



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-14443-TDI/1138

**UM ESTUDO DE MÉTODOS ESTOCÁSTICOS EM AMBIENTE
PARALELO PARA A PRODUÇÃO DE MAPAS DE
ANISOTROPIAS DA RADIAÇÃO CÓSMICA DE FUNDO EM
MICROONDAS**

Ricardo Varela Corrêa

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Airam Jônatas Preto e Stephan Stephany, aprovada em 5 de abril de 2005.

INPE
São José dos Campos
2007

519.216:523.165

Corrêa, R. V.

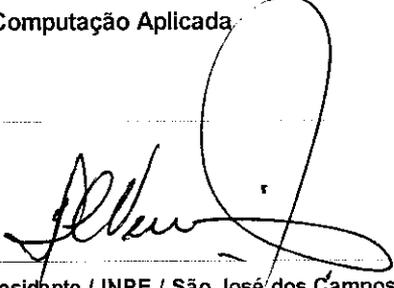
Um estudo de métodos estocásticos em ambiente paralelo para a produção de mapas de anisotropias da radiação cósmica de fundo em microondas / Ricardo Varela Corrêa. – São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), 2005.

102 p.; (INPE-14443-TDI/1138)

1.Recozimento simulado. 2.Algoritmos genéticos.
3.Astrofísica computacional. 4.Radiação cósmica de fundo em microondas. 5.Cosmologia. 6.Processamento paralelo. I.
Título

Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de Doutorado em
Computação Aplicada.

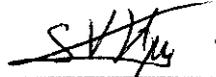
Dr. Haroldo Fraga de Campos Velho


Presidente / INPE / São José dos Campos - SP

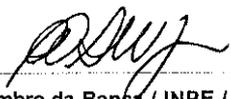
Dr. Airam Jônatas Preto


Orientador(a) / INPE / São José dos Campos - SP

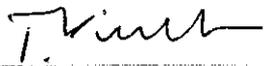
Dr. Stephan Stephany


Orientador(a) / INPE / São José dos Campos - SP

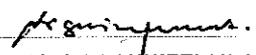
Dr. Carlos Alexandre Wuensche de Souza


Membro da Banca / INPE / São José dos Campos - SP

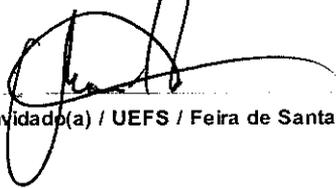
Dr. Thyrso Villela Neto


Membro da Banca / INPE / SJC Campos - SP

Dr. Newton de Figueiredo Filho


Convidado(a) / UNIFEI / Itajubá - MG

Dr. Marildo Geraldete Pereira


Convidado(a) / UEFS / Feira de Santana - BA

Aluno (a): Ricardo Varela Corrêa


São José dos Campos, 05 de abril de 2005

“Our continuing mission: to seek out knowledge of C, to explore strange Unix commands, and to boldly code where no one has man page 4”

Sve Goldt, 1995.

A Niel e Yedda
Faróis em minha vida.

AGRADECIMENTOS

À Solange, Kátia e Rodrigo, obrigado por entenderem a importância desse passo e peço desculpas pela minha ausência nos últimos anos.

Sem colocar uma ordem, esses amigos foram muito importantes tanto em minha vida profissional como pessoal. Meus sinceros agradecimentos.

Ao Dr. Stephan Stephany, agradeço a paciência e a incrível revisão de meu trabalho. Sua orientação segura, criteriosa e constante me acompanhará em todos os aspectos de minha vida profissional. Um amigo nessa cruzada.

Ao Dr. Airam Jonatas Preto, agradeço as orientações recebidas no trabalho aqui apresentado.

Ao Dr. Thyrso Villela e ao Dr. Carlos Alexandre Wuensche, agradeço o apoio sempre presente e, em especial, à amizade que nos une.

Ao Dr. José Humberto Sobral, por ter sido um grande incentivador no início dessa busca pelo doutoramento. Um amigo cujo senso crítico, humanidade e honestidade, são pontos de referência em minha vida profissional.

Ao Dr. Jorge Mejia, a quem agradeço por sua boa vontade e paciência infinita nas discussões técnicas e científicas relacionadas a este trabalho. Sem sua ajuda, o mistério da Radiação Cósmica de Fundo continuaria um mistério.

Ao Rodrigo Leonardi, agradeço a boa vontade e o auxílio em inúmeras partes deste trabalho.

Ao Dr. Celso Luiz Mendes, a quem agradeço pelo incentivo à continuidade de meus estudos.

À Maria Emília, seus ensinamentos constantes e por mostrar que dificuldades existem para serem vencidas, foi um marco nessa vida. O trabalho aqui apresentado é em sua memória.

Ao Ronaldo e Renata, irmãos de sangue e coração, eu agradeço por contar com vocês nessa vida e com certeza continuarei a contar em muitas outras vidas.

Ao Grande Arquiteto do Universo, agradeço por permitir que essas pessoas estivessem em meu caminho.

RESUMO

Mapas de anisotropia da radiação cósmica de fundo em microondas são constituídos por valores de temperaturas correspondentes a essa radiação para pontos discretos (pixels) da região observada do céu. A construção de um mapa envolve a resolução do problema inverso da estimação desses valores de temperatura a partir de séries temporais de medidas de temperatura com ruído. Uma matriz de apontamento relaciona as medidas com os pontos do céu, sendo definida pela estratégia de apontamento do instrumento. Quase todos os métodos existentes de construção de mapas são analíticos, sendo derivados do método dos mínimos quadrados. Esses métodos envolvem alguma forma de inversão de matrizes de grande dimensão que são singulares e esparsas. Devido aos custos computacionais decorrentes, utiliza-se comumente uma alternativa, o método *bin average*, no qual a temperatura estimada de cada pixel é calculada pela média ponderada de suas medidas. Este trabalho propõe uma nova alternativa a estes métodos inversos através de uma abordagem implícita (iterativa). O problema inverso associado é formulado como um problema de otimização e resolvido por um algoritmo estocástico. São avaliadas versões paralelas de algoritmos genéticos e de um algoritmo de recozimento simulado, utilizando-se dados sintéticos corrompidos com ruído. O mapa resultante obtido por este último apresentou melhor qualidade quando comparado com aquele obtido pelo método *bin average* num tempo de execução equivalente.

A STUDY OF STOCHASTIC METHODS IN A PARALLEL ENVIRONMENT FOR THE PRODUCTION OF MAPS OF THE ANISOTROPIES OF THE MICROWAVE BACKGROUND RADIATION

ABSTRACT

Anisotropy maps of the microwave background cosmic radiation are composed of temperature values that correspond to that radiation in discrete points (pixels) of the observed region of the sky. The construction of a map requires the solving of the inverse problem that is the estimation of temperature values from temporal series of noisy temperature measurements. A pointing matrix correlates the measurements to the points of the sky, being defined by the pointing strategy of the instrument. Most methods for map construction are analytical, being derived from the least square method. These methods require the inversion of sparse, singular matrixes that have large dimensions. Due to the high processing costs, a common alternative is the bin average method, in which the estimated temperature of each pixel is given by a weighted average of its temperature measurements. This work proposes a new alternative for map construction by using an implicit (iterative) inverse method. The associated inverse problem is formulated as an optimization problem that is solved by a stochastic algorithm. Parallel versions of genetic algorithms and a simulated annealing algorithm are evaluated using noise-corrupted synthetic data. The resulting map obtained by the latter algorithm presented better quality in comparison to that obtained by the bin average method using an equivalent processing time.

SUMÁRIO

	<u>Pág.</u>
LISTA DE FIGURAS	
LISTA DE TABELAS	
LISTA DE SIGLAS E ABREVIATURAS	
CAPÍTULO 1 - INTRODUÇÃO	21
1.1 - Esboço Geral dos Capítulos	27
CAPÍTULO 2 - MAPAS DA RADIAÇÃO CÓSMICA DE FUNDO EM MICROONDAS	28
2.1 - Construção de Mapas de Anisotropias da RCFM.....	28
2.2 - Breve Descrição do Experimento BEAST	32
2.3 - Formulação do Problema de Construção de Mapas de Anisotropias da RCFM	33
2.3.1 - Método <i>bin average</i>	43
CAPÍTULO 3 - FORMULAÇÃO INVERSA IMPLÍCITA PROPOSTA PARA A CONSTRUÇÃO DE MAPAS DE ANISOTROPIAS DA RADIAÇÃO CÓSMICA DE FUNDO EM MICROONDAS	45
3.1 - Problemas Inversos e a Formulação Implícita Inversa Adotada	46
3.2 - Métodos de Otimização	48
3.2.1 - Algoritmos Genéticos	50
3.2.2 - Recozimento Simulado	57
- Computação Paralela	
3.3	62

CAPÍTULO 4 - RESULTADOS NUMÉRICOS E DE DESEMPENHO	68
4.1 - Resultados Obtidos Utilizando Algoritmo Genético	70
4.2 - Resultados Obtidos Utilizando Recozimento Simulado	75
4.2.1 - Resultados Obtidos para o Algoritmo RS com Distribuição de Cauchy e Solução Candidata Inicial Gerada por Algoritmo Genético	77
4.2.2 - Resultados Obtidos para o Algoritmo RS com Distribuição de Cauchy e Solução Candidata Inicial Amostrada das Medidas de Temperatura	78
4.2.3 - Resultados Obtidos para o Algoritmo RS com Distribuição de Cauchy com <i>re-annealing</i> e Solução Candidata Inicial Amostrada das Medidas de Temperatura	80
4.3 - Análise de Desempenho Computacional	83
CAPÍTULO 5 - CONCLUSÕES	88
REFERÊNCIAS BIBLIOGRÁFICAS	95

LISTA DE FIGURAS

	<u>Pág.</u>
2.1 Mapas de anisotropias da RCFM gerados pelos experimentos COBE e HACME	24
2.2 Mapa de RCFM do experimento BEAST para um período aproximado de 30 dias de aquisição	25
2.3 Exemplo de varredura do experimento BEAST	27
3.1 Visão esquemática de problemas inversos	42
3.2 Pseudo-código do algoritmo genético canônico	48
3.3 Pseudo-código do algoritmo de recozimento simulado	56
4.1 Dados simulados de temperatura e ruído correspondente	69
4.2 Convergência dos AGs testados: valor da função objetivo em função do número de iterações	70
4.3 Diferença entre a temperatura exata e a temperatura estimada pelo AG Boltzmann global para os 100 primeiros pixels do mapa simulado	71
4.4 Convergência dos RSs testados: valor da função objetivo em função do número de iterações	73
4.5 Convergência do RS Cauchy com solução candidata inicial gerada por AG: valor da função objetivo em função do número de iterações.	75
4.6 Convergência do RS Cauchy com solução candidata inicial amostrada das medidas de temperatura: valor da função objetivo em função do número de iterações	76
4.7 Convergência do RS Cauchy com <i>re-annealing</i> e com solução candidata inicial amostrada das medidas de temperatura: valor da função objetivo em função do número de iterações	78
4.8 Módulo diferença entre a temperatura exata e a temperatura estimada para o algoritmo RS com distribuição de Cauchy com e sem <i>re-annealing</i> e para o método <i>bin average</i> , para os primeiros 10 primeiros pixels do mapa simulado	79
4.9 Módulo da diferença entre a temperatura exata e as temperaturas estimadas pelo algoritmo RS com distribuição de Cauchy com <i>re-annealing</i> e pelo método <i>bin average</i> , para uma faixa de 20 pixels do mapa simulado	80

LISTA DE TABELAS

	<u>Pág.</u>
2.1 Experimentos de RCFM no solo e a bordo de satélite e balões estratosféricos	23
2.2 Definição da matriz W para diversos métodos de construção de mapas de anisotropias da RCFM	36
2.3 Requisitos computacionais do método COBE	38
4.1 Comparação das soluções obtidas pelos diversos AGs e pelo método <i>bin average</i>	72
4.2 Comparação das soluções obtidas pelos diversos algoritmos RSs e pelo método <i>bin average</i>	73
4.3 Comparação das soluções obtidas pelos algoritmos RS Cauchy para diversos esquemas de geração de solução candidata inicial e pelo método <i>bin average</i>	77
4.4 Comparação das soluções obtidas pelo algoritmo RS Cauchy com e sem <i>re-annealing</i> e pelo método <i>bin average</i>	79
4.5 Comparação dos tempos de execução do método <i>bin average</i> e AG com distribuição Boltzmann e paralelização modelo global ou ilha	81
4.6 Comparação dos tempos de execução do método <i>bin average</i> e RS com distribuições de Boltzmann, Cauchy e Uniforme	82
4.7 Comparação dos tempos de execução do método <i>bin average</i> e RS Cauchy para diversos esquemas de solução candidata inicial	83
4.8 Comparação dos tempos de execução do método <i>bin average</i> e RS com e sem <i>re-annealing</i>	84

LISTA DE SIGLAS E ABREVIATURAS

AG	Algoritmo Genético
BEAST	<i>Background Emission Anisotropy Scanning Telescope</i>
COBE	<i>Cosmic Background Explorer</i>
DAS	Divisão de Astrofísica
FSA	<i>Fast Simulated Annealing</i>
GB	<i>Giga Byte</i>
GHz	<i>Giga Hertz</i>
GSA	<i>Generalized Simulated Annealing</i>
HACME	<i>HEMT Advanced Cosmic Microwave Explorer</i>
HBB	<i>Hot Big Bang</i>
IDL	<i>Interactive Data Language</i>
INPE	Instituto Nacional de Pesquisas Espaciais
IP	<i>Internet Protocol</i>
JPL	<i>Jet Propulsion Laboratory</i>
K	Graus Kelvin
LAC	Laboratório Associado de Computação e Matemática Aplicada
MADCAP	<i>Microwave Anisotropy Dataset Computational Analysis Package</i>
MCP	Modelo Cosmológico Padrão
MPI	<i>Message Passing Interface</i>
PC	<i>Personal Computer</i>
STD	Série Temporal de Dados
RCF	Radiação Cósmica de Fundo
RCFM	Radiação Cósmica de Fundo em Microondas
RS	Recozimento Simulado
TB	<i>Tera Byte</i>
UCSB	<i>University of California – Santa Barbara</i>

LISTA DE SÍMBOLOS

T	Temperatura do céu
S	Série temporal de dados
M	Dimensão do vetor série temporal de dados
N	Número de pixels de uma subregião
Δ_n	Ruído do pixel n
d_n	Medida de temperatura do pixel n
T_{en}	Temperatura efetiva do pixel n
$A_{M \times N}$	Matriz de apontamento
A^{-1}	Matriz inversa de A
A^T	Matriz transposta de A
P	Distribuição de probabilidade
σ	Desvio padrão
μ	Média aritmética
τ	Pseudo-temperatura
χ	Função qui-quadrado
C	Matriz de covariâncias
\bar{T}_i	Temperatura média do pixel i
m^{exp}	Vetor de medidas experimentais
m^{mod}	Vetor de solução candidata
f	Função de aptidão
\bar{f}	Média das aptidões dos indivíduos da população
p	Fator de mutação
g	Valor em ponto flutuante de um gene
K_B	Constante de Boltzmann
E	Energia
ΔE	Varição de energia
ξ	Fator de ajuste da pseudo-temperatura
t_{com}	Custo de comunicação

β	Dimensão da informação transmitida
t_l	Tempo de latência
t_w	Tempo de transmissão dos dados
$\alpha(p)$	<i>Speed up</i> para p processadores
ΔT_{seq}	Tempo de execução do programa seqüencial
ΔT_{par}	Tempo de execução do programa paralelo
$\eta(p)$	Eficiência do paralelismo
f_{seq}	Fração seqüencial do programa paralelo
f_{par}	Fração paralela do programa paralelo
ρ	Correlação de Pearson
α	Desvio Médio

CAPÍTULO 1

INTRODUÇÃO

O Modelo Cosmológico Padrão (MCP), também chamado de modelo do *Hot Big Bang* (HBB), descreve que o Universo teria começado a partir de um estado de temperatura e densidade extremamente altas, com uma posterior expansão, em que as condições existentes levariam a uma sucessão de estados de quase equilíbrio térmico nos quais a temperatura e densidade diminuiriam com o decorrer do tempo. As condições após a criação tornaram possível a formação de partículas, átomos, moléculas e finalmente a formação das estruturas cosmológicas que hoje existem [1].

O MCP apóia-se nos seguintes fatos observacionais:

- A abundância de elementos leves na atmosfera das estrelas, os quais só poderiam ter sido produzidos via nucleossíntese primordial;
- A expansão do Universo (Lei de Hubble);
- O espaço é banhado por uma radiação térmica, chamada de Radiação Cósmica de Fundo em Microondas (RCFM), cuja temperatura média atual é de $\sim 2,716$ K.

Esses fatos permitem que se estabeleça a hipótese de que as propriedades de uma determinada região observada do Universo em grande escala são as mesmas em todas as outras regiões. A melhor evidência dessa isotropia do Universo consiste na distribuição observada da temperatura de RCFM [2]. O presente capítulo apresenta uma breve introdução sobre a RCFM.

No período de até aproximadamente 380.000 anos após essa explosão primordial, haveria um plasma fortemente acoplado a uma radiação eletromagnética e sem possibilidade de combinação de prótons e elétrons. Com o resfriamento do Universo, átomos se formaram, uma vez que os prótons passaram a ter energia média do plasma

suficientemente baixa para manter os átomos de hidrogênio. Este período é designado de “período de recombinação”. Com a diminuição do número de elétrons livres, a matéria e a radiação não mais interagiram de forma significativa, caracterizando o que se denomina de desacoplamento entre a radiação e a matéria. Os fótons desacoplados da matéria formaram a RCFM, a qual ocupa uma hipersuperfície que corresponde à região onde os fótons foram espalhados pela última vez antes de serem detectados na Terra.

A RCFM foi descoberta por A. Penzias e R. Wilson em 1965 [3], e sua máxima intensidade é observada no comprimento de onda $\lambda \approx 2$ mm (50 GHz). Essa radiação fornece informação do Universo quando ele tinha cerca de 380 mil anos e pode ser estudada a partir de medidas de seu espectro, polarização e distribuição angular espacial [4].

Em particular, as medidas realizadas pelo satélite *Cosmic Background Explorer* (COBE) [5] permitiram observar flutuações que podem ser diretamente associadas a traços de perturbações da densidade primordial, o que fornece importantes informações sobre as condições iniciais do Universo [6]. As pequenas flutuações de temperatura observadas, denominadas anisotropias, estão intimamente ligadas à formação das estruturas complexas existentes no Universo observável hoje.

A partir da medida da distribuição angular da RCFM, é possível a construção de mapas de anisotropias dessa distribuição, os quais permitem que seja obtida uma imagem do estado do Universo num período passado. As regiões de temperaturas mais altas correspondem a regiões de alta densidade e regiões de temperaturas mais baixas correspondem a regiões de baixa densidade.

As observações da RCFM do HBB são obtidas por instrumentos, tipicamente radiotelescópios, a bordo de satélites, embarcados em balões estratosféricos ou no solo. A construção de mapas de anisotropias da RCFM é feita a partir das medidas de flutuações de temperatura dessa radiação em pontos discretos (pixels) do céu [7]. Essas medidas incluem diversos ruídos. A quantidade de pixels de um dado mapa está

relacionada à resolução do instrumento de medida e à área do céu observada. A resolução espacial do mapa depende da resolução angular do instrumento. Esses dados são amostrados repetidamente formando uma série temporal para cada pixel. Uma matriz de apontamento relaciona os dados com os pontos do céu.

A construção de um mapa envolve a resolução do problema inverso de estimar as temperaturas para cada pixel a partir da série temporal de dados de medidas correspondente utilizando-se uma matriz de apontamento conhecida. Quase todos os métodos de construção de mapas atuais são analíticos, sendo derivados do método dos mínimos quadrados. Esses métodos envolvem alguma forma de inversão de matrizes de grandes dimensões que são, em geral, singulares e esparsas. Devido aos custos computacionais decorrentes, é muito utilizado o método *bin average*, que emprega uma redução do volume de dados, ao estimar o valor de temperatura de cada pixel a partir das médias aritméticas das medidas de temperatura desse pixel para o período considerado. Este método é derivado do chamado método COBE [8], que é o mais referenciado na literatura dessa área.

A representação gráfica das temperaturas estimadas constitui o mapa de anisotropias da RCFM, em que cada pixel corresponde à temperatura estimada na posição correspondente à posição do céu em que o telescópio efetuou as medidas. Note-se que nesta representação gráfica utiliza-se como valor de referência a temperatura média atual da RCFM, e portanto o mapa representa as diferenças de cada pixel em relação a esse valor, ou seja, as anisotropias.

