



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Ciências Aplicadas

MÉRCIA DE SOUSA ROCHA

APLICABILIDADE DE META-HEURÍSTICAS AO PROBLEMA DE  
SEPARAÇÃO DE SINAIS NO CASO DE MODELOS LINEARES

APPLICABILITY OF META-HEURISTICS TO THE PROBLEM OF  
SIGNAL SEPARATION IN THE CASE OF LINEAR MODELS

LIMEIRA  
2016

MÉRCIA DE SOUSA ROCHA

APLICABILIDADE DE META-HEURÍSTICAS AO PROBLEMA DE  
SEPARAÇÃO DE SINAIS NO CASO DE MODELOS LINEARES

*Dissertação apresentada à Faculdade de Ciências Aplicadas da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestra em ENGENHARIA DE PRODUÇÃO E DE MANUFATURA, na Área de PESQUISA OPERACIONAL E GESTÃO DE PROCESSOS.*

*Orientador:* LEONARDO TOMAZELI DUARTE

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELA ALUNA MÉRCIA DE SOUSA ROCHA, E ORIENTADA PELO PROF. DR. LEONARDO TOMAZELI DUARTE.

LIMEIRA  
2016

Agência(s) de fomento e nº(s) de processo(s): CAPES, 01P-04386-2015

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Faculdade de Ciências Aplicadas  
Renata Eleuterio da Silva - CRB 8/9281

R582a Rocha, Mércia de Sousa, 1988-  
Aplicabilidade de meta-heurísticas ao problema de separação de sinais no caso de modelos lineares / Mércia de Sousa Rocha. – Limeira, SP : [s.n.], 2016.

Orientador: Leonardo Tomazeli Duarte.  
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Ciências Aplicadas.

1. Heurística. 2. Separação cega de fontes. 3. Evolução diferencial. 4. Otimização por enxame de partículas. 5. Otimização por colônia de formigas. 6. Pesquisa operacional. I. Duarte, Leonardo Tomazeli, 1982-. II. Universidade Estadual de Campinas. Faculdade de Ciências Aplicadas. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Applicability of meta-heuristics to the problem of signal separation in the case of linear models

**Palavras-chave em inglês:**

Heuristic

Blind source separation

Differential evolution

Particle swarm optimization (PSO)

Ant colony optimization

Operational research

**Área de concentração:** Pesquisa Operacional e Gestão de Processos

**Titulação:** Mestra em Engenharia de Produção e de Manufatura

**Banca examinadora:**

Leonardo Tomazeli Duarte [Orientador]

Cristiano Torezzan

Romis Ribeiro de Faissol Attux

**Data de defesa:** 11-04-2016

**Programa de Pós-Graduação:** Engenharia de Produção e de Manufatura

**Autora:** Mércia de Sousa Rocha

**Título:** Aplicabilidade de Meta-Heurísticas ao Problema de Separação de Sinais no caso de Modelos Lineares

**Natureza:** Dissertação de Mestrado

**Instituição:** Faculdade de Ciências Aplicadas/ UNICAMP

**Data da Defesa:** Limeira, 11/04/2016

## **BANCA EXAMINADORA**

Prof. Dr. Leonardo Tomazeli Duarte (Orientador)

---

Assinatura

Prof. Dr. Romis Ribeiro de Faissol Attux

---

Assinatura

Prof. Dr. Cristiano Torezzan

---

Assinatura

A ata da defesa com as respectivas assinaturas dos membros encontra-se no processo de vida acadêmica da aluna.

Para Christophe e Olivier

## RESUMO

A separação cega de fontes é um problema da área de processamento de sinais que surge em diversas aplicações, desde o tratamento de sinais biológicos à geofísica. Usualmente, os métodos de separação são formulados como problemas de otimização, que são abordados por meio de ferramentas clássicas da pesquisa operacional, como o método do gradiente descendente. No entanto, diante da complexidade das funções que são comumente utilizadas como critérios de separação, os métodos mais usuais são atraídos por pontos de mínimos locais e podem não encontrar boas soluções a nível global. Tal limitação abre espaço para a aplicação de meta-heurísticas ao problema, por possibilitar uma ampliação do espaço de busca, em termos de mínimos locais e, conseqüentemente, encontrar soluções mais eficazes. Diante disso, o presente trabalho se dedica à análise de aplicabilidade de algumas meta-heurísticas ao problema de separação cega de fontes no caso de modelos lineares. Uma comparação entre um conjunto representativo de meta-heurísticas, sendo elas, *particle swarm*, evolução diferencial, *simulated annealing* e otimização por colônia de formigas para domínios contínuos, indica que, de modo geral, tais soluções são consideravelmente eficazes quando aplicadas ao caso e comparadas ao método do gradiente. Os principais pontos que podem diferenciá-las são a dificuldade de implementação e de ajuste de seus parâmetros.

## ABSTRACT

Blind source separation is an important problem in the field of signal processing and arises in many applications, from biological signal analysis to geophysics. Usually, separation methods are formulated as optimization problems that are solved by classic tools of operations research, such as the gradient descent method. However, considering the complexity of the functions which are commonly used as separation criteria, the usual methods can be attracted to local minima and may not find global solutions. This limitation opens the way for the application of the meta-heuristics due to their capacity to amplify the search space locally and, consequently, provide best solutions to the problems. Thus, the present thesis is dedicated to the analysis of the applicability of some meta-heuristics to deal with the blind source separation problem in the case of linear models. A comparison between a representative set of meta-heuristics, which comprises particle swarm optimization, differential evolution, simulated annealing and ant colony optimization for continuous domains, points out that, in general, such solutions are considerably effective when applied to source separation and compared to the gradient method. The main differences found between them concern the difficulty of implementation and the stage of parameter setting.

## SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	Objetivos.....	11
1.2	Organização do trabalho.....	11
<b>2.</b>	<b>SEPARAÇÃO CEGA DE FONTES</b>	<b>13</b>
2.1	Processamento de Sinais.....	13
2.2	Separação de Sinais.....	15
2.3	Separação cega de fontes.....	17
2.4	Análise de Componentes Independentes.....	20
2.5	Estimação por máxima verossimilhança.....	23
2.6	Critério Infomax.....	24
2.7	Método do Gradiente Ascendente ou Descendente.....	26
2.7.1	Presença de mínimos locais.....	28
2.7.1.1.	Branqueamento.....	28
<b>3.</b>	<b>META-HEURÍSTICAS</b>	<b>31</b>
3.1.	Evolução diferencial (DE).....	34
3.1.1	Geração da solução inicial e definição de parâmetros.....	36
3.1.2	Mutação.....	36
3.1.3	<i>Crossover</i> .....	37
3.1.4	Seleção.....	38
3.1.5	Pseudo-código.....	40
3.2	Otimização por Colônia de Formigas para domínios contínuos (ACO <sub>R</sub> ).....	41
3.2.1	Geração da solução inicial e definição de parâmetros.....	43
3.2.2	Cálculo dos pesos de cada função.....	44
3.2.3	Geração de novas soluções viáveis.....	45
3.2.4	Atualização do feromônio.....	46
3.2.5	Pseudo-código.....	47
3.3	Otimização por <i>Simulated Annealing</i> (SA).....	49
3.3.1	Inicialização da população.....	50

3.3.2	Cálculo da nova solução.....	50
3.3.3	Análise da nova solução.....	51
3.3.4	Atualização da temperatura.....	52
3.3.5	Pseudo-código.....	53
3.4	Otimização por <i>Particle Swarm</i> (PSO).....	54
3.4.1	Geração da população de soluções iniciais.....	56
3.4.2	Atualização da posição e da velocidade.....	57
3.4.3	Análise da nova solução.....	58
3.4.4	Pseudo-código .....	58
<b>4.</b>	<b>RESULTADOS</b>	<b>60</b>
4.1.	Problema supervisionado .....	62
4.1.1.	<i>Simulated Annealing</i> .....	62
4.1.2.	Evolução Diferencial.....	63
4.1.3.	Otimização por Colônia de Formigas para domínios contínuos (ACO <sub>R</sub> ).....	64
4.1.4.	Otimização por <i>Particle Swarm</i> .....	66
4.2.	Problema não supervisionado.....	67
4.2.1.	Método Gradiente.....	69
4.2.2.	Evolução Diferencial.....	70
4.2.3.	<i>Simulated Annealing</i> .....	71
4.2.4.	Otimização por Colônia de Formigas para domínios contínuos (ACO <sub>R</sub> ).....	72
4.2.5.	Otimização por <i>Particle Swarm</i> .....	73
4.2.6.	Resultado da SIR .....	75
<b>5.</b>	<b>CONCLUSÕES E PERSPECTIVAS</b>	<b>76</b>
5.1	Considerações gerais.....	76
5.2	Perspectivas.....	78
	REFERÊNCIAS.....	80

## **CAPÍTULO 1**

### **INTRODUÇÃO**

A Pesquisa Operacional tem evoluído consideravelmente e ganhado cada vez mais visibilidade no meio científico por ser uma área do conhecimento cujas contribuições se estendem por diversos domínios, desde a engenharia até a medicina, possibilitando importantes avanços em todos eles. Entre as ferramentas de otimização de maior impacto estão as meta-heurísticas, assunto central de nosso trabalho, que se fazem extremamente importantes na resolução de problemas não lineares ou de extrema complexidade, onde métodos clássicos costumam falhar.

No entanto, ainda que muitos avanços tenham sido alcançados nessa área, diversos aspectos podem ainda ser explorados como meio de aprimorar tais ferramentas, ou mesmo buscar novos campos de aplicação, como o que propomos em nossos estudos. Nossa motivação vem do fato de que os problemas de otimização em processamento de sinais ainda são pouco explorados no âmbito da otimização por meta-heurísticas. Como exemplo, podemos citar o trabalho de Attux [1], na área de equalização e identificação de canais.

A área de processamento de sinais encontra aplicações em diversos ramos do conhecimento, da genética à geofísica. Dentre os problemas de interesse em processamento encontram-se: compressão de dados, filtragem, e interpretação da informação. Muitas das tarefas abordadas nessa área podem ser formuladas como problemas de otimização. Por exemplo, em filtragem adaptativa e em separação cega de fontes, o processo de separação culmina num processo de otimização de uma função custo que é definida segundo o problema a ser resolvido e a estratégia de separação utilizada.

Classicamente, a teoria de processamento de sinais se apoia na conjunção de critérios baseados em funções quadráticas/sistemas lineares. Tal conjunção, além de se adequar a diversas situações de interesse prático, resulta em funções objetivo convexas, o que, por sua vez, facilita a tarefa de otimização. Mais recentemente, no entanto, com o advento de novos paradigmas de adaptação e do uso de sistemas e/ou critérios não lineares, é cada vez mais frequente o surgimento de problemas de otimização desafiadores, cujas funções custo são de caráter extremamente não linear e apresentam descontinuidades e ótimos espúrios. Tais

características dificultam a aplicação de métodos de programação matemática baseados no uso do gradiente e/ou da hessiana. Além disso, dado que as aplicações geralmente estão associadas a dados de grande dimensão, o uso de métodos exaustivos torna-se inviável.

Alternativamente, surge como uma boa opção a utilização de métodos de otimização como as meta-heurísticas. Dentre as vantagens na aplicação de meta-heurísticas em processamento de sinais, podemos citar a robustez a ótimos locais e a capacidade de se operar sem a necessidade de derivadas e de evitar problemas ligados à instabilidade do sistema. Além disso, com o aumento da capacidade de processamento e memória dos computadores, o uso de meta-heurísticas é viável mesmo em problemas de grande dimensão.

## **1.1 Objetivos**

Diante das perspectivas interessantes do uso de meta-heurísticas em processamento de sinais, o presente projeto de mestrado tem como objetivo conduzir um amplo estudo dessas técnicas em um problema relevante da área de processamento de sinais: a separação cega de fontes. Procuramos, assim, realizar um comparativo de diferentes abordagens de otimização por meta-heurísticas presentes na literatura com o intuito de buscar algoritmos que forneçam um desempenho satisfatório na resolução do problema.

## **1.2 Organização do trabalho**

O presente trabalho está organizado da seguinte forma:

- Capítulo 2: nesse capítulo, é feita uma breve introdução à área de processamento de sinais, e, especificamente, ao problema de separação cega de fontes. Ênfase é dada na análise de componentes independentes, uma das principais ferramentas consideradas em separação, e na otimização dos critérios resultantes a partir de métodos de otimização baseados no gradiente. Ainda nesse capítulo,

são discutidas algumas situações encontradas em separação cega de fontes onde há indícios da presença de mínimos locais.

- Capítulo 3: As meta-heurísticas investigadas no presente trabalho são discutidas neste capítulo, que, inicialmente, apresenta uma breve introdução sobre pesquisa operacional. O objetivo é apresentar os principais operadores utilizados nessas meta-heurísticas, além de mostrar como se dá o processo de otimização nessas soluções e quais são critérios mais importantes a serem considerados no processo de implementação de cada técnica de otimização. O pseudo-código de cada método é apresentado como um complemento às considerações feitas.
- Capítulo 4: nesse capítulo, são apresentados os resultados obtidos com a aplicação de cada meta-heurística em diferentes casos de separação de sinais, com destaque para a situação dita não supervisionada.
- Capítulo 5: finalmente, apresentamos nesse capítulo as nossas observações e conclusões embasadas, principalmente, nos resultados obtidos no capítulo anterior. Algumas propostas para dar continuidade a esse trabalho ou mesmo para alavancar trabalhos futuros também são discutidas nesse capítulo.

## CAPÍTULO 2

### SEPARAÇÃO CEGA DE FONTES

Para compreender o problema de separação cega de fontes (BSS – *Blind Source Separation*), é interessante conduzir uma discussão mais geral sobre o processamento de sinais, grande área da qual a BSS faz parte. Por esse motivo, um dos objetivos iniciais do presente capítulo é prover uma breve explanação sobre sinais. Em seguida, abordaremos os principais aspectos do problema de separação de sinais, para só então dissertarmos sobre a separação cega de fontes e algumas das abordagens capazes de resolver tal problema.

#### 2.1 Processamento de Sinais

Assim como são inúmeras as formas de sinais existentes no mundo, há também diferentes formas de utilizá-los. Devido a essa considerável amplitude, há quem defenda, como em [2], que há muitas definições possíveis quando se trata de sinais, principalmente levando em conta as conotações empíricas. Assim, considerando a área de concentração deste estudo, em processamento de sinais, um sinal pode ser deliberadamente definido como uma parcela de informação que pode ser expressa como uma função do tempo ou de outras variáveis, como frequência e distância [3,4].

Isso abre caminho para a formação de um amplo conjunto de exemplos, como [3]:

- Sinais cerebrais, cardíacos e visuais;
- Música, fala ou outros sons naturais, como na Figura 2.1;
- Eletricidade e sinais magnéticos;
- Lucro em função do tempo e de variáveis de mercado;
- Taxa de desemprego em função de variáveis econômicas, sociais, e políticas;
- Densidade demográfica.

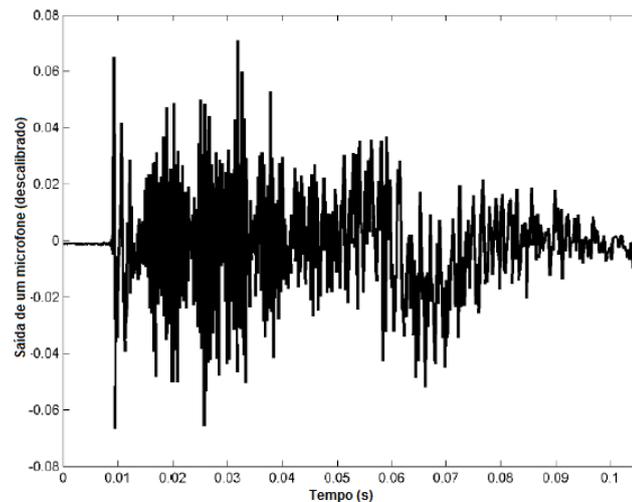


Figura 2.1: Exemplo de sinal (registro de um microfone) (adaptado de [5]).

Um sinal pode ser classificado de diversas formas. A maioria dos sinais físicos é encontrada na sua forma analógica e esses sinais são usados, por exemplo, para audição, visualização ou transmissão de informações. Porém, o processamento é, via de regra, realizado em sistemas de computação e é baseado em modelos discretos de sinais, ou seja, torna-se necessário um estágio de conversão elétrica conhecido como conversão analógico-digital (A/D). Ainda assim, mesmo após a conversão A/D, muitos dados não poderão ser imediatamente utilizáveis, devido à presença de ruídos como as flutuações aleatórias provindas dos equipamentos de medição. Casos como esses afetam a precisão dos dados e reforçam a necessidade do desenvolvimento de métodos de processamento de sinais [3,4,6].

Levando em conta as definições apresentadas, podemos dizer, de forma sucinta, que quando se fala em extração, processamento ou modificação de informações provenientes de sinais, com o objetivo de atender às nossas necessidades, fala-se em processamento de sinais [3]. Dentre os problemas de interesse em processamento, encontram-se: compressão de dados, filtragem, e interpretação da informação [14]. Sua área de aplicação abrange domínios como comunicações, medicina, economia, bioengenharia e muitas outras [7]. À medida que novas tecnologias foram surgindo e os computadores foram evoluindo, principalmente depois dos anos 1980, o campo de atuação do processamento de sinais cresceu, juntamente com sua importância [8].

Desde então, seu impacto alcança não somente áreas tradicionais como

a engenharia elétrica, mas diversos outros domínios. A abrangência da aplicação de processamento de sinais, como mostraremos a seguir, pode ser facilmente percebida quando observamos determinada disciplina e seu extenso conjunto de subgrupos, de onde é possível extrair diferentes tipos de informações que serão utilizadas para fins diversos, como [4,9]:

- Processamento de imagem: compressão, melhoria e reconhecimento de imagem e de padrões, mapa satélite meteorológico, animação etc;
- Processamento da fala: codificação e reconhecimento da fala, síntese de voz, transcrição de áudio para texto etc;
- Telecomunicação: *modems* de dados (recepção e decodificação de sinais), comunicações móveis, anulação ou redução de eco, etc;
- Industrial: controlador digital de processo, anulação de ruído, desenho assistido por computador (CAD) e ferramentas de projeto etc;
- Biomédico: eletrocardiograma, eletroencefalograma, tomografia computadorizada, imagem de ressonância magnética, radiologia, raios-X digitais, aparelhos auditivos digitais etc;

Exemplos concretos de aplicação de processamento de sinais podem ser encontrados em telecomunicações [9], engenharia, medicina e economia [10], imagem, fala e radar [11], genética [12] e geofísica [13], entre outros.

## 2.2 Separação de Sinais

Dentre os problemas tratados na área de processamento, merece destaque o problema de filtragem, que, de certo modo, busca estimar um sinal, ou um conjunto de sinais, a partir de observações correlacionadas aos sinais de interesse. Dentre o amplo universo de problemas de filtragem, um dos tópicos recorrentes na área é o problema de separação de sinais.

Com o intuito de introduzir o problema de separação de sinais, consideremos o problema conhecido como “*cocktail party problem*”. Nesse problema, encontram-se algumas pessoas conversando ao mesmo tempo. Além da fala de cada pessoa, há ainda a presença de ruídos, como música de fundo,

interferências externas, objetos sendo manipulados ou outros. Durante esse tempo toda a festa é gravada com o auxílio de diversos microfones espalhados pelo ambiente, conforme ilustrado na Figura 2.2.

