação ou levem a soluções sub-ótimas. O GEO foi proposto [160, 158] como uma generalização do método da otimização extrema (*Extremal Optimization*, EO) [21], de forma que o conceito deste pudesse ser aplicado a praticamente qualquer tipo de problema de otimização. Ambos se inspiraram em um modelo simplificado de evolução natural, criado por Bak e Sneppen [11], que buscava mostrar que ecossistemas naturais estão em um estado de Criticalidade

Auto-Organizada (*Self-Organized Criticality*, SOC).

De fácil implementação, o GEO é um método de ordem zero (não utiliza informação das derivadas da função objetivo ou de suas restrições) que pode ser aplicado a problemas com ou sem restrições, que apresentem múltiplas soluções sub-ótimas ou mesmo um espaço de projeto disjunto, com qualquer combinação de variáveis, contínuas, inteiras ou discretas. Ele é uma meta-heurística de busca global, como o algoritmo genético (*Genetic Algorithm*, GA) e o recozimento simulado (*Simulated Annealing*, SA), mas com a vantagem *a priori* sobre estes métodos de possuir apenas um parâmetro livre para ajustar. A importância desta característica do GEO, deve-se ao fato de que a eficiência no uso de algoritmos do tipo GA, SA e GEO em um dado problema é fortemente dependente dos valores atribuídos aos seus parâmetros livres. De fato, o ajuste destes parâmetros pode se tornar um problema em si mesmo, se o custo computacional na avaliação da função objetivo for alto e o algoritmo tiver muitos parâmetros para ajustar.

O GEO, assim como o GA, faz parte de uma categoria de algoritmos inspirados na natureza denominados Algoritmos Evolutivos (AE) [55]. Estes algoritmos tiveram origem nas décadas de 1950 e 1960 [57], mas foi somente a partir do final da década de 1980 que os mesmos passaram a receber grande atenção, tanto do ponto de vista de desenvolvimento de novas implementações como na sua aplicação aos mais diversos tipos de problemas em ciência e engenharia. Os algoritmos genéticos são provavelmente a classe de AE mais conhecida e utilizada, mas outras abordagens também são bastante utilizadas como as Estratégias Evolutivas (EE), a Programação Evolutiva (PE) e a Programação Genética (PG). Apesar de receberem diferentes denominações, dependendo de sua origem e de como são implementados, os algoritmos evolutivos têm como características comuns o uso de uma ou mais populações de *indivíduos* ou *espécies* (soluções candidatas ao problema), e esquemas de seleção e modificação (*mutação e/ou recombinação*) que levem a população a evoluir ao longo das *gerações* (iterações) do algoritmo. Em um problema de otimização, estes esquemas, ou operadores, permitem que uma busca global pela solução

ótima seja feita no espaço de projeto, em um processo onde, probabilisticamente, soluções melhores são identificadas continuamente à medida que novas gerações da população são criadas.

Neste Capítulo a versão original do GEO é apresentada de forma detalhada na Seção 8.2. Na Seção 8.3 é apresentada sua aplicação a um problema de transferência radiativa e na Seção 8.4 são feitas considerações finais. Cabe ressaltar que o GEO como apresentado na Seção 8.2, ou versões mais recentes do mesmo, vem sendo utilizado eficazmente para resolver diferentes tipos de problemas de projeto ótimo, inclusive multi-objetivos [4, 45, 161, 164, 159, 121, 130, 173]. Todavia, o objetivo principal deste Capítulo é apresentar os conceitos básicos do GEO e como o mesmo pode ser utilizado em problemas de transferência radiativa. Descrições mais detalhadas de versões mais recentes do algoritmo básico podem ser encontradas, por exemplo, em [60, 104].

8.2 Descrição do Algoritmo

8.2.1 O GEO canônico

No GEO a população é composta de espécies dispostas em uma cadeia onde cada nicho representa uma espécie, só que no GEO a cadeia é formada por bits. Em um problema de otimização, as variáveis de projeto são codificadas na cadeia de bits. A adaptabilidade de cada espécie é atribuída verificando-se qual o impacto que sua mutação (ou seja, o bit muda de 0 para 1 ou de 1 para 0) gerará no valor da função objetivo, em relação a um dado valor de referência.

A seguir os passos para a implementação do GEO canônico para um problema de otimização são apresentados.

Passo 1. Inicialize uma cadeia de N bits que codifique as L variáveis de projeto do problema. O número de bits que representa cada variável é definido pela resolução mínima que se quer ter em cada uma delas.

Passo 2. Para cada bit b_i , $i = 1, N$:

1. Modifique seu valor corrente (de 0 para 1 ou de 1 para 0) e calcule o valor da função objetivo FOb_i da cadeia de bits com b_i

modificado;

2. Calcule o valor da adaptabilidade de b_i usando a expressão dada pela equação (8.2.1), onde FO_{ref} é um dado valor de referência;
3. Retorne b_i para sua configuração original.

$$Adapb_i = FO_{b_i} - FO_{ref} \quad (8.2.1)$$

Passo 3. Ordene os bits de $rankb_i = 1$ a N , de acordo com suas adaptabilidades $Adapb_i$. O bit menos adaptado recebe $rankb_i = 1$, enquanto o mais adaptado $rankb_i = N$. Em um problema de minimização o bit menos adaptado será aquele que tiver o menor valor de $Adapb_i$. Já em um problema de maximização o bit menos adaptado será aquele que tiver o maior valor de $Adapb_i$.

Passo 4. Realize a mutação em uma espécie da população por meio do seguinte procedimento:

1. Escolha aleatoriamente, com distribuição uniforme, uma espécie b_i candidata a sofrer mutação;
2. Calcule o valor da probabilidade de b_i sofrer mutação, dada pela expressão:

$$Pb_i = (rankb_i)^{-\tau} \quad (8.2.2)$$

onde τ é um parâmetro de ajuste com valor positivo. Gere um número aleatório RAN , com distribuição uniforme, no intervalo $[0, 1]$. Se $Pb_i \geq RAN$, então aceite a mutação e passe para o Passo 5. Se não, volte para (1) neste passo e repita este processo até que um bit seja confirmado para sofrer mutação. A atribuição do valor de τ se reflete no grau de aleatoriedade da busca. Para $\tau = 0$, qualquer espécie escolhida para sofrer mutação terá a mesma chance de ser confirmada. Por outro lado, quanto maior o valor de τ , menor a chance de uma espécie, que não a menos adaptada, ter sua mutação confirmada, se for escolhida para tal. Note que a espécie menos adaptada ($rankb_i = 1$), se escolhida para sofrer mutação, sempre terá a mesma chance de ser confirmada, não importando o valor de τ .

O valor de τ influencia também na performance da busca. Ou seja, para um dado problema, tipicamente existe um valor para τ que faz com que a busca seja realizada de maneira mais eficiente do que se forem usados outros valores. Tem-se observado que os valores ótimos para τ encontram-se no intervalo $[0, 75; 5]$.

Passo 5. Verifique se o critério de parada foi alcançado. Este pode ser, por exemplo, um dado número máximo de avaliações da função objetivo. Se sim, pare o algoritmo e retorne o melhor valor da função objetivo encontrado ao longo da busca, e os valores das variáveis de projeto associadas ao mesmo. Se não, retorne ao Passo 2 com a nova configuração de bits da população (mutação realizada no Passo 4).

Uma pequena variação do algoritmo apresentado nos Passos 1 a 5, pode ser implementada simplesmente fazendo-se o ordenamento das espécies por variável de projeto e modificando-se L espécies (uma para cada variável) a cada iteração do algoritmo. Denominada GEO_{var} [158, 161], esta implementação mostrou ser mais eficiente do que a implementação original para problemas onde existam apenas restrições laterais (restrições do tipo $x_{min} \leq x \leq x_{max}$, onde x representa o valor da variável).

Restrições são levadas em consideração no GEO simplesmente penalizando-se o bit que, se alterado, levar a busca a uma região inviável do espaço de projeto. Note que restrições do tipo lateral são incorporadas diretamente ao problema pela codificação binária das variáveis.

Finalmente, cabe ressaltar que embora na versão original do GEO apresentada acima as variáveis estejam codificadas em uma sequência binária, outros tipos de codificação, ou uma combinação delas, podem ser utilizadas. Por exemplo, em [164] foi utilizada a codificação real em um problema com variáveis discretas.

Na Subseção 8.2.3 a seguir, o funcionamento do GEO canônico é exemplificado, por meio de sua utilização para identificar o mínimo global de uma função teste simples.

8.2.2 Algumas Considerações Sobre Novas Versões do GEO

A partir da ideia básica do GEO descrita em 8.2.3, novas versões do algoritmo vem sendo propostas de forma a buscar aumento de perfor-

mance das versões canônicas ou investigar sua utilização em diferentes tipos de problemas. Por exemplo, em [60] um amplo estudo foi realizado neste sentido, sendo propostas versões que aumentaram significativamente a performance do GEO canônico. Neste trabalho foi também proposta uma primeira versão do algoritmo para aplicação em problemas multiobjetivos (M-GEO), sendo investigadas também versões híbridas e para aplicação em sistemas computacionais com processamento paralelo (GEO_{par}). Já em [104], foi proposta uma versão do algoritmo com codificação real das variáveis (GEO_{real}) que parece ser superior em performance às canônicas e aquelas propostas em [60], para problemas com variáveis contínuas [105].

8.3 Aplicação ao Problema Inverso de Transferência Radiativa

O problema consiste em determinar-se a espessura ótica (τ_0), albedo (ω) e as refletividades difusas (ρ_1 e ρ_2) na fronteira de um meio participante homogêneo uni-dimensional, a partir de medidas da radiação emergindo do meio (Y_i). Vide equação (3.2.1). Este problema está discutido na Subseção 2.2.1 e no Capítulo 3.

Posto como um problema de otimização, deseja-se minimizar a função objetivo

$$Q(\vec{Z}) = \sum_{i=1}^{N_d} [I_{calc_i}(\tau_0, \omega, \rho_1, \rho_2) - Y_i]^2 \quad (8.3.3)$$

Sujeita a $0 \leq \tau_0 \leq 5; 0 \leq \omega, \rho_1, \rho_2 \leq 1$.

onde $I_i(\tau_0, \omega, \rho_1, \rho_2)$, com $i = 1$ a 20, representa os valores calculados para a radiação nos mesmos locais onde Y_i é medida, mas usando-se valores estimados para as propriedades radiativas por meio do GEO.

Três casos foram definidos para estudo, correspondendo a três conjuntos de propriedades a serem recuperadas pelo algoritmo de otimização, como apresentado na Tabela 8.1. Para cada caso foi considerada a leitura de Y_i sendo feita por um sensor, com e sem ruído.

O sinal de Y_i foi gerado numericamente utilizando-se, para cada caso, os valores das propriedades apresentados na Tabela 8.1. O ruído foi introduzido em Y_i utilizando-se a expressão:

$$Y_i = I_i(\tau_0, \omega, \rho_1, \rho_2) + 2,576 \cdot r_i \cdot \sigma \quad (8.3.4)$$

onde r_i é um número aleatório gerado com distribuição uniforme no intervalo $[-1, 1]$ e σ é o desvio padrão do ruído, que recebeu os valores de 0,005, 0,002 e 0,0025 para os casos 1, 2 e 3, respectivamente, correspondendo a ruídos com valores máximos da ordem de 5% da intensidade da radiação medida.

| Propriedade radiativa | Caso 1 | Caso 2 | Caso 3 |
|-------------------------------|--------|--------|--------|
| Espessura ótica τ_0 | 1,0 | 2,0 | 0,5 |
| Albedo ω | 0,5 | 0,8 | 0,3 |
| Refletividade difusa ρ_1 | 0,2 | 0,1 | 0,1 |
| Refletividade difusa ρ_2 | 0,2 | 0,8 | 0,8 |

Tabela 8.1: Valores das propriedades radiativas a serem recuperadas.

O GEO, na sua implementação GEO_{var} , foi utilizado para resolver este problema, e seus resultados comparados com os obtidos utilizando-se um SA e um GA [163]. Em um trabalho posterior [61], estes resultados foram comparados com uma versão do GEO desenvolvida por Galski [60], que introduzia elementos característicos do SA ao seu funcionamento, denominada GEO+SA. A diferença fundamental entre o GEO canônico e a versão GEO+SA é a introdução de um recozimento (*annealing*) no parâmetro τ do GEO, similarmente ao que ocorre com a temperatura no SA, de forma a transformar a busca pelo ótimo global no GEO progressivamente mais determinística. Entretanto, cabe ressaltar também que o GEO+SA, dependendo do ajuste de dois de seus parâmetros livres (n_{max} e q), pode realizar a busca pelo ótimo em ciclos de recozimento (*annealing*) e *reaquecimento* (ou seja, tornar a busca mais determinística progressivamente até um certo ponto e depois novamente aleatória em ciclos).

Nas Figuras 8.1 a 8.3 são apresentados, no espaço de função objetivo, os resultados da evolução da média dos melhores valores obtidos em 10 execuções independentes de cada algoritmo, em função do número de avaliações da função objetivo. No GEO, GEO+SA e GA

foram utilizados 10 bits para codificar cada variável, enquanto no SA foi utilizada codificação real. No GEO foi utilizado $\tau = 1, 25$. Detalhes da implementação do GEO, SA e GA ao problema e do ajuste de seus parâmetros livres pode ser encontrado em [163]. A mesma informação relativa ao GEO+SA pode ser encontrada em [61].

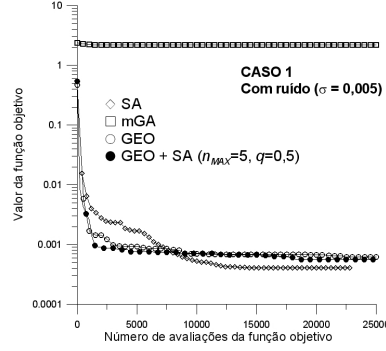


Figura 8.1: Média dos melhores valores da função objetivo em dez execuções independentes de cada algoritmo em função do número de avaliações da função objetivo, para o Caso 1, com ruído. Adaptado de [61].

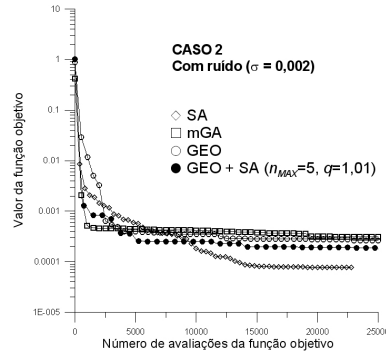


Figura 8.2: Média dos melhores valores da função objetivo em dez execuções independentes de cada algoritmo em função do número de avaliações da função objetivo, para o Caso 2, com ruído. Adaptado de [61].