Dependendo do tipo de estratégia observacional empregada por cada experimento, mapas referentes a períodos de aquisição consecutivos podem ser acumulados através de uma média ponderada dos valores das correspondentes anisotropias. Como exemplo, o mapa gerado pelo satélite COBE, lançado em 1989, representa o acúmulo de mapas diários num total de um ano de aquisição.

Desde então, novos experimentos com detectores de melhor resolução angular ou com técnicas diferentes de varredura estão em operação ou em projeto. Dentre eles, destaca-se um experimento desenvolvido para medidas da RCFM resultante de uma cooperação entre instituições norte-americanas, brasileiras e européias, denominado *Background Emission Anisotropy Scanning Telescope* (BEAST) [9]. Em razão de sua maior resolução angular, este experimento gera cerca de 100 vezes mais dados de medidas do que o experimento COBE.

Nos experimentos de RCFM, o telescópio realiza uma varredura no céu em que é associada uma temperatura para cada pixel da região do céu observada pelo instrumento. A estratégia de apontamento, ou seja, a forma como o telescópio executa essa varredura do céu, faz com que cada pixel representando uma dada posição no céu seja amostrado diversas vezes.

Tegmark [10] especifica 10 métodos inversos para solução do problema de construção dos mapas de anisotropias da RCFM. Destes, 8 são métodos analíticos lineares derivados do método dos mínimos quadrados e os demais são métodos não lineares. Alguns pacotes de aplicativos estão disponíveis na *Internet*, como no caso do aplicativo *Microwave Anisotropy Dataset Computational Analysis Package* (MADCAP) [11], desenvolvido pelo *National Energy Research Scientific Computing Center* (NERSC). A Divisão de Astrofísica (DAS) do INPE utiliza atualmente o método *bin average* implementado na forma de rotinas desenvolvidas na linguagem IDL *Interactive Data Language* (IDL) [12].

Este trabalho propõe uma nova alternativa aos métodos analíticos através de uma abordagem implícita (iterativa) para o problema inverso, o qual é formulado como um problema de otimização e resolvido por um algoritmo estocástico. Um aspecto importante é que o método proposto permite realizar estimativas a partir de todo o volume de dados, sem redução deste volume de dados, o que em tese possibilitaria mapas de melhor qualidade. Para a avaliação do método proposto, utilizaram-se dados sintéticos corrompidos com ruído, de forma a simular os dados do experimento BEAST,

que possui $O(10^4)$ pixels com $O(10^5)$ medidas para cada pixel. Este volume de dados não poderia ser tratado por métodos tradicionais sem redução de seu volume num tempo de processamento razoável [13], exceto pelo método *bin average*.

O presente trabalho simula os dados e a forma de armazenamento de dados do experimento BEAST, em que as medidas são agrupadas em séries que contêm uma hora de aquisição, sendo varrida a cada hora uma determinada região do céu.

Para geração dos dados simulados, os valores de temperatura são gerados aleatoriamente, constituindo uma solução exata que será utilizada para avaliar o novo método de construção de mapas da RCFM proposto neste trabalho. Estas temperaturas são contaminadas com ruído gaussiano reproduzindo a relação sinal/ruído das medidas reais do BEAST.

Nos experimentos de RCFM, o ruído $1/f$, causado pelas flutuações de ganho dos amplificadores, é filtrado com um filtro do tipo ‘passa-alta’, devido à sua baixa frequência. Por generalidade, neste trabalho, os dados simulados representam o sinal já filtrado.

A avaliação da qualidade das soluções obtidas foi feita por três critérios: correlação de Pearson entre a solução obtida e a exata, desvio médio da diferença entre essas soluções [14], e tempo de execução para se obter a solução.

Em razão do aumento de resolução angular de novos instrumentos de medidas e o conseqüente aumento do volume de dados de medidas, a busca por métodos computacionalmente mais eficientes é um campo de pesquisa em aberto e a nova abordagem implícita de problema inverso que é proposta pode vir a se constituir, dados os resultados obtidos, um avanço nessas investigações, uma vez que, pela primeira vez, são apresentados resultados com melhor qualidade que os obtidos pelo método *bin average*.

Para a resolução implícita do problema inverso associado à produção de mapas da RCFM, foram propostos e avaliados neste trabalho um algoritmo genético e um algoritmo de recozimento simulado. O custo computacional demandado pela resolução implícita do problema inverso por meio destes algoritmos, aliado ao fato de não ser empregada nenhuma técnica de redução do volume de dados, requereu o uso de processamento paralelo. Os algoritmos genético e de recozimento simulado foram paralelizados por meio da inclusão de chamadas à biblioteca de comunicação *Message Passing Interface* (MPI) [15] para execução numa máquina paralela de memória distribuída, um *cluster* de PCs interconectados por uma rede padrão *Fast Ethernet*. Dada sua natureza, algoritmos estocásticos normalmente permitem uma paralelização eficiente.

O algoritmo genético forneceu uma solução similar à do método *bin average* sequencial, mas com um tempo de execução paralelo muito maior. Isso levou à investigação do algoritmo de recozimento simulado, o qual foi explorado quanto ao uso de esquemas de resfriamento e de geração de solução candidata inicial. Adicionalmente, foi proposta a inserção de uma perturbação na solução candidata corrente sempre que não houver uma melhora de seu desempenho computacional, esquema conhecido por *re-annealing*. Isso levou a uma versão otimizada do algoritmo de recozimento simulado que obteve uma solução de melhor qualidade (correlação e desvio médio) e com tempo de execução próximo ao obtido pelo método *bin average*.

O método proposto é inédito e os resultados obtidos são os melhores até hoje alcançados com dados simulados. Os valores obtidos são próximos da sensibilidade dos detectores do telescópio, o que possibilita o uso de telescópios com maior resolução angular.

1.1 Esboço Geral dos Capítulos

Este trabalho está dividido, além desse capítulo introdutório, nos capítulos sucintamente descritos abaixo:

- **CAPÍTULO 2 – MAPAS DA RADIAÇÃO CÓSMICA DE FUNDO EM MICROONDAS:** Este capítulo apresenta a formulação tradicional do problema de construção de mapas de anisotropias da RCFM. São discutidos os diversos métodos existentes e, em particular, o método *bin average*, bem como aspectos de desempenho computacional desses métodos.
- **CAPÍTULO 3 - FORMULAÇÃO INVERSA IMPLÍCITA PROPOSTA PARA A CONSTRUÇÃO DE MAPAS DE ANISOTROPIAS DA RADIAÇÃO CÓSMICA DE FUNDO EM MICROONDAS:** A formulação inversa proposta para este problema é apresentada neste capítulo. São apresentados os algoritmos estocásticos utilizados, a saber, algoritmo genético e algoritmo de recozimento simulado, bem como suas respectivas implementações paralelas.
- **CAPÍTULO 4 – RESULTADOS NUMÉRICOS E DE DESEMPENHO:** Os resultados obtidos para dados simulados utilizando o algoritmo genético e o algoritmo de recozimento simulado são discutidos e comparados aos do método *bin average* quanto à qualidade da solução e ao desempenho computacional.
- **CAPÍTULO 5 – CONCLUSÕES:** O método inverso proposto é discutido, além de possíveis desenvolvimentos futuros.

CAPÍTULO 2

MAPAS DA RADIAÇÃO CÓSMICA DE FUNDO EM MICROONDAS

Este capítulo discute os métodos existentes de construção de mapas de anisotropias da RCFM, detalhando um dos métodos largamente utilizado pela comunidade científica para esse fim, o método *bin average*. Faz-se também uma breve descrição do experimento BEAST, utilizado neste trabalho como referência para a geração dos dados de medidas simulados contaminados com ruído, os quais serão utilizados para a avaliação do novo método aqui proposto.

2.1 Construção de Mapas de Anisotropias da RCFM

O MCP é sustentado, entre outras evidências, pelas observações obtidas através de experimentos no solo, embarcados em satélites ou balões estratosféricos com detectores para a medida de anisotropias da RCFM. Nestes experimentos, durante a aquisição de dados, cada região do céu é observada repetidas vezes, formando uma série temporal de medidas de temperatura, contaminadas com ruído, de forma a obter uma relação sinal/ruído suficiente para análise. A forma como é realizada a varredura do céu tem implicações diretas na análise dos dados, sendo denominada de estratégia de observação, a qual define como a região do céu escolhida será observada. O mapeamento entre as medidas obtidas na varredura e a região observada do céu é obtido por meio da matriz de apontamento. Designa-se como mapa de anisotropias da RCFM a representação gráfica da distribuição de temperaturas do céu estimada a partir dos dados coletados (medidas de temperatura) para a matriz de apontamento correspondente.

A produção de mapas de distribuição de temperatura da RCFM permite uma análise das anisotropias observadas nas medidas, isto é, das diferenças de temperatura de um pixel para outro, sendo cada pixel correspondente à menor região do céu, de acordo com a resolução do instrumento de medida. Os instrumentos atuais geralmente possuem diversos detectores que operam em diferentes frequências e geram grandes séries

temporais de dados. É a partir desses dados que se produz uma estimativa do mapa da RCFM numa determinada frequência. Após a produção de mapas para as várias frequências determina-se o espectro de potência permitindo a estimativa de alguns parâmetros cosmológicos importantes, como por exemplo, a densidade do Universo, a constante cosmológica e a constante de *Hubble*, importantes na determinação do tipo de Universo em que vivemos.

O satélite COBE foi pioneiro em apresentar um mapa completo das anisotropias intrínsecas de RCFM e portanto constitui uma referência na literatura de construção desses mapas [16]. Vários outros experimentos em diversas frequências procuraram aumentar a resolução angular e assim obter mais detalhes em relação aos mapas gerados a partir de dados do COBE, em detrimento da extensão da região do céu observada. A Tabela 2.1 mostra alguns experimentos e suas respectivas frequências utilizadas, bem como informa se o experimento é de solo, transportado em balão estratosférico ou em satélite. A quarta coluna apresenta o ano em que o experimento foi finalizado.

TABELA 2.1 – Experimentos de RCFM no solo e a bordo de satélites e balões estratosféricos.

Experimento	Embarcado	f (GHz)	Ano
COBE	Satélite	31,5; 53 e 90	1992
HACME [17]	Balão	39-43	2000
BEAST [18]	Balão/solo	30 e 41,5	2000
ACBAR [19]	Solo	150; 219; 274	2001
CBI [20]	Solo	26; 36	2002
WMAP [21]	Satélite	23;30;41;63;94	2003

O grupo de Cosmologia Experimental da Divisão de Astrofísica (DAS) do Instituto Nacional de Pesquisas Espaciais (INPE) participa de uma cooperação com a Universidade Federal de Itajubá (UNIFEI), o Departamento de Física da Universidade da Califórnia em Santa Bárbara (UCSB), o *Jet Propulsion Laboratory* (JPL) e as

Universidades de Roma e Milão, que levou ao desenvolvimento do experimento BEAST para observações de anisotropias da RCFM, tanto a bordo de balão estratosférico como para observações no solo. Este experimento é uma evolução do experimento HACME (HEMT *Advanced Cosmic Microwave Explorer*) [17] cujo objetivo consistiu em medir flutuações de temperatura da RCFM nas regiões de *Ursae Minoris* e α *Leonis*.

A Figura 2.1 mostra o mapa de anisotropias da RCFM obtido para todo o céu pelo satélite COBE, bem como os mapas obtidos com uma maior resolução angular realizados com dados obtidos pelo experimento HACME para as duas regiões mencionadas de apontamento.

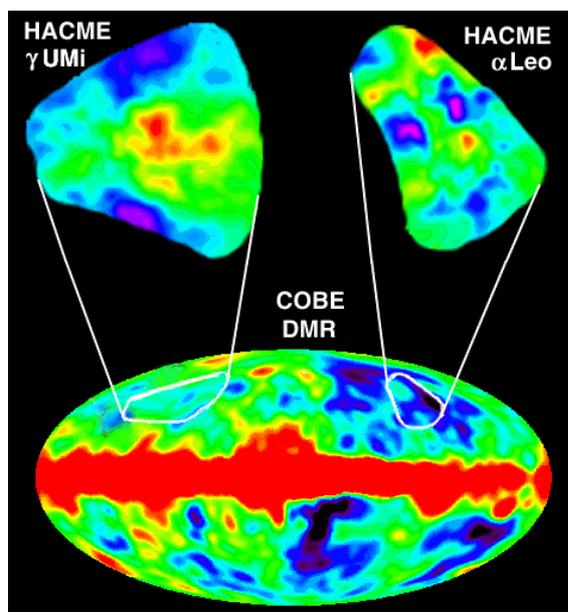


FIGURA 2.1 – Mapas de anisotropias da RCFM gerados pelos experimentos COBE e HACME.

FONTE: Adaptada de Tegmark et al. [17].

A Figura 2.2 mostra um mapa recente do BEAST feito com aproximadamente 30 dias de observações.

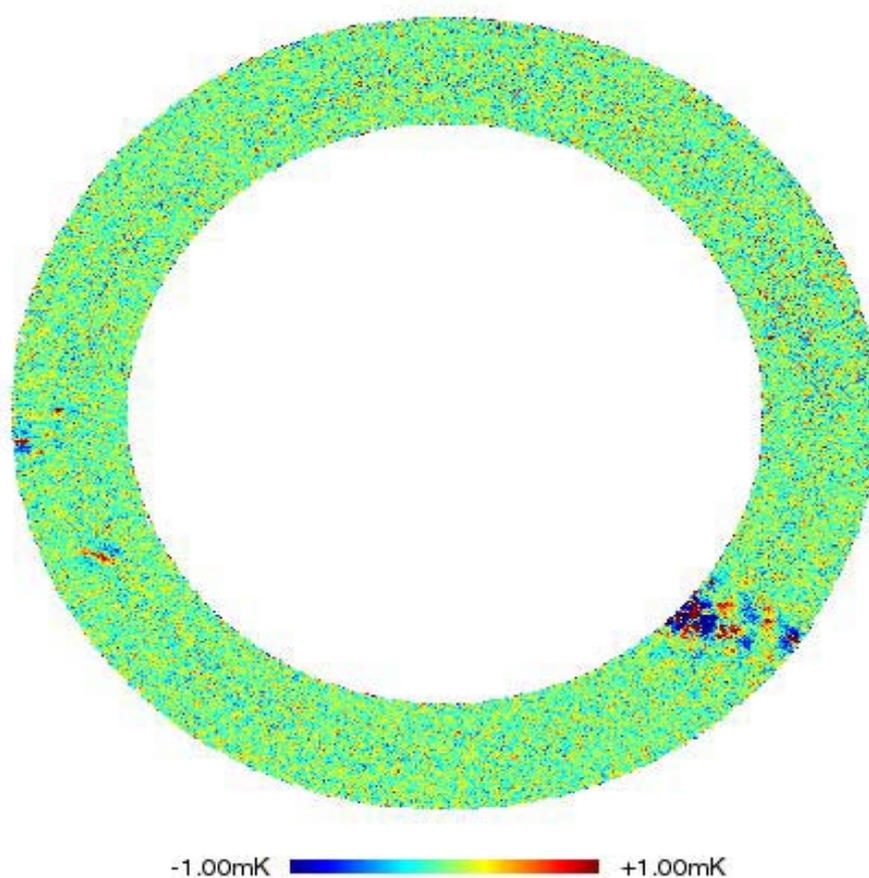


FIGURA 2.2 – Mapa de anisotropias da RCFM do experimento BEAST para um período aproximado de 30 dias de aquisição.

FONTE: Adaptada de Meinhold et al. [22].

2.2 Breve Descrição do Experimento BEAST

O presente trabalho utiliza dados simulados do experimento BEAST. O BEAST possui um sistema óptico do tipo Gregoriano não-axial. Nesta montagem, o espelho principal é um parabolóide cuja superfície refletora não intersecta o eixo óptico da própria parábola que o gerou. O espelho secundário é um elipsóide, com um dos focos superposto ao foco do espelho principal parabolóide. Os detectores de microondas utilizados para medir as flutuações da RCFM são posicionados no plano focal correspondente ao outro foco do espelho secundário elipsóide. O experimento BEAST permite a produção de mapas nas frequências de 30 e 41,5 GHz, com resolução angular de 23' e 30' e sensibilidade de 10 μ K a 20 μ K por pixel. A configuração óptica consiste em um espelho primário paraboloidal (semi-eixo maior igual a 2,2 m) e um sub-refletor elipsoidal (semi-eixo maior igual a 0,8 m). Além destes, um espelho plano giratório faz o feixe descrever uma trajetória elíptica no céu, que combinada com o movimento da Terra possibilita uma varredura capaz de cobrir regiões do céu distante de ordem de 10 graus muito rapidamente. A frequência de rotação de 2 Hz garante 2 observações por segundo por pixel. Isso permite uma forte correlação entre as medidas de forma a obter dados para a construção de mapas bidimensionais das anisotropias da RCFM. O telescópio BEAST está instalado em uma base de pesquisas da Universidade da Califórnia (*White Mountain Research Station*), tendo coletado dados de julho a dezembro de 2001 e em alguns períodos de 2002, além de diversos outros períodos [22].

Neste experimento, foram utilizadas duas estratégias de observação: a primeira realizando uma varredura de 10° a partir do zênite por meio da rotação do espelho plano e a segunda estratégia, mantendo o telescópio com apontamento fixo no zênite em que a varredura é realizada pela rotação da Terra. A Figura 2.3 ilustra uma varredura típica obtida por qualquer das estratégias.

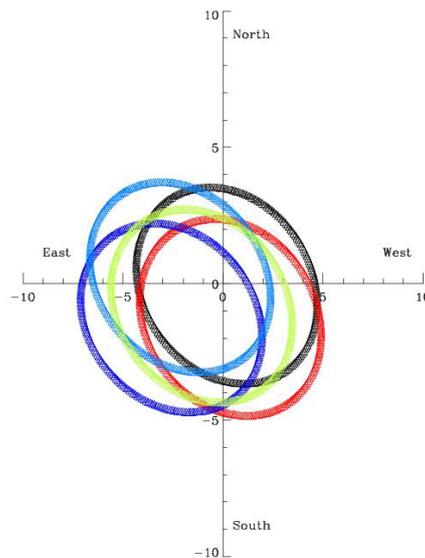


FIGURA 2.3 – Exemplo de varredura do experimento BEAST.

FONTE: Adaptada de Meinhold et al.[9].

2.3 Formulação do Problema de Construção de Mapas de Anisotropias da RCFM

A determinação de pontos na esfera celeste é feita através da adoção de um sistema de coordenadas astronômicas, dadas pelos ângulos de elevação θ e azimute ϕ do telescópio. Com base nesse sistema, associa-se à região do céu que se pretende mapear um conjunto de pixels, correspondentes aos valores discretizados dessas coordenadas, conforme a resolução do telescópio. Para a região do céu observada, uma varredura completa do telescópio é composta de sucessivas varreduras parciais de sub-região da região considerada, cada uma correspondente a um subconjunto de N pixels.

Uma varredura parcial qualquer é constituída de sucessivas medidas de temperatura de pixels a intervalos discretos de tempo para a sub-região correspondente. Entretanto, numa mesma varredura parcial, o telescópio adquire medidas das temperaturas dos pixels da sub-região correspondente de forma repetitiva. Os pixels das bordas entre sub-regiões são mais amostrados, uma vez que são medidos em varreduras parciais diferentes. Essas medidas excedentes podem, por exemplo, ser descartadas. A medida de temperatura de um determinado pixel é independente das medidas dos demais pixels da região ou sub-região considerada do céu.

A varredura parcial de uma sub-região do céu considerada fornece então diversas seqüências de medidas de temperaturas dos N pixels dessa sub-região. Várias seqüências sucessivas constituem séries temporais de medidas dessa sub-região. Define-se a série temporal S de medidas dessa sub-região, com M elementos, como sendo o conjunto de todas essas séries temporais.

Define-se por T , o vetor de dimensão N que contém as temperaturas reais dos pixels. Assim, T_n é a temperatura real de um dado pixel n . A série temporal de medidas é composta pelas M/N medidas de temperatura d_n de cada pixel n . O número total de medidas varia conforme o pixel, mas são consideradas as primeiras M/N medidas para cada pixel. Estas medidas estão contaminadas com ruído Δ_n devido não somente ao instrumento (telescópio), que é constituído pelo receptor de microondas e a eletrônica associada, mas por todas as contribuições de ruídos, como por exemplo, o ruído atmosférico, galáctico, extra galáctico. Para qualquer uma das medidas de um dado pixel n , pode-se então escrever que:

$$d_n = T_n + \Delta_n \quad . \quad (2.1)$$

As medidas d_n de cada pixel são armazenadas de forma intercalada na série temporal S , conforme abaixo:

$$S = [d_1 \quad d_2 \quad \cdots \quad d_N \quad \cdots \quad d_1 \quad d_2 \quad \cdots \quad d_N]_{M \times 1}^T \quad . \quad (2.2)$$

Note-se que, na série S acima, existem M/N seqüências de medidas de temperatura dos N pixels.