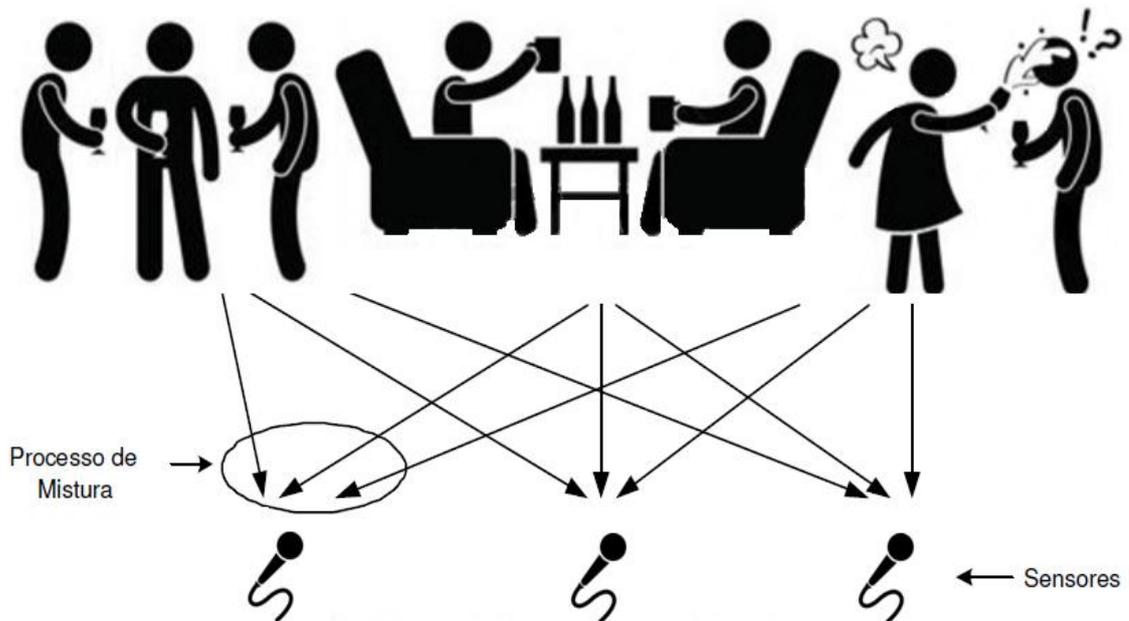


Figura 2.2: O problema "Cocktail Party".

Diante da natureza desse problema e da capacidade de processamento do cérebro humano, abrimos um questionamento a respeito da possibilidade de um sistema de processamento artificial alimentado pelas gravações feitas pelos microfones distinguir uma voz de uma pessoa qualquer assim como o faz o nosso cérebro [16]. Tal formulação é justamente aquela encontrada no problema de separação de sinais, cujo objetivo é recuperar o conjunto de sinais originais, processando as saídas dos sensores por meio da aplicação de um sistema de separação apropriado [17].

Apesar de parecer uma tarefa simples para o ser humano, o problema de separação de sinais, quando abordado por sistemas de computação, se apresenta desafiador. Em linhas gerais, a principal dificuldade é que o processo de separação geralmente é feito por um conjunto restrito de informações. Por outro lado, o fato de não se utilizar muitas informações faz com o que a separação de sinais encontre diversas aplicações. De fato, o processo de recuperação de sinais de interesse (fontes) pode ser aplicado em qualquer problema cujo sinal seja o resultado de uma combinação de sinais elementares [15].

### 2.3 Separação cega de fontes

Certos tipos de problema, altamente incidentes no processamento de sinais, não tinham solução até o surgimento da separação cega. Trata-se de determinados casos que são marcados pela falta de informações a respeito do processo de mistura e das características dos sinais fontes, o que explica a designação “cega” dada ao problema. A primeira abordagem em separação cega de fontes foi feita por Héroult, Jutten e Ans [18], em um contexto de modelagem de redes neurais artificiais, cuja motivação estava ligada à capacidade cerebral de processar diferentes sinais enviados pelo corpo para a ativação muscular [16]. Desde então, muitas contribuições foram feitas para que o sistema evoluísse e hoje a BSS ocupa papel central na teoria de processamento de sinais.

Para ilustrar o problema de separação cega de fontes desde a geração de dados até a saída das estimativas, vamos considerar o esquema representado na Figura 2.3. Se levarmos em conta o “Cocktail Party”, apresentado na seção anterior, é fácil entender a modelagem descrita nesse esquema. De fato, em tal contexto, a fala de cada pessoa é uma fonte, os microfones são sensores que captam todos os sinais ao mesmo e os dados obtidos nas gravações são as saídas do sistema misturador.

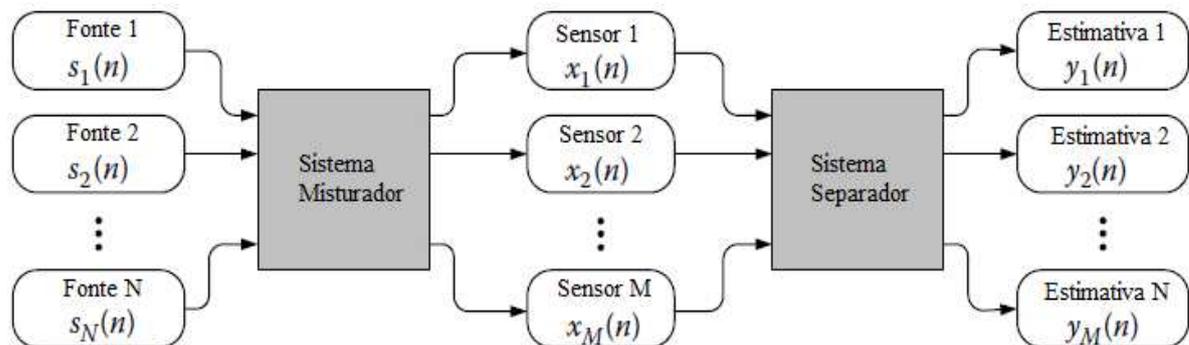


Figura 2.3: O problema de Separação Cega de Fontes (adaptado de [17]).

Em termos matemáticos, uma dada mistura  $X(n)$  pode ser modelada da seguinte maneira:

$$X(n) = [F(S(n), \dots, S(n-L), n(n), n)], \quad (2.1)$$

onde  $F(\cdot)$  é o mapeamento que descreve o processo de mistura,  $L$  corresponde à memória do sistema quando outras amostras são consideradas,  $n(n)$  corresponde ao ruído presente nos dados, e  $n$  denota o índice associado a uma dada amostra. Dadas essas variáveis, fica claro que os sistemas misturadores podem ser bastante complexos. Eles são classificados de diferentes formas:

- Linear: quando o mapeamento  $F(\cdot)$  atende ao princípio de superposição, como mostra a Equação 2.2, para quaisquer constantes  $b_1$  e  $b_2$ , e quaisquer sinais  $S_1$  e  $S_2$ :

$$F(b_1S_1(n) + b_2S_2(n)) = b_1F(S_1(n)) + b_2F(S_2(n)). \quad (2.2)$$

- Não linear: quando o mapeamento  $F(\cdot)$  não atende ao princípio de superposição apresentado na Equação 2.2.
- Instantâneo: quando o processo de mistura não está vinculado às amostras passadas, ou seja,  $L = 0$ .
- Convolutivo: quando o processo de mistura está vinculado às amostras passadas. Nesse caso temos  $L > 0$ .
- Sobredeterminado: quando o número de sensores é maior que o número de fontes, ou seja,  $M > N$ .
- Subdeterminado: quando o número de sensores é menor que número de fontes. Nesse caso, temos  $M < N$ .

É importante lembrar que, dado que ainda não há um sistema separador que seja capaz de considerar todas essas variáveis em seu processo de separação, faz-se necessário um conhecimento, ainda que mínimo, sobre a estrutura do sistema misturador. Sendo assim, a maioria das abordagens é feita considerando o número de misturas igual ao número de fontes, e sistemas de misturas lineares e instantâneos [16].

Além da separação de sinais de áudio, podemos encontrar na literatura uma ampla gama de outras aplicações de grande importância em separação de sinais ou em problemas de caráter não supervisionado, em geral. São casos como estudos em separação de imagens, em exploração geofísica, ou mesmo casos

especiais em separação de sinais de áudio, como a análise dos sons emitidos por uma máquina a fim de verificar seu estado de funcionamento [19]. Outros exemplos de aplicação incluem:

- Sinais biomédicos como eletrocardiograma (ECG) [20, 21, 22] e eletroencefalograma (EEG) [22, 23]: um mesmo sensor capta vários sinais gerados pelo corpo ao mesmo tempo. Como não há a possibilidade de filtrar apenas os sinais de interesse, é muito comum que esses exames sofram interferência de outros sinais, daí a importância da BSS nesse campo.
- Telecomunicações [23, 24, 25]: um elemento presente entre o transmissor e o receptor, chamado de canal, pode causar distorções significativas no sinal que está sendo transmitido. Nesse caso, o processo de equalização de canais se torna um problema de separação de sinais, onde a BSS pode ser aplicada no intuito de reduzir, o máximo possível, tais distorções.
- Aplicações financeiras [26, 27]: a aplicação da BSS nesse caso é muito importante quando o objetivo concentra-se na análise de resultados como oscilações de taxas de câmbio e preço e de ações. Esses tipos de dados podem carregar informações consideravelmente relevantes em relação ao contexto financeiro/econômico em análise.

Existem diferentes estratégias que são utilizadas em BSS. Entre elas temos a análise de componentes principais (PCA) [23], a análise de componentes esparsos (SCA) [28], ou ainda a abordagem bayesiana [29]. Contudo, a abordagem mais popular em separação cega de fontes é a chamada análise de componentes independentes (ICA – *Independent Component Analysis*). A ICA é uma técnica que se baseia na independência estatística dos dados e que, a princípio, era destinada a resolver problemas de separação cega de fontes. Com o tempo, sua abrangência aumentou e, atualmente, a ICA pode ser usada para resolver problemas de diversos outros segmentos da análise de dados.

## 2.4 Análise de Componentes Independentes

Assim como na separação cega de fontes, o primeiro estudo em análise de componentes independentes foi promovido por Héroult, Jutten e Ans [18]. Porém, um pouco mais tarde, recebeu grandes contribuições dos estudos realizados por Comon [32]. A ICA pode ser interpretada como uma extensão da PCA. Ao levar em conta estatísticas que vão além das medidas clássicas de média e de correlação, a ICA se mostra uma alternativa interessante à PCA, sobretudo em situações nas quais os dados não apresentam distribuição gaussiana. O principal objetivo da criação da ICA concentra-se na extração de informações úteis de determinados dados por meio da recuperação da independência estatística dos mesmos [32].

A ICA se aplica tanto a problemas lineares quanto não lineares. Contudo, sua aplicação exige o conhecimento da estrutura do sistema misturador, e supõe que as fontes são estatisticamente independentes, hipótese essa considerada bastante razoável em muitos casos [16]. Nesse sentido, é fundamental destacar que o presente estudo se direciona somente aos casos de misturas lineares, instantâneas e com número de fontes igual ao número de misturas, como mostraremos a seguir.

Antes de explicar como funciona a ICA, e de apresentar seu embasamento teórico, considere o conjunto de sinais fontes que, no contexto da ICA, são modelados pelo vetor de variáveis aleatórias:

$$S = [S_1, S_2, \dots, S_N], \quad (2.3)$$

onde  $N$  representa a quantidade de fontes existentes no problema.

Assim, considerando que cada sinal fonte estará presente em cada mistura com uma determinada proporção, uma mistura pode ser representada da seguinte maneira:

$$X_i = [a_{i1}S_1 + a_{i2}S_2 + \dots + a_{iN}S_N], \quad \text{para } i = 1, \dots, N. \quad (2.4)$$

onde  $a_{ij}$  é um coeficiente que representa o peso de cada fonte na mistura. Note que, como consideramos que o número de fontes é igual ao número de misturas, temos

$M = N$  nesse caso. Nessa situação, o processo de mistura linear pode ser representado pelo seguinte modelo matricial:

$$X = AS, \quad (2.5)$$

onde  $A$  é uma matriz de dimensão  $N \times N$ , cujos elementos são dados por  $a_{ij}$ . Obedecendo às condições estabelecidas inicialmente,  $A$  é a matriz que representa o sistema misturador.

Caso a matriz  $A$  fosse conhecida e admitisse inversa, bastaria inverter essa matriz que o problema já estaria resolvido. No entanto, conforme mencionado anteriormente, não há informação sobre a matriz de mistura em separação cega de fontes. Deste modo, torna-se necessário ajustar um sistema separador, representado pela matriz  $W$ , de modo que as estimativas das fontes ( $Y$ ) apresentem alguma propriedade de interesse, e sejam fornecidas da seguinte maneira:

$$Y = WX. \quad (2.6)$$

No caso da ICA, estamos interessados na obtenção dos dados  $Y$  que são os mais independentes entre si, pois, diante da hipótese de independência das fontes, espera-se que tal procedimento leve a estimativas que sejam as mais próximas possíveis das fontes originais.

Como mencionado anteriormente, a ICA é uma ferramenta de base estatística. Ela sugere que a recuperação das fontes pode ser alcançada pela recuperação da sua independência estatística, baseada na suposição de que diferentes processos físicos geram dados independentes entre si [33]. Isso significa dizer que a seguinte condição deve ser atendida:

$$p(Y_1, Y_2, \dots, Y_N) = p_1(Y_1)p_2(Y_2) \dots p_N(Y_N), \quad (2.7)$$

onde  $p(Y_1, Y_2, \dots, Y_N)$  representa a função de densidade de probabilidade conjunta das estimativas das fontes  $Y$  e  $p_1(Y_1)p_2(Y_2) \dots p_N(Y_N)$  representa a função de densidade de probabilidade marginal das estimativas das fontes  $Y_i$ .

De forma resumida, a independência estatística dos dados pode ser

entendida como a ausência de informação compartilhada entre fontes de origens diferentes [23]. Contudo, é essencial salientar que não é possível encontrar situações rotineiras que carreguem uma estrutura de dados puramente independentes. Todavia, é possível contornar esse problema, procurando por componentes que minimizem algum critério ligado à independência estatística [23].

Outro ponto importante associado à ICA, que torna possível solucionar o caso, leva a resolução do problema ao campo da pesquisa operacional. Para concluir a análise de componentes independentes com sucesso é preciso otimizar uma função, também conhecida em BSS como função contraste, que, de certo modo, expresse uma medida da independência estatística dos dados. Tal processo pode ser representado pelo seguinte problema de otimização:

$$\min_W \{J(Y(n)) = J(WX(n))\}, \quad (2.8)$$

onde  $J$  representa a função contraste. Note que essa equação refere-se a um caso de minimização apenas de maneira ilustrativa, pois poderia facilmente ser substituída por um problema de maximização dependendo da situação a ser abordada. Há diversas estratégias que podem ser utilizadas para gerar a função contraste. Nosso estudo aborda duas delas, a Máxima Verossimilhança e o Infomax. Apenas para citar alguns exemplos de outras estratégias, temos a Minimização da Informação Mútua [32] e Métodos Tensoriais [34].

Outra importante característica da ICA, que deve ser considerada, está relacionada às suas ambiguidades. Como não há um conhecimento prévio das fontes e da matriz de misturas, uma solução dada por essa técnica pode resultar em um conjunto de valores  $Y$  cuja ordem e escala são diferentes daquelas das fontes. Essas restrições são conhecidas como ambiguidades de escala e de permutação e, na maioria dos casos, são insignificantes [23]. Assim, é considerada solução do problema de BSS via ICA todas as matrizes  $W$  que fornecem as estimativas:

$$Y = PDS, \quad (2.9)$$

onde  $P$  é uma matriz de permutação e  $D$  é uma matriz diagonal.

Vale lembrar que diversos outros problemas podem ser analisados por

meio da ICA, como compressão de dados [30], predição de séries temporais [23], estágio de pré-processamento em sistemas classificadores [31], entre outros. Além disso, como nosso trabalho não aborda o conteúdo estatístico de forma mais aprofundada, recomendamos aos interessados nesse domínio a leitura dos trabalhos de Hyvärinen, Karhunen e Oja [23] e de Comon [32].

## 2.5 Estimação por máxima verossimilhança

A estimação por máxima verossimilhança é uma ferramenta teórica cuja proposta original destina-se à estimação de parâmetros de modelos probabilísticos, com base em amostras geradas pelo modelo probabilístico em questão [23]. Sua aplicação em análise de componentes independentes culmina na estimação de uma matriz de separação que gere sinais cuja função densidade de probabilidade (pdf – *probability density function*) se aproxime o máximo possível da distribuição das fontes verdadeiras [33]. Devemos responder a seguinte questão: “qual seria a matriz  $A$  com a maior probabilidade de ter gerado as misturas em análise?” [33]. Para resolver esse problema, consideremos a seguinte expressão:

$$\tilde{\theta} = \arg \max_{\theta} L(\theta) = \arg \max_{\theta} p_e(e|\theta), \quad (2.10)$$

onde,  $L(\tilde{\theta})$  é a função de máxima verossimilhança de  $\theta$ ,  $\theta$  é um conjunto de parâmetros  $\theta = [\theta_1, \dots, \theta_p]$  a ser estimado,  $e = [e(1), \dots, e(J)]$  é um conjunto de amostras,  $p_e$  é a pdf de  $e$ , e  $\tilde{\theta}$  é o conjunto dos parâmetros estimados pelo método. Isso implica que o objetivo dessa abordagem é buscar parâmetros que maximizem a probabilidade condicional de  $e$  dado  $\theta$ .

Considerando que os dados amostrais são independentes entre si, é possível reescrever 2.10 da seguinte maneira:

$$L(\theta) = p_e(e|\theta) = \prod_{j=1}^J p_e(e(j)|\theta), \quad (2.11)$$

onde  $J$  representa o tamanho do conjunto de amostras. No problema de separação

cega de fontes, o conjunto de amostras é dado por  $X = [X_1, \dots, X_N]$ , e o parâmetro que buscamos estimar é a matriz de misturas  $A$ . Assim, podemos reescrever 2.11 como:

$$L(A) = \prod_{j=1}^N p_x(X_j|A). \quad (2.12)$$

Além disso, no caso de modelos determinados, estimar  $A$  é equivalente a estimar a matriz de separação  $W$  (mais precisamente, a sua inversa). Assim, após um desenvolvimento matemático de 2.11 [16,23,35], que leva em conta que  $A = W^{-1}$ , a Equação 2.12 pode ser escrita da seguinte maneira:

$$\frac{1}{J} \log(L(W)) = E\left\{ \sum_{n=1}^N \log(p_{s_n}(W_n X_j)) \right\} + \log(|\det(W)|), \quad (2.13)$$

onde  $L(W)$  é a função de máxima verossimilhança em função de  $W$ ,  $p_x$  é a pdf do conjunto de amostras  $X$  e  $p_{s_n}$  é a distribuição dos sinais de origem.

A partir dessa formulação, a estimativa de  $W$  pode ser entendida como um problema de maximização, dado por:

$$\tilde{W} = \arg \max_W (L(W)). \quad (2.14)$$

No entanto, como podemos observar, resolver qualquer problema por meio da estimação por máxima verossimilhança implica em saber a função de densidade de probabilidade das fontes verdadeiras. Esse, nem sempre é o caso, principalmente quando se trata de problemas não supervisionados. Considerando os preceitos do método conhecido como *Infomax*, torna-se possível encontrar uma solução razoável para esse problema, como explicaremos a seguir.

## 2.6 Critério Infomax

Em separação de fontes, o critério infomax [35], que tem origem no

treinamento não supervisionado de redes neurais, se baseia na maximização da entropia conjunta das estimativas das fontes, também abordada como maximização da transferência de informação. Outra designação igualmente válida para esse método descreve seu objetivo como maximização da informação mútua, com base na seguinte formulação:

$$\max_W I(Z, X) = \max_W (H(Z) - H(Z|X)). \quad (2.15)$$

Onde:

$$Z = g(WX) = g(Y) = [g_1(Y_1) \dots g_N(Y_N)]^T, \quad (2.16)$$

$$g(.) = [g_1(.) \dots g_N(.)]. \quad (2.17)$$

$H(Z)$  representa a entropia conjunta, uma medida da uniformidade da distribuição de um limitado conjunto de valores [33]. Finalmente,  $H(Z|X)$  representa a entropia de  $Z$  condicionada a  $X$  que, por sua vez não depende de  $W$ , ou seja, é apenas uma constante no processo de otimização [35].

Assim,  $I(Z|X)$  corresponde à informação mútua de  $Z$  condicionada a  $X$ , sendo que  $g(.)$  são funções de ativação como  $g(.) = -\tanh(Y_i(n))$  e  $g(.) = -Y_i(n)^3$ . Para ilustrar melhor a base de funcionamento do infomax, a Figura 2.4 fornece uma visão geral do método, onde podemos observar que os dados de entrada são as misturas das fontes e que o processo continua após passar de sua fase linear.