Pelos resultados apresentados nas Figuras 8.1 a 8.3, observa-se que o SA, para o problema em questão, claramente apresentou uma per-

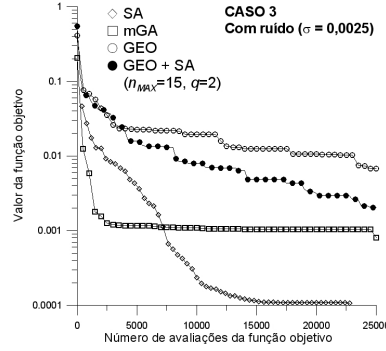


Figura 8.3: Média dos melhores valores da função objetivo em dez execuções independentes de cada algoritmo em função do número de avaliações da função objetivo, para o Caso 3, com ruído. Adaptado de [61].

formance melhor que o GEO canônico o GEO+SA e o GA. Diferentemente do que se esperava [163, 61], a aplicação de uma estratégia de recozimento (*annealing*) no parâmetro τ do GEO não resultou em uma melhora significativa na performance do algoritmo para este problema. Note-se que o GA também apresenta uma performance pobre quando comparado com o SA. Talvez esta diferença de performance se deva à codificação das variáveis. Tanto os GEOs quanto o GA usaram codificação binária, enquanto no SA foi utilizada codificação real. Como observado em [60], a codificação binária pode não ser a melhor representação para o GEO em problemas onde a mesma levar o algoritmo a “saltar” regiões promissoras do espaço de projeto devido ao caráter discreto dos incrementos no valor das variáveis. De fato, a escolha da representação correta a ser usada para codificar as variáveis de projeto, é um fator importante para se obter uma boa performance de um algoritmo evolutivo [55].

8.4 Considerações Finais

Finalmente, cabe ressaltar que o GEO é um algoritmo em evolução. A partir da sua implementação canônica descrita na Seção 8.2.2, versões mais sofisticadas ou customizadas para determinados tipos de proble-

mas, incluindo hibridização com outros algoritmos, podem ser desenvolvidas, como já demonstrado em [60] e [104]. Essa continua sendo uma das promissoras áreas de pesquisa com o GEO, assim como sua aplicação aos mais variados tipos de problema.

Capítulo 9

Algoritmo de Colisão de Partículas (*Particle Collision Algorithm*)

*Wagner Figueiredo Sacco
Diego Campos Knupp
Eduardo Fávero Pacheco da Luz
Antônio José da Silva Neto*

9.1 Motivação e Histórico do Método

Nas décadas de 1980 e 1990, a aplicação de métodos estocásticos em problemas de difícil otimização, especialmente os combinatórios, tornou-se alvo de um considerável interesse. Essas técnicas mostraram-se bem mais eficientes que os métodos determinísticos tradicionais, sendo uma das principais ferramentas empregadas, o algoritmo de minimização conhecido como recozimento simulado (*Simulated Annealing*, SA) [84]. A maior vantagem do SA sobre os outros métodos então existentes era a sua capacidade de evitar cair em armadilhas do tipo mínimos locais. O algoritmo emprega uma busca aleatória que não aceita somente mudanças que diminuam a função objetivo

Q , mas também algumas mudanças que a aumentem, seguindo o paradigma estabelecido por Metropolis [117]. O aumento do valor da função objetivo é aceito com probabilidade

$$P = e^{-(\Delta Q/T)} \quad (9.1.1)$$

onde ΔQ é o aumento de valor da função objetivo e T é um parâmetro de controle que, por analogia à aplicação original do algoritmo, é conhecido como a temperatura do sistema. No Capítulo 5 é apresentado o método SA.

Quando considerado como uma técnica para solucionar os problemas de otimização de alta multimodalidade, ou seja, com vários ótimos locais, o SA possui algumas características atraentes:

- é um “método fraco”, isto é, não é projetado para um problema específico;
- é um método comparativamente eficiente em problemas de alta dimensionalidade;
- não requer qualquer informação funcional da derivada da função [23].

No entanto, o SA possui limitações que dificultam a sua utilização [29]:

- vários parâmetros a serem especificados pelo usuário;
- o algoritmo é extremamente sensível aos valores destes parâmetros;
- tendência a convergir para um ótimo local se a taxa de resfriamento não for suficientemente baixa.

Avaliando estas limitações do SA e com a experiência adquirida na sua implementação e utilização, recentemente foi concebido um algoritmo do tipo Metropolis a partir de uma analogia com a colisão de partículas nucleares [52]. Este algoritmo, denominado algoritmo de colisão de partículas, tem como vantagem o fato de não necessitar da especificação de parâmetros além daqueles que definem o número de iterações. O algoritmo de colisão de partículas (*Particle Collision Algorithm*, PCA) [139, 140] é descrito na seção seguinte.

9.2 Descrição do Algoritmo

9.2.1 A versão canônica

O PCA é inspirado no espalhamento de uma partícula nuclear incidente (onde ela é espalhada pelo núcleo-alvo) e pela absorção (onde ela é absorvida pelo núcleo-alvo). Nesta modelagem, a partícula que atinge um núcleo com baixo valor da função objetivo é absorvida. Em contrapartida, uma partícula que atinge um núcleo com alto valor da função objetivo é espalhada para outra região. Isso permite que o espaço de busca do problema seja amplamente percorrido e que as regiões mais promissoras sejam exploradas através de eventos sucessivos de espalhamento e absorção.

Assim como no SA, primeiramente uma solução inicial é determinada e em seguida esta solução é modificada através de uma perturbação estocástica. As qualidades dessas possíveis soluções são comparadas e então é decidido pela manutenção ou alteração da solução atual por outra solução potencial.

Se a qualidade da nova solução é melhor do que aquela da solução antiga, então a partícula é absorvida e ocorre a exploração das vizinhanças para que seja encontrada uma solução ainda melhor. O Passo 4 no algoritmo descrito a seguir realiza essa busca local, gerando pequenas perturbações estocásticas na solução dentro de um processo iterativo.

Se a qualidade da nova solução é pior do que aquela da solução antiga, então a partícula é espalhada. A probabilidade de espalhamento $p_{scattering}$ é inversamente proporcional a sua qualidade, i.e. uma partícula de menor qualidade terá maior probabilidade de ser espalhada. O Passo 5 no algoritmo descrito a seguir realiza a operação de espalhamento.

Deste modo, o PCA também pode ser considerado um algoritmo do tipo Metropolis, i.e. uma solução pode ser aceita, com certa probabilidade, mesmo sendo sua qualidade menor do que aquela da solução antiga. Tal flexibilidade pode evitar a convergência para ótimos locais.

A seguir, os passos para a implementação do PCA canônico para um problema de minimização da função objetivo são apresentados. O

fluxograma do PCA é representado na Fig. 9.1.

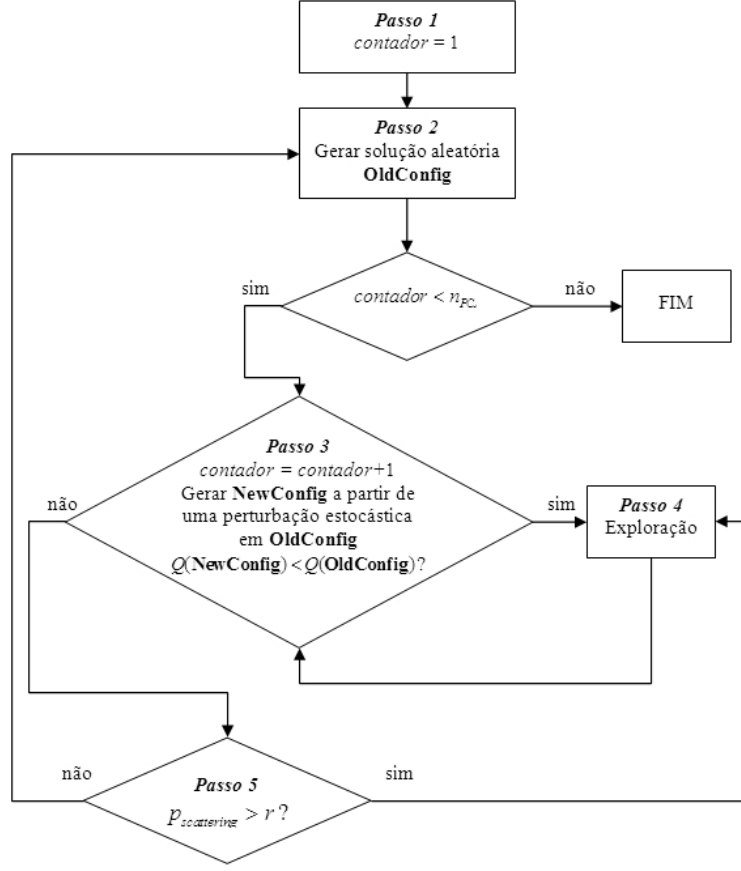


Figura 9.1: Fluxograma do PCA.

Passo 1. Atribua 1 como o valor para a variável *contador*;

Passo 2. Gere uma solução inicial aleatória dentro do domínio de busca da solução. Chame esta solução de *OldConfig*

$$\text{OldConfig} = L + (U - L)r \quad (9.2.2)$$

onde L e U são, respectivamente, o limite inferior e superior do domínio

de busca da solução e r é o gerador de um número aleatório entre 0 e 1;

Passo 3. Caso $contador \geq n_{PCA}$ encerre o PCA. Caso contrário, incremente o valor de $contador$ em uma unidade e gere uma nova solução, $NewConfig$, que é calculada a partir de uma perturbação estocástica na solução $OldConfig$ da seguinte forma:

$$\begin{cases} NewConfig = OldConfig + [(U - OldConfig) \cdot r - \\ \quad (OldConfig - L) \cdot (1 - r)] \\ NewConfig = L, \text{ se } NewConfig < L \\ NewConfig = U, \text{ se } NewConfig > U \end{cases} \quad (9.2.3)$$

Se o valor da função objetivo para $NewConfig$ for menor que o valor da função objetivo para $OldConfig$ então $OldConfig$ recebe o valor de $NewConfig$ e vá para o Passo 4. Caso contrário, vá para o Passo 5.

Obs: Mantenha uma variável $BestConfig$ que contém o valor da melhor solução até o momento. Sempre que uma nova solução for adotada para $OldConfig$ deve-se verificar se esta solução é melhor que a $BestConfig$ e então, se for o caso, fazer $BestConfig$ receber o valor de $OldConfig$.

Passo 4. Faça o seguinte procedimento $n_{Exploitation}$ vezes:

$$\begin{cases} NewConfig = OldConfig + [(U^* - OldConfig) \cdot r - \\ \quad (OldConfig - L^*) \cdot (1 - r)] \\ NewConfig = L, \text{ se } NewConfig < L \\ NewConfig = U, \text{ se } NewConfig > U \end{cases} \quad (9.2.4)$$

onde :

$$U^* = (1 + 0,2r) \cdot OldConfig$$

$$L^* = (1 - 0,2r) \cdot Oldconfig$$

Caso o valor da função objetivo para $NewConfig$ seja menor que o valor da função objetivo para $OldConfig$ então $OldConfig$ recebe o valor de $NewConfig$. Sempre que uma nova solução for adotada

como *OldConfig* deve-se verificar se esta solução é melhor que a *BestConfig* e então, se for o caso, fazer *BestConfig* receber o valor de *OldConfig*.

Depois de repetido este procedimento $n_{Exploitation}$ vezes, vá para Passo 3.

Passo 5. Calcule a probabilidade de espalhamento, $p_{scattering}$,

$$p_{scattering} = 1 - \frac{F(BestConfig)}{F(NewConfig)} \quad (9.2.5)$$

Gere um número aleatório r . Se $p_{scattering}$ for maior que r , então *OldConfig* recebe o valor de *NewConfig* e vá para o Passo 4. Caso contrário, vá para o Passo 2.

Ao término do PCA, a estimativa do mínimo da função objetivo é dado pela solução *BestConfig*.

A versão canônica do PCA descrita acima, apesar da simplicidade, foi bem sucedida em aplicações reais de engenharia [140, 142], onde superou o algoritmo genético [78] e variantes, com larga margem.

Novas versões do PCA vêm sendo implementadas de modo a buscar um melhor desempenho. A hibridização de métodos estocásticos com métodos determinísticos, na qual o método estocástico desempenha apenas um pequeno número de iterações de modo a encontrar uma estimativa inicial razoável para ser explorada pelo método determinístico tem sido aplicada com sucesso [157, 127]. Assim, foi implementada uma versão híbrida do PCA com o conhecido método de Levenberg-Marquardt (LM) [110, 148]. Esta versão híbrida foi capaz de obter resultados melhores com um esforço computacional muito menor [85]. Na Subseção 3.4 é feita uma descrição do método LM.

Outro objeto de estudo tem sido a tentativa de melhorar o mecanismo de busca local do PCA (Passo 3 do algoritmo descrito na seção 9.2.1), que é muito simples [139]. Assim, em [141] o algoritmo determinístico Nelder-Mead Simplex [123] foi utilizado para a busca local do PCA. De modo semelhante, foi utilizado o Algoritmo de Otimização por Enxame de Partículas (*Particle Swarm Optimization*, PSO) [16] e, recentemente, tem sido estudado o uso do algoritmo de busca direta

de Hooke-Jeeves [79, 143].

Mais recentemente, em [108] foi implementada uma versão do PCA com processamento paralelo e em [86] foi implementada uma versão baseada em uma análise de sensibilidade prévia, na qual apenas os dados com maior sensibilidade são utilizados.

9.2.2 Algoritmo de colisão de múltiplas partículas

A implementação desta nova versão baseada em múltiplas partículas estende o conceito básico da aplicação de uma única partícula para a exploração da totalidade do espaço de buscas em um dado tempo total de iterações. Nesta nova versão, é utilizado um conjunto de n partículas explorando, de maneira independente, porém relativamente colaborativa, este mesmo espaço de busca [108].

O algoritmo proposto para o algoritmo de colisão de múltiplas partículas, ou *Multi-Particle Collision Algorithm* (M-PCA), é, em sua essência, exatamente o mesmo do PCA canônico. Porém a introdução de n partículas efetuando a busca no mesmo espaço de soluções leva à necessidade da implementação de um mecanismo de comunicação indireta entre as partículas, para que uma coordenação das atividades seja capacitada.