O problema associado à construção de mapas de anisotropias de RCFM é constituído pela determinação de um vetor T_e com as estimativas de temperaturas dos N pixels que mais se aproximem das temperaturas reais. Esse problema é formulado como um problema de otimização a partir da Equação 2.1, a qual pode ser reescrita em função da temperatura estimada de um pixel n como:

$$d_n - T_{en} = \Delta_n . \quad (2.3)$$

No limite, considerando-se o lado direito da Equação 2.3 tendendo a zero, o primeiro membro também tende a zero e a medida d_n tende ao valor exato T_n . Assim, pode-se formular o problema de otimização como a minimização da seguinte norma para todos os N pixels:

$$\min \|d_n - T_{en}\|_2; \quad n = 1, 2, \dots, N. \quad (2.4)$$

O método COBE basicamente consiste na aplicação do método dos mínimos quadrados [23] ao problema de otimização expresso pela Equação 2.3, a qual deve ser reescrita em função da estratégia de varredura do telescópio, como exposto a seguir. Este método, desenvolvido pela equipe do satélite do mesmo nome, é também utilizado para análise dos dados de outros experimentos e pela linha de pesquisa em Cosmologia da DAS.

Define-se A , uma matriz de apontamento, de dimensão $M \times N$, que correlaciona os elementos do vetor T de temperaturas reais com os elementos do vetor S de temperaturas medidas. Os elementos de uma dada linha m da matriz A tem valor 1 se houver correlação entre os elementos correspondentes dos vetores T , e valor 0 caso contrário. No caso particular da estratégia de varredura do BEAST, cada linha de A tem um único elemento igual a 1, ou seja, não há repetição de valores numa dada linha.

A relação entre S e T é dada pela matriz A e pelo ruído das medidas, suposto independente do pixel:

$$S_{M \times 1} = A_{M \times N} T_{N \times 1} + \Delta_{M \times 1} . \quad (2.5)$$

Esta equação pode ser novamente reescrita em notação matricial como:

$$\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \\ \vdots \\ d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}_{M \times 1} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{MN} \end{bmatrix}_{M \times N} \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix}_{N \times 1} + \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_N \\ \vdots \\ \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_N \end{bmatrix}_{M \times 1} \quad (2.6)$$

Partindo-se da Equação 2.5 e supondo-se ruído gaussiano, considera-se o valor de temperatura de um determinado pixel n medido na seqüência i de medidas, o qual está armazenado no elemento $[n + (i-1)N]$ de S . A curva gaussiana correspondente à distribuição das medidas de temperatura desse pixel n terá como máximo a temperatura real desse pixel. A probabilidade P_n de se obter uma dada medida de temperatura d_{in} desse pixel n é dada pela função de densidade de probabilidade gaussiana, sendo Z uma constante de proporcionalidade:

$$P_n = \frac{Z}{\sigma_n} e^{-\frac{1}{2}g_n}, \quad g_n = \sum_{i=1}^{M/N} \left(\frac{d_{in} - \mu_n}{\sigma_n} \right)^2 \quad (2.7)$$

em que μ_n é o valor médio das medidas i do pixel n e σ_n é o correspondente desvio padrão. O valor médio, que corresponde à temperatura estimada, é dado por:

$$\mu_n = \frac{1}{(M/N)} \sum_{i=1}^{M/N} d_{in} = T_{en}, \quad (2.8)$$

A probabilidade conjunta P de ocorrer um dado conjunto de medidas para todos os N pixels é dada pelo produto das probabilidades para cada pixel (Equação 2.7):

$$P = \frac{Z^n}{\sigma_1 \sigma_2 \cdots \sigma_n} e^{-\frac{1}{2} \sum_{n=1}^N g_n} \quad (2.9)$$

Definindo-se a função $\chi_n^2 = g_n = \sum_{i=1}^{M/N} \left(\frac{d_{in} - \mu_n}{\sigma_n} \right)^2$ e substituindo-a na Equação 2.9, esta probabilidade P passa a ser expressa por:

$$P = \frac{Z^n}{\sigma_1 \sigma_2 \cdots \sigma_n} e^{-\frac{1}{2} \sum_{n=1}^N \chi_n^2} \quad (2.10)$$

Uma vez que P é uma função decrescente de cada χ_n^2 , valores máximos de P ocorrem quando os χ_n^2 tendem a zero. Dessa forma, pode-se minimizar a norma expressa pela Equação 2.3 por meio da minimização dos próprios χ_n^2 para todas as medidas de temperaturas efetuadas em todos os pixels. Esta norma pode ser reescrita para todos os N pixels utilizando-se o vetor S e correlacionando seus elementos com o vetor de temperaturas reais T_n por meio da matriz de apontamento A . Aqui, consideram-se as temperaturas estimadas como sendo as reais, para efeito da dedução a seguir. Para o pixel n , a função χ_n^2 pode ser reescrita como:

$$\chi_n^2 = \sum_{i=1}^{M/N} \frac{\left(S - \sum_{n=1}^N A_{in} T_n \right)^2}{\sigma_n^2} \quad (2.11)$$

Na expressão acima, o vetor σ de dimensão N é formado pelos desvios padrão das medidas de cada pixel. O desvio padrão σ_n de um pixel n é definido a partir do valor médio μ_n das M/N medidas desse pixel:

$$\sigma_n^2 = \sum_{i=1}^{M/N} \frac{(d_{in} - \mu_n)^2}{(M/N)}, \quad (2.12)$$

em que d_{in} é a i -ésima medida de temperatura do pixel n .

O método considera como melhor estimativa para o vetor de temperaturas T aquela que minimiza χ^2 , satisfazendo a condição clássica de minimização de uma função:

$$\frac{\partial \chi_n^2}{\partial T_n} = 0; \quad n = 1, 2, \dots, N. \quad (2.13)$$

Substituindo-se χ_n^2 nesta equação, conforme a Equação 2.11, obtém-se:

$$\frac{\partial \chi_n^2}{\partial T_n} = -2 \sum_{i=1}^{M/N} \left\{ \frac{\left[S - \sum_{n=1}^N A_{in} T_n \right] \sum_{n=1}^N A_{in}}{\sigma_n^2} \right\} = 0. \quad (2.14)$$

Esta última equação pode ser reescrita como:

$$\sum_{i=1}^{M/N} \left\{ \frac{\left[\sum_{n=1}^N A_{in} T_n \right] \sum_{n=1}^N A_{in}}{\sigma_n^2} \right\} = \sum_{i=1}^{M/N} \left[\frac{S \sum_{n=1}^N A_{in}}{\sigma_n^2} \right]. \quad (2.15)$$

Esta equação é novamente reescrita em notação matricial como:

$$A_{NxM}^T C_{MxM}^{-1} A_{MxN} T_{Nx1} = A_{NxM}^T C_{MxM}^{-1} S_{Mx1}, \quad (2.16)$$

em que C^{-1} é a matriz (quadrada) de covariância das medidas de temperatura, normalmente denominada matriz das covariâncias de ruído, dada por:

$$C^{-1} = \begin{bmatrix} \sigma_1^{-2} & 0 & \dots & & & & & \dots & 0 \\ 0 & \sigma_2^{-2} & 0 & \dots & & & & & \vdots \\ \vdots & 0 & \ddots & 0 & \dots & & & & \\ & \vdots & 0 & \sigma_N^{-2} & 0 & \dots & & & \\ & & \vdots & 0 & \sigma_1^{-2} & 0 & \dots & & \\ & & & \vdots & 0 & \sigma_2^{-2} & 0 & \dots & \\ & & & & \vdots & 0 & \ddots & 0 & \dots \\ \vdots & & & & & \vdots & 0 & \sigma_N^{-2} & 0 \\ 0 & \dots & & & & & \vdots & 0 & \ddots \\ & & & & & & & \dots & 0 & \sigma_N^{-2} \end{bmatrix}_{M \times M} \quad (2.17)$$

Note-se que todos elementos fora da diagonal são nulos, uma vez que não há correlação entre pixels diferentes e, portanto, a covariância nesses casos é nula.

Definindo-se as matrizes K e B que aparecem comumente na literatura, conforme abaixo:

$$K_{N \times N} = A_{N \times M}^T C_{M \times M}^{-1} A_{M \times N} \mathbf{e}, \quad (2.18)$$

$$B_{N \times 1} = A_{N \times M}^T C_{M \times M}^{-1} S_{M \times 1}, \quad (2.19)$$

pode-se reescrever a Equação 2.16, substituindo-se as Equações 2.18 e 2.19, obtendo-se

$$K_{N \times N} T_{N \times 1} = B_{N \times 1}. \quad (2.20)$$

Essa equação (ou a própria Equação 2.16) descreve o problema inverso associado à construção de mapas: estimar o vetor de temperaturas T dadas as matrizes K e B , as quais contém informação sobre a matriz de apontamento e as medidas de temperatura. A Equação 2.20 possibilitaria o cálculo das temperaturas exatas se todas as hipóteses consideradas fossem verificadas, o que obviamente não ocorre na prática. Entretanto, essa é formulação adotada correntemente para o problema inverso associado.

Uma limitação relevante são os recursos computacionais necessários devido ao enorme volume de dados (e a conseqüente dimensão dos vetores e matrizes decorrentes). Tegmark [24], com o objetivo de avaliar o consumo de memória e o custo computacional necessário para a solução por métodos analíticos (mostrados adiante na Tabela 2.2), reescreve a Equação 2.20 de forma a expressar a resolução como uma multiplicação de matrizes:

$$T_{Nx1} = W_{NxM} S_{Mx1}. \quad (2.21)$$

em que a matriz W é definida por conveniência de forma que $WA = I$ conforme abaixo:

$$W_{NxM} = \left(A_{NxM}^T C_{MxM}^{-1} A_{MxN} \right)^{-1} A_{NxM}^T C_{MxM}^{-1}. \quad (2.22)$$

Substituindo-se a Equação 2.5 na Equação 2.21, a qual fornece uma estimativa das temperaturas a partir de suas medidas, encontra-se que essa estimativa é dada por $[WA + W\Delta]$. Pode-se então definir um vetor ε de erros de temperatura, que mede a diferença entre as temperaturas estimadas T_e e as reais T , como sendo:

$$\varepsilon \equiv T_e - T = [WA - I]T + W\Delta \quad (2.23)$$

Como foi suposto que $WA = I$, obtém-se então uma expressão na qual o erro ($\varepsilon = W\Delta$) independe do vetor de temperaturas reais T .

Tegmark [10] especifica 10 métodos inversos para solução do problema de construção dos mapas de anisotropias da RCFM. Destes, 8 são métodos analíticos lineares derivados do método dos mínimos quadrados e os demais são métodos não lineares. Os oito métodos lineares diferem entre si quanto à forma de cálculo da matriz W , conforme especificado na Tabela 2.2. Observa-se a necessidade de inversão de matrizes de grandes dimensões. O próprio método COBE apresenta variações conforme a definição adotada para a matriz W .

TABELA 2.2 – Definição da Matriz W para diversos métodos de construção de Mapas de Anisotropias da RCFM.

	Método	Definição de W
1	COBE generalizado	$W = [A^T V A]^{-1} A^T V$
2	COBE média geral (<i>bin average</i>)	$W = [A^T A]^{-1} A^T = A^T$
3	COBE	$W = [A^T C^{-1} A]^{-1} A^T C^{-1}$
4	Wiener 1	$W = F A^T [A F A^T + C]^{-1}$
5	Wiener 2	$W = [F^{-1} + A^T C^{-1} A] A^T C^{-1}$
6	Saskatoon	$W = [H F^{-1} + A^T C^{-1} A]^{-1} A^T C^{-1}$
7	TE96	$W = \Lambda F A^T [A F A^T + C]^{-1}, WA = I$
8	TE97	$W = \Lambda [\eta F^{-1} + A^T C^{-1} A]^{-1} A^T C^{-1}, WA = I$

FONTE: Adaptada de Tegmark [24].

Nesta tabela, a matriz V é uma matriz arbitrária de dimensão $M \times M$ definida convenientemente. O método 3 (COBE propriamente dito) define V como sendo o inverso de C , a matriz de covariâncias de ruído definida na Equação 2.17. O método 2, obtido a partir do método COBE generalizado ao se adotar a matriz V como identidade, é conhecido como método da média geral (*bin average*) e será detalhado na Seção 2.3.1. Os métodos 4 e 5 são semelhantes e utilizam filtros de Wiener em que F representa o filtro aplicado. O método 6 permite a escolha de uma relação sinal/ruído através do parâmetro η aplicado ao método 5. Caso $\eta \rightarrow 0$, o método 6 reduz-se ao método COBE. O método 7 permite um reescalonamento dos pixels através da matriz diagonal Λ de forma a garantir $WA = I$. O método 8 representa uma combinação dos métodos 6 e 7.

Ainda são especificados dois métodos não lineares, um de máxima probabilidade e um de máxima entropia. Esses métodos são pouco documentados e não foram encontradas referências sobre suas possíveis implementações.

Objetivando avaliar o consumo de memória e o custo computacional necessário para a solução por esses métodos analíticos, Tegmark reescreve a Equação 2.20 utilizando duas funções (γ e Z). Para o caso particular do método 3, COBE:

$$\Upsilon = A^T C^{-1} A, \quad e \quad (2.24)$$

$$z = A^T C^{-1} S. \quad (2.25)$$

Assim, a Equação 2.20 é expressa como:

$$T = \Upsilon^{-1} z. \quad (2.26)$$

O processo para obtenção dos mapas é atualmente resolvido numericamente em 3 etapas, dadas pela solução das Equações 2.23, 2.24 e 2.25. O cálculo do vetor solução T envolve assim a inversão da matriz Υ não-quadrada, esparsa e de grandes dimensões.

A Tabela 2.3, descrita por Borril [25], apresenta uma estimativa dos recursos computacionais requeridos para solução de cada passo do algoritmo acima exposto. Na última coluna, mostra-se a complexidade algorítmica, expressando-se o fator de proporcionalidade para o número de operações em ponto flutuante. Nas demais colunas, são mostrados os fatores de proporcionalidade para o volume de armazenamento e de memória requeridos.

TABELA 2.3 – Requisitos computacionais do método COBE.

Etapa	Disco rígido	Memória principal	Operações em ponto flutuante
$\Upsilon^{-1} = A^T C^{-1} A$	$4M^2$	16M	$2NM^2$
$z = A^T C^{-1} S$	$4M^2$	16M	$2M^2$
$m = (\Upsilon^{-1})^{-1} z$	$4N^2$	$8N^2$	$8N^3/3$

FONTE: Adaptada de Borril [25].

A análise da enorme quantidade de informação nas séries temporais de dados de RCFM referente aos experimentos HACME, MÁXIMA [26], BOOMERANG ou BEAST tornou-se um grande desafio devido aos requisitos computacionais. Por exemplo, para

experimentos tais como BOOMERANG, em que $M = 2.106$ varreduras e $N = 3.104$ pixels [27], o método COBE requer 16 TB para armazenamento de dados com 4 B de precisão, 7 GB de memória principal para cálculos com 8 B de precisão e envolve $2,4 \times 10^{17}$ operações em ponto flutuante.

Assim, devido à complexidade algorítmica envolvida na construção de mapas para os próximos experimentos planejados pela comunidade científica internacional, nos quais há uma maior resolução angular do pixel e portanto mais dados, torna-se importante a otimização das operações com matrizes, típicas nos métodos analíticos e a busca por novos métodos de construção que sejam mais eficientes computacionalmente. Uma alternativa muito utilizada é o método *bin average*, descrito a seguir, que apresenta complexidade algorítmica bem menor, mas às custas de uma certa perda de informação. Nesse escopo, a formulação implícita para o problema inverso e o uso de computação paralela se apresenta como alternativas promissoras. No próximo capítulo apresenta-se uma nova abordagem para o problema de construção de mapas das anisotropias da RCFM. O problema inverso associado é resolvido por uma metodologia implícita.

2.3.1 Método *bin average*

Conforme visto na Tabela 2.2, neste método adota-se a matriz C^{-1} das covariâncias de ruído como sendo a matriz identidade. Isso se deve à adoção da hipótese de ruído. Conseqüentemente, a covariância das medidas de cada pixel é também unitária (inverso do quadrado do desvio padrão). Como exposto na mesma tabela, a matriz W resultante será a transposta da matriz de apontamento e terá então somente elementos com valor 0 ou 1. A Equação 2.21 pode então ser reescrita como:

$$T_{Nx1} = A_{NxM}^T S_{Mx1}. \quad (2.27)$$

Assim, o valor da temperatura real de cada pixel será simplesmente a média aritmética de todas as M/N medidas de temperatura do pixel correspondente. Na prática, as medidas de temperatura de todos os pixels estão armazenadas em arquivos separados

correspondentes a uma série temporal de l hora de aquisição, tipicamente. Esse cálculo da média pode ser então efetuado separadamente para cada uma dessas séries temporais e a média final para cada pixel determinada pela média dessas médias parciais ponderadas pelo desvio padrão de cada uma delas. Então, a temperatura média \bar{T}_i de cada pixel i é calculada a partir das medidas de temperatura desse pixel das L séries temporais conforme abaixo:

$$\bar{T}_i = \frac{\sum_{l=1}^L \left(\frac{\bar{T}_{il}}{\sigma_{il}^2} \right)}{\sum_{l=1}^L \left(\frac{1}{\sigma_{il}^2} \right)}, \quad i = 1, 2, \dots, N \quad (2.28)$$

em que \bar{T}_{il} representa a média de todas as medidas de temperatura do i -ésimo pixel do arquivo l e σ_{il} representa o desvio padrão desse pixel em relação a essas medidas.

Pode-se ver claramente que a complexidade algorítmica é bem menor que a dos demais métodos, uma vez que o algoritmo se resume a cálculos de médias e desvios padrão.

CAPÍTULO 3

FORMULAÇÃO INVERSA IMPLÍCITA PROPOSTA PARA A CONSTRUÇÃO DE MAPAS DAS ANISOTROPIAS DA RADIAÇÃO CÓSMICA DE FUNDO EM MICROONDAS

O capítulo anterior abordou os métodos inversos correntemente utilizados para construção de mapas de anisotropias da RCFM. Esses métodos são analíticos e derivam do método COBE, o qual é baseado no método dos mínimos quadrados. Todos estes métodos empregam alguma redução no volume dos dados relativa ao vetor S , definido na Equação 2.2, devido às necessidades em termos de recursos computacionais. Esses métodos são também limitados devido a problemas na inversão de matrizes não-quadradas, esparsas e de grandes dimensões.

Este trabalho propõe uma nova abordagem para contornar esses problemas ao resolver o problema inverso associado à construção de mapas de anisotropias da RCFM por meio de uma metodologia implícita (iterativa). Num problema inverso qualquer, dado um determinado fenômeno ou processo, busca-se determinar as causas a partir do conhecimento dos efeitos [28]. No presente caso, pode-se considerar como causa o conjunto de temperaturas reais do céu em pontos determinados, enquanto que os efeitos correspondem às temperaturas medidas pelo telescópio corrompidas com ruído. Neste trabalho, o problema inverso associado é aqui resolvido por uma metodologia implícita, na qual é formulado como um problema de otimização e resolvido iterativamente. A cada iteração, esse algoritmo gera uma nova solução candidata que é avaliada por meio de uma função objetivo. Neste trabalho, propõem-se e testam-se dois algoritmos estocásticos: um algoritmo genético e um algoritmo de recozimento simulado. Esta abordagem inversa tem a vantagem adicional de permitir a utilização de dados de medidas de temperatura sem redução no volume dos dados.

3.1 Problemas Inversos e a Formulação Inversa Implícita Adotada

Num problema inverso, dado um determinado fenômeno ou processo, busca-se determinar as causas a partir do conhecimento dos efeitos. Nesse contexto, geralmente existe uma formulação analítica que mapeia causas conhecidas para os efeitos correspondentes. Essa formulação é conhecida como o modelo direto associado a esse problema inverso. O problema direto é normalmente bem definido e contínuo, implicando na existência de um único efeito ou conjunto de efeitos para uma dada causa ou conjunto de causas, sendo que pequenas variações dos efeitos resultam em pequenas variações nas causas, ou seja, o problema direto é bem posto. A Figura 3.1 ilustra a relação entre causas e efeitos expressa pelos problemas direto e inverso associados a um dado fenômeno ou processo.