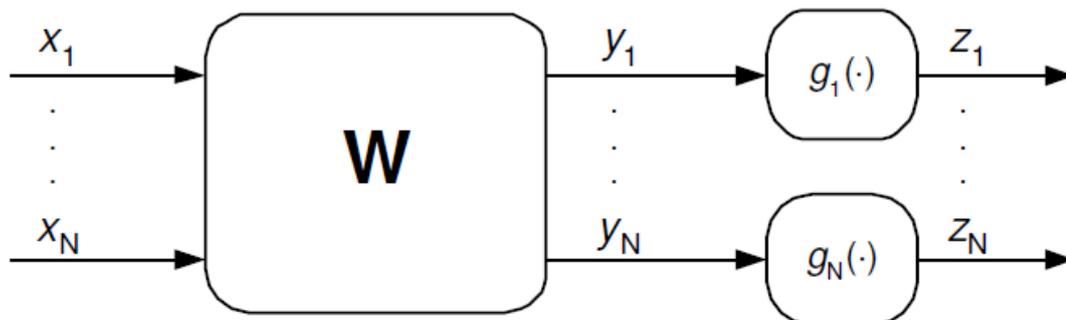


Figura 2.4: Base de funcionamento do critério *Infomax* [16].

Assim como na máxima verossimilhança, as manipulações matemáticas para obtenção da função contraste serão omitidas, podendo ser igualmente observadas com mais detalhes nos trabalhos de Hyvärinen, Karhunen e Oja [23] e de Duarte [16]. Resta então, apresentar a função contraste resultante desse processo como sendo:

$$\max_W H(z) \triangleq \max_W E \left\{ \sum_{i=1}^N \log (g'_i(W_i X)) \right\} + \log(|\det(W)|). \quad (2.18)$$

Note que a formulação final do critério *infomax* tem uma estreita relação com a máxima verossimilhança. A única diferença que pode existir entre esses dois métodos consiste na escolha da função de ativação. Enquanto o critério *infomax* propõe que a função de ativação seja uma estimativa da função densidade de probabilidade dos dados em análise, a máxima verossimilhança necessita de certeza nesse ponto. Assim, quando as essas duas funções coincidem, ambos os métodos têm a mesma formulação final.

## 2.7 Método do Gradiente Ascendente ou Descendente

Uma vez definida a função contraste a ser otimizada, resta apenas fixar um método de otimização que resolva o problema. O gradiente ascendente ou descendente (do inglês – Steepest Ascent or Descent) é um método iterativo que faz parte de um conjunto de técnicas que usam o gradiente como base de cálculo, e surgem como uma boa solução para o caso. Também podem ser usados nessa etapa, métodos como o gradiente natural [36,37,38] ou aqueles baseados em derivadas de segunda ordem [39], porém eles não se enquadram no foco do nosso estudo e, portanto, não serão abordados.

Em seu trabalho, Stone [33] explica a dinâmica do gradiente em função do cálculo da derivada local da função contraste. Dado um ponto qualquer, o resultado desse cálculo indica qual seria o melhor caminho em direção ao ponto ótimo. A cada iteração, um pequeno movimento é feito na direção do gradiente (maximização) ou na direção contrária a ele (minimização). Partindo da premissa de que pretendemos encontrar uma matriz  $W$  que otimize uma função custo  $J(W)$ , dado

um conjunto de misturas  $X$ , a regra de atualização, começando por um ponto inicial  $W(0)$ , fica da seguinte forma:

$$W(t) = W(t - 1) - \mu \frac{dJ(W)}{dW} \Big|_{W=W(t-1)}, \quad (2.19)$$

onde,  $\mu$  é passo do gradiente. Esse valor se refere ao quanto  $W$  vai se deslocar em função do resultado do gradiente, e no sentido ditado por ele.

Assim, A. J. Bell and T. J. Sejnowski [35] mostram que, aplicando o critério *infomax* ao processo de separação, e considerando a função contraste representada pela equação 2.18, cujo objetivo é a maximização da entropia conjunta, temos a seguinte regra de atualização do gradiente:

$$W(t) = W(t - 1) - \mu(E\{f(Y(n))X(n)^T\} - W(t - 1)^{-T}). \quad (2.20)$$

As iterações serão repetidas até que um determinado critério de parada seja atingido.

Como um exemplo simples de aplicação do gradiente, Stone [33] simula um processo de maximização, usando a curtose como a função contraste, supondo que essa função possa ser descrita como  $K = -(\beta + 1)^2 + 25$ . Em seguida, Stone escolhe pontos iniciais no intervalo  $[-3, 3]$ , e mostra que o resultado do gradiente, independente do ponto inicial escolhido, sempre indica a direção do ponto  $\beta = -1$ , onde a curtose atinge seu maior valor, como podemos observar na Figura 2.5.

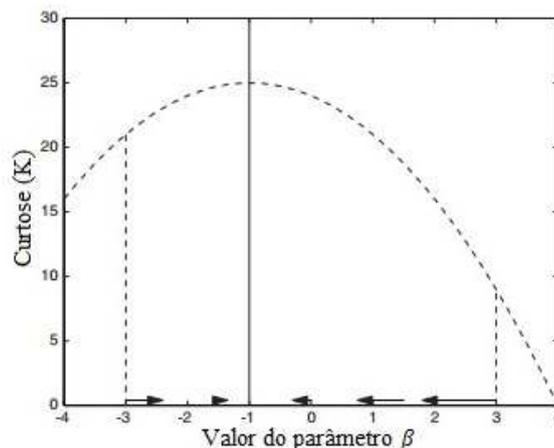


Figura 2.5: Gráfico da Curtose em função do parâmetro  $\beta$  [33].

### 2.7.1 Presença de mínimos locais

Apesar de efetivo, o método gradiente só tem garantia de convergência local. Nesse sentido, ele pode ser atraído por pontos de mínimos locais que estão distantes da solução ótima global. Alguns estudos como o de Pham e Vrins [40] ou o de Vrins e Verleysen [41] apontam que a principal causa desse problema tem relação com a etapa de branqueamento, cuja dinâmica de funcionamento explicaremos logo adiante.

A Figura 2.6 mostra um caso de minimização com tendência a convergir em um ponto de mínimo local. Nesse caso, o problema consiste no ponto inicial escolhido para o cálculo do gradiente. Na região onde se encontra o vetor gradiente há presença de mínimo local, o que pode levar à convergência de tal vetor nesse ponto, quando espera-se que a convergência aconteça no ponto de mínimo global.

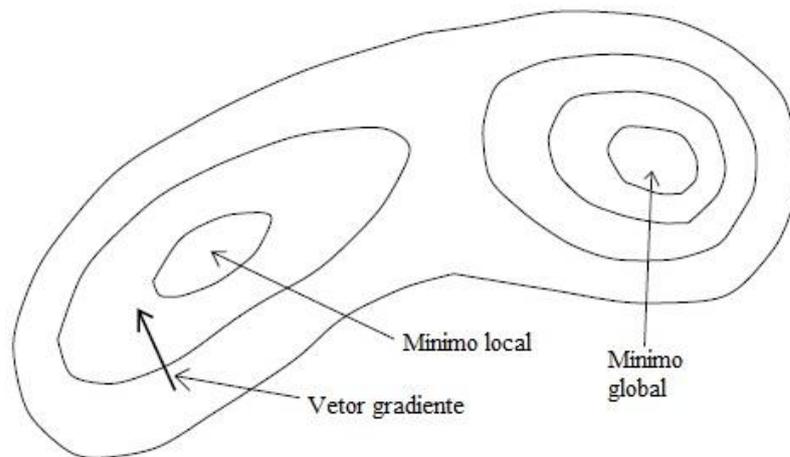


Figura 2.6: Superfície de contorno de uma função com mínimos locais (adaptado de [23]).

#### 2.7.1.1. Branqueamento

O branqueamento é uma etapa normalmente aplicada como uma ferramenta de preparação dos dados, antes da execução do processo de separação em si. Mais precisamente, o branqueamento resulta em dados que são descorrelacionados entre si. A descorrelação, por sua vez, não significa independência dos dados, mas acredita-se que facilita o processo de separação,

como explicaremos a seguir. Vamos começar pela definição matemática de descorrelação, representada pela Equação 2.21.

Duas variáveis aleatórias são ditas descorrelacionadas se sua covariância é igual a zero, ou seja:

$$\text{cov}(Y_1, Y_2) = E\{Y_1 Y_2\} - E\{Y_1\}E\{Y_2\} = 0, \quad (2.21)$$

onde,  $\text{cov}(Y_1, Y_2)$  é a covariância de  $Y_1$  e  $Y_2$  e  $E$  representa a esperança desses dados.

Para duas variáveis de média zero, a covariância é igual à correlação:

$$\text{corr}(Y_1, Y_2) = E\{Y_1 Y_2\}. \quad (2.22)$$

Ademais, a propriedade que relaciona independência e descorrelação estabelece que se duas variáveis são independentes, é válida a igualdade proposta pela seguinte equação:

$$E\{h_1(Y_1)h_2(Y_2)\} = E\{h_1(Y_1)\}E\{h_2(Y_2)\}, \quad (2.23)$$

onde,  $h_1$  e  $h_2$  são duas funções quaisquer. Note que a substituição  $h_1(Y_1)$  por  $Y_1$ , e  $h_2(Y_2)$  por  $Y_2$  implica em uma equivalência entre a Equação 2.21 e as Equações 2.22 e 2.23.

Em suma, o objetivo geral do branqueamento é descorrelacionar os dados e fazer com que eles tenham variância unitária. Matematicamente, a matriz de covariância de dados branqueadas é dada por:

$$E\{YY^T\} = I, \quad (2.24)$$

de modo que a matriz de branqueamento  $\tilde{A}$  deve atender à seguinte condição:

$$E\{ZZ^T\} = \tilde{A}E\{SS^T\}\tilde{A}^T = \tilde{A}\tilde{A}^T = I. \quad (2.25)$$

Uma das possibilidades que resultam nesse objetivo, obedecendo às

propriedades de covariância e correlação, é dada pela Equação 2.26.

$$\begin{bmatrix} Z_1(n) \\ Z_2(n) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} S_1(n) \\ S_2(n) \end{bmatrix}. \quad (2.26)$$

onde  $\theta$  é o ângulo que parametriza a rotação entre  $S(n)$  e  $Z(n)$ . Portanto, pode-se dizer que os sinais resultantes  $Z$ , enquanto elementos descorrelacionados, são uma versão rotacionada das fontes. É importante lembrar que essa intervenção é uma ilustração de um processo de branqueamento em um problema que tem apenas duas fontes. Nesse caso, após o branqueamento resta apenas o parâmetro  $\theta$  para ser estimado, em vez de quatro coeficientes de mistura no caso de problemas que não passaram pelo processo de branqueamento. Isso explica por que o branqueamento é considerado como um facilitador na busca por elementos estatisticamente independentes.

Além dessas observações, Hyvärinen, Karhunen e Oja [23] explicam que a escolha da região de busca, ou seja, do ponto de partida no processo de otimização, deve ser feita cuidadosamente. Caso contrário, ainda que não seja feito o branqueamento, há possibilidade de convergência em pontos ótimos locais, em casos cujas superfícies de custo são multimodais. Esses impasses podem ser resolvidos por meio da aplicação de métodos mais avançados de otimização como as meta-heurísticas, que vêm ganhando cada vez mais visibilidade nessa área por se mostrarem consideravelmente eficazes na maioria dos casos onde são aplicadas.

## CAPÍTULO 3

### META-HEURÍSTICAS

As meta-heurísticas fazem parte de um extenso conjunto de técnicas que abordam problemas de otimização no campo da pesquisa operacional, também conhecida como a “ciência da tomada de decisão”. Isso porque muitas operações, como controle de estoque ou transporte de mercadorias, envolvem objetivos e restrições que podem ser descritos como modelos matemáticos e acabam resultando em problemas decisórios de ordem quantitativa cuja solução deve ser a melhor possível, ou seja, aquela que otimiza o sistema em questão [42].

Existe uma série de problemas, de origens diversas, que podem ser descritos como um problema de pesquisa operacional, dentre eles: processos estocásticos, problemas de fila, processamento de ordens, produção e estoque, decisões multi-critérios, alocação e programação linear, e planejamento de rede. Observe, por exemplo, a Figura 3.1 que mostra diversos possíveis domínios de um problema de pesquisa operacional, onde estão incluídos os casos supracitados, deixando evidente a complexidade com a qual podemos nos deparar nesse campo.

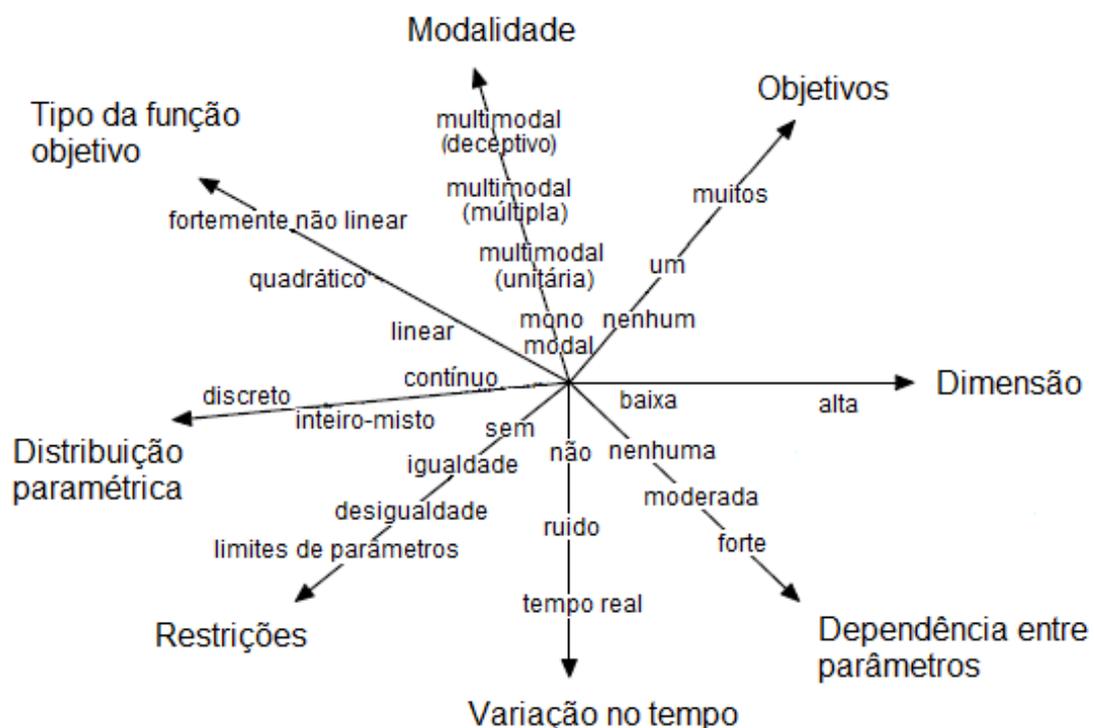


Figura 3.1: Diferentes domínios de problemas de otimização (adaptado de [43]).

Percebe-se que, no campo da pesquisa operacional, há um extenso conjunto de problemas que não podem ser resolvidos por métodos exatos. Isso se deve, por exemplo, à não linearidade de uma função objetivo ou a problemas que podem ter um aumento exponencial de sua complexidade à medida que crescem suas proporções. A não linearidade implica na possível presença de mínimos locais e, por este motivo, impede ou dificulta a busca por uma solução ótima.

Nesse contexto, as meta-heurísticas fornecem uma maneira eficaz para abordar cenários complexos em problemas de otimização. Elas combinam busca local com outros procedimentos de alto nível estratégico com o objetivo principal de fugir de mínimos locais [44], funcionando como um amplo campo de busca cujo resultado seja o ótimo ou o mais próximo possível dele. Para que isso aconteça, uma meta-heurística trabalha de forma iterativa, com uma estratégia que muda segundo a técnica escolhida, e com substancial aporte computacional. Em geral, a cada iteração, as novas soluções geradas são avaliadas segundo o resultado obtido em uma função que representa a essência do problema, ou seja, que o resume. Essa função é chamada de “função objetivo”.

Em meio à extensa ramificação de meta-heurísticas e suas variantes existentes atualmente, as semelhanças e diferenças entre elas são inúmeras, mas, de modo geral, o ciclo de aplicação dessas técnicas é similar e pode ser representado pelo fluxograma da Figura 3.2.

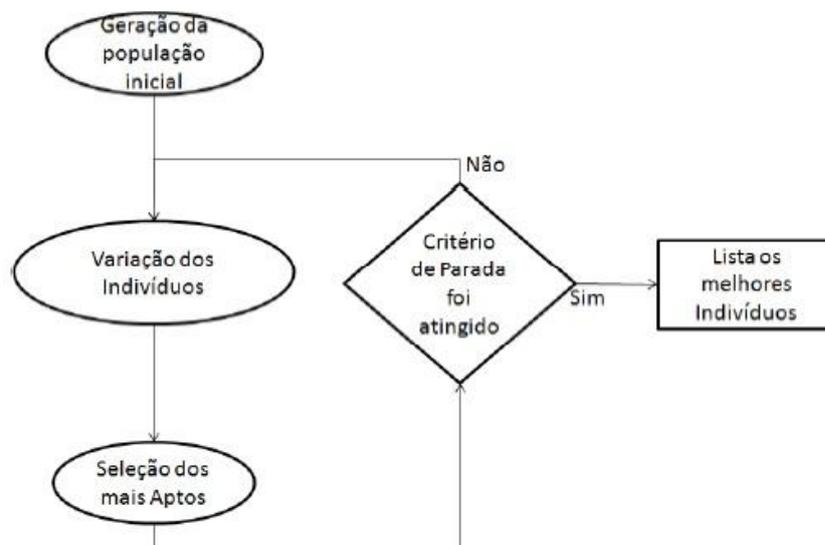


Figura 3.2: Estrutura básica de funcionamento de uma meta-heurística [45].

As diferenças entre as meta-heurísticas concentram-se, principalmente, em suas origens e suas estruturas de cálculo de possíveis novas soluções rumo ao ponto ótimo global. Isso implica que o critério de escolha de uma determinada meta-heurística está relacionado, sobretudo, ao tipo de problema com o qual estamos lidando, à facilidade de aplicação do método, e à sua demanda computacional.

Com o objetivo de proporcionar uma maior abrangência do estudo, neste trabalho serão consideradas meta-heurísticas de diferentes origens, tais como evolutivas e inteligência de enxame. A meta-heurística de base evolutiva que será aplicada ao problema é a evolução diferencial (DE, do inglês *Differential Evolution*). Já em inteligência de enxame, aplicaremos a *particle swarm optimization* (PSO) e a otimização por colônia de formigas para domínios contínuos (ACO<sub>R</sub>, do inglês *Ant Colony Optimization For Continuous Domains*). Além desses métodos, aplicaremos ainda o *simulated annealing*, uma meta-heurística clássica cuja base será explicada mais adiante.

As técnicas de origem evolutiva são aquelas baseadas no modelo de evolução orgânica, e que contam com um aporte estatístico que represente sua base populacional [46]. A base populacional de meta-heurísticas de base evolutiva tem grande importância pelo fato de que a manipulação de diversos vetores ao mesmo tempo, por meio de parâmetros adequadamente ajustados, é uma forma eficaz de ajudar outros vetores a fugirem de mínimos locais [47].

As técnicas com base em inteligência de enxame [48], em sua definição literária, são aquelas que objetivam implementar um robusto sistema artificial para resolver problemas, inspiradas no comportamento de insetos sociais como as formigas e certas abelhas e vespas. De maneira mais abrangente, podemos considerar que essas técnicas bio-inspiradas se resumem em um conjunto que procede do comportamento social coletivo dos organismos como, por exemplo, o *particle swarm optimization* [49].