Esta coordenação foi viabilizada através da implementação de uma técnica de *blackboard*, onde o *BestFitness*, o melhor resultado obtido pela partícula do PCA agora seja atualizado de maneira constante, com a melhor posição dentre todas as partículas envolvidas no processo de busca. Ou seja, as partículas trabalham de maneira independente no processo de busca da solução, porém no momento em que alguma partícula identifica uma nova melhor posição, esta partícula “escreve” no *blackboard* do M-PCA para que as outras partículas tomem conhecimento deste novo valor, e então os seus critérios de busca, tais como a definição de um novo $p_{scattering}$, variável em uso na função de espalhamento e que leva em consideração o melhor fitness da melhor solução atual, sejam baseados neste valor, definido como sendo o melhor mínimo atualmente em uso, e de conhecimento disseminado, pelo conjunto de partículas.

A redução de tempo proporcionada pelo M-PCA advém da capaci-

dade de dividir o total de iterações imposta ao algoritmo pelo número de partículas que atuarão de maneira conjunta no espaço de busca, ou seja, para uma inicialização do PCA com 10^5 iterações, uma instância do M-PCA com 10 partículas terá o número de iterações reduzidas para 10^4 , caso 100 partículas sejam usadas, 10^3 iterações serão suficientes para um número equivalentes de avaliações no espaço de buscas.

9.3 Aplicação ao Problema Inverso de Transferência Radiativa

Como não se tinha disponível dados experimentais reais, dados experimentais sintéticos foram gerados, usando

$$Y_i = I_{calc_i} \left(\vec{Z}_{exato} \right) + \sigma_e \cdot r \quad (9.3.6)$$

onde r é um número aleatório de uma distribuição gaussiana normal, i.e. com média zero e desvio padrão unitário, \vec{Z}_{exato} é o vetor contendo os valores exatos das variáveis que serão estimadas e σ_e simula o desvio padrão dos erros de medição. Este procedimento está descrito na Subseção 3.3.

Para o caso de transferência radiativa em um meio homogêneo unidimensional, o número de dados experimentais utilizados em todos os resultados foi $N_d = 20$. Para o caso de transferência radiativa em um meio heterogêneo unidimensional em duas camadas, $N_d = 40$, o dobro do primeiro caso citado. Isso se deve à presença de detectores internos na interface das duas camadas, para garantir a unicidade da solução [156].

Para todos os casos apresentados nesta seção as intensidades das fontes de radiação externa foram consideradas como $A_1 = 1, 0$ e $A_2 = 0, 0$. Vide Capítulos 2 e 3 para a descrição dos problemas diretos e dos problemas inversos considerados nesta seção.

Todos os resultados foram gerados em uma máquina equipada com Processador AMD TurionTM 64 X2 Mobile 1,6 GHz, 1,37 GB de RAM.

9.3.1 Meio homogêneo unidimensional

Conforme descrito no Capítulo 3, o problema inverso de transferência radiativa, tal como descrito na Subseção 2.2.1 e no Capítulo 3, pode ser enunciado como: utilizando medidas experimentais $Y_i = 1, 2, \dots, N_d$, deseja-se determinar os elementos do vetor de incógnitas \vec{Z} definido como

$$\vec{Z} = \{\tau_0, \omega, \rho_1, \rho_2\}^T \quad (9.3.7)$$

onde τ_0 é a espessura óptica do meio, ω o albedo de espalhamento simples, ρ_1 e ρ_2 as refletividades difusas das paredes internas do meio.

Como consideramos o número de dados experimentais, N_d , maior que o número de incógnitas, N_u , i.e. $N_d > N_u$, formulamos o problema inverso implicitamente, como um problema de otimização, no qual se deve minimizar a função objetivo, dada por

$$Q(\vec{Z}) = \sum_{i=1}^{N_d} [I_i(\vec{Z}) - Y_i]^2 \quad (9.3.8)$$

onde $I_i(\vec{Z})$ representa os valores calculados para a radiação nos mesmos locais onde Y_i é medida. O objetivo do PCA é estimar o vetor de propriedades radiativas, \vec{Z} , de modo a minimizar a função objetivo $Q(\vec{Z})$. O problema inverso de transferência radiativa é descrito com um maior nível de detalhe no Capítulo 3.

Três casos foram propositalmente escolhidos por apresentarem propriedades radiativas difíceis de serem recuperadas pelo algoritmo de otimização. Estes casos estão apresentados na Tabela 9.1.

| Propriedade radiativa | Caso 1 | Caso 2 | Caso 3 |
|----------------------------------|--------|--------|--------|
| Espessura ótica, τ_0 | 1,0 | 0,5 | 2,0 |
| Albedo de espalhamento, ω | 0,5 | 0,1 | 0,9 |
| Refletividade difusa, ρ_0 | 0,1 | 0,1 | 0,5 |
| Refletividade difusa, ρ_1 | 0,1 | 0,9 | 0,5 |

Tabela 9.1: Valores das propriedades radiativas a serem recuperadas.

O desvio padrão dos erros de medição, σ_e (vide equação (9.3.6)), recebeu os valores $\sigma_e = 0,0005$, $0,0004$ e $0,004$ para os Casos 1, 2 e 3,

respectivamente, de modo que os erros são menores ou iguais a 5,5%.

O PCA foi configurado com $M_{PCA} = 200$ e $n_{Exploitation} = 500$. Com esta configuração o PCA avalia, em média, 40.000 vezes a função objetivo.

Com exceção de τ_0 , temos um intervalo de busca de $[0, 1]$ para todas as variáveis, que representa, de fato, os limites físicos para essas variáveis. Para τ_0 foi considerado um espaço de busca de $[0, 3]$. Na realidade, a espessura ótica pode ser maior que o limite superior considerado, porém para $\tau_0 > 3$ a radiação transmitida é muito pequena, o que afeta a qualidade das estimativas obtidas.

Para cada caso da Tabela 9.1 foram realizadas dez execuções com o PCA. Nas Tabelas 9.2, 9.3 e 9.4, a seguir, são apresentadas a melhor e a pior estimativa, \vec{Z}_{best} e \vec{Z}_{worst} , respectivamente, a média e o desvio padrão das 10 execuções, μ_Z e σ_Z , respectivamente, e o valor da função objetivo no final da execução, $Q(\vec{Z})$. Cada execução levou, em média, 2h40min de tempo de processamento.

| | τ_0 | ω | ρ_1 | ρ_2 | $Q(\vec{Z})$ eq. (3.2.4) |
|---------------------------------------|----------|----------|----------|----------|-----------------------------|
| \vec{Z}_{exato} | 1,0 | 0,5 | 0,1 | 0,9 | - |
| PCA \vec{Z}_{worst} | 1,003 | 0,515 | 0,145 | 0,901 | $3,80E - 05$ |
| \vec{Z}_{best} | 1,004 | 0,507 | 0,116 | 0,902 | $6,44E - 06$ |
| μ_Z | 1,002 | 0,498 | 0,092 | 0,899 | |
| σ_Z | 0,006 | 0,010 | 0,032 | 0,002 | |
| $\frac{\sigma_Z}{\mu_Z} \times 100\%$ | 0,6% | 1,9% | 34,6% | 0,3% | |

Tabela 9.2: Resultados obtidos com o PCA para o meio homogêneo unidimensional - Caso 1. $\sigma_e = 0,0005$ (5,5%).

Pode ser observado nas Tabelas 9.2 a 9.4 que o PCA foi capaz de obter boas estimativas para todos os casos considerados. Entretanto, observa-se nos Casos 1 e 2 que o desvio padrão para a refletividade difusa ρ_1 foram relativamente altos. Isso indica que, embora a média possa estar próxima do valor exato, as estimativas não são precisas. Isso era previsto, uma vez que a análise de sensibilidade destes casos mostra grande dificuldade de se estimar essa incógnita.

No Caso 3, os desvios padrão para todas as incógnitas, com exceção de ρ_1 , foram maiores que nos Casos 1 e 2. A principal dificuldade neste

| | τ_0 | ω | ρ_1 | ρ_2 | $Q(\vec{Z})$ eq. (3.2.4) |
|---------------------------------------|----------|----------|----------|----------|-----------------------------|
| \vec{Z}_{exato} | 0,5 | 0,1 | 0,1 | 0,9 | - |
| PCA \vec{Z}_{worst} | 0,470 | 0,124 | 0,140 | 0,906 | $2,22E-04$ |
| \vec{Z}_{best} | 0,499 | 0,115 | 0,124 | 0,899 | $4,18E-05$ |
| μ_Z | 0,485 | 0,110 | 0,113 | 0,904 | |
| σ_Z | 0,029 | 0,010 | 0,077 | 0,006 | |
| $\frac{\sigma_Z}{\mu_Z} \times 100\%$ | 6,0% | 9,4% | 47,7% | 0,7% | |

Tabela 9.3: Resultados obtidos com o PCA para o meio homogêneo unidimensional - Caso 2. $\sigma_e = 0,0004$ (5,5%).

| | τ_0 | ω | ρ_1 | ρ_2 | $Q(\vec{Z})$ eq. (3.2.4) |
|---------------------------------------|----------|----------|----------|----------|-----------------------------|
| \vec{Z}_{exato} | 2,0 | 0,9 | 0,5 | 0,5 | - |
| PCA \vec{Z}_{worst} | 2,180 | 0,920 | 0,549 | 0,440 | $1,79E-03$ |
| \vec{Z}_{best} | 1,952 | 0,897 | 0,495 | 0,511 | $2,97E-04$ |
| μ_Z | 2,114 | 0,907 | 0,517 | 0,488 | |
| σ_Z | 0,159 | 0,012 | 0,032 | 0,029 | |
| $\frac{\sigma_Z}{\mu_Z} \times 100\%$ | 7,5% | 1,4% | 6,2% | 6,0% | |

Tabela 9.4: Resultados obtidos com o PCA para o meio homogêneo unidimensional - Caso 3. $\sigma_e = 0,004$ (5,5%).

caso é o alto valor da espessura ótica, τ_0 , o que traz dificuldades quando se quer considerar informações da radiação transmitida na solução do problema inverso.

9.3.2 Meio heterogêneo unidimensional em duas camadas

No problema inverso de interesse nesta seção, previamente descrito na Subseção 2.2.2 e Capítulo 3, deseja-se determinar os elementos do vetor de incógnitas \vec{Z} , definido como

$$\vec{Z} = \{\sigma_{s1}, k_{a1}, \sigma_{s2}, k_{a2}\}^T \quad (9.3.9)$$

onde σ_{si} são os coeficientes de espalhamento e k_{ai} são os coeficientes de absorção para as camadas $i = 1, 2$.

Como consideramos o número de dados experimentais, N_d , maior que o número de incógnitas, N_z , i.e. $N_d > N_z$, formulamos o problema inverso implicitamente, como um problema de otimização, no qual se

deve minimizar a função objetivo, dada pela equação (9.3.8), onde \vec{Z} é dado pela equação (9.3.9) e $I_i(\vec{Z})$ representa os valores calculados para a radiação nos mesmos locais onde Y_i é medida.

Como caso teste para esse problema, foi considerado um meio composto por duas camadas diferentes, isto é com as propriedades radiativas diferentes.

| Propriedade | Valor |
|-------------------------|-------|
| $L_1 (cm)$ | 0,8 |
| $L_2 (cm)$ | 3,2 |
| $\sigma_{s1} (cm^{-1})$ | 0,8 |
| $k_{a1} (cm^{-1})$ | 0,5 |
| $\sigma_{s2} (cm^{-1})$ | 0,9 |
| $k_{a2} (cm^{-1})$ | 0,3 |
| ρ_1 | 0,1 |
| ρ_2 | 0,0 |
| ρ_3 | 0,0 |
| ρ_4 | 0,6 |

Tabela 9.5: Propriedades do meio participante contendo duas camadas diferentes.

Para este caso, o domínio de busca foi considerado como $[0, 1]$ para todas as propriedades radiativas. Na Tabela 9.6, a seguir, são mostrados os resultados obtidos pelo PCA configurado com $n_{PCA} = 100$, $n_{Exploitation} = 100$ para sete execuções do algoritmo.

| $n_{PCA} = 100, n_{Exploitation} = 100$ | | | | | | |
|---|-------------------------|--------------------|-------------------------|--------------------|-----------------------------|-----------------|
| Exec. | $\sigma_{s1} (cm^{-1})$ | $k_{a1} (cm^{-1})$ | $\sigma_{s2} (cm^{-1})$ | $k_{a2} (cm^{-1})$ | $Q(\vec{Z})$ eq. (3.2.4) | Tempo (min)* |
| 1 | 0,791 | 0,549 | 0,910 | 0,280 | $2,48E-06$ | 13,9 |
| 2 | 0,791 | 0,467 | 0,861 | 0,325 | $2,38E-07$ | 13,4 |
| 3 | 0,741 | 0,513 | 0,878 | 0,325 | $3,76E-06$ | 15,4 |
| 4 | 0,864 | 0,484 | 0,854 | 0,316 | $9,90E-06$ | 11,6 |
| 5 | 0,701 | 0,483 | 0,985 | 0,295 | $4,19E-06$ | 10,4 |
| 6 | 0,781 | 0,481 | 0,826 | 0,324 | $2,91E-06$ | 11,1 |
| 7 | 0,915 | 0,474 | 0,978 | 0,342 | $6,69E-06$ | 12,6 |
| μ | 0,797 | 0,493 | 0,898 | 0,315 | | |
| σ | 0,072 | 0,028 | 0,062 | 0,021 | | |
| $\frac{\sigma}{\mu} \times 100\%$ | 9,0% | 5,8% | 6,9% | 6,6% | | |

* Processador Intel Core i5, 4 GB RAM

Tabela 9.6: Resultados obtidos com o PCA para o meio heterogêneo unidimensional em duas camadas. Foram considerados até 5% de erro no dado experimental. $\vec{Z}_{exato} = \{0,8; 0,5; 0,9; 0,3\} cm^{-1}$.

Dos resultados apresentados na Tabela 9.6, pode se observar que o

PCA foi capaz de obter boas estimativas para as propriedades radiativas também para o problema em um meio heterogêneo unidimensional em duas camadas. É importante ressaltar, mais uma vez, que neste caso detetores internos na interface foram utilizados de modo a garantir a unicidade da solução [156].