Normalmente, no caso de um fenômeno investigado, as causas correspondem às condições iniciais e de contorno e às propriedades intrínsecas do sistema [29], sendo os efeitos expressos por dados observados, por exemplo, correspondentes a um determinado parâmetro (temperatura, luz, calor, etc.).

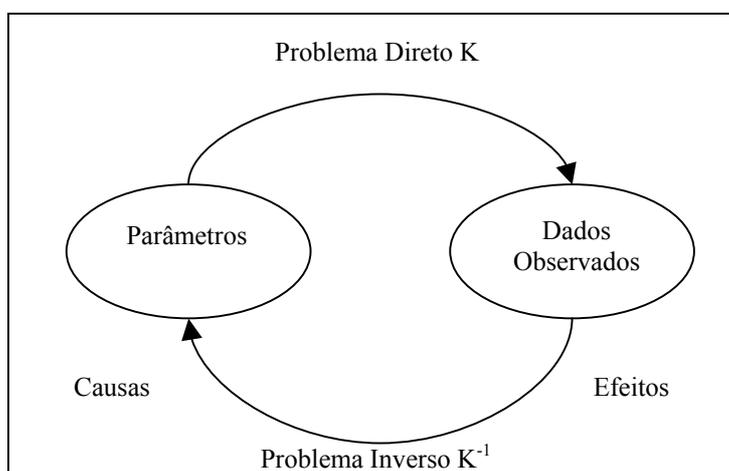


FIGURA. 3.1 – Visão esquemática de problemas inversos.

Os problemas inversos são tipicamente de difícil solução, uma vez que são mal-postos. Hadamard [30] define as propriedades necessárias e suficientes para que um problema seja bem posto:

- 1) Existência - a solução existe;
- 2) Unicidade - a solução é única;
- 3) Estabilidade - a solução é estável, isto é, tem dependência contínua e suave com os dados de entrada.

Os problemas inversos podem ser agrupados em categorias, embora não exista uma classificação padrão. Uma divisão possível [31] classifica três tipos de problemas inversos: identificação de parâmetros, determinação do estado inicial do sistema e determinação de condições de contorno. Outra classificação [32] caracteriza o problema inverso em função do número de parâmetros desconhecidos do modelo direto (N_{par}) e do número de dados experimentais/observacionais (N_{dados}). A relação entre essas quantidades pode ser:

- $N_{dados} = N_{par}$ - o problema inverso pode ser resolvido analiticamente (se existir tal formulação) ou implicitamente.
- $N_{dados} > N_{par}$ - o problema inverso deve ser resolvido implicitamente.
- $N_{dados} < N_{par}$ - o problema inverso não pode ser resolvido.

Problemas inversos podem ser resolvidos explicitamente, quando existe uma formulação analítica (como no caso dos métodos mostrados no capítulo anterior), ou implicitamente, quando são geradas e avaliadas sucessivas soluções candidatas, num processo iterativo. Na metodologia implícita, o problema inverso é formulado como um problema de otimização, em que se procura minimizar (ou maximizar) uma determinada função de custo, ou função objetivo. Para cada solução candidata, correspondente a um ponto no espaço de busca (domínio do problema), o valor da função objetivo é calculado aplicando-se uma determinada norma para quantificar a diferença entre os efeitos (dados m_{mod}) causados pela solução candidata e os efeitos observados experimentalmente (dados m_{exp}). Note-se que, como no caso do presente trabalho, os dados m_{exp} podem ser amostrados repetidas vezes. A função objetivo é muitas vezes dada pela norma 2, conforme abaixo:

$$\text{função_objetivo} = \|m^{\text{exp}} - m^{\text{mod}}\|_2 = \sum_{i=1}^{N_{\text{dados}}} (m_i^{\text{exp}} - m_i^{\text{mod}})^2. \quad (3.1)$$

No problema inverso aqui abordado e expresso pela Equação 2.20 ($K_{N \times N} T_{N \times 1} = B_{N \times 1}$), a causa é o conjunto de temperaturas reais T , enquanto que o efeito é o conjunto de temperaturas medidas S . Assim, note-se que não há necessidade de se aplicar um modelo direto. Como as medidas de temperatura são independentes, esse problema inverso é resolvido independentemente para cada pixel, sendo a função objetivo acima expressa para cada pixel n para o conjunto de M/N medidas correspondente àquele pixel, como a norma 2 da diferença entre a temperatura estimada T_n dessas medidas.

3.2 Métodos de Otimização

Para um determinado problema de otimização, a escolha do método depende de características específicas, na tentativa de se avaliar o comportamento da função associada. Entretanto, na prática, muitas vezes essa escolha acaba sendo empírica.

Existem métodos de otimização determinísticos e estocásticos [33]. Os primeiros caracterizam-se por sua eficiência na localização de mínimos partindo de um ponto próximo do espaço de busca, mas podem justamente convergir num mínimo local e não no mínimo global desejado. Os algoritmos estocásticos são bem mais robustos, no sentido de que conseguem explorar melhor o espaço de busca, sendo menos suscetíveis à convergência num mínimo local. Há também métodos híbridos que utilizam um algoritmo estocástico para uma busca global e executam uma busca local utilizando um algoritmo determinístico. Os métodos de otimização ainda podem ser classificados como os inspirados pela natureza, como algoritmos genéticos e os algoritmos de colônia de formigas, ou métodos diversos como busca tabu ou recozimento simulado.

Os métodos determinísticos são baseados na geração sucessiva de pontos no espaço de busca. A geração de um novo ponto é baseada numa direção de busca, determinada com

base no gradiente da função considerada. Assim, esta função deve ser avaliada para cada um destes pontos. Numa dada iteração, a procura pelo ponto ótimo usa o ponto corrente como ponto de partida para a determinação do próximo ponto. Dado o uso de gradientes, estes métodos unicamente produzem bons resultados para funções contínuas, convexas e unimodais.

Existem diversos métodos determinísticos, tais como: Descida Máxima [34], Gradiente Conjugado [35], Levenberg-Marquadt [36], Hookes e Jeeves [39], Simplex [37] e Redes Neurais[38].

Os métodos estocásticos têm como principal característica a busca pelo ponto ótimo usando alguma informação do ponto corrente, mas determinando o próximo ponto de maneira estocástica. Tais métodos demandam também a avaliação da função considerada para cada ponto gerado no espaço de busca. Entretanto, não dependem de gradientes ou derivadas, sendo portanto aplicáveis a funções descontínuas. Destacam-se, entre os métodos estocásticos mais conhecidos, as chamadas estratégias evolucionárias como os algoritmos genéticos [40], além de outras estratégias tais como recozimento simulado [41], Otimização Extrema [42] e [43], Busca Tabu [44] e Colônia de Formigas [45].

A seguir, são expostos dois métodos estocásticos de otimização, juntamente com suas características principais, utilizados no problema exposto neste trabalho: um algoritmo genético e um algoritmo de recozimento simulado.

Os algoritmos genéticos são baseados na capacidade natural de evolução dos indivíduos mais adaptados ao meio, modelando processos evolutivos. A cada iteração do algoritmo, correspondente a uma geração de indivíduos, certos operadores evolutivos são aplicados a

os indivíduos da população corrente, determinando assim os indivíduos que irão compor a próxima geração.

O algoritmo de recozimento simulado é inspirado na metalurgia, na qual um metal é aquecido e fundido, sendo gradualmente esfriado até atingir um estado mais estável, ou seja, um estado de menor energia termodinâmica.

Os métodos de otimização são iterativos e portanto demandam o estabelecimento de um critério de parada. Os algoritmos genéticos apresentados neste trabalho utilizam dois critérios de parada: (i) se for atingido o número máximo de iterações, valor obtido empiricamente, e (ii) se não houver uma redução de pelo menos 1% do valor da aptidão. No algoritmo de recozimento simulado, utilizam-se três critérios de parada simultaneamente: (i) se for atingido o número máximo de iterações, obtido empiricamente, (ii) se a pseudo-temperatura for reduzida a zero e (iii) se o valor da função objetivo não for reduzido em pelo menos 1% a cada 100 iterações.

3.2.1 Algoritmos Genéticos

Os algoritmos genéticos (AGs) compreendem uma família de modelos computacionais inspirados na teoria da evolução inicialmente proposta por Darwin [46]. As células de um ser vivo possuem vários cromossomos, que são cadeias de DNA. Cada uma destas cadeias é composta por seqüências chamadas genes que contém a informação genética desse ser. Os AGs utilizam estruturas de dados correspondentes a genes para caracterizar os indivíduos de uma dada população. Operadores de seleção, cruzamento e mutação são aplicados a essas estruturas de forma a permitir a geração de novos indivíduos num processo evolutivo no qual se busca gerar um indivíduo que corresponda à solução ótima. O termo foi cunhado e o tema investigado pioneiramente por John Holland em 1975 [47]. O uso de AGs se difundiu consideravelmente ao longo dos últimos anos [48].

Um AG começa com a geração de uma população inicial aleatória de indivíduos caracterizados pelos seus respectivos genes. Esses genes constituem uma codificação (binária, inteira ou real) de parâmetros associados ao problema. Os indivíduos são avaliados segundo uma métrica (dita aptidão) associada à solução do problema obtida

com aquele conjunto de parâmetros. São geradas sucessivas populações. Os indivíduos de cada nova população são gerados a partir dos indivíduos da geração anterior. Para tal, cada AG define um conjunto específico de operadores de cruzamento e mutação, os quais são aplicados aos indivíduos melhor avaliados segundo uma estratégia de seleção.

O algoritmo AG dito simples ou canônico, que serve de base para qualquer outro AG, pode ser descrito nas seguintes etapas:

- 1) Geração aleatória de uma população, na qual cada indivíduo corresponde a uma solução candidata ao problema;
- 2) Cálculo de um valor associado à qualidade de cada indivíduo através de uma função de aptidão (*fitness*) e teste de condição de parada;
- 3) aplicação de um operador de seleção dos melhores indivíduos;
- 4) Aplicação de um operador de cruzamento aos indivíduos selecionados na etapa anterior, para gerar os indivíduos da nova população, que estatisticamente podem ser melhores que os indivíduos da população corrente;
- 5) Aplicação de um operador de mutação aos indivíduos gerados na etapa anterior, de forma a evitar a perda de diversidade desses novos indivíduos;
- 6) Retorno à etapa (2).

O pseudo-código correspondente (AG canônico) está apresentado na Figura 3.2 abaixo.

```

Início:
  geração := 0;
  população_inicial(geração);
  avaliar_população(geração);
  enquanto(condição_de_parada ≠ 0)
    faça
      geração := geração + 1;
      nova_população(geração) = população(geração - 1);
      modificar_nova_população(geração); aplicar { operadores genéticos };
      avaliar_nova_população(tempo);
      população(geração) = nova_população(geração);
    fim
  fim
fim

```

FIGURA. 3.2 – Pseudo-código do algoritmo genético canônico.

Um algoritmo genético típico utiliza então três operadores: seleção, cruzamento e mutação. O operador de seleção tem a finalidade de aplicar uma pressão evolutiva na população de modo similar aos seres vivos. Indivíduos mais aptos possuem maior chance de sobrevivência e portanto maior capacidade de transmissão de suas características no processo evolutivo. O cruzamento permite que a informação genética dos indivíduos seja trocada, da mesma forma que na reprodução sexuada dos seres vivos. Em seguida, o operador de mutação altera de forma aleatória a informação genética resultante da combinação. A cada geração, todos os indivíduos da população correspondente são avaliados através de uma função de aptidão (*fitness*). O algoritmo é interrompido quando um critério de convergência é atendido ou o número total de gerações é atingido.

No algoritmo genético canônico, a função $f(t)$ que mede a aptidão de um indivíduo de uma população/geração t , denominada função de aptidão (*fitness*) é dada por:

$$f(t) = \frac{f_i}{f} \quad , \quad (3.2)$$

em que,

f_i = aptidão do i -ésimo indivíduo, e

\bar{f} = média das aptidões dos indivíduos da população.

Num AG, a aptidão pode ser calculada de acordo com o problema específico abordado. Na resolução de problemas inversos formulados como problemas de otimização, a aptidão de um indivíduo é calculada a partir do valor da função a ser minimizada (função objetivo), que por sua vez é calculado pelo modelo direto a partir dos parâmetros associados a esse indivíduo.

Após o cálculo de $f(t)$ para todos os indivíduos na população corrente, aplica-se um operador genético de seleção, geralmente aos indivíduos mais aptos. Entretanto, alguns operadores não necessariamente selecionam os indivíduos mais aptos, com a finalidade de manter a diversidade da população. Há diversos operadores de seleção, como a seleção por roleta (mais utilizado) e a seleção Boltzmann, explicados a seguir e outros, como a seleção por escalada sigma [49] e a seleção por *rank* [50].

Na seleção por roleta, os indivíduos de uma população/geração são escolhidos para fazer parte da próxima população/geração através de um sorteio similar ao de uma roleta. Cada indivíduo é representado por uma fração correspondente à sua aptidão, gerando um mapa similar às divisões de uma roleta. Calcula-se assim uma probabilidade P para cada indivíduo que é proporcional à sua aptidão. A rodada da roleta é feita através da geração de um número aleatório que vai indicar indivíduo a ser selecionado. A probabilidade P_i para cada um dos L indivíduos avaliados é dada por:

$$P = \frac{f_i}{\sum_{i=1}^L f_i}. \quad (3.3)$$

A seleção Boltzmann permite também que indivíduos com aptidão baixa sejam selecionados, mantendo a diversidade da população e evitando assim convergências

prematuras. O método utiliza a distribuição de Boltzmann para avaliar a probabilidade P de seleção de cada indivíduo i , numa população de L indivíduos, em função da aptidão f_i , e de uma pseudo-temperatura T escolhida arbitrariamente, conforme a expressão abaixo:

$$P = \frac{e^{-\frac{f_i}{T}}}{\sum_{i=1}^L e^{-\frac{f_i}{T}}}. \quad (3.4)$$

A temperatura T é inicializada com um valor alto, sendo gradativamente reduzida até a convergência do algoritmo. Esse processo é similar ao encontrado em algoritmos de recozimento simulado.

O operador genético de cruzamento é responsável pela recombinação da informação genética dos indivíduos de uma população, contida nos genes, permitindo que os indivíduos da geração seguinte herdem essas características. Esse operador genético pode ser implementado de várias maneiras, sendo as mais conhecidas:

- Operador de cruzamento com um ponto de recombinação: Gera-se um número aleatório para determinar a posição do cromossomo (ponto de cruzamento) a partir da qual os genes dos indivíduos pais serão permutados, gerando um novo indivíduo com herança genética dos pais.
- Operador de cruzamento com múltiplos pontos de recombinação [51]: Vários pontos de recombinação são gerados, ou seja, vários pontos de cruzamento no cromossomo, que determinam vários trechos nos quais os genes dos pais serão permutados.
- Operador de cruzamento uniforme [52]: Não utiliza pontos de recombinação nos indivíduos, mas determina através da geração de um número aleatório quais genes dos pais devem ser permutados.

O operador genético de mutação causa uma alteração nos genes dos indivíduos que irão compor a nova população. Um indivíduo qualquer pode ou não sofrer mutação, condicionado a uma probabilidade de mutação que tem um valor tipicamente entre 0 e

0,01. Esta probabilidade determina o número de genes que sofrerão mutação. A mutação contribui para evitar o confinamento a mínimos locais do espaço de busca na otimização, pois promove alterações que direcionam a busca para novos pontos desse espaço.

No caso de genes representados por números reais g , a mutação consiste na alteração do valor correspondente ao gene que vai sofrer mutação pela multiplicação do mesmo por um fator de mutação p conforme abaixo:

$$g = g \pm p \times g. \quad (3.6)$$

Esse fator p pode ser gerado aleatoriamente ou utilizando-se uma distribuição gaussiana.

No AG canônico, os operadores genéticos de seleção, cruzamento e mutação são aplicados aos indivíduos indiscriminadamente, sendo que não há restrições ao cruzamento de indivíduos similares, cujo cruzamento poderia ser considerado “incesto” e que poderia levar à propagação de possíveis seqüências genéticas defeituosas.

A idéia de prevenção de incesto é discutida por Eschelman [53] e Craighurst [54], que a aplicaram pioneiramente. No primeiro trabalho, que utiliza codificação binária, o cruzamento é permitido a indivíduos cujos genes apresentem uma distância de Hamming [55] acima de um certo valor pré-determinado. No segundo, critérios de prevenção específicos são aplicados para níveis de prevenção determinados. O primeiro nível de prevenção, nível 0, permite o cruzamento entre qualquer par de indivíduos, independente de parentesco. O nível 1 proíbe o cruzamento do próprio indivíduo. O nível 2 proíbe o cruzamento entre indivíduos pais, filhos e irmãos, enquanto o nível 3 restringe ainda mais o cruzamento, definindo indivíduos avós, netos, tios e sobrinhos.

Um AG é intrinsecamente passível de paralelização e o aumento da complexidade dos problemas causado por estruturas de dados cada vez maiores tem motivado o

desenvolvimento de implementações paralelas [56]. Podem ser citadas três abordagens na paralelização de AGs:

- No modelo ilha, cada processador da máquina paralela gera independentemente uma população isolada (às vezes dita subpopulação), sendo que há troca de indivíduos entre processadores segundo uma política de migração. Cada processador é portanto análogo a uma ilha. As subpopulações evoluem independentemente (exceto pela migração) e, para tal, devem possuir um número suficiente de indivíduos, o que faz com que a granularidade desse algoritmo seja grossa (ou seja, boa razão processamento/comunicação) para as políticas de migração mais comuns [57].
- No modelo global, uma única população é gerada e evoluída num processador mestre. Os demais processadores, escravos, executam as tarefas associadas à aplicação de um ou mais dos operadores genéticos (seleção, cruzamento e mutação) ou executam a avaliação da função de aptidão num subconjunto de indivíduos correspondente a cada processador. A comunicação de indivíduos entre o processador mestre e os escravos faz com que a granularidade desse algoritmo seja fina (ou seja, baixa razão processamento/comunicação), a não ser quando o cálculo da aptidão é custoso em relação às demais etapas, como é o caso do presente trabalho.
- Modelos híbridos são também possíveis, nos quais, por exemplo, uma única população é gerada num processador mestre, mas esta é dividida entre subpopulações independentes que são evoluídas pelos processadores escravos [58].

O AG do presente trabalho utiliza o modelo canônico com 100 indivíduos por geração/população. Devem ser estimados os valores de temperatura para 8.000 pixels e assim, cada indivíduo é representado por um conjunto de 8.000 temperaturas, correspondente a uma solução candidata possível. É utilizada codificação real, sendo

cada uma dessas temperaturas correspondente a um gene do indivíduo. O cálculo da aptidão de cada indivíduo consiste na comparação do valor estimado de cada pixel com a série temporal de medidas correspondente a esse pixel, conforme a Equação 3.1 (valor da função objetivo).

A implementação paralela segue o modelo global ou o modelo ilha. No primeiro caso, o processador mestre executa o AG e o cálculo da aptidão de cada indivíduo é dividido entre os demais processadores, i.e. cada processador fica a cargo do cálculo parcial da aptidão para seu conjunto de pixels. No modelo ilha, cada processador avalia os indivíduos de sua subpopulação localmente.

O AG utilizado foi implementado por meio de uma biblioteca de funções específicas para AG denominada PGAPack [59], desenvolvida pelo *Argonne National Laboratory* (ANL). Esta biblioteca fornece suporte à paralelização, que pode ser então efetuada com chamadas a diretivas da biblioteca de comunicação por troca de mensagens MPI.

3.2.2 Recozimento Simulado

O método de recozimento simulado (RS) tem seu nome inspirado num processo comum em metalurgia, no qual um metal é fundido e gradualmente esfriado até atingir um estado cristalino, ou seja, um estado mínimo global de energia termodinâmica. A idéia subjacente é quebrar o estado corrente de acomodação dos átomos para depois permitir, no resfriamento gradual, uma redistribuição mais homogênea dos átomos. O RS foi desenvolvido para problemas de otimização com espaço de busca muito extenso e com presença de mínimos locais, muitas vezes não tratáveis por métodos determinísticos. Em 1983, Kirpatrick *et al.* [60] resolveram um problema de otimização utilizando técnicas de mecânica estatística, tais como o algoritmo de Monte Carlo desenvolvido por Metropolis *et al.* [61].

Define-se uma pseudo-temperatura τ no escopo do algoritmo de RS, o qual é descrito a seguir. Esta pseudo-temperatura aparece explicitamente nas Equações 3.4 a 3.8 e na

Figura 3.2, não estando relacionada, obviamente, com as temperaturas correspondentes aos pixels do mapa do céu.