A técnica de otimização conhecida como *simulated annealing* foi inspirada no processo físico de recozimento dos sólidos, onde sua configuração de rede é a mais estável possível apenas quando o resfriamento acontece de forma extremamente lenta [50]. Ou seja, cada pequena perturbação feita nas soluções representa o decaimento gradual da temperatura. De todos os métodos estudados, o SA é o único que não exige uma população de soluções em sua base de cálculos.

### 3.1. Evolução diferencial (DE)

Criada por Price e Storn [47], a evolução diferencial (DE, do inglês Differential Evolution) consiste em uma meta-heurística de base evolutiva que foi projetada para ser facilmente implementável, ter boas propriedades de convergência, avaliar de forma eficiente mesmo em casos de funções objetivo com importante demanda computacional, e ter capacidade de resolver tanto problemas não lineares quanto aqueles não diferenciáveis ou multimodais [47].

Como o próprio nome sugere, o modelo de evolução natural e biológica formulado por Charles Darwin é o ponto de partida de métodos de base evolutiva. Na teoria da evolução proposta por ele, acontece a seleção natural, onde apenas as espécies mais bem adaptadas têm mais chances de sobreviver e, de certa forma, evoluem [46]. Atualmente, sabe-se que fazem parte desse processo, etapas como mutação e seleção, que representam a hereditariedade e a sobrevivência de apenas alguns dos indivíduos da população. Além da evolução diferencial, outros métodos compartilham a mesma base, como programação genética [51], estratégias evolutivas [52], e algoritmo genético [53].

A meta-heurística evolução diferencial busca o ponto ótimo fazendo combinações das soluções viáveis encontradas, de maneira que apenas as melhores continuem na população; elas serão recombinadas por diversas gerações até que o ciclo evolutivo seja concluído. Em resumo, a evolução diferencial segue uma sequência de quatro etapas:

- i. Na primeira é feita a geração aleatória da população de soluções iniciais. Apesar de ser um método de base populacional, o tamanho da população é sempre restrito para que ela não cresça de forma exponencial, como aconteceria na vida real [46];
- ii. Na segunda etapa acontece a mutação, que consiste na escolha de três vetores aleatórios, dois deles são subtraídos entre si e em seguida ponderados e o resultado é somado ao terceiro vetor;
- iii. Na terceira etapa, chamada de *crossover* (ou cruzamento), há a troca de elementos aleatórios do vetor mutado com o vetor que pertence atualmente à população de soluções (chamado de vetor base);
- iv. Já na quarta etapa, o vetor base e o novo vetor (vetor experimental)

são avaliados de acordo com a função objetivo. Aquele cujo resultado é melhor permanece na população de soluções. Essa etapa é chamada de seleção.

É importante lembrar que as etapas dois, três e quatro se repetem diversas vezes, inicialmente para cada vetor do conjunto de soluções e, em seguida, até que seja atingindo determinado número de iterações ou algum outro critério de parada pré-estabelecido. Espera-se que esses movimentos levem ao ponto ótimo, como é ilustrado na Figura 3.3.

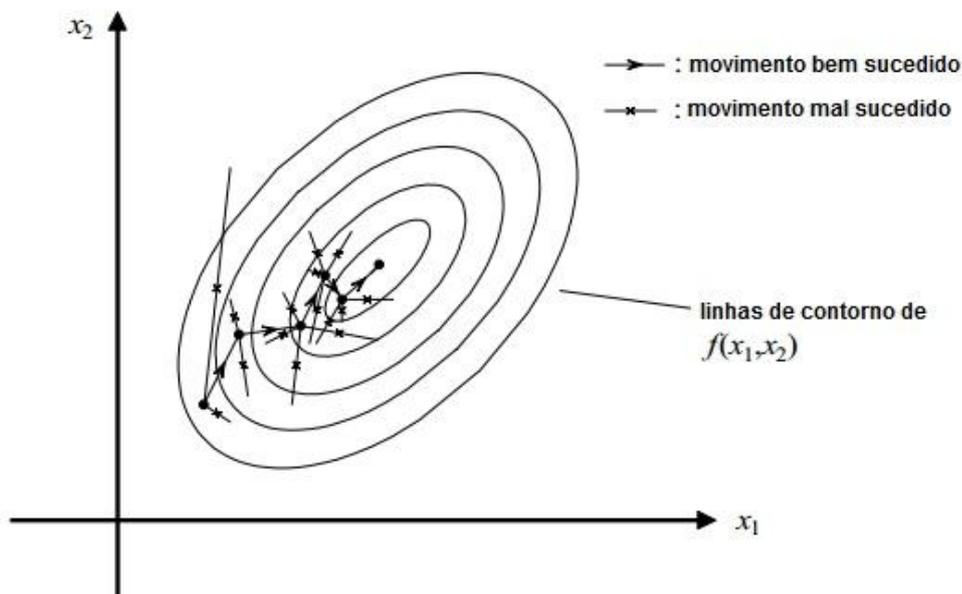


Figura 3.3: Representação dos movimentos de busca em Evolução Diferencial (Adaptado de [54]).

É possível observar que o princípio desse método é similar ao algoritmo genético (GA, do inglês *Genetic Algorithm*). Porém, uma das principais diferenças é que em GA, a mutação é feita por meio de pequenas perturbações em genes individuais, enquanto que a DE realiza combinações aritméticas entre diferentes genes [46]. Outra significativa diferença é que GA trabalha com números binários. De acordo com Price e Storn [54], a evolução diferencial exige menos esforço computacional, utiliza sua memória de cálculo de forma mais eficiente, e é mais fácil de implementar mesmo quando se trata de problemas mais complexos.

Na sequência, detalhamos as quatro etapas que constituem o algoritmo evolução diferencial.

### 3.1.1 Geração da solução inicial e definição de parâmetros

Antes de iniciar o processo de otimização em si, é necessário estabelecer quatro parâmetros: o número de iterações desejado ( $it$ ), o tamanho da população de soluções ( $B \geq 4$ ), o fator de ponderação que atuará na etapa de mutação ( $f$ ), e a quantidade de elementos ( $k$ ) do vetor analisado que substituirá os elementos de mesma posição do vetor mutado.

Uma população de  $N$  soluções é gerada de forma aleatória, de acordo com uma distribuição uniforme de probabilidade; tal distribuição também é considerada nas etapas de mutação e *crossover*. Dessa operação um conjunto de soluções iniciais, representado por  $X = [X_1, X_2, \dots, X_N]$ , é obtido de modo que cada uma dessas soluções é uma solução viável para o problema, onde  $X_i = [x_{1,i}, x_{2,i}, \dots, x_{m,i}]$  é um vetor solução com dimensão  $m$ .

### 3.1.2 Mutação

Nessa etapa são escolhidos, de forma aleatória, três vetores do conjunto de soluções, com exceção do vetor que está sendo analisado. Dois desses vetores serão subtraídos e o resultado da soma será ponderado e somado ao terceiro vetor, de acordo com as Equações 3.1 e 3.2:

$$V_i = X_{r_0} + f \cdot (X_{r_1} - X_{r_2}); \quad (3.1)$$

$$V_i = [v_{1,i}, v_{2,i}, \dots, v_{m,i}], \quad (3.2)$$

Onde  $r_0, r_1$  e  $r_2$  correspondem aos índices dos vetores selecionados,  $f$  representa o fator de ponderação que controla a amplitude da diferença entre os vetores  $X_{r_1}$  e  $X_{r_2}$ , podendo assumir qualquer valor no intervalo  $[0,2]$ , segundo Storn e Price [47]. Na Figura 3.4, podemos observar o processo de mutação por meio da manipulação aritmética dos três vetores. A partir dessa figura, fica claro que a mutação é uma das formas de explorar o máximo possível o espaço de busca.

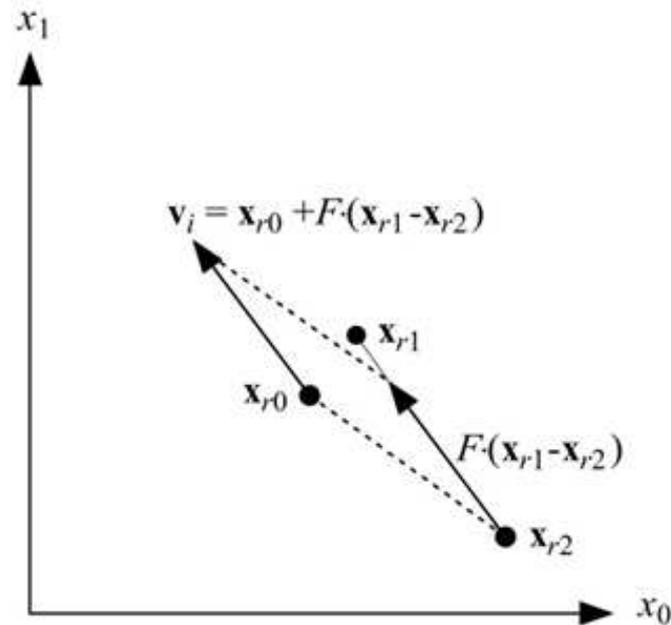


Figura 3.4: Representação do processo de mutação (adaptado de [54]).

### 3.1.3 Crossover

Com o objetivo de aumentar a diversidade dos vetores perturbados [47], o vetor  $V_i$ , após ser gerado pelo processo de mutação, é submetido a um processo de troca de elementos (cruzamento) entre ele e o vetor base, conhecido como *crossover*. Para tanto, são escolhidas aleatoriamente  $k$  constantes, representadas por:

$$C = C_1, C_2, \dots, C_k \quad (3.3)$$

que representarão quais elementos do vetor  $V_i$  serão substituídos pelos elementos de mesma posição do vetor base. Assim, levando em conta os valores dessas constantes, obtêm-se um novo vetor (vetor experimental), dado por:

$$U_i = [X_{c1,i}, V_{2,i}, X_{c2,i}, \dots, X_{ck,i}]. \quad (3.4)$$

Note que os elementos de  $X$  e de  $V$  (vetor mutado) estão misturados em um único vetor.

O processo de *crossover* descrito acima é ilustrado na Figura 3.5. Essa figura mostra qual seria o resultado do *crossover* entre dois vetores cuja dimensão é  $m = 8$ , aplicando  $k = 3$ . Ou seja, caso três dos oito elementos fossem trocados aleatoriamente, mantendo apenas a mesma posição no vetor receptor.

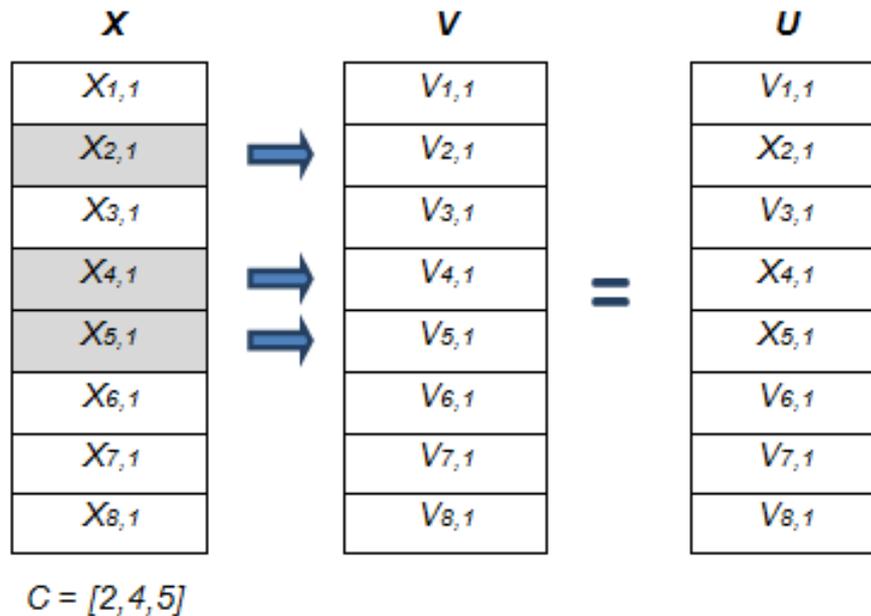


Figura 3.5: Representação do processo de *crossover*.

#### 3.1.4 Seleção

Concluídas as etapas anteriores, temos como resultado dois vetores, o vetor base e o vetor experimental. Resta agora avaliar qual dos dois oferece a melhor solução. Se for o vetor base, o processo de otimização continua normalmente, sem que nenhuma ação seja tomada. Caso contrário, o novo vetor substituirá o vetor base, ou seja:

$$X_i = \begin{cases} U_i & \text{se } F(U_i) \leq F(X_i) \\ X_i & \text{(caso contrário)} \end{cases} \quad (3.5)$$

Em seguida, todo o processo é repetido para o vetor seguinte ( $X_{i+1}$ ) até que todos os vetores da população de soluções tenham sido avaliados e/ou transformados e até que o critério de parada seja atingido. Vale lembrar que a

Equação 3.5 representa o processo de decisão de um problema de minimização. No caso de maximização,  $F(U_i)$  deve ser maior ou igual a  $F(X_i)$  para que o vetor experimental  $U_i$  substitua o vetor base  $X_i$  na população de soluções.

A sequência de passos seguida pela meta-heurística evolução diferencial, descrita anteriormente, pode ser observada de uma maneira mais generalizada na Figura 3.6.

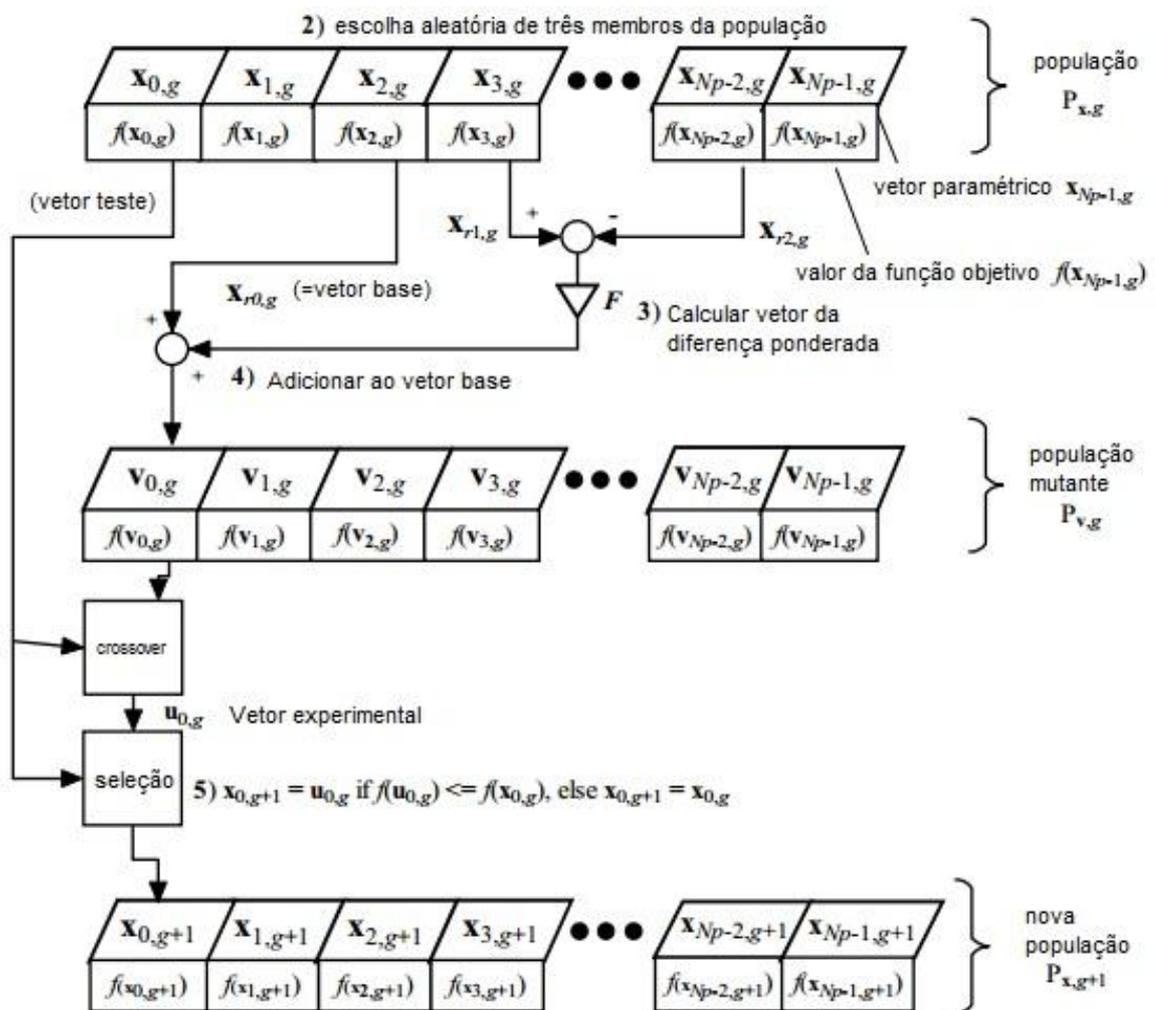


Figura 3.6: Funcionamento geral da Evolução Diferencial (adaptado de [54]).

É importante observar que a notação usada por Price e Storn [54] na Figura 3.6 acrescenta a geração  $g$  em seus vetores. Note que essa diferença não interfere no funcionamento do método em si, ela apenas permite identificar quantas vezes o processo de busca já foi repetido, ou seja, em qual iteração ele se encontra.

## 3.1.5 Pseudo-código

Na Tabela 3.1 apresentamos o pseudo-código da evolução diferencial.

FUNÇÃO = DE (X)

Definir parâmetros:  $it, k, f, B$

Gerar população de soluções iniciais aleatórias

ENQUANTO (critério de parada não for atingido) FAÇA

PARA  $i = 1$  até  $M$  (dimensão do vetor solução)

- (i) Selecione aleatoriamente 3 vetores  $(r_0, r_1, r_2)$  da população de soluções, com exceção do vetor  $i$
- (ii) Faça a mutação:  $V_i = X_{r_0} + f \cdot (X_{r_1} - X_{r_2})$
- (iii) Selecione aleatoriamente  $k$  constantes (e.g.  $C1 = 1, C2 = 3$ )
- (iv) Faça o cruzamento e obtenha o vetor experimental  $U_i$
- (v) Faça a seleção do melhor vetor:

SE  $(F(U_i) \leq F(X_i))$  (ou  $\geq$  se for maximização)

$X_i = U_i$

SE NÃO

$X_i = X_i$

FIM SE

FIM PARA

FIM ENQUANTO

FIM FUNÇÃO

Tabela 3.1: Pseudo-código da meta-heurística Evolução Diferencial.

### 3.2 Otimização por Colônia de Formigas para domínios contínuos (ACO<sub>R</sub>)

Antes de descrever a otimização por colônia de formigas para domínios contínuos (ACO<sub>R</sub>), é preciso lembrar que ela possui uma estreita ligação com a otimização por colônia de formigas (ACO) [55], com a diferença primordial de que esta última foi elaborada para problemas combinatórios. Por outro lado, ambos os métodos foram criados com base no comportamento social das formigas, sendo que o ACO<sub>R</sub> é considerado como uma extensão da ACO [56].

Apesar de ser conhecida como uma importante ferramenta de otimização para problemas combinatórios, a ACO é considerada como uma técnica pouco estudada, e de difícil aplicação, especialmente em problemas contínuos [57]. É possível encontrar diversas propostas de meta-heurísticas de mesma base para problemas contínuos. A primeira delas foi a CACO (do inglês – *Continuous ACO*) [58,59,60,61], depois vieram outras propostas como a CIAC (do inglês – *Continuous Interacting Ant Colony*) [62,63], a DACO (do inglês – *Direct Application of Ant Colony Optimization*) [57], ou ainda a ACO expandida [64], que pode ser considerada como uma das principais predecessoras da ACO<sub>R</sub> por ter apresentado um trabalho baseado em função de densidade de probabilidade.

A ACO<sub>R</sub>, objeto de nosso estudo, foi proposta por Socha e Dorigo [56], e é considerada na literatura como aquela que mais se aproxima da formulação para problemas combinatórios originada no comportamento social das formigas. Tal como seu predecessor, esse método faz parte da chamada inteligência de enxame [48], um conjunto de meta-heurísticas baseado no comportamento social dos organismos.