9.4 Considerações finais

O futuro do PCA parece residir em duas vertentes: na paralelização, seguindo o trabalho pioneiro em [108] e na hibridização [85, 141].

No futuro, pode ser testado um esquema paralelo do PCA associado a mecanismos de otimização determinísticos. Outro possível melhoramento é a introdução de mecanismos de perturbação estocásticas mais inteligentes, talvez com heurísticas embutidas para problemas específicos.

Capítulo 10

Evolução Diferencial (*Differential Evolution*)

*Fran Sérgio Lobato
Valder Steffen Jr.
Antônio José da Silva Neto*

10.1 Motivação e Histórico do Método

O algoritmo de Evolução Diferencial (*Differential Evolution*, DE) foi proposto em 1995 por Storn e Price [167] com o objetivo de estimar os parâmetros do Polinômio de Chebyshev. A principal motivação para o seu desenvolvimento foi a lenta taxa de convergência e a dificuldade na determinação dos parâmetros exibida pelo algoritmo denominado Recozimento Genético (RG) (*Genetic Annealing Algorithm*), proposto por Price em 1994 e publicado no *Dr. Dobb's Journal* (DDJ). O Recozimento Genético é uma abordagem de otimização combinatória baseada numa população inicial e que alia características dos Algoritmos Genéticos (*Genetic Algorithms*, GA) e do Algoritmo de Recozimento Simulado (*Simulated Annealing*, SA).

Com as dificuldades encontradas durante a resolução do problema de estimativa de parâmetros, Storn decidiu modificar o algoritmo de

RG para trabalhar com codificação de ponto flutuante e com operações aritméticas. Neste sentido, ele acabou por desenvolver o operador de mutação diferencial, no qual o algoritmo de DE é baseado. Daí em diante, foi dado um passo a mais até se chegar à primeira versão do algoritmo de DE. Posteriormente, Storn sugeriu diferentes configurações para a determinação de candidatos em potencial, de tal forma a melhor acomodar a sua arquitetura em máquinas paralelas [132].

Outro aspecto, segundo Price, Storn e Lampinen [132], que motivou o desenvolvimento do algoritmo de DE, foi o tratamento de problemas reais através de uma abordagem que fosse conceitualmente simples e eficiente ao mesmo tempo. Essa abordagem, segundo aqueles autores, deveria ser capaz de ultrapassar obstáculos, tais como: tratamento de funções não diferenciáveis, descontínuas ou até mesmo desconhecidas a priori; lidar com funções não lineares com grande potencial de apresentarem multi-modalidade; de trabalhar com variáveis de projeto discretas, contínuas, inteiras e/ou binárias ao mesmo tempo, sem ter que reestruturar o problema original conforme ocorre quando métodos clássicos são utilizados.

A apresentação do algoritmo de DE no meio científico foi realizada no *ICEO - International Contest on Evolutionary Optimization*, no qual ficou demonstrada a sua simplicidade conceitual e o seu desempenho em uma série de funções matemáticas com diferentes graus de dificuldade e complexidade. Essa primeira aparição foi consolidada com a publicação do artigo intitulado *Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces* no *Journal of Global Optimization*, em 1997 [132].

A partir da publicação deste trabalho, a comunidade científica passou a se interessar pelo algoritmo de DE. Neste sentido, hoje em dia, pode-se encontrar inúmeras aplicações da técnica em vários campos da ciência. Na solução de problemas inversos encontra-se aplicação na estimativa de parâmetros cinéticos no processo de fermentação alcoólica [175, 98]; na determinação de propriedades radiativas em problemas de transferência de calor por radiação [103, 102]; na estimativa de parâmetros cinéticos em um secador rotatório [97]; na estimativa de parâmetros de um filtro adaptativo utilizado como ferramenta para

controle de qualidade em linhas de montagem de câmbios automotivos [99]; na estimativa da difusividade térmica de frutas [109]. Na área de controle ótimo, pode-se mencionar a determinação do perfil ótimo de alimentação de substrato em um fermentador de batelada alimentada [82] e a resolução de problemas de controle ótimo clássicos com restrições de desigualdade [100]. Aplicações em robótica podem ser encontradas no trabalho de Oliveira [126]. No contexto multi-objetivo, destacam-se trabalhos como a otimização estrutural de vigas [95], a determinação do perfil ótimo de alimentação de substrato e dos eventos em um problema de controle ótimo com índice flutuante [101], a otimização das condições operacionais de um reator industrial utilizado para a produção de estireno [10] e o uso de técnicas de meta-modelagem associada ao enfoque multi-objetivo para o tratamento de um problema de interação fluido-estrutura [96]. Várias outras aplicações do algoritmo de DE podem ser encontradas em [96, 132] e no site: <http://www.icsi.berkeley.edu/~storn/code/html>.

10.2 Descrição do Algoritmo

De maneira geral, o algoritmo de DE apresenta as seguintes operações: a inicialização da população, o operador de mutação, o operador de cruzamento, o operador de seleção e o critério de parada do algoritmo. A seguir são descritos, com detalhes, cada uma destas operações.

10.2.1 A Inicialização do Algoritmo de DE

Assim como em outras estratégias evolutivas, o processo de inicialização da população no algoritmo de DE consiste na geração de indivíduos de forma aleatória. A partir daí, geram-se números aleatórios que serão aplicados ao domínio das variáveis de projeto, obtendo-se assim um vetor de indivíduos da população, como mostrado a seguir

$$x_{i,j} = x_{i,L} + rand(x_{i,U} - x_{i,L}) \quad (10.2.1)$$

onde $x_{i,L}$ e $x_{i,U}$ são os limites inferiores e superiores das j -ésimas variáveis de projeto, respectivamente e $rand$ é um gerador de números

aleatórios entre 0 e 1. Na literatura são encontrados outros mecanismos para a inicialização da população, como, por exemplo, o uso de distribuição Gaussiana, onde são gerados indivíduos com média e desvio padrão definidos pelo usuário para essa finalidade [3].

10.2.2 O Operador de Mutação

O algoritmo de DE realiza as operações de mutação e recombinação para a geração de uma nova população com NP indivíduos. Em particular, o operador de mutação diferencial adiciona um vetor referência, escolhido aleatoriamente na população, a um vetor diferença obtido a partir de outros dois vetores também escolhidos aleatoriamente na população. A equação (10.2.2) mostra como essa combinação é realizada para gerar o novo vetor (candidato) $v_{i,g}$,

$$v_{i,g} = x_{r0,g} + F (x_{r1,g} - x_{r2,g}) \quad (10.2.2)$$

O escalar F , denominado taxa de perturbação, é um número real que controla a magnitude do vetor diferença obtido em cada operação aritmética realizada. O vetor referência, $x_{r0,g}$, é geralmente escolhido aleatoriamente na população, apesar de poder ser definido de outras formas, como, por exemplo, usando o melhor indivíduo da geração anterior. De maneira similar, os vetores $x_{r1,g}$ e $x_{r2,g}$ são também escolhidos de forma aleatória na população.

Como dito anteriormente, o vetor referência $x_{r0,g}$ pode ser escolhido aleatoriamente na população ou pode ser definido segundo um outro critério.

A convenção utilizada pelos autores é DE/x/y/z, onde x representa o vetor que será perturbado. Por exemplo, optando-se por *rand*, o vetor que será perturbado é escolhido aleatoriamente na população. y é o número de pares de vetores que são considerados durante a perturbação e z é o tipo de cruzamento usado para a geração do candidato, *bin* no caso binomial e *exp* no caso exponencial. Recentemente, novas estratégias para a representação do mecanismo de mutação no algoritmo de DE têm sido propostas. Dentre estas, pode-se citar o trabalho de Fan e Lampinen [56]. Em tal abordagem, fatores de ponderação

para cada um dos vetores que serão utilizados para a geração do novo indivíduo são calculados a partir dos indivíduos da geração corrente. Esse mecanismo foi testado com sucesso, em termos de tempo de processamento, quando comparado com os mecanismos originais do algoritmo de DE em problemas inversos [7].

10.2.3 O Operador de Cruzamento

Para complementar a operação de mutação, o algoritmo de DE emprega o operador de cruzamento do tipo binomial (outros operadores podem ser encontrados em [126]). Nesta operação, o vetor $v_{i,g}$ gerado anteriormente pode ou não ser aceito na próxima geração de acordo com a seguinte relação:

$$u_{i,g} = \begin{cases} v_{i,g} & \text{se } rand \leq Cr \text{ ou } j = j_{rand} \\ x_{i,g} & \text{caso contrário} \end{cases} \quad (10.2.3)$$

onde Cr é denominada probabilidade de cruzamento definida pelo usuário e contida no intervalo $[0, 1]$. Esse parâmetro controla as informações dos pais que serão transmitidas aos filhos. Para determinar qual a contribuição de um determinado vetor gerado, o cruzamento compara Cr com o gerador de números aleatórios $rand$. Se o número aleatório gerado é menor ou igual a Cr , o vetor $v_{i,g}$ (com posição j_{rand}) é aceito, caso contrário, o vetor $x_{i,g}$ é mantido na população atual.

10.2.4 O Operador de Seleção

Se o vetor $u_{i,g}$ tem melhor valor de função objetivo (f_{obj}) com relação ao vetor $x_{i,g}$, ele o substitui na próxima geração, caso contrário, $x_{i,g}$ é mantido na população por mais uma geração como mostrado a seguir:

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{se } f_{obj}(u_{i,g}) \leq f_{obj}(x_{i,g}) \\ x_{i,g} & \text{caso contrário} \end{cases} \quad (10.2.4)$$

Uma vez completado o processo de atualização da população que será considerada na próxima geração, todo o processo descrito acima é repetido até que um determinado critério de parada seja satisfeito.

10.2.5 O Critério de Parada

O principal critério de parada utilizado nos métodos evolutivos é o número máximo de gerações, definido previamente pelo usuário. Todavia, outros mecanismos podem ser utilizados para finalizar o processo evolutivo, dentre os quais pode-se citar [132]: o tempo de processamento, o número de avaliações da função objetivo, o uso de um valor de referência obtido da literatura para essa finalidade e o monitoramento humano. Em qualquer uma destas estratégias procura-se obter convergência à custa de baixo esforço computacional, mensurado pelo número de avaliações da função objetivo.

Com relação à escolha dos parâmetros do algoritmo de DE, Storn e Price [132] aconselham o uso dos seguintes valores: número de indivíduos da população como sendo um valor entre 5 e 10 vezes o número de variáveis de projeto, taxa de perturbação F entre 0,2 e 2,0 e probabilidade de cruzamento Cr entre 0,1 e 1,0. Com relação à escolha da estratégia DE/x/y/z, Angira e Babu [6], Babu e Anbarasu [9] e Babu *et al.* [10] constataram em seus trabalhos que os melhores resultados, em termos de convergência e diversidade para os problemas estudados por esses autores, foram obtidos quando se utiliza a estratégia DE/rand/1/bin. Em contrapartida, Oliveira [126] verificou em seu trabalho que a escolha da estratégia pouco influenciou os resultados dos casos analisados. É evidente que uma conclusão definitiva sobre a influência da escolha da estratégia precisa ser melhor investigada para uma gama de estudos de caso que representem um domínio importante de problemas de interesse. Além disso, deve ser ressaltado que, para um determinado problema, podem ser escolhidos outros valores para cada um desses parâmetros. Uma vez destacada a fundamentação teórica do algoritmo de DE e as principais características dos seus operadores, cabe ressaltar que no site <http://www.icsi.berkeley.edu/~storn/code/html> pode ser feito o download do algoritmo de DE implementado em várias linguagens de programação, tais como C++, FORTRAN 90, SCILAB®, MATLAB® e MATHEMATICA®, entre outras.

10.3 Aplicação ao Problema Inverso de Transferência Radiativa

Com o sucesso do algoritmo de DE observado em várias aplicações em áreas distintas da ciência e da tecnologia, a sua utilização na resolução de problemas inversos se deu naturalmente. Neste sentido, esta seção é dedicada à resolução do problema inverso de transferência radiativa usando o algoritmo de DE. São tratados dois estudos de caso: um com um meio participante heterogêneo unidimensional em duas camadas [155, 34, 103], e outro considerando um meio participante heterogêneo unidimensional em duas camadas [156, 102]. Os problemas diretos e os problemas inversos correspondentes são descritos, respectivamente, nos Capítulos 2 e 3.

Antes da resolução propriamente dita dos estudos de caso, deseja-se elucidar alguns pontos referentes a estas aplicações. Como não são conhecidos os dados experimentais do presente fenômeno, são admitidos valores para as variáveis de projeto a serem determinadas a fim de resolver o problema direto. Assim, conhecido o perfil da intensidade da radiação em alguns pontos do domínio e com o acréscimo de um ruído previamente definido pelo usuário, definem-se os dados experimentais sintéticos que serão utilizados neste capítulo. Esse acréscimo de ruído ao ponto calculado procura reproduzir os erros de medição que são inerentes a qualquer procedimento experimental. Desse modo, tal aspecto será considerado neste trabalho através da seguinte relação:

$$Y_i = I_{exp_i} = I_{cal_i} + l_i \sigma \quad (10.3.5)$$

onde I_{cal} são os valores calculados para as intensidades da radiação com os valores exatos das incógnitas que se deseja determinar considerados em cada um dos estudos de caso, l_i é um número aleatório entre -1 e 1 e σ é o desvio padrão do erro associado ao procedimento experimental, tal como o apresentado na equação (3.3.6).

Para a resolução do problema direto será considerado, em ambos os estudos de caso, o Método da Colocação Normal com 10 pontos [172, 182]. O termo da integral encontrado do lado direito de cada uma das equações (2.2.6) e (2.2.9, 2.2.10) serão substituídos pela for-

mula de Quadratura de Gauss-Legendre [182]. Desse modo, o problema integro-diferencial original é transformado em um sistema de Equações Diferenciais Ordinárias (EDO's) através do uso de aproximações polinomiais para as variáveis dependentes.

10.3.1 Meio homogêneo unidimensional

Este primeiro estudo de caso considera uma placa plana feita de um material espalhador isotrópico, cujas propriedades não dependem do comprimento de onda da radiação, as superfícies não refletem radiação e estão sujeitas a radiação externa isotrópica. O modelo que descreve esse fenômeno foi apresentado na Subseção 2.2.1.