O RS emprega conceitos de mecânica estatística. O algoritmo define um estado inicial S_0 de energia E_0 associada a uma pseudo-temperatura τ_0 . A cada iteração i do RS, uma perturbação aleatória no posicionamento das partículas do metal é realizada alterando seu estado de energia de $E_{(i-1)}$ para E_i . A variação resultante de energia será $\Delta E = E_i - E_{(i-1)}$. Caso $\Delta E \leq 0$, o novo estado é aceito e caso contrário, um teste probabilístico é realizado e o novo estado pode ser aceito ou não. Para tal, um número aleatório r (entre 0 e 1) é gerado e comparado com $P(\Delta E)$. Caso $r < P(\Delta E)$, o novo estado é aceito. Isso possibilita evitar mínimos locais. A probabilidade de aceitar ou não o novo estado é dada pela distribuição de Boltzmann:

$$P(\Delta E) = e^{-\frac{\Delta E}{k_B \tau}}, \quad (3.4)$$

em que,

k_B = constante de Boltzmann;
 τ = pseudo-temperatura.

Esta equação mostra que pseudo-temperaturas maiores implicam em maior probabilidade de aceitar um novo estado. Por outro lado, quanto maior for o valor de ΔE menor a probabilidade do novo estado ser aceito. Nesse processo de recozimento o sistema é representado pelos parâmetros da solução candidata. Sucessivas perturbações são aplicadas ao sistema, de forma a executar uma varredura no espaço de busca, em particular na vizinhança em que se localizam pontos de mínimo da função analisada [62], condicionadas à redução gradual da pseudo-temperatura. Há vários esquemas que podem ser adotados para esta redução.

O modelo acima exposto recebe a denominação de recozimento simulado clássico ou ainda máquina de Boltzmann, no qual a pseudo-temperatura é reduzida segundo um resfriamento logarítmico dado em função da pseudo-temperatura inicial para uma iteração i por:

$$\tau_i = \frac{\tau_0}{\ln i}. \quad (3.5)$$

Alguns trabalhos utilizam uma redução exponencial para a temperatura, dada por:

$$\tau_i = \tau_0 e^{((c-1)i)}, \quad (3.6)$$

em que o parâmetro c tem um valor determinado empiricamente.

Szu et al., substituíram a distribuição de Boltzman por uma distribuição de Cauchy/Lorentz, método denominado de *Fast Simulated Annealing* (FSA) [63]. Nesse método, o resfriamento é linear, sendo a pseudo-temperatura na iteração i , dada por:

$$\tau_i = \frac{\tau_0}{i}. \quad (3.7)$$

No chamado *Adaptive Simulated Annealing* (ASA) [64], Jones substitui a distribuição de Boltzmann por uma distribuição uniforme:

$$\tau_i = \tau_0 e^{-ci^{1/N}}, \quad (3.8)$$

em que N representa o total de parâmetros e o parâmetro c tem um valor determinado empiricamente.

O pseudo código [65] do algoritmo de RS clássico é ilustrado na Figura 3.3.

```
Inicializa ( $S_0, \tau_0$ );
E := energia ( $S_0$ );
i := 0;
  Enquanto(condição_de_parada  $\neq 0$ )
     $S_i$  :=novo_estado( $S, \tau$ );
     $E_i$  := energia( $S_i$ );
    Se ( $E_i < E$ ) ou {aleatório()  $<$  prob( $E, \tau$ )};
      S :=  $S_i$ ;
      E :=  $E_i$ ;
       $\tau$  := temperatura( $\tau, i$ );
      i := i + 1;
  fim
fim
fim
```

FIGURA 3.3 – Pseudo-código do algoritmo de recozimento simulado.

Inúmeras versões paralelas do RS existem e estão disponíveis, porém a estratégia de paralelização difere em cada trabalho estudado. Por exemplo, o método implementado por Hamma , explora o paralelismo em termos de ciclos de pseudo-temperaturas: vários RS são executados independentemente pelos processadores, com pseudo-temperaturas iniciais diferentes [66]. Nesse mesmo método, a versão dada por Jones *et al.* propõe que um processador mestre determine o estado atual a cada iteração aos demais processadores que então geram novos estados independentemente utilizando suas próprias pseudo-temperaturas [67]. Os novos estados e suas correspondentes probabilidades de aceitação geradas pelos processadores escravos nessa iteração são comunicados ao processador mestre, que procede à seleção do novo estado e o comunica, de forma a que seja utilizado por todos os processadores na iteração seguinte.

Cheng *et al.* compararam duas implementações, uma de RS e outra de AG, quando aplicadas independentemente a um problema de identificação de imagens em tomografia. Os resultados mostram que ambos os métodos podem trazer bons resultados a um custo computacional equivalente [68].

No caso específico do problema de construção dos mapas de anisotropias da RCFM, os parâmetros a serem estimados são as temperaturas reais da região observada do céu. Cada solução candidata corresponde a um conjunto possível de temperaturas dos pixels do mapa. Similarmente à aptidão do AG, o cálculo da energia é expresso pela diferença quadrática entre o valor estimado de cada pixel com a série temporal de medidas correspondente a esse pixel, conforme a Equação 3.1.

O RS do presente trabalho emprega um resfriamento de pseudo-temperatura seguindo o modelo FSA. Foram definidos dois critérios de parada, prevalecendo o que ocorrer primeiro: um critério baseado num número limite de iterações e outro baseado na avaliação da função objetivo. Neste último critério, emprega-se um esquema adicional, denominado *re-annealing*. Caso a função objetivo não sofra variações superiores a um certo limite (no caso deste trabalho, 1%) durante um determinado número de iterações (no caso, 10 iterações), o algoritmo causa uma perturbação externa na solução candidata corrente.

Para aplicação dessa perturbação foi proposto neste trabalho um novo esquema, inspirado no ruído gaussiano considerado. A perturbação é determinada em função do desvio padrão σ dos 8.000 valores de temperatura correspondentes à solução candidata considerada e de uma constante $\xi = 0,4$. A pseudo-temperatura τ_i de cada pixel i é perturbada, obtendo-se uma nova pseudo-temperatura τ_i^n dada por:

$$\tau_i^n = \tau_i(1 - \sigma\xi). \quad (3.9)$$

Na implementação paralela deste trabalho, o processador mestre executa o RS sequencialmente e, similarmente à implementação paralela do AG, divide a avaliação de cada solução candidata entre os demais processadores. Cada processador calcula a diferença quadrática expressa pela Equação 3.1 para seu conjunto de pixels.

3.3 Computação Paralela

A seguir, são apresentados os conceitos relativos ao paralelismo implementado no presente trabalho e aspectos funcionais da biblioteca de comunicação por troca de mensagens MPI [69].

As dimensões das matrizes e vetores envolvidos no problema de construção de mapas das anisotropias da RCFM constituem uma limitação para qualquer método de solução. No caso específico do experimento BEAST, o número de medidas de temperatura M da série temporal S é de ordem $O(10^5)$, enquanto que o número de pixels N é $O(10^4)$. Assim, a paralelização constitui um auxílio importante para se obter mapas em tempo viável, principalmente no caso de construção de mapas utilizando métodos inversos implícitos ou iterativos.

Na computação paralela há dois paradigmas de programação: paralelismo de dados e paralelismo de tarefas [70]. No primeiro, cada processador executa o mesmo conjunto de instruções em seus dados locais, definidos pela partição do domínio entre os processadores. No paralelismo de tarefas, cada processador executa conjuntos diferentes de instruções. Pode-se, obviamente, ter uma forma de programação híbrida. Por exemplo, num esquema mestre-escravo, um processador mestre distribui os dados aos demais processadores, que são processados em paralelo e então esses processadores enviam seus respectivos resultados parciais ao processador mestre.

O paradigma de paralelismo de dados é normalmente implementado pelo esquema de programação paralela *Single Program, Multiple Data* (SPMD) [71], em que cada processador executa o mesmo conjunto de instruções em dados locais que são definidos em função do *rank* (identificação) do processador.

Em termos de arquitetura, as máquinas paralelas podem ser divididas em duas categorias: máquinas de memória compartilhada e máquinas de memória distribuída. As máquinas da primeira categoria se caracterizam pelo acesso comum a um único espaço

de endereços de memória pelos processadores. Essas máquinas são comumente denominadas multiprocessadores. Tipicamente, o tempo de acesso à memória é uniforme e cada processador executa uma cópia idêntica do sistema operacional, caso em que são denominadas máquinas *symmetric multiprocessor* (SMP). Nas máquinas da segunda categoria, cada processador acessa seu próprio espaço de endereços da memória. Essas máquinas, comumente denominadas multicomputadores, são tipicamente constituídas por nós independentes e autônomos interligados por uma rede de comunicação. Cada nó é uma máquina seqüencial ou SMP. Como os espaços de memória são disjuntos, a dependência de dados entre processadores requer comunicação entre os nós, o que normalmente é implementado por uma biblioteca de comunicação por troca de mensagens.

De maneira geral, o uso dos paradigmas de programação paralela requer, correntemente, que o programador desenvolva uma versão do código que contenha diretivas específicas de paralelismo [72]. Assim, o desempenho paralelo depende do algoritmo, o qual pode ser paralelizável ou não - neste último caso, no sentido de que não haveria redução do tempo de execução com a paralelização, ou então uma redução muito pequena. Esse desempenho depende também da arquitetura paralela considerada e da habilidade do programador.

Num programa paralelo existem pontos em que obrigatoriamente deve haver um sincronismo entre processadores para preservar a lógica do programa ou a coerência dos dados. Neste último caso, isso está relacionado a dependências de dados entre processadores. Por exemplo, um determinado dado somente pode ser lido por um processador após ser atualizado por outro. As instruções executadas entre esses pontos de sincronismo não são necessariamente executadas de forma síncrona devido a problemas tais como atrasos na comunicação e desbalanceamento de carga entre processadores.

A grande restrição à paralelização num dado algoritmo é devida à dependência de dados entre processadores, a qual gera problemas de contenção (e consistência) no acesso à

memória compartilhada de um multiprocessador ou então necessidades de comunicação entre os nós de um multicomputador. Em ambos os casos o desempenho paralelo é penalizado devido ao tempo gasto no acesso à memória ou na comunicação, sendo esses tempos rotulados genericamente como sendo um custo de comunicação, em qualquer das arquiteturas. Assim, um algoritmo paralelo deve minimizar as dependências de dados de forma a minimizar custos de comunicação e permitir um bom desempenho paralelo.

O custo de comunicação t_{com} para envio de uma mensagem num multicomputador pode ser dividido num tempo de latência t_l associado ao estabelecimento da conexão entre os nós, e o tempo de transmissão t_w para o envio dos dados, o qual depende do tamanho da mensagem e da taxa de transmissão (largura de banda) da rede de interligação e β representa a dimensão da informação a ser transmitida.

$$t_{com} = t_l + t_w \beta. \quad (3.10)$$

Uma métrica comum para avaliação de desempenho de um programa executado numa máquina paralela é o *speedup* ($\alpha(p)$), calculado a partir do tempo de execução do programa seqüencial (ΔT_{seq}) e o tempo de execução do programa paralelo ($\Delta T_{par}(p)$) para a mesma instância e tamanho de problema em função do número p de processadores utilizados. O *speedup* é definido por:

$$\alpha(p) = \frac{\Delta T_{seq}}{\Delta T_{par}(p)}. \quad (3.11)$$

A eficiência $\eta(p)$ definida para um programa executado numa máquina paralela é definida como a razão entre o *speedup* e o número de processadores p :

$$\eta(p) = \frac{\alpha(p)}{p}. \quad (3.12)$$

Numa situação ideal, o *speedup* tende a p e a eficiência, tende a 1. Um *speedup* igual a p é dito linear. Esses valores não são limitantes, pois a otimização do acesso à memória cache pode levar a valores um pouco superiores. Esta otimização é devida à divisão dos dados entre processadores, o que obviamente ocasiona uma quantidade de dados menor para cada processador, em relação ao programa seqüencial.

O tempo de execução seqüencial de um programa pode ser dividido numa fração seqüencial (f_{seq}), correspondente às instruções que tem que ser executadas seqüencialmente, e numa fração paralela (f_{par}), correspondente às instruções que poderiam ser executadas em paralelo, de forma que $f_{seq} + f_{par} = 1$. A Lei de Amdahl estabelece um limite para o *speedup* a partir das frações f_{seq} e f_{par} [73]:

$$\alpha(p) = \frac{\Delta T_{exe}}{\Delta T_{par}(p)} = \frac{f_{seq} + f_{par}}{\left(f_{seq} + \frac{f_{par}}{p}\right)} = \frac{1}{\left(f_{seq} + \frac{f_{par}}{p}\right)} \quad (3.13)$$

Assim, dispondo-se de um número de processadores p muito grande, o tempo (f_{par}/p) correspondente à fração paralelizável tenderia a zero e o *speedup* tenderia ao limite ($1/f_{seq}$). Ou seja, o *speedup* é limitado pela fração seqüencial do programa.

Uma outra métrica para se avaliar um dado algoritmo, implementado no correspondente programa paralelo, é a granularidade, que avalia, considerando a parte paralela do código, a razão entre a quantidade de processamento “útil” e a quantidade de comunicação requerida. A granularidade de um algoritmo é genericamente classificada em três valores relativos: fina, média e grossa, embora um algoritmo possa ter diferentes granularidades numa análise que considere diferentes trechos do mesmo. A granularidade grossa é a desejável, sendo que corresponde a alta razão processamento/comunicação, associada a valores altos de *speedup*, próximos ao

equivalente seqüencial. Em oposição, a granularidade fina corresponde a uma baixa razão processamento/comunicação.

Neste trabalho, na abordagem proposta de problema inverso resolvido por um método implícito ou iterativo, buscou-se avaliar a qualidade de solução e o desempenho dos algoritmos estocásticos referidos. O desempenho computacional foi unicamente avaliado em termos de tempo de execução paralelo destes algoritmos, sempre utilizando o mesmo número de processadores. Assim, as métricas acima não foram necessárias, uma vez que o objetivo era se obter um tempo de execução equivalente ao método correntemente utilizado, o método *bin average*.

Uma forma eficiente e de baixo custo de se implementar uma máquina paralela consiste na utilização de microcomputadores como nós independentes de uma máquina paralela de memória distribuída. A rede de interconexão é tipicamente de padrão *Fast Ethernet* (largura de banda nominal de 100×10^6 bits/s) ou *Gigabit Ethernet* (1×10^3 Mbits/s) e utiliza sempre um comutador (*switch*) que permite múltiplas conexões simultaneas entre os nós. Quando se utiliza o sistema operacional Linux em cada nó, essa máquina é tipicamente denominada *Cluster Beowulf*, em referência ao primeiro multicomputador com essas características, desenvolvido pelo *Center of Excellence in Space Data and Information Sciences* (CESDIS) em 1994 [74].

A máquina paralela utilizada neste trabalho é um cluster [75] composto por 16 nós de monoprocessadores com arquitetura Intel IA-32, Pentium III, com frequência de relógio de 800 MHz. Cada nó tem 256 MB de memória principal e 20 GB de disco rígido. A rede de interconexão e o comutador seguem o padrão *Fast Ethernet*. O sistema operacional adotado é o Linux, *kernel 2.2* [76]. Na paralelização do código foi empregada a biblioteca de comunicação por troca de mensagens MPI. Como em bibliotecas similares, as diretivas do MPI permitem implementar o paradigma de paralelismo de dados e/ou de paralelismo de tarefas, sendo o programador responsável pela distribuição dos dados/tarefas e pela especificação e controle da comunicação de dados e do sincronismo entre processadores.

No caso deste trabalho, o algoritmo AG modelo global ou o algoritmo de RS é executado seqüencialmente num processador mestre, sendo a avaliação da qualidade da solução dividida entre os demais processadores. Como já exposto, essa avaliação consiste em calcular a diferença quadrática entre o valor da temperatura estimada para um dado pixel e as medidas contidas na série temporal desse pixel. Cada processador efetua esse cálculo para seu conjunto de pixels. No AG modelo ilha, cada processador evolui sua própria subpopulação e avalia os indivíduos da mesma.

CAPÍTULO 4

RESULTADOS NUMÉRICOS E DE DESEMPENHO

Este capítulo apresenta os resultados numéricos e de desempenho obtidos através da implementação dos algoritmos estocásticos discutidos no capítulo 3. Na abordagem proposta neste trabalho, estes algoritmos permitem resolver o problema inverso associado à construção de mapas de anisotropias da RCFM por uma metodologia implícita (iterativa). Estes algoritmos foram implementados e testados, sendo apresentados a seguir. Em todos os casos, utilizou-se compilador *C* da Intel e o cluster da DAS, descrito na seção 3.3.

Para testar o algoritmo inverso proposto, foram utilizados dados simulados referentes ao telescópio BEAST para uma sub-região. Nos experimentos relativos a este instrumento, os dados adquiridos são armazenados em séries temporais resultantes de sessões consecutivas de 1 hora de aquisição diária. Em cada determinada hora de um dia são executadas varreduras parciais da correspondente sub-região do céu. Os dados simulados correspondem a sessões horárias de 160.000.000 de medidas da sub-região considerada, ou 20.000 medidas de temperatura por pixel para $N = 8.000$ pixels. Foi simulado um período de 30 dias, num total de $M = 4.800.000.000$ medidas ou 600.000 medidas por pixel, que correspondem a medidas dessa sub-região, sempre efetuadas na mesma hora do dia.

Para a construção do mapa simulado de anisotropias correspondente às temperaturas observadas, é utilizado o aplicativo SYNFAST [77]. O vetor de temperaturas simuladas é denominado aqui vetor inicial e corresponde à solução exata procurada, é retirado de uma faixa do mapa gerado pelo SYNFAST. Um gerador de números aleatórios [78] fornece os valores de ruído gaussiano para contaminar as temperaturas do vetor inicial. O ruído é ajustado de forma que a relação sinal/ruído das medidas simuladas de temperatura fique próxima ao valor $0,1$ e que sua média seja.

A avaliação da qualidade da solução obtida pelo algoritmo inverso proposto é feita, além do tempo de execução do código implementado, por meio de dois parâmetros, a correlação e o desvio médio, explicados a seguir. Avalia-se a correlação de Pearson, ρ , entre o vetor de temperaturas simuladas sem ruído (vetor inicial T) e o vetor de temperaturas estimadas T_e conforme abaixo:

$$\rho = \frac{\text{cov}(T, T_e)}{\sigma_T \sigma_{T_e}}, \quad (4.1)$$

em que,

$$\text{cov}(T, T_e) = \frac{\sum_{i=1}^N (T_i - \bar{T})(T_{ei} - \bar{T}_e)}{N}, \quad (4.2)$$

e,

$$\bar{T} = \frac{\sum_{i=1}^N T_i}{N} \text{ e } \bar{T}_e = \frac{\sum_{i=1}^N T_{ei}}{N}. \quad (4.3)$$

O desvio médio α_i refere-se às diferenças entre vetor inicial T e a média das temperaturas para o pixel i . Quanto menor o desvio médio, menor a diferença entre esses vetores e, portanto, melhor a solução obtida. O desvio médio é dado por:

$$\alpha_i = \left\langle \left| T_i - \bar{T} \right| \right\rangle = \frac{1}{N} \sum_{i=1}^N (T_i - \bar{T}), \quad \bar{T} = \langle T_i \rangle = \frac{1}{N} \sum_{i=1}^N T_i. \quad (4.4)$$

Para efeito de comparação, o algoritmo inverso proposto é comparado a um método comumente utilizado, o método *bin average*, no qual calcula-se, para cada pixel, a média de todos os valores de temperatura medidos, referentes ao período de aquisição

considerado. No caso do experimento BEAST, essas temperaturas correspondem a séries temporais de 1 hora de aquisição, cada uma armazenada num arquivo separado, e a média é calculada a partir das médias parciais (Equação 2.27).

Convém ressaltar que o método *bin average* é normalmente implementado por meio de *scripts* que são interpretados pelo aplicativo IDL. Devido à sua natureza interpretativa, estes *scripts* implicam em tempo elevados de execução. Então, para efeito de comparação com o método proposto, o método *bin average* foi implementado na linguagem C, executado seqüencialmente, o que implicou obviamente um tempo de execução consideravelmente inferior ao obtido no IDL, aproximadamente 5 vezes menor.

O algoritmo inverso implícito proposto foi implementado por meio de algoritmo genético, com resultados expostos na Seção 4.1, e por meio de algoritmo de recozimento simulado, com resultados expostos na Seção 4.2. Os resultados apresentados referentes à qualidade da solução e de desempenho são médias aritméticas de execuções repetidas para cada algoritmo testado.