Para explicar melhor a origem do método, consideremos um grupo de formigas trabalhando à procura de alimentos. Assim que um alimento é encontrado, é preciso buscar o caminho de volta e, de uma forma particular, as formigas acabam por traçar o trajeto mais curto até o formigueiro. Isso acontece porque durante esse processo, por onde passam, as formigas depositam uma substância biologicamente ativa, chamada de feromônio, em quantidades que variam proporcionalmente à quantidade e qualidade da comida, funcionando como um meio de comunicação entre elas [48,65]. Quanto mais curto for o caminho, quanto mais alimento for encontrado e quanto melhor ele for, maior a probabilidade de outras formigas passarem por ele e reforçarem a substância deixada anteriormente antes que ela evapore.

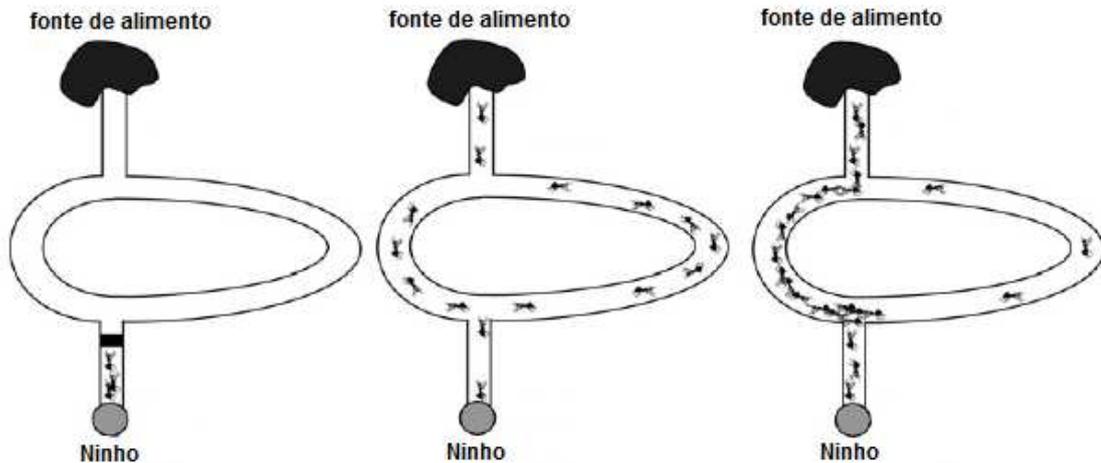


Figura 3.7: Representação do experimento mostrando a organização das formigas no processo de busca por alimentos (adaptado de [66]).

Castro [66] mostrou a organização das formigas *I. Humilis* com um experimento que liberava todas as formigas ao mesmo tempo e as deixava livres em sua busca por alimentos, como mostra a Figura 3.7. É possível perceber que dentro de pouco tempo todas as formigas se organizam em um mesmo trajeto pelo simples fato de seguirem o caminho com maior quantidade de feromônio. Em termos gerais, esse modelo é usado como base da sequência de cálculo para a construção probabilística do espaço de busca [56]. O primeiro método baseado nessa organização social foi o *Ant System* (AS) [55, 67, 68, 69], depois outras técnicas foram surgindo, entre elas a ACO.

Podemos então dizer que, nesse sistema, as formigas representam as soluções, o feromônio representa as memórias direta e indireta do sistema, e o caminho mais curto, claramente, é o objetivo, é a função que define o problema que queremos otimizar. Em resumo, a otimização por colônia de formigas para domínios contínuos acontece da seguinte forma:

- i. Na primeira etapa é feita a geração aleatória da população de soluções iniciais e elas são ordenadas, segundo a função objetivo  $F(X)$ , começando pela melhor solução.
- ii. Na segunda etapa, o peso de cada solução é calculado segundo uma fórmula específica que será apresentada mais adiante. Esse peso é usado, inicialmente para o cálculo da probabilidade de escolha de uma função base utilizada na geração de novas

soluções.

- iii. Na terceira etapa, são geradas novas soluções em função do peso calculado e de outros dados retirados da população de soluções, e dos parâmetros ajustados para o problema.
- iv. Já na quarta etapa, chamada de atualização do feromônio, as soluções geradas na etapa anterior são adicionadas à população de soluções para, em seguida, passarem pelo processo de seleção das melhores soluções, sempre organizando em ordem dos melhores resultados.

Na sequência, os passos descritos acima serão detalhados.

### 3.2.1 Geração da solução inicial e definição de parâmetros

São cinco os parâmetros que devem ser definidos antes da implementação dessa meta-heurística: o tamanho da população de soluções iniciais ( $k$ ), o número de iterações desejadas ( $it$ ), o fator multiplicador do desvio padrão ( $\varepsilon$ ), um fator multiplicador utilizado no cálculo dos pesos de cada solução ( $q$ ), e a quantidade de novas soluções desejadas ( $m$ ) que serão geradas a cada iteração.

É importante ressaltar que o fator  $q$ , usado no cálculo dos pesos, influencia na escolha da população de soluções que será a base de cálculo de novas soluções para o problema. Quando  $q$  é pequeno as melhores soluções têm maior probabilidade de serem escolhidas. Quando  $q$  é grande, a probabilidade se torna mais uniforme [56]. Já o fator  $\varepsilon$  é um parâmetro que deve ser maior que zero e que influencia na velocidade de convergência do algoritmo, sendo inversamente proporcional a ela.

Definidos os parâmetros, é preciso considerar as seguintes entradas:

$$X = X_1, X_2, \dots, X_k, \quad (3.6)$$

onde  $X$  é um conjunto de soluções geradas de forma aleatória, com distribuição uniforme, sendo que  $X_i = [x_{1,i}, x_{2,i} \dots x_{N,i}]$  e  $N$  é a dimensão de cada vetor. É preciso

considerar também um conjunto de soluções  $S$ , da seguinte forma:

$$S = S_1, S_2, \dots, S_k, \quad (3.7)$$

onde  $S_i = [s_{1,i}, s_{2,i}, \dots, s_{N,i}]$  e pode ser obtido inicialmente por meio de  $X$ , sendo  $S_{inicial} = X_{inicial}$ .

É importante lembrar que, considerando a organização das soluções,  $F(X_1) > F(X_2) > F(X_3) > \dots > F(X_k)$ . Essa mesma propriedade vale também para o conjunto  $S_i$ .

### 3.2.2 Cálculo dos pesos de cada função

Antes de explicar como são obtidos os pesos de cada função, vamos entender em que ponto eles são importantes na busca pelo ótimo global nesse método. O primeiro ponto se trata da escolha da solução que servirá como base de cálculo de novas soluções. Tal solução é escolhida de forma aleatória, segundo a probabilidade obtida, da seguinte maneira:

$$p_l = \frac{\omega_l}{\sum_{r=1}^k \omega_r}, \quad (3.8)$$

onde  $\omega_l$  é o peso da solução  $l$  e  $p_l$  é a probabilidade de escolher a solução  $l$  como a base de cálculo de novas soluções. O segundo ponto está relacionado diretamente ao cálculo de novas soluções, pois o peso das soluções também é utilizado na equação principal desse processo, como será mostrado logo adiante.

Sendo assim, a obtenção desse fator, é alcançada da seguinte maneira:

$$\omega_l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}}, \quad (3.9)$$

onde  $l$  é o ranque da solução, considerando que as soluções estejam organizadas da melhor para a pior.

### 3.2.3 Geração de novas soluções viáveis

A geração de novas soluções usa o conceito de função de densidade de probabilidade. Em seu trabalho, Socha e Dorigo [56] fazem uso de uma função kernel gaussiana, definida por eles como uma função de densidade de probabilidade que agrupa os dados analisados como uma soma ponderada de diversas funções gaussianas de uma dimensão. Na Figura 3.8, ilustramos as funções gaussianas individuais que são a base para a formação de uma função de kernel gaussiana ( $G$ ).

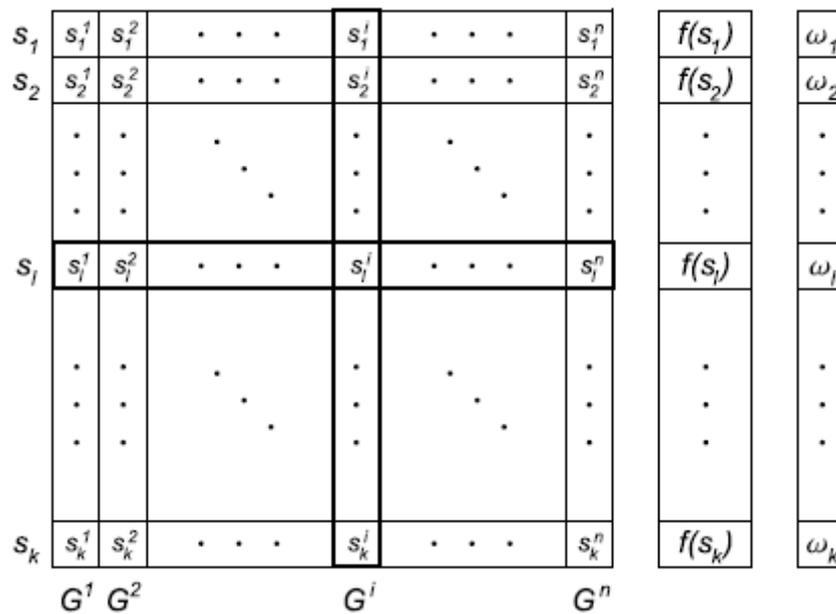


Figura 3.8: Funções gaussianas de uma dimensão ( $G^i$ ) em função das soluções individuais [56].

Nesse sentido, a função de kernel gaussiana escolhida para representar estatisticamente os dados e permitir sua manipulação, é definida como:

$$G(x) = \sum_{l=1}^k \omega_l g_l^i(x) = \sum_{l=1}^k \omega_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-\frac{(x-\mu_l^i)^2}{2\sigma_l^i{}^2}}, \quad (3.10)$$

onde  $\mu_l^i$  é considerado igual a  $s_l^i$  e  $\sigma$  é obtido da seguinte maneira:

$$\sigma_l^i = \varepsilon \sum_{e=1}^k \frac{|S_e^i - S_l^i|}{k-1}. \quad (3.11)$$

Nessa equação,  $S_l^i$  representa a solução base escolhida segundo a probabilidade calculada, como mostra a Equação 3.8.

A Figura 3.9 mostra como funcionam a adaptação e a representatividade geral dos dados segundo a função de kernel gaussiana.

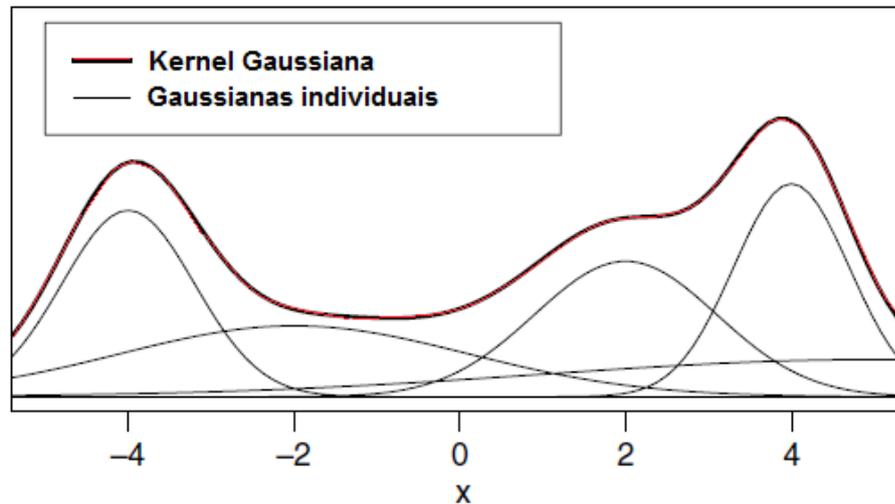


Figura 3.9: Curva da função de kernel gaussiana ( $G$ ) gerada a partir de gaussianas individuais [56].

No cenário apresentado nas Figuras 3.8 e 3.9, podemos observar, de modo geral, de que forma uma população de soluções, seus resultados em termos de função objetivo e seus pesos são utilizados, segundo a lógica da  $ACO_R$ , na busca pelo ponto ótimo global. Em resumo, uma nova solução calculada, representada pela curva da função de kernel gaussiana, de certa forma, é uma combinação das outras soluções, representadas pelas gaussianas individuais, de forma que diversos valores ou funções possam ser representados por um único valor ou função.

### 3.2.4 Atualização do feromônio

Ao término de uma iteração temos, como resultado,  $m$  novas soluções que precisam ser analisadas e comparadas com aquelas que já fazem parte da população de soluções. Essa etapa é chamada de atualização do feromônio justamente pelo fato de apenas as melhores soluções ficarem na população, servindo como indício do melhor caminho para as próximas buscas e guardando, intrinsecamente, a memória de cálculo. Para isso, basta juntar as  $m$  novas soluções

àquelas já existentes e deixar apenas as  $k$  melhores na população de soluções de forma que o tamanho desse arquivo nunca mude, como mostra a Figura 3.10.

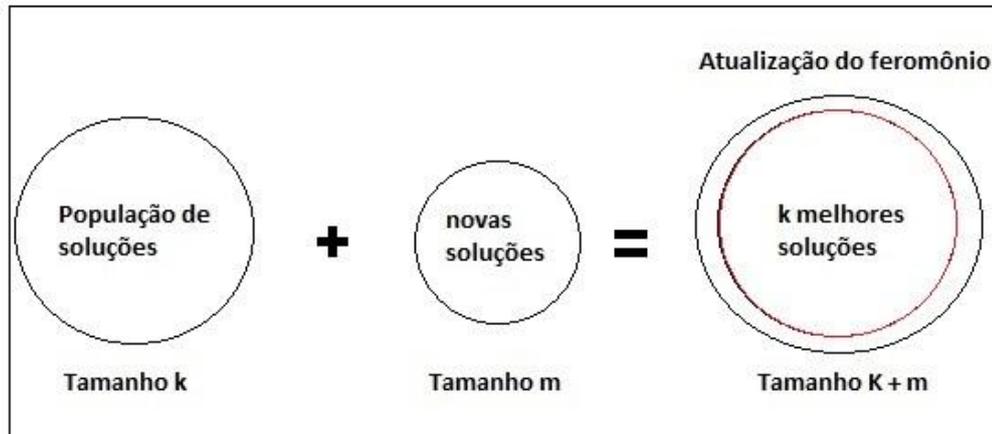


Figura 3.10: Representação da junção dos conjuntos na fase de atualização do feromônio.

Feita a seleção, a nova população de soluções deve ser novamente ordenada, e os pesos devem ser atribuídos a cada uma dessas soluções.

Levando esse processo em consideração, o método pode ser descrito como uma sequência de passos onde soluções candidatas são construídas seguindo um modelo de feromônio cuja base se concentra na função de distribuição de probabilidade. Essas soluções candidatas, em seguida, podem ser usadas para atualizar o valor do feromônio, contribuindo para a geração de futuras amostragens de soluções que sejam melhores que as atuais [70].

### 3.2.5 Pseudo-código

Na Tabela 3.2 apresentamos o pseudo-código da meta-heurística colônia de formigas para domínios contínuos.

FUNÇÃO = ACO<sub>R</sub> (X)

Gerar população de “ $k$ ” soluções iniciais aleatórias

Calcular o peso “ $\omega$ ” de cada solução gerada:  $\omega_l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}}$

Avaliar cada solução e ordená-las começando pela melhor

ENQUANTO (critério de parada não for atingido) FAÇA

PARA  $i = 1$  até  $m$  (quantidade desejada de novos vetores)

PARA  $l = 1$  até  $D$  (dimensão do vetor solução)

- (i) Escolha a solução base ( $S_l$ ):  $p_l = \frac{\omega_l}{\sum_{r=1}^k \omega_r}$
- (ii) Calcule o desvio padrão de cada solução em relação à solução base escolhida
- (iii) Multiplique o desvio padrão calculado pelo fator “e”
- (iv) Calcule o novo valor de cada elemento:

$$G(x) = \sum_{l=1}^k \omega_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-\frac{(x-\mu_l^i)^2}{2\sigma_l^{i^2}}}$$

FIM PARA

FIM PARA

- (i) Adicione as novas soluções à população de soluções
- (ii) Reavalie todo o grupo de soluções de acordo com a função contraste
- (iii) Ordene as soluções começando pela melhor
- (iv) Redefina o tamanho da população de soluções para “ $k$ ”
- (v) Redefina o peso de cada função

FIM ENQUANTO

FIM FUNÇÃO

Tabela 3.2: Pseudo-código da meta-heurística ACO<sub>R</sub>

### 3.3 Otimização por *Simulated Annealing* (SA)

A otimização por *simulated annealing* (recozimento simulado) leva esse nome por fazer uma analogia ao processo de recozimento de sólidos. Uma técnica onde o estado de energia mínima, situação em que a estrutura cristalina do material é mais regular e não possui defeitos, é atingido quando o sólido se resfria de forma lenta e gradual [71]. Um modelo de representação e simulação do resfriamento, conhecido como estatística mecânica, foi proposto por Metropolis et al. [72]. E foi a partir desse modelo que o método de otimização foi desenvolvido.

As primeiras versões dessa meta-heurística foram apresentadas por Kirkpatrick, Gelatt e Vecchi [73] e um pouco mais tarde por Cerny [74]. Esses autores propõem que os diversos parâmetros do sistema que se deseja otimizar são equivalentes às partículas em um sistema de resfriamento imaginário, e que o estado de energia é dado pela função objetivo. Para isso, eles utilizam o modelo de mecânica estatística que descreve o resfriamento, com certo grau de liberdade, em função do equilíbrio térmico até que uma determinada temperatura seja atingida. Eles propõem ainda que a perturbação de uma solução equivale à perturbação das partículas em determinado estado físico, e que um ótimo local pode ser comparado ao resfriamento excessivamente rápido do sistema.

Após esse trabalho pioneiro, o SA foi objeto de diversas pesquisas e testes e até hoje pode ser encontrado em diversos estudos, de diferentes formas de implementação, dando liberdade de escolha quanto à sua aplicação. A maioria dos problemas que têm sido resolvidos por meio da aplicação do SA são do tipo discreto, o que não impede que ele seja útil e eficaz em casos contínuos [71]. No trabalho de Tan [75], estão disponíveis diversos estudos no domínio do *simulated annealing* e suas diferentes aplicações.

Aguiar e Oliveira Junior et al. [76] explicam que, de maneira geral, o SA é uma ferramenta de otimização capaz de evitar perturbações demasiadamente grandes e desnecessárias. Sua forma de funcionamento busca regiões onde há, pelo menos, um mínimo local (ou máximo, dependendo do problema), e em seguida as exploram detalhadamente em busca de valores ótimos locais ou mesmo globais. Em resumo, podemos definir a sequência principal de passos dada pelo SA como:

- i. Inicialização de uma solução inicial de forma aleatória;

- ii. Cálculo da nova solução por meio da perturbação aleatória da solução anterior, cuja fórmula será apresentada mais adiante;
- iii. Verificação da aceitação da solução obtida em função do resultado da função objetivo. Essa etapa marca a capacidade dessa meta-heurística em fugir de mínimos locais em função da aceitabilidade, calculada de forma estatística, de soluções que podem ser piores que a atual [47];
- iv. Atualização da temperatura. O que, na realidade, significa dizer que há uma redução do número de iterações restantes.

Na sequência, detalharemos os passos descritos acima.

### 3.3.1 Inicialização da população

Na otimização por *simulated annealing*, essa etapa pode ser executada de forma simples e direta, pois é preciso considerar apenas uma solução inicial aleatória, representada por:

$$X_1 = x_{1,1}, x_{2,1}, \dots, x_{N,1}. \quad (3.12)$$

Essa etapa faz referência ao início do processo de resfriamento. Logo depois que o sólido é aquecido até passar para o seu estado líquido, todas as partículas se arranjam de forma aleatória, sendo que sua reorganização acontece no decorrer do resfriamento que, quando suficientemente lento, tende para um estado de energia mínima, ou estado ótimo [77].