Para a discretização da variável espacial μ , são utilizados os seguintes valores: $\{-0,9983 -0,9830 -0,9426 -0,8765 -0,7864 -0,6750 -0,5451 -0,4003 -0,2446 -0,0823 0,0823 0,2446 0,4003 0,5451 0,6750 0,7864 0,8765 0,9426 0,9830 0,9983\}$ [34]. Para se comparar a qualidade da solução obtida pelo algoritmo de DE será utilizado o algoritmo de Recozimento Simulado (*Simulated Annealing*, SA), com os parâmetros dados na Tabela 10.1.

| Parâmetro | SA | DE |
|---|--------------------|---------------|
| Número de iterações/gerações | 100 | 100 |
| Tamanho da população | 10 | 10 |
| Probabilidade de cruzamento | - | 0,5 |
| Taxa de Perturbação | - | 0,8 |
| Estratégia | - | DE/rand/1/bin |
| Número de Temperaturas | 50 | - |
| Número de iterações para cada temperatura | 10 | - |
| Temperatura Inicial/Final | 0,5/0,01 | - |
| Estimativa Inicial | [0,25 4,5 0,5 0,5] | aleatória |

Tabela 10.1: Parâmetros utilizados para a resolução do estudo de Caso 1.

Neste estudo, foram admitidos os seguintes valores para as variáveis de projeto para a determinação dos dados experimentais sintéticos: $\omega = 0,1$; $\tau_0 = 5,0$; $A_1 = 1$ e $A_2 = 0$. O espaço de projeto para as variáveis ω , τ_0 , A_1 e A_2 é assim definido: $0 < \omega < 1$, $4 < \tau_0 < 6$, $0 < A_1 < 1,5$ e $0 < A_2 < 1$. Neste caso, Ω pertence ao conjunto $\{1$

2 3 4}, que são os denominados detetores internos, considerados no trabalho com o objetivo de evitar as dificuldades relacionadas à não-unicidade de solução.

A Tabela 10.2 apresenta os resultados obtidos pelos algoritmos evolutivos sem ruído ($\sigma = 0$) e com ruído ($\sigma = 0,005$, o que representa um erro menor que 5%), vide equação (10.3.5). Nesta tabela, NF é o número médio de avaliações da função objetivo e $cputime$ é o tempo médio de execução do algoritmo em minutos utilizando um micro-computador PENTIUM IV com 3,2 GHz e 2 GB de memória RAM. Deve-se destacar que os algoritmos foram implementados em ambiente MATLAB® e que cada um foi executado 10 vezes para a obtenção dos valores médios apresentados nesta tabela. Também são apresentados o melhor (*best*) e o pior (*worst*) resultados obtidos.

| | | ω | τ_0 | A_1 | A_2 | f_{obj} eq. (3.2.4) | |
|-----------------|----------------|---------------|---------------|---------------|---------------|--------------------------|-------------------|
| | | 0,1 | 5,0 | 1,0 | 0,0 | - | |
| DE [§] | $\sigma=0$ | <i>Pior</i> | 0,1024 | 4,9982 | 0,9988 | 0,0013 | 6,35E - 06 |
| | | <i>Média</i> | 0,1004 | 4,9976 | 0,9992 | 0,0000 | 2,61E - 06 |
| | | <i>Melhor</i> | 0,0998 | 5,0036 | 1,0008 | 0,0000 | 1,18E - 07 |
| | $\sigma=0,005$ | <i>Pior</i> | 0,0978 | 4,9438 | 0,9844 | 0,0007 | 8,03E - 04 |
| | | <i>Média</i> | 0,0984 | 4,9470 | 0,9847 | 0,0008 | 8,03E - 04 |
| | | <i>Melhor</i> | 0,0983 | 4,9494 | 0,9850 | 0,0010 | 8,03E - 04 |
| SA [‡] | $\sigma=0$ | <i>Pior</i> | 0,0997 | 5,0097 | 1,0026 | 0,0004 | 8,64E - 07 |
| | | <i>Média</i> | 0,0998 | 4,9981 | 0,9995 | 0,0003 | 7,72E - 07 |
| | | <i>Melhor</i> | 0,0994 | 4,9956 | 0,9988 | 0,0005 | 7,16E - 07 |
| | $\sigma=0,005$ | <i>Pior</i> | 0,0929 | 4,9487 | 0,9789 | 0,0009 | 9,47E - 03 |
| | | <i>Média</i> | 0,0971 | 4,9256 | 0,9848 | 0,0005 | 8,09E - 03 |
| | | <i>Melhor</i> | 0,0987 | 4,9390 | 0,9841 | 0,0004 | 8,06E - 04 |

[§] $NF=1010$, $cputime=21,3065$ min

[‡] $NF=5348$, $cputime=57,4895$ min

Tabela 10.2: Resultados obtidos para o Caso 1 - meio homogêneo unidimensional. NF é o número médio de avaliações da função objetivo.

10.3.2 Meio heterogêneo unidimensional em duas camadas

O próximo estudo de caso considera um meio participante com duas camadas sujeito à radiação externa incidente em ambos os lados conforme descrito na Subseção 2.2.2.

Na presente aplicação são usados os seguintes parâmetros no algoritmo de ED: 10 indivíduos, 100 gerações, taxa de perturbação e probabilidade de cruzamento iguais a 0,8 e a estratégia DE/RAND/1/BIN para a geração de candidatos em potencial.

As propriedades consideradas no presente estudo de caso para a resolução do problema direto são: $L_1=0,8$ cm; $L_2=3,2$ cm; $\sigma_{s1}=0,8$ cm⁻¹; $k_{a1}=0,5$ cm⁻¹; $\sigma_{s2}=0,9$ cm⁻¹; $k_{a2}=0,3$ cm⁻¹; $\rho_1=0,1$; $\rho_2=0,0$; $\rho_3=0,0$; $\rho_4=0,6$; $f_1(\mu)=0,3$ e $f_2(\mu)=1,0$ (vide equações (2.2.9b) e (2.2.10c)). O seguinte espaço de projeto é definido: $0,4 \leq \sigma_{s1} \leq 1,2$, $0 \leq k_{a1} \leq 1$, $0,4 \leq \sigma_{s2} \leq 1,5$ e $0 \leq k_{a2} \leq 1$.

A Tabela 10.3 apresenta os resultados para o problema do meio semitransparente em duas camadas para os dados sem ruído.

Tabela 10.3: Resultados para o problema do meio semitransparente em duas camadas (dados sem ruído).

| | σ_{s1} | k_{a1} | σ_{s2} | k_{a2} | Q (Eq.3.2.4) | DI |
|-------|---------------|----------|---------------|----------|----------------|-----|
| Exato | 0,8 | 0,5 | 0,9 | 0,3 | - | - |
| ED | 0,7999 | 0,5000 | 0,9000 | 0,3000 | 1,14E-10 | Sim |

Na Tabela 10.4, quando ruído é acrescentado (0,1% e 0,2%), o algoritmo de ED consegue obter bons resultados quando detectores internos são usados.

Este problema inverso foi proposto e resolvido em [156] usando as seguintes técnicas: Recozimento Simulado (RS), o Método de Levenberg-Marquardt (LM) e um híbrido envolvendo o Recozimento Simulado e Levenberg-Marquardt (RS-LM). Entretanto, é importante ressaltar que os autores utilizaram outras condições na interface entre os dois meios, o que não permite, portanto, a comparação com os resultados obtidos neste capítulo pelo algoritmo de ED.

Mais detalhes sobre essa aplicação podem ser encontrados em [102].

Tabela 10.4: Resultados para o problema do meio semitransparente em duas camadas (dados com ruído).

| | σ_{s1} | k_{a1} | σ_{s2} | k_{a2} | Q (Eq.3.2.4) | Solução | DI |
|------------|---------------|----------|---------------|----------|----------------|------------------|-----|
| Ruído 0,1% | | | | | | | |
| Exato | 0,8 | 0,5 | 0,9 | 0,3 | - | - | - |
| ED | 0,8010 | 0,4990 | 0,8980 | 0,2990 | 1,96E-06 | Boas estimativas | Sim |
| Ruído 0,2% | | | | | | | |
| Exato | 0,8 | 0,5 | 0,9 | 0,3 | - | - | - |
| ED | 0,7989 | 0,4979 | 0,8960 | 0,2983 | 4,56E-04 | Boas estimativas | Sim |

10.4 Considerações Finais

Como pode ser observado neste capítulo, o algoritmo de DE se configura como uma alternativa bastante interessante para o tratamento de problemas inversos de transferência radiativa com diferentes graus de complexidade e dificuldade. Nas aplicações apresentadas, o algoritmo de DE se mostrou eficiente quando comparado com outras técnicas clássicas e heurísticas, além de ser verificada uma redução no número de avaliações da função objetivo quando comparado com o algoritmo de SA. Esta redução é um fator muito importante, já que esse é o critério mais utilizado pela literatura especializada para fazer comparação entre os inúmeros algoritmos heurísticos que vêm sendo propostos.

Capítulo 11

Projetos de Aplicação

São apresentados a seguir alguns projetos que têm por objetivo a aplicação dos conceitos e dos métodos apresentados nos Capítulos 5 a 10 deste livro.

P1. Utilize os métodos apresentados neste livro para resolver um problema de otimização unidimensional no qual se busca o máximo da função de Easom. Sua expressão, para maximização, é dada por:

$$\max f(x) = -\cos(x)e^{-(x-\pi)^2} \quad (11.0.1)$$

com o valor de x variando no intervalo $[-10; 10]$. O máximo global está localizado em $x^* = \pi$ e possui $f_{\max} = 1$.

P2. Utilize os métodos apresentados neste livro para resolver o problema de otimização bidimensional no qual se busca o máximo da função conhecida como *Four Peaks*. Sua expressão é dada por:

$$\max f(x, y) = e^{-(x-4)^2-(y-4)^2} + e^{-(x+4)^2-(y-4)^2} + 2 \left[e^{-x^2-y^2} + e^{-x^2-(y+4)^2} \right] \quad (11.0.2)$$

onde $(x, y) \in [-5; 5] \times [-5; 5]$. Esta função possui quatro picos, sendo que dois picos correspondem a ótimos locais com $f = 1$ e localizados

em $(-4; 4)$ e $(4; 4)$, e dois outros picos, que correspondem aos ótimos globais, com $f_{max} = 2$ e localizados em $(0; 0)$ e $(0; -4)$. A Fig. 11.1 apresenta os picos da função.

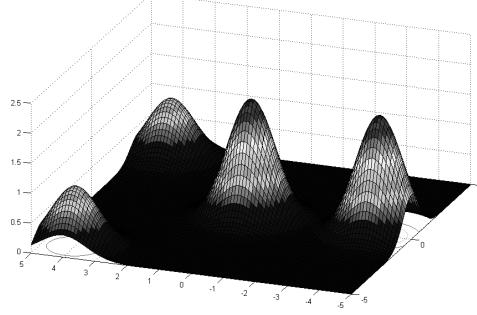


Figura 11.1: Função de otimização com quatro picos.

P3. Utilize os métodos apresentados neste livro para resolver o problema de otimização bidimensional no qual se busca o mínimo da função banana de Rosenbrock. Sua expressão, para minimização, é dada por:

$$\min f(x, y) = (1 - x)^2 + 100(y - x^2)^2 \quad (11.0.3)$$

sabendo que o seu mínimo global encontra-se na posição $(1; 1)$ e possui valor $f_{min} = 0$. Adote como limites do espaço de busca $(-100 \leq x \leq 100)$ e $(-100 \leq y \leq 100)$.

P4. Utilize os métodos apresentados neste livro para resolver o problema de otimização bidimensional no qual se busca o mínimo da função caixa-de-ovo (“eggcrate”):

$$f = x^2 + y^2 + 25(\sin^2 x + \sin^2 y) \quad (11.0.4)$$

com $(x, y \in [-2\pi; 2\pi] \times [-2\pi; 2\pi])$. O mínimo da função está localizado em $(0; 0)$ com $f_{min} = 0$.

P5. Utilize os métodos apresentados neste livro para resolver o problema de otimização no qual se busca o mínimo da função 2D de Michalewicz:

$$f(x, y) = - \left(\sin(x) \left(\sin \left(\frac{x^2}{\pi} \right) \right)^{2m} + \sin(y) \left(\sin \left(\frac{2y^2}{\pi} \right) \right)^{2m} \right) \quad (11.0.5)$$

onde $m = 10$. Adote $(x, y) \in [0, 5] \times [0, 5]$. O mínimo da função ocorre em $(2, 20319; 1, 57049)$ com $f_{\min} \approx -1, 8013$.

P6. Utilize os métodos apresentados neste livro para resolver o problema de otimização n -dimensional no qual se busca o mínimo da função de Griewank. Sua expressão, para minimização, é dada por:

$$\min f(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (11.0.6)$$

e seus limites estão entre $(-600 \leq x_i \leq 600)$ com ótimo global localizado em $x^* = (0; \dots; 0)$ e $f_{\min} = f(x^*) = 0$. Tente resolver esta função com $n \geq 10$, i.e., 10 dimensões ou mais.

P7. Utilize os métodos apresentados neste livro para resolver o problema de otimização bidimensional no qual se busca o mínimo da função “*six-hump camel back*”:

$$f(x_1; x_2) = (4 - 2, 1x_1^2 + x_1^{4/3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \quad (11.0.7)$$

com $x_1 \in [-3; 3]$ e $x_2 \in [-2; 2]$. Os mínimos da função estão localizados em $(x_1; x_2) = (-0, 0898; 0, 7126)$ e $(0, 0898; -0, 7126)$ com $f_{\min} = -1, 0316$.

P8. Utilize os métodos apresentados neste livro para resolver o pro-

blema de otimização bidimensional no qual se busca o mínimo da função Goldstein-Price:

$$f(x_1; x_2) = (1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) \times (30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)) \quad (11.0.8)$$

limitada em $x_i \in [-2; 2], i = 1, 2$. O mínimo da função está localizado em $(x_1; x_2) = (0; -1)$ e $f_{\min} = 3$.