4.1 Resultados Obtidos Utilizando Algoritmo Genético

Um AG canônico paralelo foi implementado e tomado como referência em termos da qualidade da solução obtida e de tempo de execução para comparação com outro AG canônico, que teve duas implementações paralelas diferentes. Este AG canônico de referência tem população de 100 indivíduos, apresenta seleção do tipo roleta, cruzamento com um único ponto de recombinação e mutação fixa de 1 % e foi paralelizado segundo o modelo global, descrito na Seção 3.2.1. Cada um dos 8000 genes que compõem um indivíduo é codificado como um número real do intervalo definido por $-999 \mu K$ e $+999 \mu K$.

Este AG e algumas variações do mesmo foram implementados utilizando-se funções do PGAPack e paralelizados por meio de chamadas à biblioteca MPI. O tamanho da população foi determinado com base em testes preliminares, levando em conta a

qualidade da solução obtida e o tempo de execução. Estes testes indicaram um tamanho ótimo de população entre 100 e 200 indivíduos, tendo-se optado pelo limite inferior para se obter tempos de execução menores.

Uma variação deste AG canônico de referência utiliza o mesmo tamanho de população, seleção por distribuição de Boltzmann, cruzamento com 2 pontos de recombinação e mutação fixa de 0,5 % aplicada apenas aos genes que não sofreram cruzamento. O total de genes que sofrem mutação nesse caso, é da ordem de 40 pixels. Esta variação foi denominada AG Boltzmann, sendo distinguidas duas implementações paralelas correspondentes: uma, segundo o modelo global (AG Boltzmann global) e outra segundo o modelo ilha (AG Boltzmann ilha).

Nos AGs implementados, utilizou-se a codificação genética com números reais. Assim, cada indivíduo é caracterizado por um conjunto de números reais associados à temperatura de todos os pixels do mapa que representa uma possível solução.

Teoricamente, uma codificação binária ou inteira seria menos custosa em termos de acesso à memória, mas testes preliminares demonstraram que, na prática e para este problema, não houve diferença significativa em relação à codificação real adotada [79].

Para efeito de comparação da qualidade da solução, foram utilizados os parâmetros de avaliação já descritos, a correlação e desvio médio das temperaturas estimadas em relação ao chamado vetor inicial, que corresponde à solução exata.

Os dados simulados representam uma determinada hora de aquisição amostrada diariamente para um período de 30 dias, representando assim um total de 8.000 pixels com 600.000 valores de temperatura para cada pixel. A Figura 4.1 mostra os valores de temperatura simulados, sem ruído, para os primeiros 50 pixels de um mapa relativo ao experimento BEAST, gerados pelo SYNFAST e o correspondente ruído adicionado a essas temperaturas. O eixo das ordenadas mostra os valores de temperaturas, expressos

em mK , em relação à temperatura média de 2,716 K. Pode-se observar a variação do ruído em relação ao sinal (temperaturas), típica nesse tipo de dados.

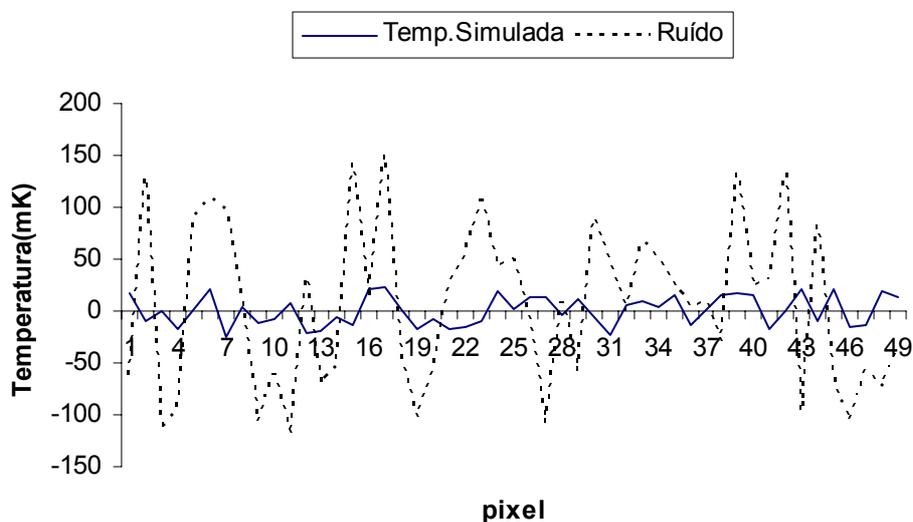


FIGURA 4.1 – Dados simulados de temperatura e ruído correspondente.

Para análise da convergência, os AGs paralelos implementados foram comparados quanto ao valor da função objetivo em função do número de iterações, conforme a Figura 4.2 para o AGs canônico de referência, Boltzmann global e modelo Boltzmann Ilha. Pode-se observar que a convergência do AG canônico tem valores próximos à convergência do AG Boltzmann ilha.

O modelo ilha utiliza todos os processadores do cluster, evoluindo assim 16 subpopulações com 100 indivíduos cada. À semelhança do AG global, este mostrou ser o tamanho ótimo de subpopulação em termos de qualidade de solução e desempenho. A política de migração baseia-se na seleção dos 10 melhores indivíduos de todas as subpopulações a cada 100 iterações/gerações. Esses conjuntos de 10 melhores indivíduos são enviados ao processador mestre que então os distribui aleatoriamente para os processadores, sendo incluído o próprio processador mestre nesse esquema. Dessa forma, cada processador recebe os 10 melhores indivíduos de algum outro processador e os utiliza para formar a subpopulação correspondente à iteração seguinte daquela da migração.

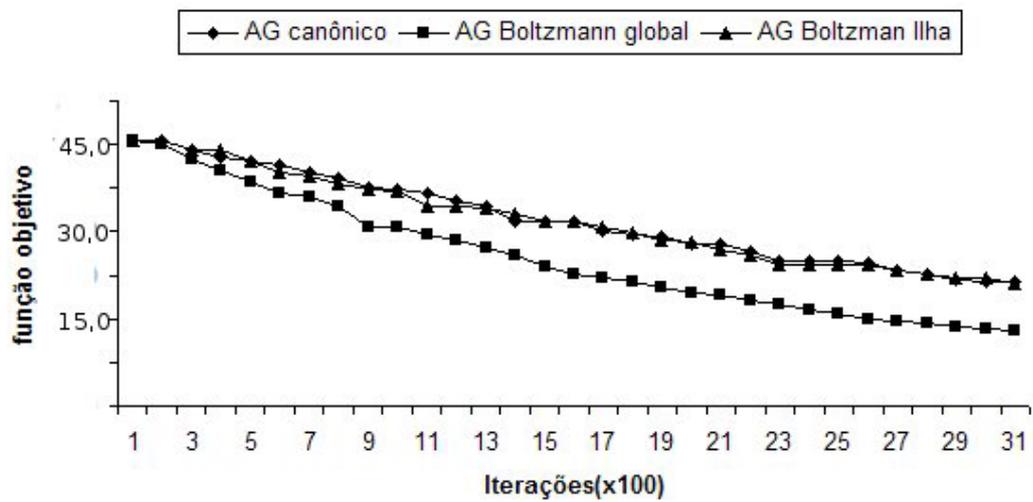


FIGURA 4.2 – Convergência dos AGs testados: valor da função objetivo em função do número de iterações.

A Figura 4.3 compara a qualidade da solução obtida pelo AG Boltzmann global para os 100 primeiros pixels com a solução exata: é mostrada, para cada pixel, a diferença em μK entre a temperatura exata e a estimada. Note-se que essas diferenças variam muito, ou seja, a qualidade da solução é muito irregular em função do pixel considerado. Esse comportamento foi também observado nos demais AGs aqui analisados, mas optou-se por mostrar apenas o AG Boltzmann global, que foi o que apresentou a melhor solução.

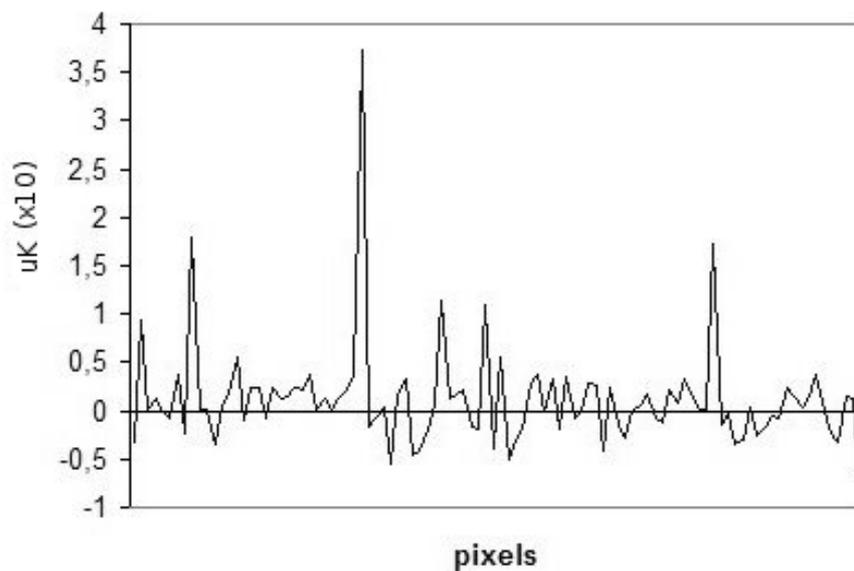


FIGURA 4.3 – Diferença entre a temperatura exata e a temperatura estimada pelo AG Boltzmann global para os 100 primeiros pixels do mapa simulado.

Finalmente, os AGs paralelos implementados são comparados quanto à qualidade da solução obtida, sendo também comparados a um método tradicionalmente usado, o *bin average*. A Tabela 4.1 apresenta os valores obtidos para a correlação e desvio médio, os quais demonstram que todos forneceram soluções aceitáveis. Os AGs Boltzmann global e ilha apresentam resultados bons e melhores que o AG de referência. A tabela mostra também os valores de convergência da função objetivo e o número de iterações até a convergência. Entretanto, o melhor resultado ainda é o obtido pelo método *bin average*, e adicionalmente com menor tempo de execução, como exposto na Seção 4.3.

TABELA 4.1 – Comparação das soluções obtidas pelos diversos AGs e pelo método *bin average*.

Algoritmo	Desvio Médio	Correlação	Função Objetivo (conv.)	Iterações (conv.)
AG canônico global	0,354	0,956	25,05	3.112
AG Boltzmann global	0,300	0,986	15,12	3.098
AG Boltzmann ilha	0,320	0,980	22,31	3.115
<i>bin average</i>	0,290	0,995	-	-

4.2 Resultados Obtidos Utilizando Recozimento Simulado

Foram implementados algoritmos RS que utilizam as distribuições de Boltzmann, Cauchi (*FSA*) e uniforme (*ASA*). A solução candidata inicial foi gerada aleatoriamente para as três implementações, conforme é feito usualmente em RS. Nestas implementações, a pseudo-temperatura inicial adotada foi $\tau = 5.0$. Para os mesmos dados simulados utilizados na seção anterior, essas três implementações forneceram valores muito bons de desvio médio e correlação. Estes valores são bem próximos, conforme mostrado na Tabela 4.2 que também inclui estes valores para a solução obtida pelo método *bin average*. A tabela mostra também os valores de convergência da função objetivo e o número de iterações até a convergência. Entretanto, dos três algoritmos RS, aquele com distribuição Cauchy obteve o menor tempo de execução conforme discutido na Seção 4.3.

TABELA 4.2 – Comparação das soluções obtidas pelos diversos algoritmos RSs e pelo método *bin average*.

Algoritmo	Desvio Médio	Correlação	Função Objetivo (conv.)	Iterações (conv.)
RS distribuição Boltzmann	0,285	0,999	18,51	1.514
RS distribuição Cauchy	0,250	0,999	15,48	1.497
RS distribuição Uniforme	0,290	0,999	19,47	1.511
<i>bin average</i>	0,290	0,995	-	

A Figura 4.4 apresenta a convergência de cada RS implementado. Observa-se que, para os três casos, que após aproximadamente 1.500 iterações a função objetivo começa a convergir para um valor estável.

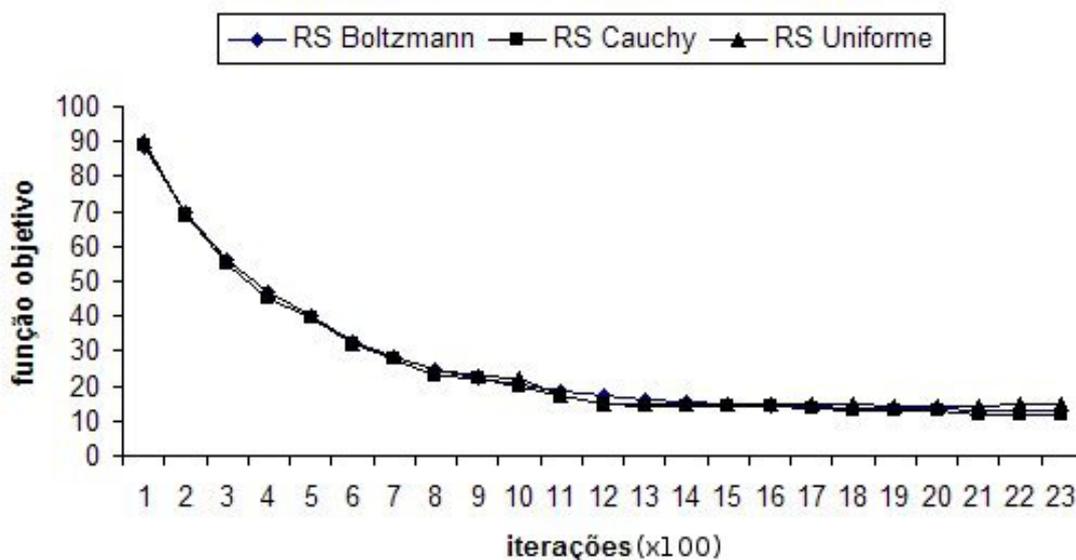


FIGURA 4.4 – Convergência dos RSs testados: valor da função objetivo em função do número de iterações.

Como o número total de iterações é elevado nos três RSs implementados, buscou-se explorar neste trabalho alternativas para geração da solução candidata inicial de forma a

diminuir o número total de iterações e obter um tempo de execução compatível com o do método *bin average*. A Seção 4.2.1 apresenta os resultados obtidos para solução candidata inicial obtida através do AG Boltzmann global, enquanto que a Seção 4.2.2 apresenta os resultados obtidos para uma solução candidata inicial obtida amostrando-se aleatoriamente, para cada pixel, medidas da série temporal simulada de medidas de temperatura.

4.2.1 Resultados Obtidos para o Algoritmo RS com Distribuição de Cauchy e Solução Candidata Inicial Gerada por Algoritmo Genético

Apresenta-se aqui a solução obtida pelo RS Cauchy descrito anteriormente quando se utiliza, conforme proposto neste trabalho, uma solução candidata gerada pelo AG Boltzmann global descrito na Seção 4.1. Este AG apresenta convergência conforme mostrado na Figura 4.2, para ~ 3.100 iterações. Pode-se observar que o valor da função objetivo decai continuamente com o número de iterações. Tentativamente, optou-se por utilizar a solução candidata da iteração 700 deste AG como solução candidata inicial para o algoritmo RS Cauchy, que convergiu com aproximadamente 1.000 iterações (além das 700 do AG), em vez das cerca de 1.500 iterações requeridas quando a solução candidata inicial foi gerada aleatoriamente. A convergência ocorreu para um valor semelhante da função objetivo.

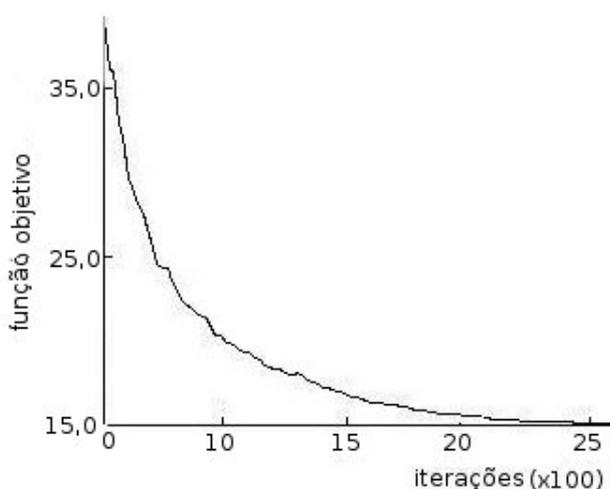


FIGURA 4.5 – Convergência do RS Cauchy com solução candidata inicial gerada por AG: valor da função objetivo em função do número de iterações.

A solução obtida apresentou qualidade similar, em termos de desvio médio (no caso 0,250) e correlação (0,999), que a obtida pelo RS com distribuição Cauchy com solução candidata inicial gerada aleatoriamente, mostrada na Tabela 4.2.

4.2.2 Resultados Obtidos para o Algoritmo RS com Distribuição de Cauchy e Solução Candidata Inicial Amostrada das Medidas de Temperatura

Apresenta-se aqui a solução obtida pelo RS Cauchy descrito anteriormente quando se utiliza, conforme também proposto neste trabalho, uma solução candidata inicial constituída por valores amostrados das séries temporais simuladas de medidas de temperatura. Nesta solução inicial cada pixel recebe o valor de uma de suas medidas correspondentes, sendo este valor escolhido aleatoriamente. Este esquema busca explorar a suposta distribuição gaussiana do ruído. A Figura 4.6 apresenta o gráfico de convergência do algoritmo RS Cauchy para essa solução candidata inicial.

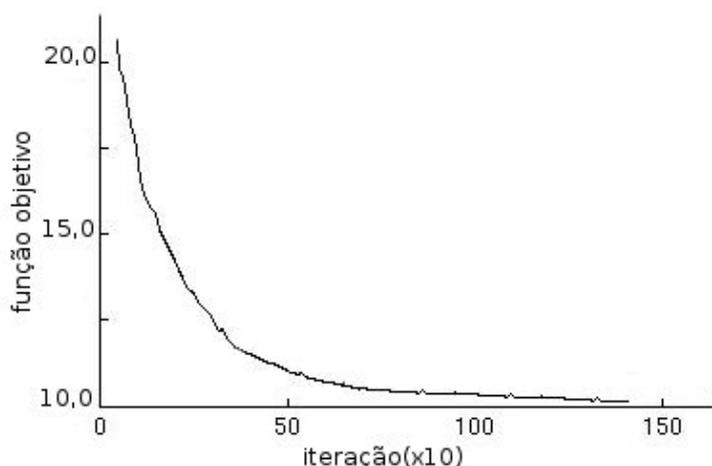


FIGURA 4.6 – Convergência do RS Cauchy com solução candidata inicial amostrada das medidas de temperatura: valor da função objetivo em função do número de iterações.

Observa-se neste gráfico que a função objetivo decaiu de maneira muito mais acentuada, ou seja, foram requeridas bem menos iterações para a convergência do algoritmo (uma ordem de grandeza abaixo). Em adição, o valor da função objetivo obtido na convergência é também menor.

Uma alternativa seria gerar a solução candidata inicial por meio da aplicação do método *bin average* em uma das séries temporais da região avaliada. A solução obtida pelo RS Cauchy com solução inicial assim gerada apresentou qualidade equivalente àquela obtida com solução candidata inicial amostrada da série temporal. Entretanto, o tempo de execução foi maior, devido à aplicação do método *bin average*.

A Tabela 4.3 apresenta os valores de correlação e desvio médio da solução obtida neste caso, comparando esta solução com as obtidas pelo mesmo algoritmo utilizando solução candidata inicial gerada aleatoriamente ou por AG, casos que foram analisados anteriormente, além da obtida como método *bin average*. A tabela mostra também os valores de convergência da função objetivo e o número de iterações até a convergência.

TABELA 4.3 – Comparação das soluções obtidas pelos algoritmos RS Cauchy para diversos esquemas de geração de solução candidata inicial e pelo método *bin average*.

Algoritmo	Desvio Médio	Correlação	Função Objetivo (conv.)	Iterações (conv.)
RS Cauchy (sol. inicial gerada aleatoriamente)	0,250	0,999	15,48	1.497
RS Cauchy (sol. inicial gerada por AG)	0,250	0,999	15,75	1.723
RS Cauchy (sol. inicial gerada por amostragem)	0,200	0,999	10,24	994
<i>bin average</i>	0,290	0,995	-	-

4.2.3 Resultados Obtidos para o Algoritmo RS com Distribuição de Cauchy com *re-annealing* e Solução Candidata Inicial Amostrada das Medidas de Temperatura

Apresenta-se aqui o algoritmo de RS com distribuição de Cauchy melhorado por meio do esquema de *re-annealing*, já descrito na Seção 3.2. A solução inicial é também gerada pela amostragem das medidas de temperatura, como na seção anterior. Para o *re-annealing*, aplica-se o fator de correção proposto neste trabalho e expresso pela Equação 3.9 à solução candidata corrente, sempre que não houver diminuição do valor da função objetivo após 10 iterações. A convergência deste algoritmo é mostrada na Figura 4.7.