### 3.3.2 Cálculo da nova solução

Nessa etapa, é feita a perturbação das partículas, ou seja, a perturbação da solução anterior, que, matematicamente, é descrita da seguinte maneira:

$$X(t + 1) = X(t) - f \cdot R, \quad (3.13)$$

onde  $X(t + 1)$  é a nova solução,  $X(t)$  é a solução anterior,  $f$  pode ser considerado como um fator de ponderação e  $R$  é um valor aleatório. Essa etapa é considerada como o deslocamento para um novo estado de energia por meio de um mecanismo de perturbação que altera levemente o atual estado de energia do sólido [77].

### 3.3.3 Análise da nova solução

A análise da nova solução é considerada como a grande vantagem desse método. Sua particularidade consiste em aceitar como solução ótima, uma solução que não é, necessariamente, melhor que aquela que será substituída. Isso ajuda a evitar que a busca convirja para mínimos locais, como ilustrado na Figura 3.11.

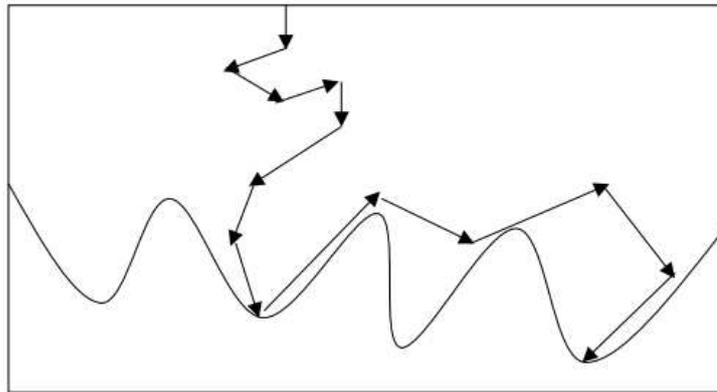


Figura 3.11: Simulação do processo de desvio de um mínimo local [78].

O processo de decisão é simples, se  $F(X(t + 1))$  for melhor que  $F(X(t))$ ,  $F(X(t + 1))$  passa a ocupar o lugar de melhor solução global encontrada até o momento, sendo que  $F$  representa a função objetivo. Caso contrário, a nova solução poderá ou não ser aceita. Para tanto, é necessário definir dois parâmetros, dados por:

$$P = e^{-\Delta F / K_B.T}. \quad (3.14)$$

Onde:

$$\Delta F = F(X(t + 1)) - F(X). \quad (3.15)$$

E ainda:

$T$  = temperatura atual do sistema;

$K_B$  = Constante de Boltzmann.

Sendo que  $K_B$  é uma constante física cuja formulação, proposta por Ludwig Boltzmann, busca relacionar a temperatura ao movimento dos átomos.

Em seguida, esses valores serão analisados considerando a seguinte regra de análise das soluções:

$$Se \begin{cases} P > U; & \text{Solução será aceita} \\ P < U; & \text{Solução será recusada} \end{cases} , \quad (3.16)$$

onde  $U$  é um número aleatório do conjunto  $[0,1]$ .

É importante observar que a Equação 3.14 foi elaborada para que uma solução pior seja aceita apenas no caso em que ainda falte muito tempo para o resfriamento total do sólido acontecer, ou seja, caso ainda restem muitas iterações para que o processo de otimização se complete. Caso contrário, a probabilidade de fazer um movimento em sentido oposto a um mínimo será quase nula, pois há um risco maior de o processo de busca chegar ao fim em um estado indesejado.

### 3.3.4 Atualização da temperatura;

Como o *simulated annealing* se baseia no processo de recozimento de sólidos e que a estrutura cristalina perfeita acontece quando o resfriamento se dá de forma lenta e gradual, é necessário representar essa etapa de forma que o método de otimização siga o mesmo caminho que o sistema físico real.

Nesse sentido, o SA representa o sistema de resfriamento como o processo, também lento e gradual, de geração de novas soluções que acontece continuamente até que uma temperatura mínima seja alcançada. Ou seja, a temperatura é atualizada a cada nova solução gerada, da seguinte maneira:

$$T = c.T, \quad (3.17)$$

onde  $c$  é uma constante pertencente ao conjunto  $[0,1]$ , responsável pelo

resfriamento do sistema. Quanto maior essa constante, mais lento será o resfriamento e, conseqüentemente, maior será a quantidade de novas soluções geradas em todo o processo.

### 3.3.5 Pseudo-código

Na Tabela 3.3 apresentamos o pseudo-código da meta-heurística *simulated annealing*.

FUNÇÃO = SA (X)

Definir parâmetros:  $t, T, it, c, f$

Gerar uma solução inicial aleatória

ENQUANTO ( $T > t$ ) FAÇA:

PARA  $i = 1$  até  $it$  (quantidade de iterações desejadas)

Calcule uma nova solução:  $X(t + 1) = X(t) - f.R$

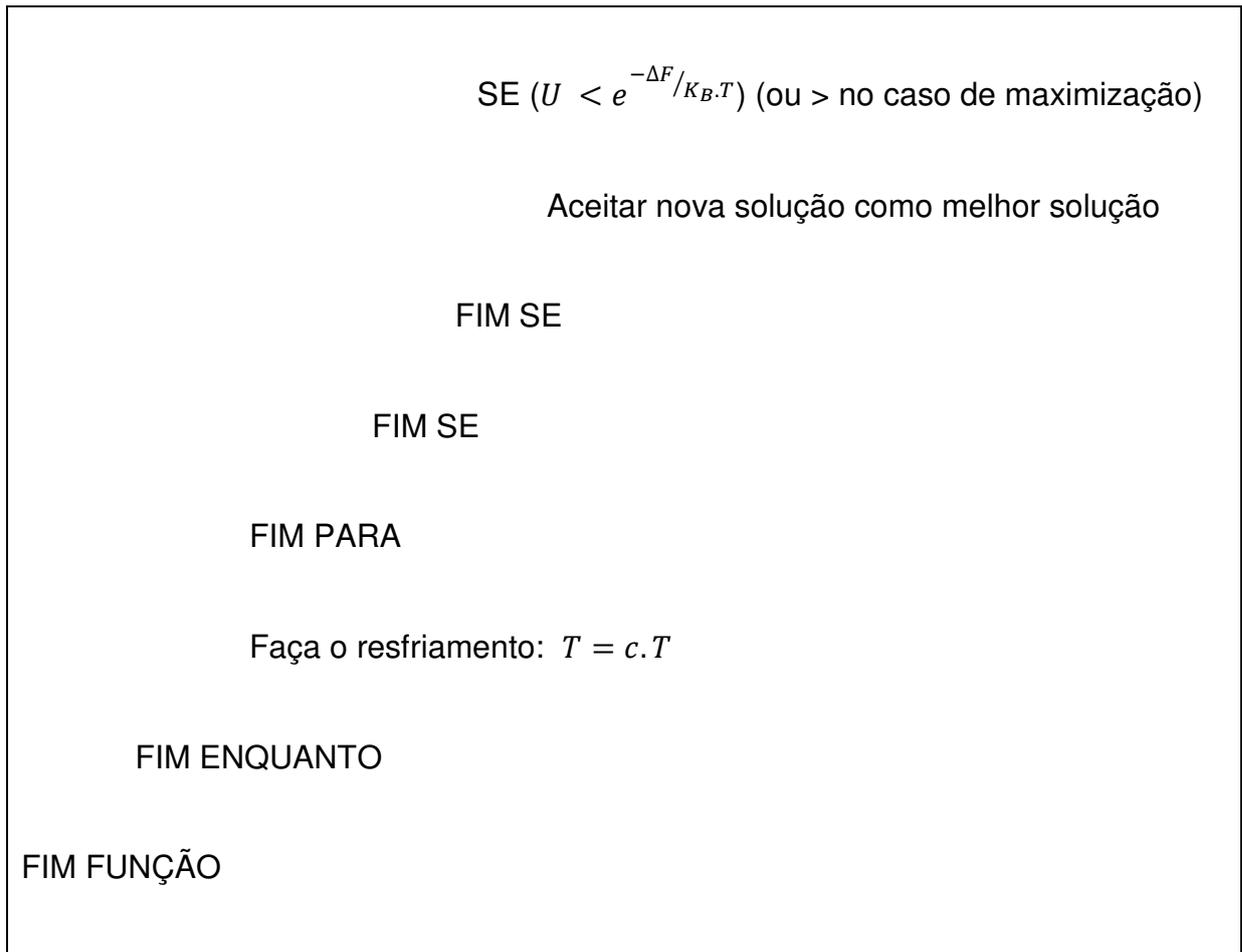
SE ( $F(X(t + 1))$ ) é melhor que  $F(X(t))$

$X(t) = X(t + 1)$

SE NÃO

Verifique se é um passo (solução) aceitável:

- i. Gerar um número aleatório  $U \in [0,1]$
- ii. Avaliar:

Tabela 3.3: Pseudo-código da meta-heurística *Simulated Annealing*

### 3.4 Otimização por *Particle Swarm* (PSO)

Motivados pelo voo sincronizado e resistente à colisão praticado pelos bandos de pássaros, o que era apenas uma simulação de suas coreografias se transformou em um método de otimização de funções contínuas não lineares [79]. Assim como a ACO, a PSO faz parte do conjunto de meta-heurísticas conhecido como inteligência de enxame, explicado anteriormente. Seus autores, Kennedy e Eberhart [79], foram inspirados principalmente pelos trabalhos de Reynolds [80] e Heppner e Grenander [81], que mostraram que cada pássaro mantém um controle da distância de forma que ela seja ótima entre ele e seus vizinhos, mantendo uma relação de sincronia no bando.

A Figura 3.12 fornece uma breve representação do sistema de voo sincronizado de um bando de pássaros no momento em que eles fazem curvas,

observando apenas um integrante do bando. Esse comportamento os ajuda a otimizar a distância entre os pássaros que, nesse meio, significa maximizar as chances de se proteger de predadores, otimizar parâmetros ambientais como a temperatura, e, sobretudo, encontrar alimentos de forma mais rápida [80].

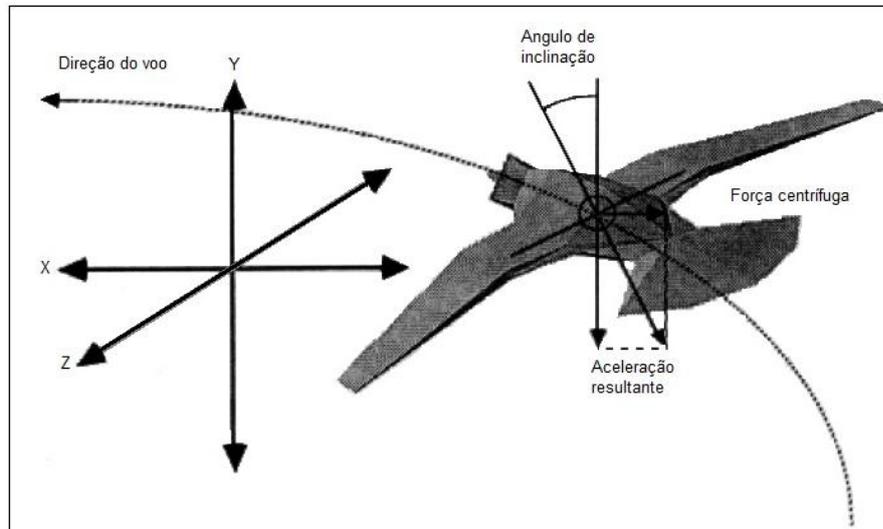


Figura 3.12: Representação do voo sincronizado de um pássaro em relação ao seu bando [80].

A PSO é uma meta-heurística de baixa dificuldade de entendimento e implementação, que mostrou estabilidade e boa convergência em diversos casos testados [82]. Em resumo, a busca pelo ponto ótimo acontece em três etapas:

- i. Na primeira etapa é feita a geração aleatória da população de soluções iniciais, e suas respectivas posições;  
Vale destacar que cada solução representa um pássaro e é chamada de partícula. Considerando, assim, que a velocidade de cada partícula é utilizada como elemento de perturbação nas tabelas de posição.
- ii. Na segunda etapa, uma nova solução é obtida segundo uma fórmula específica que será apresentada mais adiante. Essa etapa é chamada de atualização da velocidade e da posição, pois faz referência ao comportamento dos pássaros em relação aos seus vizinhos em pleno voo; Nesse momento são consideradas as posições e velocidades atuais, aquelas que resultaram na melhor posição global do bando, e aquelas que resultaram na melhor

posição individual de cada partícula.

- iii. Já na terceira etapa, a nova solução é avaliada segundo a função objetivo que está sendo otimizada, podendo, ou não, assumir o papel de melhor posição global até o momento.

As etapas II e III se repetem até que algum critério de parada seja atingido.

Assim, podemos observar que cada pássaro (ou partícula, como foram chamados pelos autores do método) voa em uma determinada velocidade e se move gradualmente para a posição ótima. Cada partícula representa uma solução potencial da população e sua velocidade é calculada em relação às outras partículas, sendo a principal responsável pela perturbação de cada solução (posição da partícula) em busca da posição ótima (ponto ótimo global do problema) no espaço de busca [83].

#### 3.4.1 Geração da população de soluções iniciais

Para que esse método se adeque e possa ser utilizado como busca do ponto ótimo do problema que está sendo analisado, é preciso definir, antes de mais nada, os parâmetros que fazem parte de sua linha de cálculo. São eles: o número de iterações desejadas (*it*) ou outro critério de parada, duas constantes positivas ( $c_1$  e  $c_2$ ) que serão utilizadas no cálculo de novas soluções, o tamanho da população de soluções ( $b$ ), e o peso de inércia ( $\omega$ ).

Com relação a  $\omega$ , é importante explicar que foi um parâmetro adicionado ao método PSO apenas mais tarde por Shi e Eberhart [84] com o intuito de equilibrar as buscas nos meios local e global de forma mais proveitosa e eficaz. De acordo com os autores dessa alteração, para chegar ao ponto ótimo com menos iterações possíveis, o valor ideal de  $\omega$  seria entre 0,9 e 1,2. Outra possibilidade seria começar por um valor mais alto, como 1,4, e fazê-lo decrescer no decorrer do tempo.

Finalmente, os dados de entrada estão dispostos de acordo com a estrutura apresentada na Figura 3.13. Nesta figura, os vetores  $X$  representam as posições e os vetores  $V$  representam as velocidades. Vale lembrar que, inicialmente, as melhores posições de cada vetor são iguais às posições iniciais, e que a melhor

solução global é aquela, dentre todas as posições iniciais, que possui o melhor  $F(X)$ .

$$\begin{array}{|cccccccc|} \hline x_{1,1} & x_{1,2} & \dots & x_{1,n} & v_{1,1} & v_{1,2} & \dots & v_{1,n} \\ \hline x_{2,1} & x_{2,2} & \dots & x_{2,n} & v_{2,1} & v_{2,2} & \dots & v_{2,n} \\ \hline \dots & \dots \\ \hline x_{N,1} & x_{N,2} & \dots & x_{N,n} & v_{N,1} & v_{N,2} & \dots & v_{N,n} \\ \hline \end{array}$$

Figura 3.13: Entrada de soluções aleatórias (partículas) e suas respectivas velocidades.

### 3.4.2 Atualização da posição e da velocidade

Essa etapa, apesar de sua extrema importância, é considerada como uma parte de fácil entendimento e aplicação. É por meio dela que uma nova solução é gerada para cada partícula da população de soluções, enquanto o critério de parada não é atingido. Consideremos então a seguinte equação:

$$v_{ij}(t+1) = \omega \cdot v_{ij}(t) + c_1 \cdot r_1 \cdot (P_{ibj}(t) - x_{ij}(t)) + c_2 \cdot r_2 \cdot (G_{ib}(t) - x_{ij}(t)), \quad (3.18)$$

onde  $v_{ij}(t+1)$  é a nova velocidade da partícula  $j$  na dimensão  $i$ ,  $r_1$  e  $r_2$  são números aleatórios uniformes com valores entre zero e um,  $P_{ibj}$  é a melhor posição já obtida pela partícula  $j$  na dimensão  $i$ , e  $G_{ib}$  é a melhor posição já obtida por todo o grupo de partículas na dimensão  $i$ . Com relação aos parâmetros, é preciso acrescentar que  $c_1$  e  $c_2$  podem ser consideradas como funções que ponderam os termos obtidos na equação 3.19. Em seu trabalho, Kennedy e Eberhart [79], mostram que  $c_1$  e  $c_2$ , em muitos casos, tem valores ideais no ponto  $c_1 = c_2 = 2$ .

Calculada a velocidade, resta apenas a atualização da posição da partícula, obtida da seguinte maneira:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (3.19)$$

onde  $x_{ij}(t+1)$  é a nova posição.

### 3.4.3 Análise da nova solução

Como esse método faz uso de diversos pontos da função objetivo em sua base de cálculo, é preciso que esses valores estejam corretamente atualizados. A atualização da melhor posição individual deve ser feita a cada geração de uma nova solução, considerando a seguinte regra de decisão:

$$P_{b_j}(t+1) = \begin{cases} P_{b_i}(t) & F(x_1(t+1), x_2(t+1), \dots, x_n(t+1)) < P_{b_i}(t) \\ x_i(t+1) & F(x_1(t+1), x_2(t+1), \dots, x_n(t+1)) \geq P_{b_i}(t) \end{cases} \quad (3.20)$$

Já o ponto ótimo global, que deve ser atualizado ao final de cada iteração, pode ser calculado de diversas formas, segundo sua vizinhança ou ponto de referência escolhido. A abordagem seguida em nosso trabalho implica que a melhor solução global para cada iteração é aquela que apresenta o melhor resultado na função objetivo, entre os pontos ótimos locais, da seguinte maneira:

$$G_b(t) = \max \{F(P_{b_1}(t)), F(P_{b_2}(t)), \dots, F(P_{b_n}(t))\}. \quad (3.21)$$

### 3.4.4 Pseudo-código

Na Tabela 3.4 apresentamos o pseudo-código da meta-heurística *particle swarm*.

FUNÇÃO = PSO (X)

Definir parâmetros:  $b, c_1, c_2, \omega, it$

Gerar população de “ $b$ ” soluções iniciais aleatórias

ENQUANTO (critério de parada não for atingido) FAÇA

PARA  $i = 1$  até  $b$

PARA  $j = 1$  até  $D$  (dimensão do vetor solução)

(i) Calcule a velocidade de adaptação:

$$v_{ij}(t + 1) = \omega \cdot v_{ij}(t) + c_1 \cdot r_1 \cdot (P_{ib_j}(t) - x_{ij}(t)) + c_2 \cdot r_2 \cdot (G_{ib}(t) - x_{ij}(t))$$

(ii) Atualize a posição:  $x_{ij}(t + 1) = x_{ij}(t) + v_{ij}(t + 1)$

FIM PARA

Avalie a nova solução (em nível local)

SE  $(X_j(t + 1))$  é melhor que  $P_{bj}$

$$P_{bj} = X_j(t + 1)$$

FIM SE

FIM PARA

Avalie todas as soluções e estabeleça a melhor solução global ( $G_b$ )

FIM ENQUANTO

FIM FUNÇÃO

Tabela 3.4: Pseudo-código da meta-heurística de otimização por *Particle Swarm*

## CAPÍTULO 4

### RESULTADOS

Para testar a aplicação de cada meta-heurística ao problema de separação de sinais, elaboramos um processo onde, em um primeiro momento, elas são implementadas considerando um cenário onde conhecemos as fontes e seu processo de mistura, o problema supervisionado. Em seguida, testamos esses ajustes resolvendo um problema não supervisionado, cuja solução também pode ser obtida mediante aplicação do método gradiente. É importante lembrar que esse último caso foi resolvido considerando as abordagens máxima verossimilhança e critério infomax, apresentadas nas Seções 2.5 e 2.6, respectivamente.

Para todos esses casos, a base de dados foi gerada da seguinte forma:

- Primeiramente foi gerada uma matriz aleatória, com duas mil colunas representando as amostras, onde os dados possuem média zero e cada linha representa um sinal, sendo eles  $s_1$  e  $s_2$ . O resultado dessa aplicação pode ser observado na Figura 4.1, onde está representada a distribuição conjunta dessas fontes.