P9. Projeto ótimo de uma mola. Utilize os métodos de otimização apresentados neste livro para resolver o problema de projeto ótimo de uma mola de tensão e compressão, onde o objetivo é minimizar o peso sujeito a restrições de deflexão, estresse, frequência de onda e geometria [183]. Este problema envolve a determinação de três variáveis: o diâmetro do fio x_1 ; o diâmetro da bobina x_2 ; e o comprimento da bobina x_3 . Sua expressão é apresentada a seguir:

$$\min f(\vec{x}) = x_1^2 x_2 (2 + x_3) \quad (11.0.9a)$$

sujeito a

$$g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \quad (11.0.9b)$$

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566 (x_1^3 x_2 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0 \quad (11.0.9c)$$

$$g_3(\vec{x}) = 1 - \frac{140,45 x_1}{x_2^2 x_3} \leq 0 \quad (11.0.9d)$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1,5} - 1 \leq 0 \quad (11.0.9e)$$

onde $\vec{x} = (x_1, x_2, x_3)$. Os limites das variáveis são $0,05 \leq x_1 \leq 2,0$; $0,25 \leq x_2 \leq 1,3$; e $2,0 \leq x_3 \leq 15,0$. Um limite inferior para a solução é: $f_{\min} = f(x^*) = 0,012665$ obtido com os valores $x^* = (0,051690; 0,356750; 11,287126)$.

Sugestão: considere o uso do método das penalidades [42] para inviabilizar as soluções que violam as restrições.

P10. Problema do Caixeiro Viajante (PCV). Conforme descrito na Seção 7.2, o PCV é estabelecido do seguinte modo: um caixeiro viajante deve visitar n cidades e retornar à cidade de onde partiu, sendo que, com exceção da cidade inicial, cada cidade deve ser visitada apenas uma vez. O objetivo é minimizar o custo dessa viagem.

Um grafo deve representar o conjunto de cidades, onde cada nó representa uma única cidade do problema e os arcos representam a união entre as cidades, sendo que, cada arco que une duas cidades, é valorado pelo custo da viagem entre essas cidades. Utilize o Algoritmo de Colônia de Formigas (Capítulo 7) para obter uma boa solução para este problema.

Nas Tabelas 11.1 e 11.2 são apresentadas respectivamente as coordenadas cartesianas e as distâncias para um exemplo com quatro cidades. Na Fig. 11.2 é apresentada uma representação deste exemplo com quatro cidades na forma de um grafo. Use este caso para um teste preliminar para o PCV, considerando que a distância entre as cidades representa o custo para cada arco.

| Cidade | x | y |
|--------|-----|-----|
| A | 1 | 1 |
| B | 3 | 2 |
| C | 2 | 4 |
| D | 5 | 5 |

Tabela 11.1: Coordenadas cartesianas de quatro cidades.

| Distâncias | A | B | C | D |
|------------|-------|-------|-------|-------|
| A | 0,0 | 2,236 | 3,162 | 5,657 |
| B | 2,236 | 0,0 | 2,236 | 3,606 |
| C | 3,162 | 2,236 | 0,0 | 3,162 |
| D | 5,657 | 3,606 | 3,162 | 0,0 |

Tabela 11.2: Distâncias entre as cidades da Tabela 11.1.

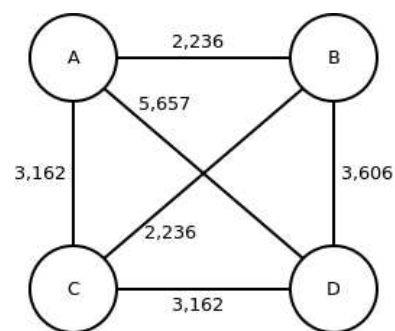


Figura 11.2: PCV da Tabela 11.1 na forma de grafo. As distâncias não estão representadas em escala.

Capítulo 12

Considerações Finais

Antônio José da Silva Neto
José Carlos Becceneri

Ao se concluir a leitura de um livro que relata os resultados obtidos por um conjunto de pesquisadores que vêm trabalhando já há algum tempo na aplicação de técnicas de otimização a problemas inversos em uma determinada classe de problemas, antecipamos que surja naturalmente para o leitor a pergunta sobre qual seria o melhor método a aplicar.

Obviamente não existe um método que seja o melhor para qualquer problema, e nem mesmo para problemas dentro de uma mesma classe. Somente para citar um exemplo, o próprio avanço da computação torna atraentes hoje métodos que anteriormente eram inviáveis devido ao elevado esforço computacional requerido.

A experiência dos autores tem levado à observação de que uma estratégia promissora é aquela que emprega uma hibridização de métodos, buscando aproveitar as melhores características de cada método envolvido. Esta seria a nossa melhor aposta.

Terminado este trabalho, já estamos buscando testar novas meta-heurísticas, como por exemplo aquelas que consideram vermes luminosos [90], abelhas [2], dentre muitos outros. É a eterna curiosidade humana, que leva sempre à busca por algo novo e que potencialmente permita se fazer algo melhor.

Bibliografia

- [1] E. Aarts e J. Korst, “Simulated Annealing and Boltzmann Machines”, John Wiley and Sons Editors, 1989.
- [2] H. A. Abbass, MBO: Marriage in Honey Bees Optimization, A Haplometrosis Polygynous Swarming Approach. *Proceedings of the IEEE Congress on Evolutionary Computation*, (2001), 207–214.
- [3] H. A. Abbass, R. Sarker e C. Newton, PDE: A pareto-frontier differential evolution approach for multi-objective optimization problems. *Proceedings of the 2001 Congress on Evolutionary Computation*, (2001).
- [4] B. T. Abreu, E. Martins e F. L. de Sousa, Generalized Extremal Optimization: a competitive algorithm for test data generation, *Anais do XXI Simpósio Brasileiro de Engenharia de Software*, 15-19 de outubro, João Pessoa, Brasil, 2007.
- [5] D. H. Ackley, G. E. Hinton e T. J. Sejnowski, A Learning Algorithm for Boltzmann Machines, *Cognitive Science*, **9** (1985), 147-169.
- [6] R. Angira e B. V. Babu, Non-dominated sorting differential evolution (NSDE): An extension of differential evolution for multi-objective optimization. *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*. India: IICAI-2005.

- [7] R. Angira e A. Santosh, Optimization of dynamic systems: a trigonometric differential evolution approach, *Computers and Chemical Engineering*, **31** (2007), 1055-1063.
- [8] M. Arenales, V. Armentano, R. Morabito e H. Yanasse, "Pesquisa Operacional para cursos de Engenharia". Elsevier Editora Ltda. Ed. 2006.
- [9] B. V. Babu e B. Anbarasu, Multi-objective differential evolution (MODE): An evolutionary algorithm for multi-objective optimization problems (MOOPS). *Proceedings of International Symposium and 58th Annual Session of IChE*. Indian, CHEMCON-2005, 2005 .
- [10] B. V. Babu, P. G. Chakole e J. H. S. Mubeen, Multiobjective differential evolution (mode) for optimization of adiabatic styrene reactor, *Chemical Engineering Science*, **60** (2005), 4822-4837.
- [11] P. Bak e K. Sneppen, Punctuated equilibrium and criticality in a simple model of evolution, *Physical Review Letters*, v. 71, **24** (1993), 4083-4086.
- [12] A. Battaglia e S. Mantovani, Forward Monte Carlo computations of fully polarized microwave radiation in non-isotropic media, *J. Quant. Spectrosc. Radiat. Transfer*, **95** (2005), 285-308.
- [13] J. C. Becceneri e S. Sandri, Function optimization using ant colony systems with pheromone dispersion. *Proceedings of the XI International Conference on Information, Proceedingsessing and Management of Uncertainty in Knowledge-based Systems (IPMU)*, França, 2006.
- [14] J. C. Becceneri, S. Sandri e E. F. P. Luz. Using Ant Colony Systems with Pheromone Dispersion in the Traveling Salesman Problem, *Proceedings of 11th International Conference of the Catalan Association for Artificial Intelligence*, Espanha, 2008.
- [15] J. C. Becceneri e A. Zinober, Extraction of Energy in a Nuclear Reactor. XXXIII Simpósio Brasileiro de Pesquisa Operacional,

- Campos do Jordão, SP, 2001. *Anais do XXXIII Simpósio Brasileiro de Pesquisa Operacional*, em CD-ROM.
- [16] J. C. Becceneri, S. Stephany, H. F. Campos Velho e A. J. Silva Neto, Solution of the inverse problem of radiative properties estimation with the particle swarm optimization technique, *Proceedings of 14th Inverse Problems in Engineering Seminar*, Ames, EUA, 2006.
- [17] J. V. Beck, Combined parameter and function estimation in heat transfer with application to contact conductance, *J. Heat Trans.*, **110** (1998), 1046-1058.
- [18] R. Bennartz e R. Preusker, Representation of the photon path-length distribution in a cloudy atmosphere using finite elements, *J. Quant. Spectrosc. Radiat. Transfer*, **58** (2006), 202-219.
- [19] L. N. Biondi, Inteligência Computacional Aplicada a Problemas Inversos de Transferência Radiativa, *Anais do II Workshop de Problemas Inversos e Inteligência Computacional*, (2008), Rio de Janeiro. II Workshop de Problemas Inversos e Inteligência Computacional, 2008.
- [20] C. Blum, Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews*, **2** (2005), 353-373.
- [21] S. Boettcher e A. Percus, Optimization With Extremal Dynamics, *Physical Review Letters*, **86** (2001), 5211-5214.
- [22] E. Bonabeau, M. Dorigo e G. Theraulaz, "From natural to artificial swarm intelligence". Oxford University Press, 1999.
- [23] R. P. Brent, "Algorithms for Minimization without Derivatives", Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [24] D. S. Broomhead e D. G. Lowe, Radial basis functions, multivariable interpolation and adaptive networks, *RSRE memo 4148*, Royal Signals Radar Est., Malvern, 1988.

- [25] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz e E. Bonabeau, "Self-organization in Biological Systems". Princeton University Press, 2003.
- [26] H. F. Campos Velho, "Problemas Inversos em Pesquisa Espacial", SBMAC - Sociedade Brasileira de Matemática Aplicada e Computacional, São Carlos, 2008.
- [27] R. F. Carita Montero, N. C. Roberty e A. J. Silva Neto, Reconstruction of a combination of absorption and scattering coefficients with a discrete ordinates method consistent with the source-detector system, *Inverse Prob. Eng.*, **12** (2004), 103-121.
- [28] G. A. Carpenter e S. Grossberg, The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network, *Computer*, v.21, **3** (1988), 77-88.
- [29] J. N. Carter, Genetic Algorithms for Incore Fuel Management and other recent developments in optimization, *Advances in Nuclear Science and Technology, Volume 25*, Plenum Press, New York, 1997.
- [30] A. C. Carvalho, H. F. Campos Velho, S. Stephany, R. P. Souto, J. C. Becceneri e S. Sandri, Fuzzy Ant Colony Optimization for Estimating Chlorophyll Concentration Profile in Offshore Sea Water, *Proceedings of Inverse Problems, Design and Optimization Symposium*, Florida, EUA, 2007.
- [31] E.S. Chalhoub, Discrete-Ordinates solution for radiative transfer problems, *J. Quant. Spectrosc. Radiat. Transfer*, **76** (2003), 193-206.
- [32] E.S. Chalhoub, Discrete-Ordinates solution for uncoupled multi-wavelength radiative transfer problems, *J. Quant. Spectrosc. Radiat. Transfer*, **92** (2005), 335-349.
- [33] E. S. Chalhoub, H. F. Campos Velho, R. D. M. Garcia e M. T. Vilhena, A comparison of radiances generated by selected methods of solving the radiative-transfer equation, *Transport Theor. Stat. Phys.*, **32** (2003), 473-503.

- [34] E. S. Chalhoub, H. F. Campos Velho e A. J. Silva Neto, A comparison of the one-dimensional radiative transfer problem solutions obtained with the monte carlo method and three variations of the discrete ordinates method. *Proceedings of 19th International Congress of Mechanical Engineering - Cobem 2007*, Brasília-DF.
- [35] S. Chandrasekhar, “Radiative Transfer”, Oxford University Press, Londres, 1950.
- [36] E. Charniak e D. McDermott, “Introduction to Artificial Intelligence”, Reading Mass, Addison-Wesley, 1985.
- [37] L. D. Chiwiacowsky e Campos Velho, Different Approaches for the Solution of a Backward Heat Conduction Problem, *Inverse Problems in Engineering*, v.11, **3** (2003), 471-494.
- [38] G. A. G. Cidade, A. J. Silva Neto e N. C. Roberty, “Restauração de Imagens com Aplicações em Biologia e Engenharia - Problemas Inversos em Nanociência e Nanotecnologia”, SBMAC - Sociedade Brasileira de Matemática Aplicada e Computacional, São Carlos, 2003.
- [39] W. F. Clocksin e C. S. Mellish, “Programming in Prolog”, Springer-Verlag Berlin Heidelberg, New York, 1984.
- [40] A. Colmerauer, H. Kanoui et al. e V. Caneghem, Prolog, bases théoriques et développements actuels, *Technique et science informatiques*, Hermès, v. 2, **4** (1982).
- [41] A. Corana, M. Marchesi, C. Martini e S. Ridella, Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing, *ACM Transactions on Mathematical Software*, **13** (1987), 262-280.
- [42] A. Correia, J. Matias e C. Sorôndio, Métodos de Penalidade Exacta para Resolução de Problemas de Optimização não Linear, *Investigação Operacional*, **28** (2008), 17-30

- [43] J. D. Cowan e S. H. David, “The artificial intelligence debate: false starts, real foundations”, MIT Press, Cambridge, MA, USA, 1989.
- [44] A. P. C. Cuco, A. J. Silva Neto, H. F. Campos Velho e F. L. de Souza, Solution of an inverse adsorption problem with an Epidemic Genetic Algorithm and the Generalized Extremal Optimization Algorithm, *Inverse Problems in Science and Engineering*, **17** (2009), 289-302.
- [45] A. P. C. Cuco, F. L. de Sousa, V. V. Vlassov e A. J. Silva Neto, Multi-Objective Design Optimization of a New Space Radiator, *Proceedings of International Conference on Engineering Optimization - EngOpt2008*, Rio de Janeiro, 01-05 Junho, 2008.
- [46] C. Darwin, “On the origin of Species by Means of Natural Selection”, John Murray, Londres, 1859.
- [47] S. K. Das, C.-W. Chiang e J.-B. Nee, Characteristics of cirrus clouds and its radiative properties based on lidar observation over Chung-Li, Taiwan, *Atmospheric Research*, (2009), no prelo.
- [48] L. P. Deutsch, A LISP machine with very compact programs, *IJCAI*, (1973), 697-703.
- [49] M. Dorigo “Ottimizzazione, Apprendimento Automático, Ed Algoritmi Basati su Metafora Naturale”, Ph. D. Dissertation, Politécnico di Milano, Italy, 1992.
- [50] M. Dorigo, V. Maniezzo e A. Coloni. “Positive Feedback as a Search Strategy”. Tech. Rep. No. 91-016, Politecnico di Milano, Italy, 1991.
- [51] M. Dorigo, V. Maniezzo e A. Coloni, The ant system: optimization by a colony of cooperating agents. *IEEE T. on Syst. Man Cy. B*, v. 26, **2** (1996), 29-41.
- [52] J. J. Duderstadt e L. J. Hamilton, “Nuclear Reactor Analysis”, John Wiley and Sons, New York, 1976.