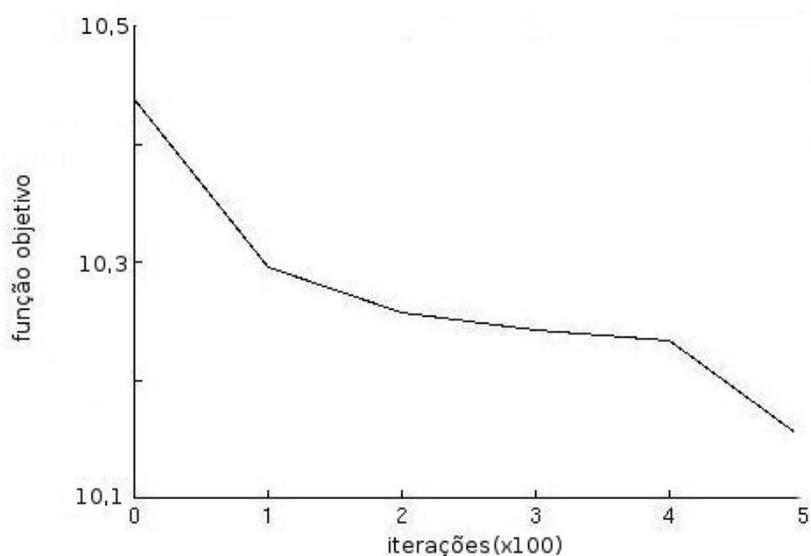


FIGURA 4.7 – Convergência do RS Cauchy com *re-annealing* e com solução candidata inicial amostrada das medidas de temperatura: valor da função objetivo em função do número de iterações.

O *re-annealing*, e sua conseqüente perturbação na solução candidata, pode ser observado neste gráfico aproximadamente na iteração 40. Nas iterações seguintes, o valor da função objetivo diminuiu sensivelmente.

A Tabela 4.4 mostra os valores de correlação, desvio médio da solução e o total de iterações executadas para algoritmo RS com distribuição de Cauchy com solução inicial gerada por amostragem das medidas, com e sem *re-annealing*, comparando-os novamente com o método *bin average*. A tabela mostra também os valores de convergência da função objetivo e o número de iterações até a convergência.

TABELA 4.4 – Comparação das soluções obtidas pelo algoritmo RS Cauchy com e sem *re-annealing* e pelo método *bin average*.

Algoritmo	Desvio Médio	Correlação	Função Objetivo (conv.)	Iterações (conv.)
RS Cauchy com re-annealing	0,150	0,999	10,05	522
RS Cauchy sem re-annealing	0,200	0,999	10,24	994
<i>Bin average</i>	0,290	0,995	-	-

Essas mesmas soluções são comparadas quanto ao módulo da diferença entre a temperatura exata e a estimada para os 10 primeiros pixels na Figura 4.8.

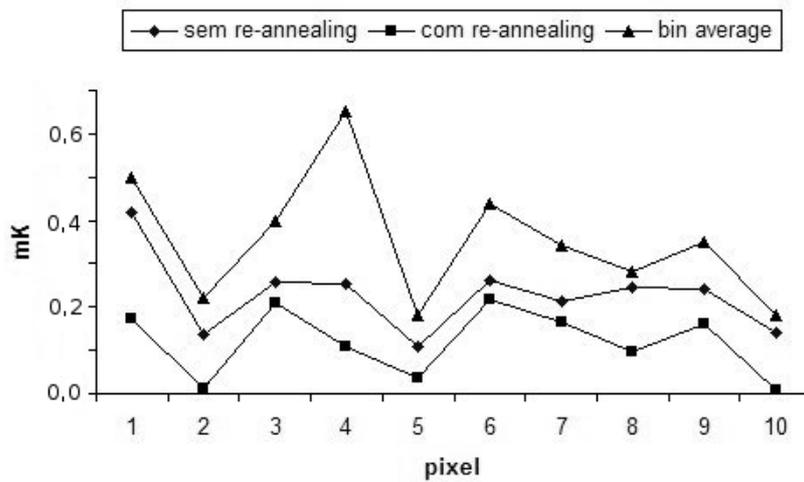


FIGURA 4.8 – Módulo da diferença entre a temperatura exata e a temperatura estimada para o algoritmo RS com distribuição de Cauchy com e sem *re-annealing* e para o método *bin average*, para os primeiros 10 primeiros pixels do mapa simulado.

A Figura 4.9 mostra os módulos da diferença entre a temperatura exata e as temperaturas estimadas pelo RS com distribuição Cauchy e estimativa inicial por amostragem com *re-annealing* e pelo método *bin average* para uma faixa selecionada de 20 pixels do mapa simulado. Um comportamento similar pode ser observado para os demais pixels.

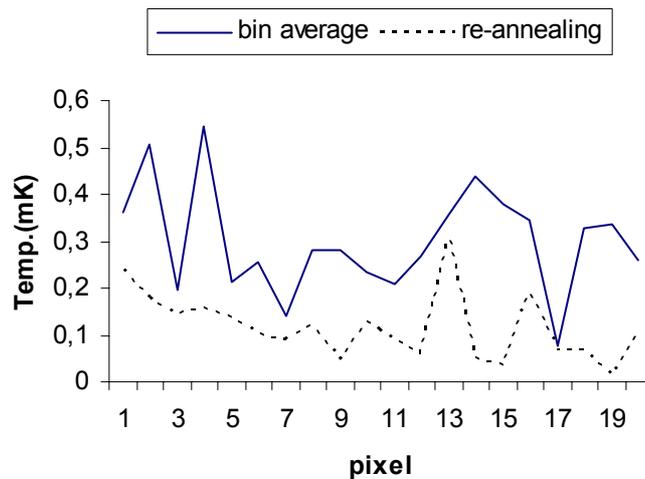


FIGURA 4.9 – Módulo da diferença entre a temperatura exata e as temperaturas estimadas pelo algoritmo RS com distribuição de Cauchy com *re-annealing* e pelo método *bin average*, para uma faixa de 20 pixels do mapa simulado.

4.3 Análise de Desempenho Computacional

Discutem-se a seguir os resultados de desempenho computacional obtidos pelos AGs e algoritmos de RS paralelos implementados. Como a proposta deste trabalho é propor uma formulação inversa implícita, utilizando AG ou RS paralelos, como alternativa ao método *bin average*, o desempenho dessas novas implementações foi comparado àquele do método *bin average* em termos de tempo de execução para a versão seqüencial deste último. Obviamente, o método *bin average* poderia ser paralelizado o que diminuiria significativamente seu tempo de execução, mas o objetivo deste trabalho foi propor uma alternativa que possibilitasse soluções com melhor qualidade, mesmo a um custo computacional maior. Como mencionado, este custo foi amenizado com o uso de processamento paralelo.

Esses tempos de execução foram obtidos, nas implementações paralelas de AG e RS, utilizando os 16 processadores do cluster da DAS, descrito na Seção 3.3. O tempo de

execução seqüencial do método *bin average* foi obtido utilizando-se um nó desse cluster.

A Tabela 4.5 mostra os tempos de execução obtidos para as três implementações, o método *bin average* (seqüencial), o AG Boltzmann global (paralelo) e o AG Boltzmann ilha (paralelo). O tempo de execução para o método *bin average*, mesmo considerando-se sua versão seqüencial é bem inferior aos tempos para os demais. O número de iterações até a convergência do algoritmo também é mostrado na tabela.

TABELA 4.5 – Comparação dos tempos de execução do método *bin average* e AG com distribuição Boltzmann e paralelização modelo global ou ilha.

Algoritmo	Tempo de execução (min.)	Iterações (conv.)
<i>bin average</i> (seqüencial)	110	-
AG Canônico global (paralelo)	1.095	3.112
AG Boltzmann global (paralelo)	1.021	3.098
AG Boltzmann Ilha (paralelo)	1.089	3.115

Entretanto, o tempo de execução do algoritmo de RS Cauchy paralelo diminuiu significativamente com o uso solução inicial candidata obtida por amostragem da série temporal simulada de medidas de temperatura. O uso de *re-annealing* neste algoritmo demandou um pouco mais de tempo de execução, mas forneceu, como visto na seção anterior, uma solução melhor.

A Tabela 4.6 compara o tempo de execução para os método *bin average* (seqüencial) com os tempos obtidos pelo algoritmo RS (paralelo) utilizando as distribuições de Boltzmann, Cauchy ou Uniforme. Nos três casos, a geração da solução inicial candidata

é realizada aleatoriamente. O número de iterações até a convergência do algoritmo também é mostrado na tabela.

TABELA 4.6 – Comparação dos tempos de execução do método *bin average* e RS com distribuições de Boltzmann, Cauchy ou uniforme.

Algoritmo	Tempo de execução (min.)	Iterações (conv.)
<i>bin average</i> (seqüencial)	110	-
RS Boltzmann (paralelo)	801	1.514
RS Cauchy (paralelo)	792	1.497
RS Uniforme (paralelo)	799	1.511

A Tabela 4.7 compara o tempo de execução para os método *bin average* (seqüencial) e os tempos obtidos utilizando o algoritmo RS com distribuição de Cauchy (paralelo) para a seleção de solução candidata inicial gerada por AG, gerada aleatoriamente ou gerada por amostragem das medidas da série temporal de dados. O número de iterações até a convergência do algoritmo também é mostrado na tabela.

TABELA 4.7 – Comparação dos tempos de execução do método *bin average* e RS Cauchy para diversos esquemas de geração de solução candidata inicial.

Algoritmo	Tempo de execução (min.)	Iterações (conv.)
<i>bin average</i> (seqüencial)	110	-
RS Cauchy (paralelo, solução inicial gerada por AG)	1.227	1.723
RS Cauchy (paralelo, solução inicial gerada aleatoriamente)	792	1.497
RS Cauchy (paralelo, solução inicial por amostragem)	201	994

No algoritmo RS Cauchy com solução inicial gerada por AG, observa-se um tempo de execução bem superior aos demais, uma vez que este tempo inclui a geração da solução inicial pelo AG (o número de iterações também inclui aquelas do AG).

A Tabela 4.8 compara o tempo de execução para os métodos *bin average* (seqüencial) e RS (paralelo) com distribuição Cauchy, com solução inicial candidata obtida por amostragem, com e sem o uso de *re-annealing*. O número de iterações até a convergência do algoritmo também é mostrado na tabela.

TABELA 4.8 – Comparação dos tempos de execução do método *bin average* e RS com e sem *re-annealing*.

Algoritmo	Tempo de execução (min.)	Iterações (conv.)
<i>bin average</i> (seqüencial)	110	-
RS Cauchy (paralelo) com <i>re-annealing</i>	122	552
RS Cauchy (paralelo) sem <i>re-annealing</i>	201	994

Pode-se observar que o algoritmo RS Cauchy paralelo com ou sem *re-annealing* constitui uma alternativa viável, em termos de tempo de execução, ao método *bin average*, com a vantagem de fornecer uma solução de melhor qualidade.

CAPÍTULO 5

CONCLUSÕES

A Radiação Cósmica de Fundo em Microondas (RCFM) constitui a principal evidência experimental que sustenta a teoria de evolução do Universo correntemente aceita. Esta teoria apóia-se em três fatos observacionais: o Universo é homogêneo e isotrópico em grande escala, encontra-se em expansão e existe abundância de elementos leves na atmosfera das estrelas, sendo o espaço banhado por uma radiação térmica, a RCFM, a qual possui uma temperatura atual média de $\sim 2,716$ K. Esses fatos permitem que se estabeleça a hipótese de que as propriedades e as distribuições de matéria de uma determinada região observada do Universo são as mesmas em todas as outras regiões. A melhor evidência dessa isotropia do Universo consiste na distribuição observada das medidas de temperaturas da RCFM.

Experimentos embarcados em satélites e em balões estratosféricos transportam instrumentos que fazem medidas de temperatura associada à RCFM de determinadas regiões do céu. Os dados adquiridos formam uma série temporal de medidas de temperatura contaminadas com ruído. O mapa de anisotropias da RCFM é a representação gráfica da distribuição dessas medidas de temperatura.

Correntemente, para resolução do problema inverso associado à construção dos mapas de anisotropia da RCFM, existem vários métodos analíticos derivados do método dos mínimos quadrados, os quais envolvem matrizes de difícil inversão e demandam muito processamento devido ao número de pixels e ao número de medidas de temperatura correspondente a cada pixel. Um método muito utilizado, o *bin average*, utiliza uma simplificação que permite reduzir o cálculo da temperatura estimada para cada pixel ao cálculo da média aritmética das medidas de temperatura desse pixel.

O presente trabalho propõe uma nova alternativa, utilizando uma abordagem implícita para resolução do problema inverso associado à construção dos mapas de anisotropias

da RCFM. Esse problema inverso é formulado como um problema de otimização e resolvido por um algoritmo estocástico. Um aspecto importante é que a abordagem proposta permite realizar estimativas de temperatura utilizando todo o volume de dados, o que em tese possibilitaria a geração de mapas de melhor qualidade.

Neste trabalho foram avaliados, para a resolução do problema de otimização associado, os algoritmos estocásticos correspondentes a variações de um algoritmo genético (AG) e de um algoritmo de recozimento simulado (RS). Estes algoritmos foram paralelizados por meio de chamadas à biblioteca de comunicação por troca de mensagens MPI e executados numa máquina paralela de memória distribuída, um *cluster* de 16 nós.

Para fins desta avaliação, foram gerados dados simulados de temperaturas do experimento BEAST referentes a um caso de teste, compostos por 600.000 medidas de temperatura para cada um dos 8.000 pixels do mapa correspondentes a um período de aquisição de 30 dias. Esses dados foram contaminados com ruído gaussiano com relação sinal/ruído igual a 0,1.

Métodos determinísticos, tais como o método Levenberg-Marquadt e o método Hookes e Jeeves, foram implementados para resolução do problema inverso associado à construção de mapas de anisotropias de RCFM e avaliados em relação ao método *bin average*, utilizado como referência de qualidade da solução e de desempenho. Esses métodos apresentaram um custo computacional bem maior que o demandado pelo método *bin average*, devido principalmente às dimensões das matrizes envolvidas. E as soluções obtidas apresentaram qualidade inferior àquela obtida pelo método *bin average* para os mesmos dados simulados. Em particular, o método de Hookes e Jeeves apresentou uma solução melhor que aquela obtida com o método *bin average* para dados com relação sinal/ruído superior a 0,5. Entretanto, dados típicos de temperatura associada à RCFM apresentam relação sinal/ruído da ordem de 0,1. Assim, esses métodos não foram mais explorados no presente trabalho.

Uma abordagem baseada em Redes Neurais também foi tentada, no escopo deste trabalho, para a solução do problema inverso associado à construção de mapas de anisotropias da RCFM. Uma Rede Neural do tipo *backpropagation* foi implementada e diversas soluções foram obtidas para os mesmos dados simulados, dependendo dos dados utilizados para treinamento dos neurônios. Notou-se uma grande influência da série temporal de dados utilizada para treinamento na qualidade da solução, razão pela qual essa abordagem também foi abandonada.

As soluções obtidas pela abordagem de resolução implícita do problema inverso associado, proposta neste trabalho, utilizando AG e RS, foram comparadas com a solução obtida pelo método *bin average*. Este método é muito empregado em Cosmologia, sendo normalmente implementado na forma de *scripts* do aplicativo IDL. O método *bin average* executa a média aritmética das medidas de cada pixel a partir dos dados armazenados nas séries temporais de medidas. São efetuadas, para cada pixel as médias das medidas desse pixel para cada série temporal. O valor estimado da temperatura do pixel é então dado pela média aritmética dessas médias, ponderado pelas variâncias das medidas de cada série para o mesmo pixel.

A avaliação das soluções obtidas foi feita por meio da correlação e do desvio médio das soluções obtidas em relação à solução exata. Tipicamente, avalia-se a solução pela correlação. Se os valores desta forem iguais, avalia-se então pelo desvio médio. No caso do método *bin average*, a solução obtida apresentou correlação de 0,995 e desvio médio de 0,290.

O desempenho computacional também foi comparado, em relação ao tempo de execução do método *bin average*, mas tomando-se uma versão seqüencial deste desenvolvida neste trabalho, a qual foi escrita na linguagem C, uma vez que o uso da versão IDL implica em tempos de execução muito maiores.

Primeiramente foi implementado um AG canônico paralelo com população de 100 indivíduos, cada um correspondente a um conjunto possível de temperaturas dos pixels,

seleção tipo torneio, cruzamento com um único ponto de recombinação e taxa de mutação de 1%. Em seguida, foi implementada uma variação desse AG, com mesmo tamanho de população, seleção por distribuição de Boltzmann, e estratégias alteradas para cruzamento e mutação. Essa variação foi paralelizada segundo o modelo global, em que o cálculo da aptidão de cada solução candidata é dividido entre os processadores, e segundo o modelo ilha, no qual subpopulações são evoluídas independentemente em cada processador, e uma política de migração de indivíduos entre processadores é estabelecida.

Estes AGs foram implementados por meio da biblioteca de funções PGAPack, que possibilita a escolha de diferentes operadores para seleção, cruzamento e mutação, além de suporte para paralelização. As soluções obtidas pelos AGs demonstram que estes resolvem o problema da construção de mapas com uma qualidade similar (ligeiramente inferior) ao obtido pelo método *bin average*. A melhor solução foi obtida pelo AG com distribuição Boltzmann e paralelização pelo modelo global, com correlação de 0,986 e desvio médio de 0,300. Entretanto o tempo execução desses AGs, todos eles em versões paralelas, é muito superior ao obtido pela implementação sequencial do método *bin average*, da ordem de 10 vezes.

Face ao custo computacional elevado do AG, investigou-se o uso de um algoritmo de RS, também paralelo, com distribuições Boltzmann, Cauchy ou uniforme, todos com solução candidata inicial gerada aleatoriamente. Esses RSs levaram a soluções parecidas e ligeiramente melhores que o método *bin average*. Os tempos de execução foram similares e um pouco menores que os tempos dos AGs, porém ainda muito altos se comparados com o método *bin average*. Assim, optou-se por implementar e avaliar novas variações do algoritmo RS com distribuição Cauchy, que foi o que deu o melhor resultado entre as 3 distribuições avaliadas inicialmente: correlação de 0,999 e desvio médio de 0,250.

O algoritmo RS com distribuição Cauchy foi avaliado para dois outros esquemas de geração da solução candidata inicial: o uso de uma solução gerada pelo AG Boltzmann

modelo global após 700 iterações e o uso de uma solução obtida por amostragem de medidas armazenadas na série temporal de dados. Neste último caso, a qualidade da solução foi ainda um pouco melhor (correlação de 0.999 e desvio médio de 0,200) e o tempo de execução bem menor que os outros RSs ou AGs, sendo aproximadamente o dobro que aquele do método *bin average*. No caso da solução candidata inicial gerada por AG, o tempo de execução piorou sensivelmente, sem melhorar a solução.

Uma última modificação foi então implementada no RS com distribuição Cauchy com solução obtida por amostragem de medidas armazenadas na série temporal de dados. Foi utilizado o esquema de *re-annealing*, no qual introduz-se uma perturbação na solução corrente quando não ocorre uma melhora do valor da função objetivo após um certo número de iterações. Obteve-se uma solução com qualidade ainda melhor (correlação de 0.999 e desvio médio de 0,150) com um tempo de execução bem próximo ao do método *bin average* (seqüencial). Em relação ao módulo das diferenças de temperatura entre a temperatura exata e a estimada, o valor médio foi de 30 μK para o método *bin average* e se reduz para 10 μK nesse RS, avaliados com todos os 8.000 pixels.

O presente trabalho mostra que a abordagem proposta de resolução implícita do problema inverso associado com uso de algoritmo estocástico paralelo resolve o problema de construção de mapas de anisotropias da RCFM a um custo computacional compatível em relação à implementação seqüencial do método *bin average*, usualmente utilizado. Em particular, o uso do algoritmo de RS com distribuição de Cauchy e solução candidata inicial obtida por amostragem das medidas de temperatura e uso de *re-annealing* resultou numa solução melhor que aquela do método *bin average*, fornecendo assim um mapa com melhor resolução que permite a identificação de estruturas menores da região observada do céu.

O método *bin average*, implementado seqüencialmente na linguagem C e aqui empregado como referência em termos de qualidade da solução e desempenho, poderia

ainda ser objeto de uma otimização de código e de uma paralelização, uma vez que o cálculo da média ponderada de medidas é intrinsecamente paralelizável.