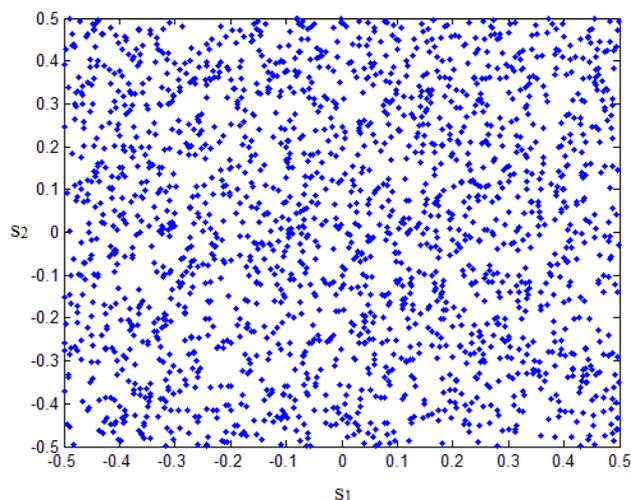


Figura 4.1: Distribuição conjunta das fontes

- Em seguida, os dados gerados anteriormente foram multiplicados, de acordo com o modelo de mistura linear instantâneo,  $X = AS$ , conforme

explicado na Seção 2.4, pela matriz de mistura:

$$A = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}. \quad (4.1)$$

Tal operação gera um conjunto de misturas  $X$ , cuja distribuição conjunta pode ser observada na Figura 4.2.

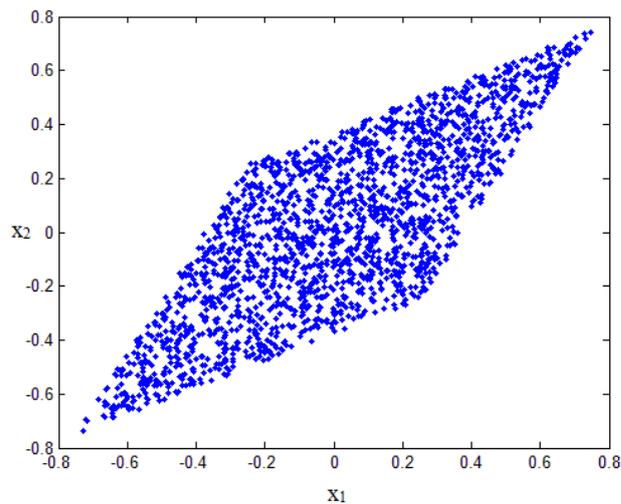


Figura 4.2: Distribuição conjunta das misturas

Considerando essa estrutura de dados, os resultados alcançados serão apresentados, em um primeiro momento, graficamente, com ênfase nas seguintes informações:

- i. Evolução da função contraste ( $F(W)$ ) em função do ponto ótimo detectado (o que chamamos de curva de convergência em nosso trabalho), considerando 100 simulações seguidas;
- ii. Distribuição de probabilidade conjunta das estimativas das fontes ( $Y$ ), considerando apenas uma simulação representativa.

Em seguida, o tempo médio, considerando 100 simulações seguidas, gasto por cada meta-heurística até sua convergência também foi apresentado como uma forma de reforçar a base de dados para comparação entre os métodos estudados, quando se trata de sua eficiência.

#### 4.1. Problema supervisionado

O nosso propósito, ao aplicar as meta-heurísticas em um caso supervisionado, é de potencializar a implementação dos métodos de otimização, resolvendo um problema cuja solução é conhecida. Dessa forma, os parâmetros de cada método poderão ser ajustados até chegarmos à solução desejada. Afinal, é importante destacar que, normalmente, os parâmetros são ajustados com base em tentativa e erro por não possuírem nenhuma regra de ajuste pré-estabelecida.

Para tanto, assumimos como critério de separação a função conhecida como norma de Frobenius, dada por:

$$J = \|E\|_p = \left( \sum_{i=1}^m \sum_{j=1}^n |e_{ij}|^p \right)^{1/p}, \quad (4.2)$$

onde  $E = Y - S$  representa o erro, e  $p = 2$ . Além disso, como já sabemos,  $Y$  é a estimativa das fontes, obtida a partir da operação  $Y = WX$ , onde  $W$  é a matriz de separação obtida no processo de otimização. Nesse sentido, a matriz de separação que minimiza essa função é dada pela inversa da matriz de mistura,  $A$ , sendo:

$$W = \begin{bmatrix} 1.3333 & -0.6667 \\ -0.6667 & 1.3333 \end{bmatrix}. \quad (4.3)$$

Dessa forma, apresentamos a seguir os resultados obtidos da aplicação de cada meta-heurística a esse problema.

##### 4.1.1. *Simulated Annealing*

Antes de apresentar os resultados obtidos com a aplicação dessa meta-heurística ao caso supervisionado, consideramos os ajustes dos seguintes parâmetros desse método:

- Temperatura mínima ( $t$ ) = 500;

- Temperatura máxima ( $T$ ) = 1000;
- Constante de resfriamento ( $c$ ) = 0,95;
- Fator de ponderação da matriz aleatória ( $f$ ) = 0,003;
- Número de iterações ( $it$ ) = 9000.

Na Figura 4.3, apresentamos a curva de convergência em função da matriz de separação, considerando a aplicação repetida do método ao mesmo problema (situação (a)), e a distribuição conjunta das estimativas das fontes (situação (b)), considerando apenas uma simulação representativa. A distribuição conjunta das estimativas representa com fidelidade aquela das fontes. Além disso, nota-se que em todas as simulações feitas há convergência no ponto de mínimo global. No entanto, na maioria das vezes, o método precisa de, no mínimo, 1000 iterações para convergir nesse ponto, para uma média de tempo de convergência de 5,45 segundos:

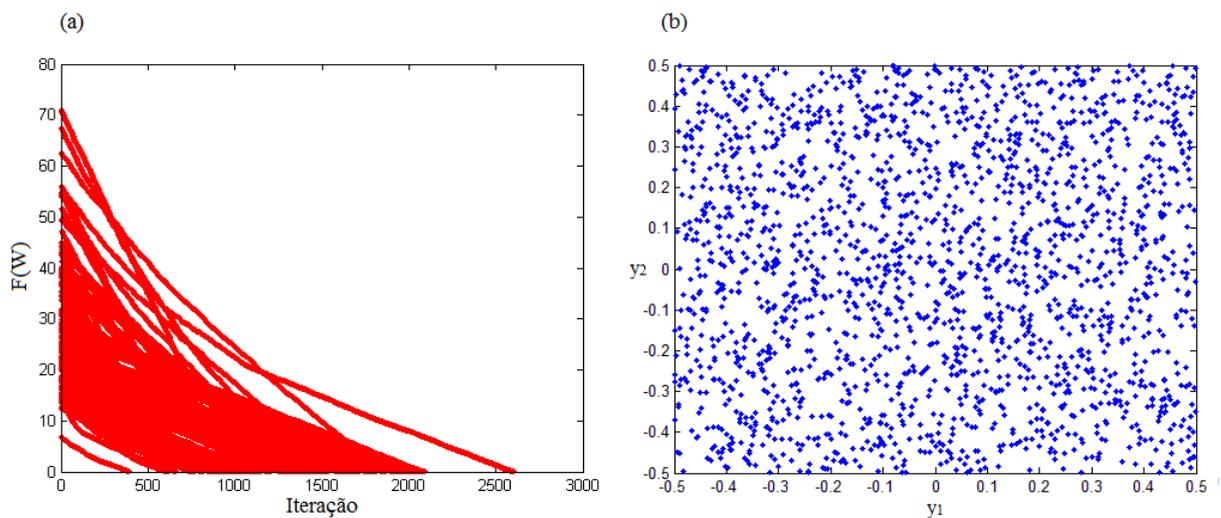


Figura 4.3: (a) Curva de convergência de  $F(W)$  (com simulações);  
 (b) Distribuição conjunta das estimativas das fontes (uma simulação).

#### 4.1.2. Evolução Diferencial

A evolução diferencial, cuja base teórica foi apresentada na Seção 3.1, é uma meta-heurística que usa poucos parâmetros em seu sistema de busca. Suas principais regras de atualização de  $W$ , mutação e *crossover*, foram processadas

segundo o seguinte ajuste de parâmetros:

- Tamanho da população de soluções ( $B$ ) = 10;
- Fator de ponderação ( $f$ ) = 1,2;
- Fator de *crossover* ( $k$ ) = 2;
- Número de iterações ( $it$ ) = 500.

Os resultados obtidos da aplicação dessa meta-heurística, apresentados na Figura 4.4, são a evolução da curva de convergência da função contraste em função da matriz de separação  $F(W)$  para cada repetição feita (situação (a)), e a distribuição conjunta das estimativas das fontes (situação (b)). Podemos observar que, nesse caso, a evolução diferencial alcança uma convergência consideravelmente rápida, em menos de trezentas iterações, para um tempo médio de convergência de 2,42 segundos, e que a distribuição conjunta das estimativas representa com exatidão aquela das fontes.

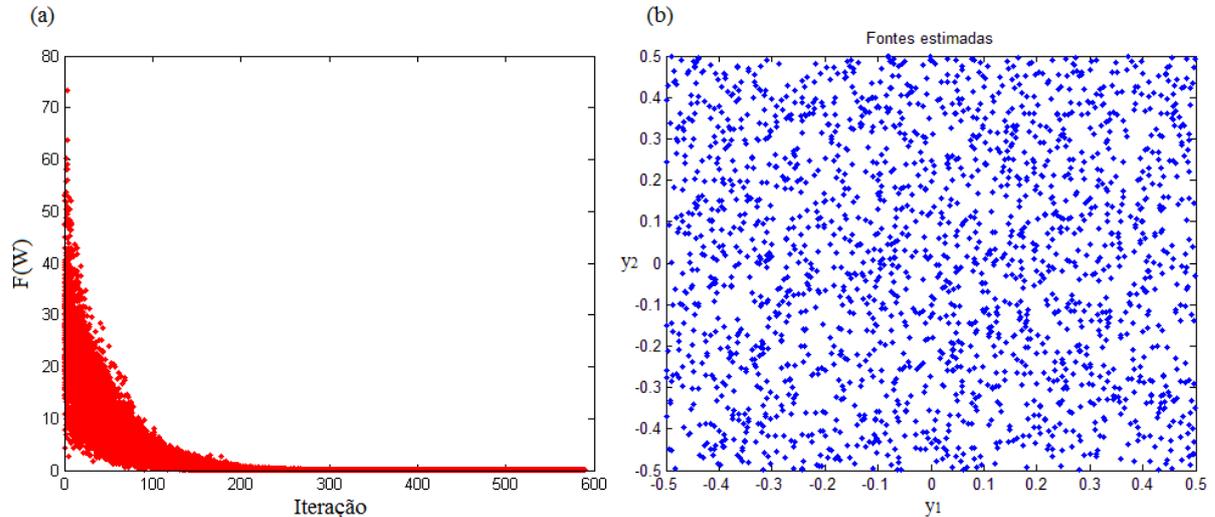


Figura 4.4: (a) Curva de convergência de  $F(W)$  (com simulações);  
(b) Distribuição conjunta das estimativas das fontes (uma simulação).

#### 4.1.3. Otimização por Colônia de Formigas para Domínios Contínuos

Em otimização por colônia de formigas para domínios contínuos, apesar de toda a complexidade envolvida na fase de implementação e de cálculos, e da

importante quantidade de variáveis envolvidas, há apenas cinco parâmetros a serem ajustados. No presente trabalho, o ajuste final desses parâmetros para o problema supervisionado ficou da seguinte forma:

- Tamanho da população de soluções iniciais ( $k$ ) = 30;
- Número de iterações desejadas ( $it$ ) = 2000;
- Fator multiplicador do desvio padrão ( $\varepsilon$ ) = 0.95;
- Fator de cálculo dos pesos de cada solução ( $q$ ) = 6000;
- Quantidade de novas soluções desejadas ( $m$ ) = 18.

Ao aplicar essa meta-heurística, considerando os parâmetros acima, temos os seguintes resultados: a curva de convergência para cada uma das cem repetições feitas, como mostra a Figura 4.5 (situação (a)), e a distribuição conjunta das estimativas (situação (b)), onde temos uma fiel representação das fontes. Já as curvas de convergência mostram, claramente, uma tendência aproximadamente linear, homogênea e estável, levando à convergência em menos de mil iterações. Apesar disso, o tempo médio de convergência gasto por este método é de 10,53 segundos. O que, comparado aos outros métodos, parece um valor bastante elevado.

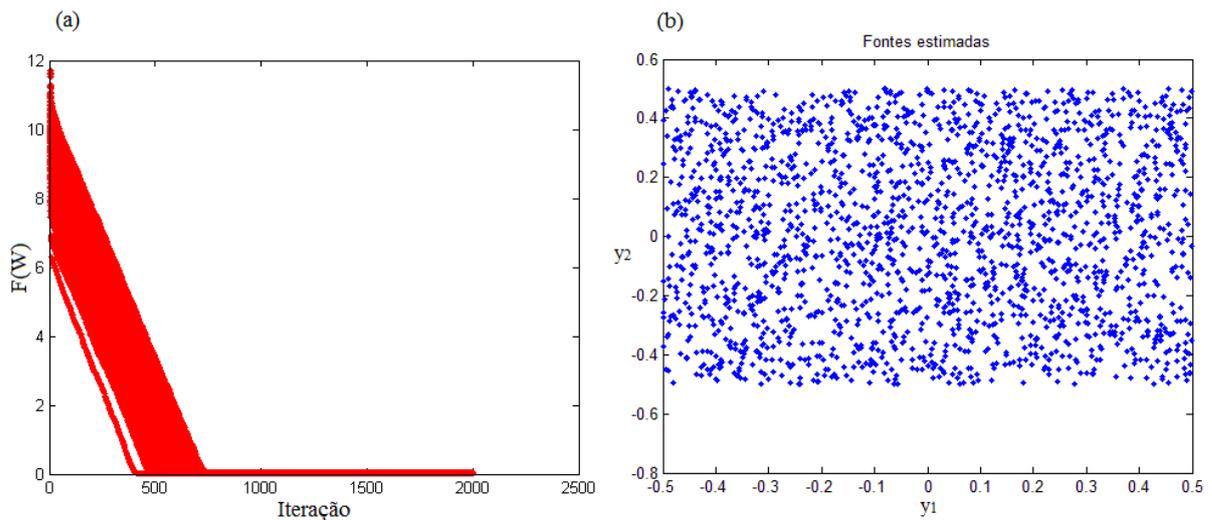


Figura 4.5: (a) Curva de convergência de  $F(W)$  (100 simulações);  
 (b) Distribuição conjunta das estimativas das fontes (uma simulação).

#### 4.1.4. Otimização por *Particle Swarm*

Para finalizar as aplicações das meta-heurísticas no caso supervisionado, apresentaremos os resultados obtidos com a implementação da otimização por *particle swarm*. É importante lembrar que sua principal regra de atualização de  $W$ , apresentada na Seção 3.4 deste trabalho, conta com a definição de alguns parâmetros, ajustados para esse problema pontual da seguinte forma:

- Número de iterações desejadas ( $it$ ) = 300;
- Constante que multiplica a melhor solução individual ( $c_1$ ) = 0,2;
- Constante que multiplica a melhor solução populacional ( $c_2$ ) = 0,2;
- Tamanho da população de soluções ( $b$ ) = 9;
- Peso de inércia ( $\omega$ ) = 0,9.

Podemos observar a seguir que a aplicação da otimização por *particle swarm* segue a mesma tendência dos outros métodos estudados neste trabalho, ou seja, são capazes de gerar estimativas das fontes cuja estrutura conserva as mesmas características que as fontes originais. Tanto essa situação como o fato do *particle swarm* ser um método bastante eficaz podem ser verificadas por meio dos gráficos apresentados na Figura 4.6 (situação(a)) e (situação(b)) e confirmadas pelo tempo de convergência médio de 1,15 segundos.

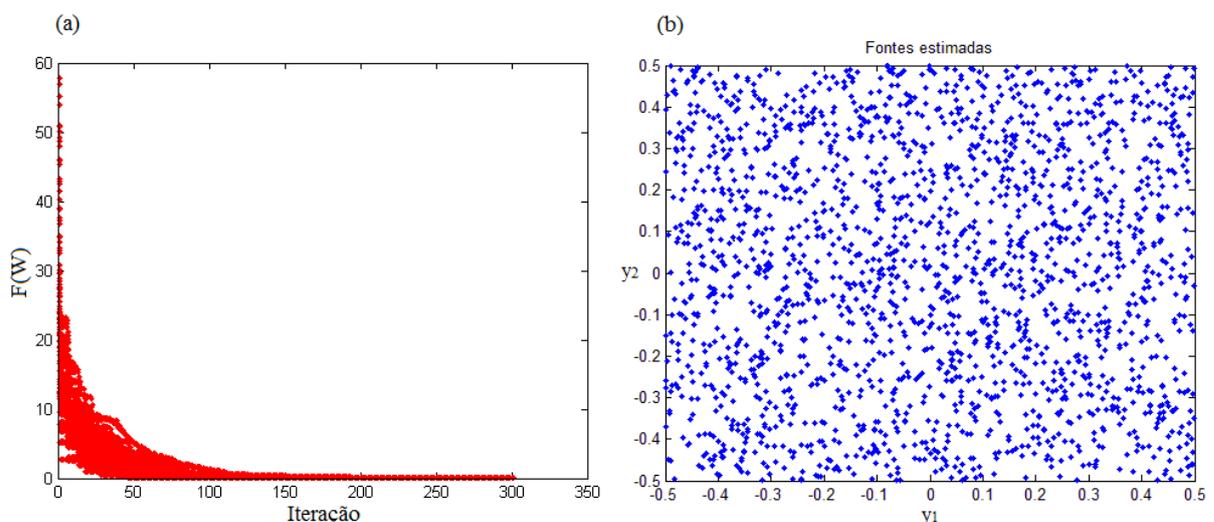


Figura 4.6: (a) Curva de convergência de  $F(W)$  (100 simulações);  
 (b) Distribuição conjunta das estimativas das fontes (uma simulação).

## 4.2. Problema não supervisionado

Outra etapa muito importante no processo de implementação das meta-heurísticas consiste em testá-las e reajustar seus parâmetros por meio de sua aplicação em um problema de separação de sinais cego (não supervisionado). A especificidade desse teste se encontra em como os dados serão tratados. Os cálculos serão feitos desconsiderando a matriz de misturas e as fontes geradas inicialmente, pois em um problema de separação cega, os únicos dados disponíveis, normalmente, são as misturas das fontes.

Nesse caso, vamos usar o critério *Infomax* como base da função contraste que descreve a independência estatística dos dados, como explicado na Seção 2.6. Para tanto, a função não linear que será usada é a seguinte:

$$g = \frac{-(Y)^4}{4}, \quad (4.4)$$

onde,  $Y = WX$ . Temos, assim, o seguinte critério de separação:

$$J = E \left\{ \sum_{i=1}^N \log \left( \frac{-(W_i X)^4}{4} \right) \right\} + \log(|\det(W)|). \quad (4.5)$$

Devemos ressaltar, no entanto que, embora a base de dados deste problema seja a mesma do caso supervisionado, os resultados obtidos podem não ser os mesmos devido à alteração da função objetivo, que agora passa a ter a função contraste em sua base de cálculo.

Além disso, uma particularidade em separação cega de fontes, que deve ser lembrada nessa etapa, é a possibilidade de validar a solução encontrada por meio da multiplicação da matriz de mistura ( $A$ ) pela matriz de separação ( $W$ ) encontrada no processo de otimização. Essa multiplicação deve resultar em uma matriz diagonal (a menos de um ganho de escala e de uma possível permutação), partindo do princípio de que as matrizes  $A$  e  $W$  são regidas pelas propriedades matemáticas relacionadas a uma matriz e sua inversa.