- [53] G.S. Dulikravich, T.J. Martin e B.H. Dennis, Multidisciplinary inverse design, Proceedings Mini-Symposium on Inverse Problems - State of the Art and Future Trends, *Anais do XXIV Congresso Nacional de Matemática Aplicada e Computacional*, Belo Horizonte, 24-33, 2001.
- [54] R.C. Eberhart e Y. Shi, Particle swarm optimization: developments, applications and resources, *Proceedings of Congress on Evolutionary Computation*, (2001), Seoul, Korea, Piscataway, NJ: IEEE, 2001, p. 81-86.
- [55] A. E. Eiben e J.E. Smith, "Introduction to Evolutionary Computing", Springer-Verlag, Berlin, 2003.
- [56] H. Y. Fan e J. Lampinen, A trigonometric mutation operation to differential evolution, *Journal of Global Optimization*, **27** (2003), 105-129.
- [57] D. B. Fogel (Editor), "Evolutionary Computation, The Fossil Record", IEEE Press, New York, 1998.
- [58] L. R. Foulds, "Combinatorial Optimization for Undergraduates". Springer-Verlag, New York, 1984.
- [59] F. L. G. Frege, "Die Grundlagen der Arithmetik". 1884. Eine logisch mathematische Untersuchung über den Begriff der Zahl, Breslau, 1884.
- [60] R. L. Galski, "Desenvolvimento de versões aprimoradas, híbridas, paralela e multiobjetivo do método da otimização extrema generalizada e sua aplicação no projeto de sistemas espaciais", Tese de Doutorado, Curso de Pós-graduação em Computação Aplicada, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2006.
- [61] R. L. Galski, F. L. de Sousa, F. M. Ramos e A. J. Silva Neto, Application of a GEO+SA hybrid optimization Algorithm to the Solution of an Inverse Radiative Transfer Problem, *Inverse Problems in Science and Engineering, Special Issue: Selected papers of IPDO 2007*, v. 17, **3** (2009), 321-334.

- [62] M. R. Garey e D. S. Johnson, “Computers and Intractability: a Guide to the Theory of NP-Completeness”, W. H. Freeman, 1979.
- [63] M. C. C. Gil, A. J. Silva Neto, L. N. Biondi e F. J. C. P. Soeiro, Determinação do Albedo de Espalhamento Simples em Um Meio Participante Homogêneo Unidimensional através de Redes Neuro-Fuzzy, *Anais do V Congresso Nacional de Engenharia Mecânica*, Salvador, 2008.
- [64] M. C. C. Gil, A. J. Silva Neto, L. N. Biondi e F. J. C. P. Soeiro, Determinação da Espessura Óptica de Um Meio Participante Homogêneo Unidimensional através de Redes Neuro-Fuzzy, *Anais do XL Simpósio Brasileiro de Pesquisa Operacional*, João Pessoa, 2008.
- [65] F. Glover, Tabu Search, part I. *ORSA Journal on Computing*, v. 1, **3** (1989), 190-206.
- [66] W. L. Goffe, G. D. Ferrier, e J. Rogers, Global Optimization of Statistical Functions with the Simulated Annealing, *Journal of Econometrics*, **60** (1992), 65-99.
- [67] D. E. Goldberg, “Genetic algorithms for search, optimization, and machine learning”, Addison-Wesley, Reading, MA, 1989.
- [68] S. Goonatilake e S. Khebbal, “Intelligent Hybrid Systems”, John Willey and Sons, Chichester, 1995.
- [69] S. Grossberg, “Neural networks and natural intelligence”, MIT Press, Cambridge, MA, USA , 1988.
- [70] P. Hajela, Genetic Search - An Approach to the Nonconvex Optimization Problem, *AIAA Journal*, v. 26, **7** (1990), 1205-1210.
- [71] N.P. Hanan, Enhanced two-layer radiative transfer scheme for a land surface model with a discontinuous upper canopy, *Agr. Forest Meteorol.*, **109** (2001), 265-281.

- [72] R. Hassan, B. Cohanin, O. L. de Weck e G. Venter, A comparison of particle swarm optimization and the genetic algorithm, *Proceedings of Multidisciplinary Design Optimization Specialist Conference*, 1., 2005, Austin, EUA, Austin:AIAA, 2005, p. 1-13.
- [73] T. Hastie, R. Tibshirani e J. Friedman, "The elements of statistical learning", New-York, Springer-Verlag, 2001.
- [74] S. Haykin, "Neural Networks A Comprehensive Foundation" , Second Edition, Prentice Hall, 1999.
- [75] D. O. Hebb, "The organizations of Behavior", Wiley Eds., 1949.
- [76] J. K. Hillier, Shadow-hiding opposition surge for a two-layer surface, *Icarus*, **128** (2001), 265-281.
- [77] A. L. Hodgkin e A. F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, *Journal of Physiology*, **117** (1952), 500-544.
- [78] J. H. Holland, "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, MI, 1975.
- [79] R. Hooke e T.A. Jeeves, Direct Search Solution of Numerical and Statistical Problems, *Journal of the ACM*, **8** (1961), 212-229.
- [80] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences, USA*, pp. 2554-2558, 1982.
- [81] P. Jackson, "Introduction to Expert Systems", Addison-Wesley, Second Edition, 1990.
- [82] M. D. Kapadi e R. D. Gudi, Optimal control of fed-batch fermentation involving multiple feeds using differential evolution, *Proceedings Biochemistry*, **39** (2004), 1709-1721.
- [83] A.T. Kauati, A.J. Silva Neto e N.C. Roberty, A source-detector methodology for the construction and solution of the one-dimensional inverse transport equation, *Inverse Prob. Eng.*, **9** (2001), 45-66.

- [84] S. Kirkpatrick, C. D. Gelatt e M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, v. 220, **4598** (1983), 671 - 680.
- [85] D. C. Knupp, A.J. Silva Neto e W. F. Sacco, Radiative Properties Estimation with a Combination of the Particle Collision Algorithm and the Levenberg-Marquardt Method, *Proceedings of Inverse Problems Symposium*, East Lansing, USA, 2007.
- [86] D. C. Knupp, A.J. Silva Neto e W. F. Sacco, Radiative properties estimation with the Particle Collision Algorithm based on a sensitivity analysis, High Temperatures - High Pressures, (2009), aceito para publicação.
- [87] D. E. Knut, "The Art of Computer Programming". Addison-Wesley Professional, vol. 1-3 Boxed Set, 2a ed., 1998.
- [88] T. Kohonen, An adaptive associative memory principle, *IEEE Transactions on Computers*, v. C-24, **4** (1974), 444-445.
- [89] T. Kohonen, "Self-Organizing Maps", Berlin: Springer, 1997.
- [90] K. N. Krishnanand e D. Ghose, Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm Intelligence*, (2008). DOI: 10.1007/s11721-008-0021-5.
- [91] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan e D. B. Shmoys, "The Traveling Salesman Problem". Chichester. A Wiley Interscience Publication, John Wiley and Sons, 1985.
- [92] R. Lent, "Cem Bilhões de Neurônios, Conceitos Fundamentais de Neurociência", Editora Atheneu, 2001.
- [93] C. Y. Lin e P. Hajela, Genetic Algorithms in Optimization Problems with Discrete and Integer Design Variables, *Journal of Engineering Optimization*, v. 19, (1992), 309-327.
- [94] W. Little e M. L. G. Shaw, A statistical theory of short and long term memory, *Behavioral, Biology*, **14** (1975), 115-133.

- [95] F. S. Lobato e V. Steffen Jr, Engineering system design with multi-objective differential evolution. *Proceedings of 19 International Congress of Mechanical Engineering - Cobem 2007*, Brasília-DF.
- [96] F. S. Lobato, “Otimização Multi-objetivo para o Projeto de Sistemas de Engenharia”, Tese de Doutorado, FEMEC, UFU, Universidade Federal de Uberlândia, 2008.
- [97] F. S. Lobato, E. B. Arruda, M. A. S. Barrozo e V. Steffen Jr, V., Estimation of drying parameters in rotary dryers using differential evolution, *Journal of Physics Conference Series*, **135** (2008), doi:10.1088/1742-6596/135/1/012063.
- [98] F. S. Lobato, P. C. S. Borges, E. J. Ribeiro e V. V. Murata, Kinetic models evaluation for fed-batch alcoholic fermentation. *Proceedings of 10th International Chemical and Biological Engineering Conference (CHEMPOR 2008)*, Braga-Portugal, 2008.
- [99] F. S. Lobato, R. H. Oliveira Filho, M. A. V. Duarte e V. Steffen Jr, Estimação de parâmetros de um filtro adaptativo utilizado como ferramenta para controle de qualidade em linha de montagem de câmbios automotivos através da técnica de evolução diferencial. *Anais do V Congresso Nacional de Engenharia Mecânica*, 2008, Salvador-Ba.
- [100] F. S. Lobato, L. C. Oliveira-Lopes, V. V. Murata e V. Steffen Jr, Solution of multi-objective optimal control problems with index fluctuation using differential evolution. *Proceedings of 6th Brazilian Conference on Dynamics, Control and Applications - DINCON*, 2007, São José do Rio Preto-SP.
- [101] F. S. Lobato, L. C. Oliveira-Lopes, V. V. Murata e V. Steffen Jr., Evolução diferencial aplicada à solução de problemas de controle ótimo com restrições de desigualdade, *Anais do X Encontro de Modelagem Computacional*, CD-ROM, Nova Friburgo-RJ, 2007. ISBN 978-85-85769-35-2.

- [102] F. S. Lobato, V. Steffen Jr e A. J. Silva Neto, Solution of inverse radiative transfer problems in two-layer participating media with differential evolution. *Inverse Problems in Science and Engineering*, **18** (2010), 183-195.
- [103] F. S. Lobato, V. Steffen Jr. e A. J. Silva Neto, Um estudo comparativo usando evolução diferencial e recozimento simulado aplicado a um problema inverso de transferência radiativa, *Anais do X Encontro de Modelagem Computacional*, CD-ROM, Nova Friburgo-RJ, 2007. ISBN 978-85-85769-35-2.
- [104] I. M. L. Lopes, “Controle de Atitude de Satélites Rígido-Flexíveis Usando a Otimização Extrema Generalizada (GEO) com Abordagem Multi-Objetiva”, Dissertação de Mestrado, Curso de Pós-graduação em Engenharia e Tecnologia Espacial, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, 2008.
- [105] I. M. L. Lopes, F.L. de Sousa e L.C.G. Souza, The Generalized Extremal Optimization With Real Codification, *Proceedings of International Conference on Engineering Optimization - EngOpt2008*, Rio de Janeiro, 01-05 June 2008.
- [106] J. Lupo, Defense application of Neural networks, *IEEE Communications magazine*, (1989), 82-88.
- [107] E. F. P. Luz, “Estimação de fonte de poluição atmosférica usando otimização por enxame de partículas”, Dissertação de Mestrado, São José dos Campos: INPE, 2008.
- [108] E. F. P. Luz, J. C. Becceneri e H. F. Campos Velho, A new multi-particle collision algorithm for optimization in a high-performance environment, *Journal of Computational Interdisciplinary Sciences*, **1** (2008), 1-7.
- [109] V. C. Mariani, A. G. Lima e L. S. Coelho, Apparent thermal diffusivity estimation of the banana during drying using inverse method, *Journal of Food Engineering*, **85** (2008), 569-579.

- [110] D. W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, *J. Soc. Industr. Appl. Math.*, **11** (1963), 431-441.
- [111] J. McCarthy, Programs with Common Sense. In Mechanisation of Thought Proceedings, *Proceedings of the Symposium of the National Physics Laboratory*, 77-84, London, U.K., Her Majesty's Stationery Office, 1959.
- [112] J. McCarthy, "A basis for a mathematical theory of computation, Computer Programming and formal systems". North-Holland, 1963.
- [113] J. McCarthy, M. Minsky, N. Rochester e C. Shannon, "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence", 1955, <http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>, .
- [114] W. S. McCulloch e W. Pitts, A Logical Calculus of Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, **5** (1943), 115-133.
- [115] F. L. L. Medeiros, "Algoritmo Genético Híbrido com um Esquema de Busca de Estado Estacionário de Sistemas Dinâmicos". Tese de Mestrado m Computação Aplicada (CAP), Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos (SP), Brasil. 2002.
- [116] "Metaheuristics Network", <http://www.metaheuristics.net/index.php?main=1>, visitado em 26/11/2007.
- [117] N. C. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller e E. Teller, Equation of state calculations by fast computing machines, *Journal of Chemical Physics*, v. 21, **6** (1953), 1087-1092.
- [118] M. Minsky, "Heuristic Aspects of the Artificial Intelligence Problem", MIT Lincoln Laboratory Report 34-55, December, 1956.