Como sugestão de trabalho futuro, pode-se preparar um pacote de software para construção dos mapas de anisotropias da RCFM baseado na implementação proposta neste trabalho, ou seja, o algoritmo RS com distribuição Cauchy, com solução obtida por amostragem de medidas armazenadas na série temporal de dados e com o esquema de *re-annealing*. Este pacote de software seria disponibilizado para distribuição à comunidade científica.

Entretanto, com o crescente número de experimentos para medidas das anisotropias da RCFM e com o aperfeiçoamento dos detectores utilizados pelos mesmos, o volume de dados armazenados nas séries temporais de medidas tende a crescer substancialmente, da mesma forma como o número de pixels de cada mapa. Isso leva à necessidade de recursos computacionais cada vez maiores para a construção dos mapas correspondentes.

Uma possibilidade para prover esses recursos computacionais é o uso de tecnologias de processamento distribuído, como por exemplo a computação em grade (*Grid Computing*) [80]. Trata-se de uma camada de software que permite o uso integrado de computadores de alto desempenho, bases de dados e instrumentos científicos pertencentes a diversas organizações ligadas em rede. Suas aplicações freqüentemente envolvem grandes quantidades de dados e/ou grandes necessidades de processamento. Existem, no Brasil, algumas grades computacionais já implementadas, inclusive com tecnologia nacional, como por exemplo o OurGrid, um projeto coordenado pela Universidade Federal da Paraíba e desenvolvido em cooperação com a empresa norte-americana Hewlett Packard (HP), que já conta com 20 instituições brasileiras de pesquisa compartilhando recursos computacionais [81]. Outro exemplo, o GridRio [82], formado por 94 máquinas, interliga 7 instituições de pesquisa e universidades do Rio de Janeiro. Há também a AccessGrid [83], interligando máquinas da Universidade Federal do Rio Grande do Sul (UFRGS) com Universidade da Flórida (EUA) por meio

da rede de alta velocidade (*internet2*). Uma grade computacional, implementada especificamente ou não para construção de mapas, poderia ser utilizada para esse fim, utilizando máquinas de universidades e institutos de pesquisa brasileiros.

Além de suprir as necessidades computacionais de alto desempenho, a computação em grade poderia também servir como suporte para um ambiente colaborativo [84], no qual usuários habilitados, situados em pontos geograficamente diferentes, poderiam remotamente ter acesso a dados, requisitar a construção de mapas ou a visualização dos mesmos. Ou seja, esses usuários teriam acesso a serviços de processamento, visualização, etc. implementados por meio de uma grade computacional [85].

Uma das necessidades de computação em grade é a interligação entre as máquinas que a compõe por meio de redes de comunicação rápidas e confiáveis. Nesse sentido, em particular, a Rede Nacional de Ensino e Pesquisa (RNP) está implantando uma rede *gigabit* (largura de banda de 10 Gbits/s) conectando instituições e universidades das cidades de São Paulo (USP), Campinas (Unicamp), São José dos Campos (INPE), Cachoeira Paulista (CPTEC/INPE), LNCC (Petrópolis) e Rio de Janeiro (UFRJ). Obviamente, já existem recursos computacionais de alto desempenho nessas instituições e universidades, as quais poderiam ser mais bem aproveitadas através de uma grade computacional, a qual poderia atender diversos serviços e áreas de pesquisa, notadamente a construção de mapas de anisotropias da RCFM.

Outra opção de trabalho futuro seria uma implementação baseada em *hardware* com o uso de conjuntos *Field-Programmable Gate Array* (FPGA), como alternativa à solução proposta em *software*, aqui apresentada. Um conjunto FPGA poderia, por exemplo, efetuar a avaliação de cada solução candidata gerada num tempo muito menor que o demandado por *software*.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 - Peacock, J. **Cosmology physics**. Cambridge: MIT Press, 1999. 600p.
- 2 - Peebles, P. J. E. **Principles of physical cosmology**. New York: PJ Editors, 1993. 736p.
- 3 - Penzias, A. A.; Wilson, R. W. A Measurement of excess antenna temperature at 4080 Mc/s. **The Astrophysical Journal**, v. 142, p.419-421, May 1965.
- 4 - Padmanabham, T. **Structure formation in the Universe**. New York: Cambridge, 1993. 499p.
- 5 - Wright, E.L. Scanning and mapping strategies for CMB experiments. **The Astrophysical Journal**, astro-ph/9612006, 1996. Disponível em: <http://xxx.lanl.gov>. Acesso em: 10 jan. 2003.
- 6 - Boesgaard, A. M.; Steigman, G. Big-Bang nucleosynthesis: theories and observations. **Annual Review of Astronomy and Astrophysics**, v.23, n. 1, p.319-378. 1985.
- 7 - Smoot et al. Structure in the COBE differential microwave radiometer first-year maps. **The Astrophysical Journal**, v. 396, n. 1, p. L1-L5. 1992.
- 8 - Smoot, G. F. COBE DMR results and implications. In: The infrared and submillimetre sky after COBE. March 20-30, 1991, Les Houches, France. **Proceedings...** France: NATO Advanced Study Institute, p. 331.
- 9 - Meinhold, P.R. et al. A map of the cosmic microwave background from the BEAST experiment. **The Astrophysical Journal**, astro-ph/0302034, 2003. Disponível em: <http://xxx.lanl.gov>. Acesso em: 11 maio 2003
- 10 - Tegmark, M. CMB mapping experiments: a designer's guide. **The Astrophysical Journal**, astro-ph/9705188, 1997. Disponível em <http://xxx.lanl.gov>. Acesso em: 01 set. 2002.
- 11 - Borrill, J. MADCAP – The microwave anisotropy dataset computational analysis package. **The Astrophysical Journal**, astro-ph/9911398, 1999. Disponível em: <http://xxx.lanl.gov>. Acesso em: 05 set. 2002.
- 12 - Research System Incorporated. The Interactive Data Language. Disponível em: <<http://www.rsinc.com/idl/>>. Acesso em: 05 set. 2002.
- 13 - Natoli, P. et al. Imaging the Cosmic Microwave Background: The BEAST experiment. In: AIP Conference Proceedings. 4-6 april 2001, Baltimore, EUA. **Proceedings...**Melville: American Institute of Physics, 2001. p.324. (ISBN 0-7354-0027-X).
- 14 - Bethea, R.M.; Duran, B.S.; Boullin, T.L. **Statistical methods for engineers and scientists**. New York: Marcel Dekker, 1995. 695p.
- 15 - Foster, I.; Kohr, D. R.; Krishnaiyer, R.; Choudhary, A. A library-based approach to task parallelism in a data-parallel language. **Journal of Parallel and Distributed Computing**, v 45, n.2, p.148-158, 1997.
- 16 - Mather, J. C. Measurement of the cosmic microwave background spectrum by the COBE FIRAS instrument. **The Astrophysical Journal**, v. 420, n. 2, p. 439-444, 1994.

- 17 - Tegmark, M.; Oliveira-Costa, A.; Staren, J.; Meinhold, P.; Lubin, P.; Childers, J.; Figueiredo, N.; Gaier, T.; Lim, M.; Seiffert, M.; Villela, T.; Wuesnche, C.A. Cosmic microwave background maps from the HACME experiment. **The Astrophysical Journal**, astro-ph/9711076, 2000. Disponível em: <<http://xxx.lanl.gov>>. Acesso em: 05 set. 2002.
- 18 - Silva, A. P. **Medidas de anisotropia da radiação cósmica de fundo em microondas em média escala angular nas frequências de 30 GHz e 41 GHz**. 2002. 160p. Tese (Doutorado em Astrofísica) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2002.
- 19 - Kuo, C.L. et al. High resolution observations of the CMB power spectrum with ACBAR. **The Astrophysical Journal**, v.600, p.32-51, 2004.
- 20 - Mason, B. S. The anisotropy of the microwave background to $l = 3500$: Deep Field Observations with the Cosmic Background Imager. **The Astrophysical Journal**, v. 591, p.540-555, May 2003.
- 21 - Bennett, C. L.; Hill, R. S.; Hinshaw, G.; Nolta, M. R.; Odegard, N.; Page, L.; Spergel, D. N.; Weiland, J. L.; Wright, E. L.; Halpern, M.; Jarosik, N.; Kogut, A.; Limon, M.; Meyer, S. S.; Tucker, G. S.; Wollack, E. First-year Wilkinson Microwave Anisotropy Probe (WMAP). Observations: Foreground emission. **The Astrophysical Journal Supplement Series**, v.148, n.1 , p. 97-117. 2003.
- 22 - Staren, J.; Meinhold, P.; Childers, J.; Lim, M.; Levy, A.; Lubin, P.; Seiffert, M.; Gaier, T.; Figueiredo, N.; Villela, T.; Wuensche, C.A.; Tegmark, M.; Oliveira-Costa, A. A spin modulated telescope to make two dimensional CMB maps. **The Astrophysical Journal**, astro-ph/9912212, 1999. Disponível em: <<http://xxx.lanl.gov>>. Acesso em: 05 set. 2002.
- 23 - Vuolo, J.H. **Fundamentos da teoria de erros**. São Paulo: Edgard Blücher, 1996. 236p.
- 24 - Tegmark, M. **How to make maps from CMB data without losing information**. **The Astrophysical Journal**, v. 480, p.87-90, 1997.
- 25 - Borrill, J. The challenge of data analysis for future CMB observations. **The Astrophysical Journal**, astro-ph/9903204, 1999. Disponível em: <<http://xxx.lanl.gov>>. Acesso em 03 out. 2002.
- 26 - Lee, A.T. et al. A High resolution analysis of the MAXIMA-1 cosmic microwave background anisotropy data. **The Astrophysical Journal**, v.561, p.1-6, 2001.
- 27 - Netterfield, C. B. et al. A measurement by BOOMERANG of multiple peaks in the angular power spectrum of the cosmic microwave background. **The Astrophysical Journal**, v. 571, p.604, 2002.
- 28 - Engl, H.W.; Hanke, M.; Neubauer, A. **Regularization of Inverse Problems**. Norwell: Kluwer Academic Publisher, 2000. 513 p.
- 29 - Campos Velho, H.F. Problemas inversos: conceitos básicos e aplicações. In: Encontro de Modelagem Computacional, 4., 2001, Nova Friburgo, RJ. **Anais...** Rio de Janeiro: IPRJ-UERJ, 2001. Mini-curso.
- 30 - Hadamard, J. **Lectures on Chauchy's problem in linear differential equations**. EUA: Dover Publications. 2003. 315 p.
- 31 - Tarantola, A. **Inverse problem theory**. Amsterdam: Elsevier Science Publishers, 1988. 613p.

- 32 - Silva Neto, A.J.; Moura Neto, F.D. Escolha de modelos: problemas inversos em engenharia. In: Congresso Nacional de Matemática Aplicada e Computacional, 2000, Santos, SP. **Anais...** São Paulo: SBMAC, 2000. Mini curso.
- 33 - Kon, A.H.G.R.; Timmer, G.T. Stochastic global optimization methods, Part I: clustering methods. **Mathematical Programming**, v.39, p.27-56, 1987.
- 34 - Ma, C.K.; Chan, C.K. Maximum descent method for image vector quantization. **Electronics Letters**, v.27, n.12, p.1772-1773, 1991.
- 35 - Van de Velde, E.F. **Concurrent scientific computing**. New York: Springer Verlag, 1994. 356p.
- 36 - Gill, P.R.; Murray, W.; Wright, M.H. **Practical optimization**. London: Academic Press. 1981. 420p.
- 37 - Tokhomirov, V.M. The evolution methods of convex optimization. **American Mathematics Monthly**, v. 103, p.76-71, 1996.
- 38 - Shiguemori, E.H. **Redes neurais e novos métodos de regularização em problemas inversos de condução de calor**. 2002. 93 p. Dissertação (Mestrado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2002.
- 39 - Hooke, R.; Jeeves, T. Direct search solutions of numerical and statistical problems. **Journal of the Association for Computing Machinery**, v.8, n.2, p.212-229, 1961.
- 40 - Goldberg, D.E. **Genetic algorithms in search, optimization and machine learning**. Reading: Addison-Wesley Publishers, 1989. 432 p.
- 41 - Beck, J. Arnold, K. **Parameter estimation in engineering and science**. New York: Wiley and Sons, 1977. 501 p.
- 42 - Sousa, F. L.; Vlassov, V.V.; Ramos, F.M. Generalized extremal optimization: an application in heat pipe design. **Applied Mathematical Modelling**, v. 28, p. 911-931, 2004.
- 43 - Sousa, F. L. ; Vlassov, V.v. ; Ramos, F. M. Generalized extremal optimization for solving complex optimal design problems. **Lecture Notes in Computer Science**, v. 2723, p. 375-376, 2003.
- 44 - Glover, F.; Laguna, M. **Tabu Search**. Boston: Kluwer Publishers. 1997. 408 p.
- 45 - Dorigo, M.; Gambardella, L.M. Ant colonies for the traveling salesman problem. **Biosystems**, v.43, p.73-81, 1997.
- 46 - Coley, D.A. **An introduction to genetic algorithms for scientists and engineers**. Singapore: World Scientific Publishing, 1999. 255 p.
- 47 - Holland, J.H. **Adaptation in natural and artificial system**. Cambridge: MIT Press, 1975. 221p.
- 48 - Cantú-Paz, E. A survey of parallel genetic algorithms. **Illinois Genetic Algorithms Laboratory**, report 97003, 1997. Disponível em <<http://neo.lcc.uma.es/cEA-web/documents>>. Acesso em: 16 mar. 2003.
- 49 - Yang, J.; Hanovar, V. Feature subset selection using genetic algorithm. **IEEE Intelligent Systems**, v. 1094, n. 7160, p.44-49, 1998.
- 50 - Mitchell, M. **An introduction to genetic algorithms**. Cambridge: MIT Press, 1996. 210 p.

- 51 - Chambers, L. **Practical handbook of genetic algorithms**. Boca Raton: CRC Press, 1991. 448 p.
- 52 - Syswerda G. Uniform crossover in genetic algorithms. In: International Conference on Genetic Algorithms, 3. 29-26 July 1989, San Mateo, EUA. **Proceedings...** San Mateo: ICGA/IEEE, 1989. p.2-29.
- 53 - Eschelmann L.J.; Schaffer J.D. Preventing premature convergence in genetic algorithms by preventing incest. In: International Conference on Genetic Algorithms, 4., 1991. **Proceedings...** [S.l.]: ICGA, 1991. p.115-122
- 54 - Craighurst R., Martin W. Enhancing GA Performance through crossover prohibitions based on ancestry. In: International Conference on Genetic Algorithms, 6., 15-19 July 1995, Pittsburgh, EUA. **Proceedings...** Pittsburgh: ICGA/IEEE, 1995. p. 130-135. (ISBN 1-55860-370-0).
- 55 - The Hamming distance. Disponível em: <http://www.nist.gov/dads/HTML/hammingdist.html>. Acesso em: 20 mar. 2003
- 56 - Nowostawski, M.; Poli, Ricardo. Parallel genetic algorithm taxonomy. In: Fuzzy Techniques in Image Processing Theory and Applications. 31 August – September 1 1999, Sidney, Australia. **Proceedings...** Sidney: KES/IEEE Computer Society. P.88-92.
- 57 - Tanese, R. **Distributed genetic algorithm for function optimization**. 1989. Tese (PhD Computer Sciences) – Michigan University, Michigan, 1989.
- 58 - Cantú-Paz, E. Designing Scalable multi-population parallel genetic algorithms.. **The University of Illinois**, Report 97004, 1998. Disponível em: <http://leitl.org/oldcontent/docs/>. Acesso em 18 fev. 2003.
- 59 - Parallel Genetic Algorithm Library. Disponível em: http://www-fp.mcs.anl.gov/CCST/research/reports_pre1998/comp_bio/stalk/pgapack.html. Acesso em: 12 fev. 2003
- 60 - Kirpatrick, S. Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. **Science**, v. 220, n. 4598, p.671-680, 1983.
- 61 - Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A .H.; Teller, E. Equation of state calculations by fast computing machines. **Journal of Chemical Physics**, v.21, p.1087-1092, 1953.
- 62 - Van Laarhoven, P.J.M.; Aarts, E.H.L. **Simulated annealing: theory and applications**. New York: Kluwer Academic Publishers, 1987. p.231.
- 63 - Szu, H.; Hartley, R. Fast simulated annealing. **Physical Letters A**, v.122, n.3/4, p.157-162, 1987.
- 64 - Atiya, A.F.; Parlos, A.G.; Ingber, L. A reinforcement learning method based on adaptive simulated annealing. Disponível em: <http://ideas.repec.org/p/lei/ingber/03rl.html#provider>. Acesso em 15 abr. 2003.
- 65 - Azencott, R. **Simulated annealing parallelization techniques**. New York: Wiley and Sons, 1992. 256 p.
- 66 - Hamma, B.; Viitanen, S.; Törn, A. Parallel continuous simulated annealing for global optimization. **Optimization Methods and Software**, v.13, n. 2. p. 93-116. 2000.

- 67 - Jones, A.E.W.; Forbes, G.W. An adaptive simulated annealing algorithm for global optimization over continuous variables. **Journal of Global Optimization**, v. 6, p. 157-181, 1993.
- 68 - Cheng, K.; Chen, B. Simulated annealing and genetic algorithms based methods for impedance image reconstruction. In: International Conference on Bioelectromagnetism, 2, 15-18 Feb. 1998, Melbourne, Australia. **Proceedings...** ICB/IEEE, 1998. p.60-64. (ISBN 07803-3867-7).
- 69 - Gropp, W.; Lusk, E.; Doss, N.; Skjellum, A. A high-performance, portable implementation of the MPI message passing interface standard. **Journal of Parallel Computing**, v.22, n.6, p.789-828, 1996.
- 70 - Hwang, K.; Briggs, F. A. **Computer architecture and parallel processing**. New York: McGraw-Hill Book, 1984. 846 p.
- 71 - Foster, I. **Designing and building parallel programs**. EUA: Addison-Wesley, 1995. p.381.
- 72 - Gropp, W.; Lusk E.; Skjellum, A. **Using MPI**. Boston: The MIT Press, 1994. 371p.
- 73 - Amdahl's Law. Disponível em: <http://home.wlu.edu/~whaley/classes/parallel/topics/amdahl.html>. Acesso em: 01 jul. 2002.
- 74 - Reschke, C. et al. **A design study of alternative network topologies for the beowulf parallel workstation**. Disponível em: <http://www.phy.duke.edu/resources/computing/brahma/Resources/beowulf/papers/HPDC96/hpdc96.html>. Acesso em: 18 set. 2003.
- 75 - Spector, D.H.M. **Building Linux Clusters**. Sebastopol: O'Reilly and Associates, 2000. 220 p.
- 76 - Bovet, D.P.; Cesati, M. **Understanding the Linux kernel**. Sebastopol: O'Reilly and Associates, 2001. 530 p.
- 77 - Górski, K.M; Hivon, E.; Wandelt, B.D. Analysis issues for large CMB data sets. In: MPA/ESO Cosmology Conference on evolution of large-scale structure. 2-7 August 1988, Garching, EUA. **Proceedings...** EUA: MPA/ESO, 1988. p. 37-42.
- 78 - Press et al. **Numerical recipes in C: the art of scientific computing**. 2.ed. EUA: Cambridge University Press, 1992. p.1020.
- 79 - Leonardi, R. **Produção de mapas de anisotropia da radiação cósmica de fundo em microondas utilizando um algoritmo genético**. 2002. 129p. Dissertação (Mestrado em Astrofísica) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2002.
- 80 - Foster, I.; Kesselman, C. **The GRID: blueprint for a new computing infrastructure**. San Francisco: Morgan Kaufmann Publishers, 1999. 677 p.
- 81 - OurGrid. Disponível em: <http://dsc.ufcg.edu.br/ourgrid/>. Acesso em: 20 jul. 2004
- 82 - GridRio. Disponível em: <http://easygrid.ic.uff.br/grid/GridRio.html>. Acesso em: 20 jul. 2004.
- 83 - Access Grid. Disponível em: <http://www.inf.ufgrs.br/procpar/hetnos/grid/scglobal01/>. Acesso em: 20 jul. 2004.

- 84 - Czajkowski, K.; Foster, I.; Karonis, N.; Kesselman, C.; Smith, W.; Tuecke, S. **Job scheduling strategies for parallel processing**. Berlin: Springer, 1954. p.62-82.
- 85 - Foster, I.; Kesselman, C. The globus project: a progress report. In: Heterogeneous Computing Workshop, 12., 22 Apr. 2003, Nice, France. **Proceedings...** Nice: IEEE Computer Society Press, 2003. p.4-18.