- [119] M. Minsky e S. Papert, “Perceptrons”, MIT Press, Cambridge, 1969.
- [120] C. D. Mobley, “Light and Water - Radiative Transfer in Natural Waters”, Academic Press, San Diego, 1994.
- [121] I. Muraoka, R. L. Galski, F. L. de Sousa e F. M. Ramos. Stochastic Spacecraft Thermal Design With Low Computational Cost. *Journal of Spacecraft and Rockets*, v. 43, **6** (2006), 1248-1257.
- [122] M. S. Nagano, M. S, R. Ruiz e L. A. N. Lorena, A Constructive Genetic Algorithm for Permutation Flowshop Scheduling, *Computers and Industrial Engineering*, v. 55 (2008), 195-207.
- [123] J.A. Nelder e R. Mead, A Simplex Method for Function Minimization, *Computer Journal*, **7** (1965), 308-313.
- [124] A. Newell e S. Herbert, “A Human problem solving”, Englewood Cliffs, NJ, USA, Prentice Hall, 1972.
- [125] N. Nilsson, “Principles of Artificial Intelligence”, Tioga Publishing Company, 1980.
- [126] G. T. S. Oliveira, “Estudo e Aplicações da Evolução Diferencial”, Tese de Mestrado, FEMEC, Universidade Federal de Uberlândia, 2006.
- [127] A. C. M. Oliveira e L. A. N. Lorena, Hybrid Evolutionary Algorithms and Clustering Search, *Studies in Computational Intelligence*, v. 75, Springer Verlag, Berlin, 2007.
- [128] M. N. Ozisik, “Radiative Transfer and Interactions with Conduction and Convection”, John Wiley, Nova York, 1973.
- [129] E. L. Passos, “Inteligência Artificial e Sistemas Especialistas”, Livros Técnicos e Científicos - Editora Ltda., 1989.
- [130] H. Patire Júnior, R. L. Galski, F. L. de Sousa, J. N. Hinckel, P. T. Lacava e F. M. Ramos, Film Cooling Temperature Estimation of a 200N Hydrazine Thruster by an Inverse Problem

- Approach, *Anais do 12o Brazilian Congress of Thermal Sciences and Engineering*, ENCIT 2008, Belo Horizonte, 10-14 November, 2008.
- [131] D. Poole, A. Mackworth e R. Goebel, “Computational Intelligence: A Logical Approach”, Oxford University Press, 1998.
- [132] K. V. Price, R. M. Storn e J. A. Lampinen, “Differential Evolution - A Practical Approach to Global Optimization”, Springer - Natural Computing Series, 2005.
- [133] R. Rank Junior, J. C. Becceneri, J. D. S. Silva, Extração de energia de um reator nuclear utilizando Simulated Annealing e Algoritmos Genéticos, *Anais do Simpósio de Pesquisa Operacional e Logística da Marinha*, SPOLM 2007, Rio de Janeiro, 2007.
- [134] V. J. Rayward-Smith, I. H. Osman, C. R. Reeves e G. D. Smith, “Modern Heuristic Search Methods”. John Wiley and Sons Ltd, Baffins Lane, Chichester, West Sussex, England, 1996.
- [135] C. Reeves, Genetic algorithms, *Handbook of Metaheuristics*, F. Glover e G. A. Kochenberger (Ed.), International Series in Operations Research and Management Science, v. 57. New York: Springer, 2003.
- [136] E. Rich e K. Knight, “Artificial Intelligence”, New York, McGraw Hill, 1992.
- [137] F. Rosenblatt, The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review*, v. 65, 6 (1958), 386-408.
- [138] D. E. Rumelhart, G. E. Hinton, e R. J. William, “Learning Internal Representation by Error Propagation in Parallel Distributed Proceedings”, D. E. Rumelhart and J. L. McClelland Eds., MIT Press, Cambridge, M. A., 1986.
- [139] W. F. Sacco e C. R. E. de Oliveira, A New Stochastic Optimization Algorithm based on Particle Collisions, 2005 ANS Annual

- Meeting, *Transactions of the American Nuclear Society*, American Nuclear Society, San Diego, CA, 2005.
- [140] W. F. Sacco, C. R. E. de Oliveira e C. M. N. A. Pereira, Two stochastic optimization algorithms applied to nuclear reactor core design, *Progress in Nuclear Energy*, **48** (2006), 525-539.
 - [141] W. F. Sacco, H. Alves Filho, N. Henderson e C.R.E. de Oliveira, A Metropolis algorithm combined with Nelder-Mead Simplex applied to nuclear reactor core design, *Annals of Nuclear Energy*, **35** (2008), 861-867.
 - [142] W. F. Sacco, C. M. F. Lapa, C. M. N. A. Pereira e H. Alves Filho, A Metropolis Algorithm applied to a Nuclear Power Plant Auxiliary Feedwater System Surveillance Tests Policy Optimization, *Progress in Nuclear Energy*, **50** (2008), 15-21.
 - [143] W. F. Sacco, A. C. Rios-Coelho, N. Henderson e M. M. Ali, The Particle Collision Algorithm, *International Workshop on Stochastic and Applied Global Optimization (SAGO 2008) - Book of Abstracts* (M.M. Ali, E.M.T. Hendrix, A.O. Adewumi, eds.), University of Witwatersrand, Johannesburg, 2008.
 - [144] C. E. Shannon, Programming a Computer for Playing Chess, *Philosophical Magazine*, s.7, v. 41, **314** (1950).
 - [145] A. J. Silva Neto, Problemas inversos: aplicações em Engenharia e Medicina, em “Cápsulas de Ciência” (F.D. Moura Neto, Ed.), 57-86.
 - [146] A. J. Silva Neto, Explicit and implicit formulations for inverse radiative transfer problems, *Proceedings 5th World Congress on Computational Mechanics*, Mini-Symposium MS125 - Computational Treatment of Inverse Problems in Mechanics, Vienna, Austria, 2002. (Invited Lecture)
 - [147] A. J. Silva Neto e N. J. McCormick, An explicit formulation based on the moments of the exit radiation intensity for the one-dimensional inverse radiative transfer problem, *Proceedings 4th*

- International Conference on Inverse Problems in Engineering*, Angra dos Reis, 2002.
- [148] A. J. Silva Neto e F. D. Moura Neto, “Problemas Inversos - Conceitos Fundamentais e Aplicações”, EdUERJ - Editora da Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2005.
- [149] A. J. Silva Neto e M. N. Özisik, An inverse problem of simultaneous estimation of radiation phase function, albedo and optical thickness, *J. Quant. Spectrosc. Radiat. Transfer*, **53** (1995), 397-409.
- [150] A. J. Silva Neto, N. C. Roberty, R. P. F. Pinheiro e N. I. Alvarez Acevedo, Inverse problems explicit and implicit formulations with applications in engineering, biophysics and biotechnology, *Inverse Probl. Sci. Eng.*, **15** (2007), 373-411.
- [151] A. J. Silva Neto e F. J. C. P. Soeiro, The Solution of an Inverse Radiative Transfer Problem with the Simulated Annealing and Levenberg-Marquardt Methods, *Boletim da SBMAC*, v. VII, **1** (2006), 17-30.
- [152] A. J. Silva Neto e F. J. C. P. Soeiro, Solution of Implicitly Formulated Inverse Heat Transfer Problems with Hybrid Methods, Mini-Symposium Inverse Problems from Thermal/Fluids and Solid Mechanics Applications - *Proceedings of 2nd MIT Conference on Computational Fluid and Solid Mechanics*, Cambridge, EUA, 2003.
- [153] A. J. Silva Neto e F. J. C. P. Soeiro, A Combination of Gradient Based and Global Optimization Methods for the Solution of Inverse Heat Transfer Problems, IPES-2001, *Proceedings of Inverse Problems in Engineering Symposium*, College Station, Texas, 2001. (Resumo)
- [154] A. J. Silva Neto e F. J. C. P. Soeiro, Solution of Inverse Heat Transfer Problems Using a Combination of Global and Local Optimization Methods, *Anais do V Encontro de Modelagem Computacional*, 82-92, Nova Friburgo, 2002. ISBN 85-903068-1-X.

- [155] C. A. Silva Neto e A. J. Silva Neto, Estimation of optical thickness, single scattering albedo and diffuse reflectivities with a minimization algorithm based on an interior points method. *Proceedings of 17th International Congress of Mechanical Engineering*, ABCM, São Paulo, Brazil, 2003.
- [156] F. J. C. P. Soeiro e A. J. Silva Neto, Inverse Radiative Transfer Problem in Two-Layer participating Media, *Proceedings of III European Conference on Computational Mechanics Solids, Structures and Coupled Problems in Engineering*, C.A. Mota Soares et.al. (eds.), Lisboa, Portugal, 2006.
- [157] F. J. C. P. Soeiro, P. O. Soares e A. J. Silva Neto, Solution of Inverse Radiative Transfer Problems with Artificial Neural Networks and Hybrid Methods, *Proceedings of 13th Inverse Problems in Engineering Seminar*, 163-169, Cincinnati, EUA, 2004.
- [158] F. L. de Sousa, “Otimização Extrema Generalizada: Um novo algoritmo estocástico para o projeto ótimo”. Tese de Doutorado, Curso de Pós-graduação em Computação Aplicada, Instituto Nacional de Pesquisas Espaciais, INPE-9564-TDI/836, São José dos Campos, 2003.
- [159] F. L. de Sousa, I. Muraoka e R. L. Galski, On The Optimal Positioning of Electronic Equipment In Space Platforms, *Anais do 19th International Congress of Mechanical Engineering - COBEM*, 5-9 November, Brasília, Brasil, 2007.
- [160] F. L. de Sousa. e F. M. Ramos, Function optimization using extremal dynamics. *Proceedings of 4th International Conference on Inverse Problems in Engineering - ICIPE*, Rio de Janeiro, Brazil, May, 2002.
- [161] F. L. de Sousa, F. M. Ramos, P. Paglione e R. M. Girardi, New Stochastic Algorithm for Design Optimization, *AIAA Journal*, v. 41, **9** (2003) 1808-1818.
- [162] F. L. de Sousa, F. M. Ramos, R. L. Galski e I. Muraoka, Generalized Extremal Optimization: A New Meta-heuristic Inspired by

- a Model of Natural Evolution, “Recent Developments in Biologically Inspired Computing” (L.N. De Castro e F.J. Von Zuben, eds.), 41-60, Idea Group Inc, London, 2005.
- [163] F. L. de Sousa, F. J. C. P Soeiro, A. J. Silva Neto e F. M. Ramos, Application of the generalized extremal optimization algorithm to an inverse radiative transfer problem, *Inverse Problems in Science and Engineering*, v. 15, **7** (2007), 699-714.
- [164] F. L. de Sousa e W. K. Takahashi, Generalized Extremal Optimization Applied to Three-Dimensional Truss Design, *Proceedings of 18th International Congress of Mechanical Engineering (COBEM2005)*, CDROM, November 6-11, Ouro Preto, Brasil, 2005.
- [165] R. P. Souto, S. Stephany, J. C. Becceneri, H. F. Campos Velho e A. J. Silva Neto. Reconstruction of Spatial Dependent Scattering Albedo in a Radiative Transfer Problem Using a Hybrid Ant Colony System Implementation and a Pre-Regularization Scheme, *Proceedings 6th World Congress on Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brasil, 2005.
- [166] R. P. Souto, S. Stephany, J. C. Becceneri, H. F. Campos Velho, A. J. Silva Neto, On the use of the ant colony system for radiative properties estimation . 5th International conference on Inverse Problems in Engineering - Theory and Practice (V ICIPE), 2005, Cambridge, Inglaterra. *Proceedings of the 5th ICIPE*. Leeds, Inglaterra : Leeds University Press, 2005. v. 3. p. 1-10.
- [167] R. Storn e K. Price, Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces, *Journal of Global Optimization*, **12** (1997), 1-16.
- [168] A. Turing, “Computing Machinery and Intelligence”, *Mind* LIX, 1950.
- [169] A. Turing, “Can Automatic Calculating Machines be Said to Think?”, Copeland, B. Jack, *The Essential Turing: The ideas that gave birth to the computer age*, 1952.

- [170] V. Vapnik, “Estimation of Dependences Based on Empirical Data”, Springer Series in Statistics, Springer-Verlag New York, Inc., Secaucus, NJ, 1982.
- [171] W. Verhoef e H. Bach, Simulation of hyperspectral and directional radiance images using coupled biophysical and atmospheric radiative transfer models, *Remote Sens. Environ.*, **87** (2003), 23-41.
- [172] J. Villadsen e M.L. Michelsen, “Solution of Differential Equation Models by Polynomial Approximation”, Prentice - Hall, Englewood Clifff, 1978.
- [173] V. V. Vlassov, F. L. de Sousa e W. K. Takahashi. Comprehensive Optimization of a Heat Pipe Radiator Assembly Filled with Ammonia or Acetone. *International Journal of Heat and Mass Transfer*, **49** (2006), 4584-4595.
- [174] C. von der Malsburg, Self-organization of orientation selective cells in the striate cortex, *Kybernetik*, **14** (1973).
- [175] F. S. Wang, T. L. Su e H. J. Jang, Hybrid differential evolution for problems of kinetic parameter estimation and dynamic optimization of an ethanol fermentation Proceedings, *Industry Engineering Chemical Research*, **40** (2001), 2876-2885.
- [176] P. D. Wasserman, “Neural Network Computing”, New York, Van Nostrand Reinhold, 1989.
- [177] P. J. Werbos, “Beyond regression: New tools for prediction and analysis in the behavioral sciences”, PhD. thesis, Harvard University, 1974.
- [178] G.C. Wick, Über ebene Diffusionsproblem, *Z. Phys.*, **120** (1943), 702-705.
- [179] B. Widrow, e S. D. Stearns, “Adaptive Signal Proceedings”, New Jersey, Prentice-Hall, Inc., 1985.

- [180] D. J. Willshaw e C. von der Malsburg, How patterned neural connections can be set up by self-organization, *Proceedings of the Royal Society London B*, **194** (1976), 431-445.
- [181] P. H. Winston, “Artificial Intelligence”, Third Edition, Addison-Wesley, 1992.
- [182] C. R. Wylie e L. C. Barrett, “Advanced Engineering Mathematics”, 5th Edition, McGraw-Hill, Singapura, 1985.
- [183] X.-S. Yang, “Nature-inspired metaheuristic algorithms”. Second edition. Frome, UK: Luniver Press, 2010.
- [184] N. Ziviani, “Projeto de Algoritmos com implementações em Pascal e C”. Editora Thomson Pioneira, 2004.

Índice

- Boltzmann
 - constante de, 50, 51
 - equação de, 21, 25, 27
 - probabilidade de, 50
- contorno
 - condições de, 22, 26, 28
 - superfícies de, 27, 28, 32, 38
- Darwin
 - teoria de, 59
- diferença quadrática, 73, 81
- Gauss-Legendre
 - quadratura de, 114
- hibridização, 92, 98, 105
- Hooke-Jeeves
 - método de, 99
- Levenberg-Marquardt
 - método de, 34–36, 57, 67, 98
- mínimo
 - global, 40, 55, 87
 - local, 40, 50, 54, 56, 93
- Monte Carlo
 - método de, 19
- Ordenadas Discretas
 - método de, 19
 - otimização combinatória, 49, 107
 - paralelização, 80, 105
 - problemas inversos, 19, 26, 29, 39, 48, 74, 78, 81, 82, 100, 108, 111, 113, 117