Apesar de essa propriedade ter sido usada ao final de cada simulação, como forma de verificar se houve convergência do método, os resultados finais

obtidos em nossos estudos, no caso não supervisionado, são apresentados com base na relação sinal-interferente (SIR, do inglês *signal-to-interference ratio*), em um quadro comparativo entre todas as técnicas de separação utilizadas, considerando a média obtida em 100 simulações seguidas.

A SIR é uma função, comumente usada para avaliar numericamente a proximidade entre as estimativas e as fontes. Assim, podemos dizer que essa função pode ser usada como uma medida da convergência de cada método. A SIR é definida da seguinte maneira:

$$SIR_i = 10 \log \left( \frac{E\{s_i(n)^2\}}{E\{(s_i(n) - y_i(n))^2\}} \right), \quad (4.6)$$

onde,  $s_i$  representa a fonte  $i$ , e  $y_i$  representa a estimativa dessa fonte. Note que, por ser uma medida logarítmica do inverso do erro quadrático médio, quanto maior for o resultado dessa aplicação, mais eficaz terá sido o método de otimização aplicado.

É importante ainda nessa etapa, considerar as interferências de certas ambiguidades quando se trata de análise de componentes independentes, como explicamos na Seção 2.4. A ambiguidade de permutação pode fazer com que a matriz de separação tenha seus componentes desordenados entre si, provocando a existência de mais de uma possibilidade de solução do problema. Um exemplo do efeito dessa ambiguidade, em um problema de separação cega que envolve duas fontes pode ser observado na seguinte equação:

$$W = \begin{bmatrix} a & b \\ b & a \end{bmatrix} \text{ ou } W = \begin{bmatrix} b & a \\ a & b \end{bmatrix}. \quad (4.7)$$

Onde  $a$  e  $b$  são as constantes que resolvem o problema de otimização. Portanto, antes do cálculo da SIR, torna-se necessário determinar se houve permutação, além de readequar a escala das estimativas das fontes à escala das fontes originais.

Definido o problema, resta agora, apresentar os resultados obtidos com a implementação das meta-heurísticas ao caso não supervisionado. Antes disso, no entanto, apresentaremos os resultados obtidos com a aplicação do método gradiente, principal base de comparação em nosso estudo.

#### 4.2.1. Método Gradiente

Os parâmetros de cálculo do método gradiente, apresentado na Seção 2.7, foram definidos com base em tentativa e erro, começando por um valor suficientemente pequeno até que o método convergisse. Assim, temos:

- Número de iterações ( $it$ ) = 5000;
- Passo de adaptação do gradiente ( $\mu$ ) = 0,09;

Os resultados obtidos, curva de convergência de  $F(W)$  e distribuição conjunta das estimativas das fontes, podem ser observados a seguir na Figura 4.7, (situação (a)) e (situação (b)), respectivamente.

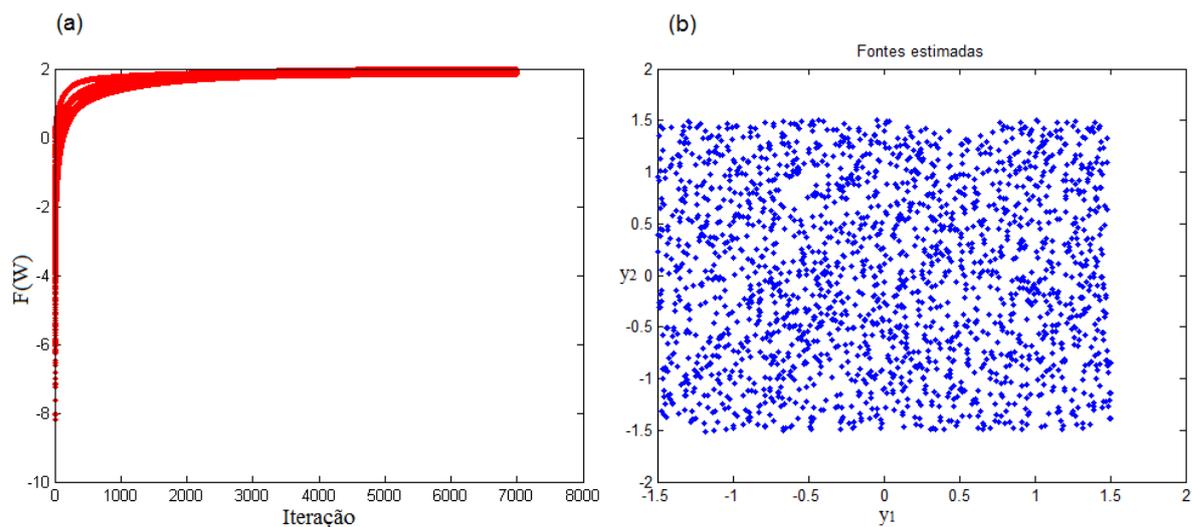


Figura 4.7: (a) Curvas de convergência de  $F(W)$  (com simulações);  
(b) Distribuição conjunta das estimativas das fontes (uma simulação).

Podemos verificar que o método gradiente pode oferecer, nesse caso, um resultado aceitável, principalmente, tendo como referência sua capacidade de recuperação das fontes apresentada no gráfico de distribuição conjunta das fontes. É importante ressaltar que, para chegar a tais resultados, foram necessários 9,41 segundos, em média, para que o método alcançasse a convergência, como esperado.

#### 4.2.2. Evolução Diferencial

A evolução diferencial foi uma das meta-heurísticas que mais sofreram alterações em seus parâmetros quando comparada com sua aplicação no caso supervisionado. Não há uma explicação exata para esse fato, mas conjectura-se que isso tenha ocorrido devido à possível ambiguidade das matrizes de separação, apresentada anteriormente. Logo adiante explicaremos de forma mais detalhada os efeitos desse problema e o que foi feito para mitigá-los. Assim, temos:

- Tamanho da população de soluções ( $B$ ) = 100;
- Fator de ponderação ( $f$ ) = 0,98;
- Fator de *crossover* ( $K$ ) = 1;
- Número de iterações ( $it$ ) = 1600.

É importante salientar que o elevado número de iterações se deve ao fato de a população de soluções também ter aumentado consideravelmente, alcançando um tamanho dez vezes maior que no caso supervisionado. Essa, porém, foi a medida usada para contornar o problema da ambiguidade que, por sua vez, é uma característica da separação cega de fontes que implica, indiretamente, na existência de mais de uma matriz de separação como solução para o problema, como explicamos anteriormente.

No processo de otimização, os componentes da população de soluções, tendem a se concentrar em diferentes regiões ótimas após certo tempo. Na evolução diferencial, durante o processo de busca por novas soluções, a regra de atualização da matriz de separação se baseia na manipulação direta de tais componentes. Entretanto, a combinação de vetores de regiões provavelmente opostas pode gerar resultados consideravelmente distorcidos e fora do cenário desejado; o feito de aumentar o tamanho da população de soluções amplia o campo de busca da meta-heurística, gerando mais possibilidades de recombinações entre os vetores. Com isso, o ponto ótimo global pode ser encontrado antes que os componentes se concentrem em regiões completamente opostas umas das outras.

Diante dessa alteração de parâmetros, os resultados nos mostram que foram necessárias aproximadamente 1600 iterações, cinco vezes mais que no caso

supervisionado, e um tempo médio de 45, 67 segundos para que o método convergisse em um ponto ótimo global. Ainda assim, a otimização por evolução diferencial pode ser considerada uma técnica estável e eficaz. Como prova, sua curva de convergência e distribuição conjunta das estimativas das fontes podem ser observadas na Figura 4.8, (situação (a)) e (situação (b)) respectivamente.

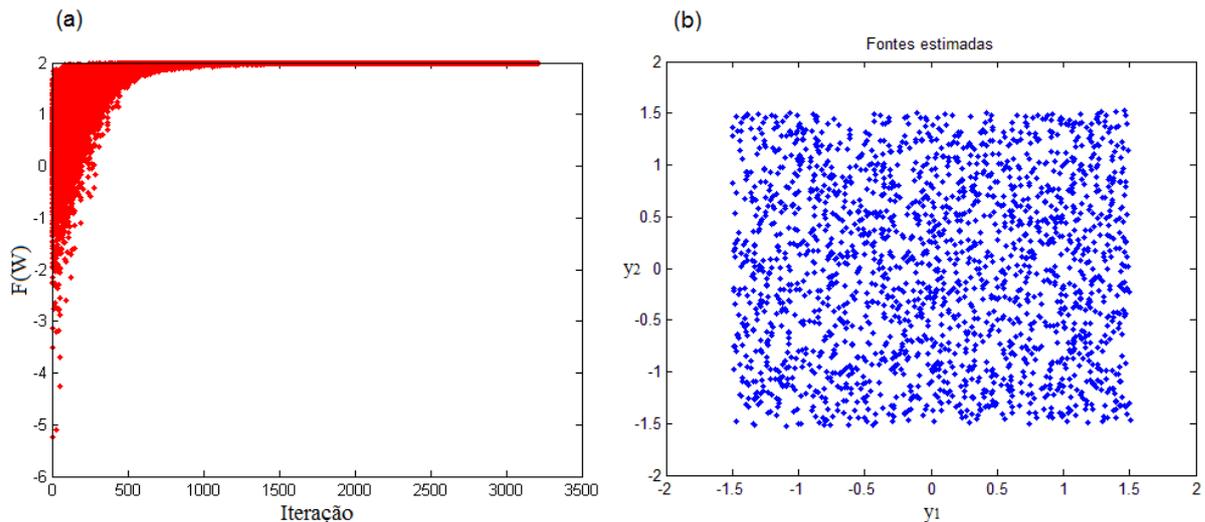


Figura 4.8: (a) Curva de convergência de  $F(W)$  (com simulações);  
(b) Distribuição conjunta das estimativas das fontes (uma simulação).

Vale lembrar que a ambiguidade nos resultados pode acometer também outras meta-heurísticas, porém com menos intensidade que no caso da evolução diferencial.

#### 4.2.3. *Simulated Annealing*

Os ajustes dos parâmetros para aplicação do *simulated annealing*, quando comparados com os dados do problema supervisionado, não sofreram grandes mudanças, como mostramos a seguir:

- Temperaturas mínima ( $t$ ) e máxima ( $T$ ) = [500,1000];
- Constante de resfriamento ( $c$ ) = 0,95;
- Fator de ponderação da matriz aleatória ( $f$ ) = 0,005;
- Número de iterações ( $it$ ) = 5000.

Os resultados da aplicação desse método, Figura 4.9, apontam que as curvas de convergência apresentam uma leve divergência entre si (situação (a)), quando comparadas a outros métodos. Porém, o processo de otimização manteve uma boa estabilidade e o resultado final recuperou, como esperado, a mesma forma das fontes, como podemos observar na distribuição conjunta das estimativas das fontes (situação (b)). O tempo médio, gasto por esta meta-heurística para alcançar a convergência foi de 3,45 segundos.

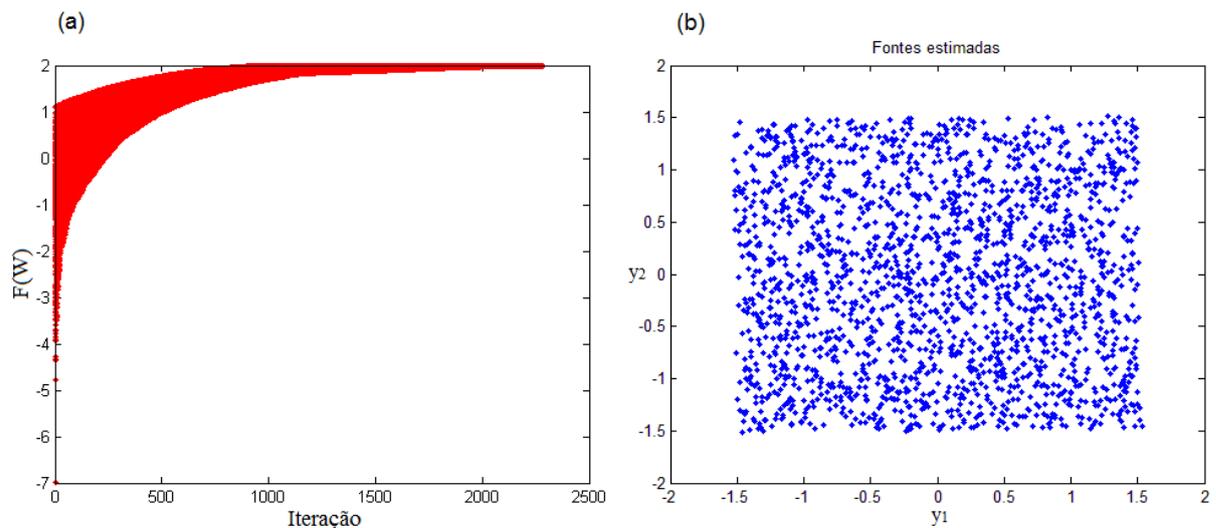


Figura 4.9: (a) Curva de convergência de  $F(W)$  (com simulações);  
(b) Distribuição conjunta das estimativas das fontes (uma simulação).

#### 4.2.4. Otimização por Colônia de Formigas para domínios contínuos

A otimização por colônia de formigas para domínios contínuos mostrou-se eficaz e bastante eficiente quando aplicada ao problema não supervisionado. Chama a atenção, nesse caso, o fato de o número de iterações ter sido reduzido a mais da metade de seu valor definido no caso supervisionado para uma população de soluções, pelo menos, duas vezes maior, sendo:

- Tamanho da população de soluções iniciais ( $k$ ) = 80;
- Número de iterações desejadas ( $it$ ) = 800;
- Fator multiplicador do desvio padrão ( $\varepsilon$ ) = 0.9;
- Fator de cálculo dos pesos de cada solução ( $q$ ) = 200;

- Quantidade de novas soluções desejadas ( $m$ ) = 40.

Observando a Figura 4.10, podemos perceber que o método apresenta boa estabilidade em seu processo de busca. Isso pode ser confirmado pelas curvas de convergência da ACO<sub>R</sub> (situação (a)), onde todos os testes realizados resultam em curvas semelhantes e que se sobrepõem na maioria dos casos. Já a distribuição conjunta das estimativas das fontes (situação (b)) aponta que as fontes foram recuperadas com fidelidade. Note que a convergência em um ponto ótimo global acontece em aproximadamente 500 iterações. No entanto, o tempo médio gasto para resolver o problema não supervisionado foi de 22,68 segundos.

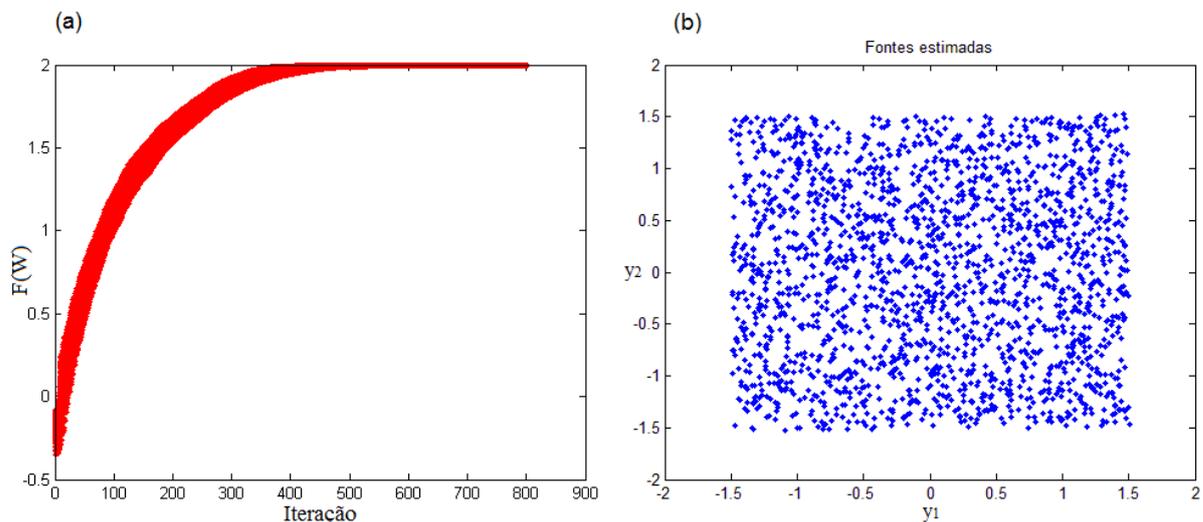


Figura 4.10: (a) Curvas de convergência (cem simulações);  
(b) Distribuição conjunta das estimativas das fontes (uma simulação).

#### 4.2.5. Otimização por *Particle Swarm*

A otimização por *particle swarm*, quando aplicada ao caso não supervisionado, chama a atenção pelo fato de convergir com um número consideravelmente baixo de iterações quando comparada aos demais métodos de otimização que aplicamos ao mesmo problema. Temos, neste caso, os seguintes parâmetros:

- Número de iterações desejadas ( $it$ ) = 60;

- Constante que multiplica a melhor solução individual ( $c_1$ ) = 0,9;
- Constante que multiplica a melhor solução populacional ( $c_2$ ) = 0,9;
- Tamanho da população de soluções ( $b$ ) = 100;
- Peso de inércia ( $\omega$ ) = 0,9.

Na Figura 4.11, podemos observar a curva de convergência da função objetivo (situação (a)), onde ficam nítidas a eficácia e a estabilidade da otimização por *particle swarm*, cuja convergência se dá com menos de 30 iterações na maior parte dos testes realizados, em um tempo médio de 2,71 segundos. Podemos observar também a distribuição conjunta das estimativas das fontes (situação (b)), onde a forma das fontes originais foi recuperada com a devida exatidão proporcionada por esse método de otimização.

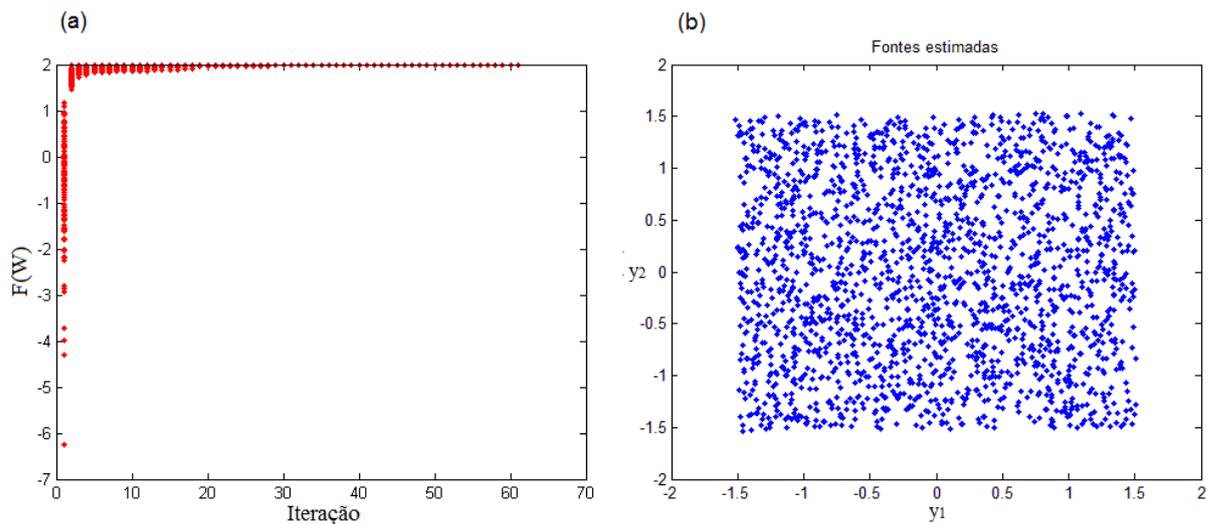


Figura 4.11: (a) Curva de convergência (cem simulações);  
(b) Distribuição conjunta das estimativas das fontes (uma simulação).

Assim como na evolução diferencial, o *particle swarm* pode apresentar problemas de convergência devido à possível ambiguidade das matrizes de separação. O fato de aumentar o tamanho da população de soluções neste caso continua sendo uma boa solução para o problema da ambiguidade e não altera significativamente a velocidade de convergência desse método, pois ele necessita de poucas iterações para convergir.

#### 4.2.6. Resultado da SIR

Com o objetivo de reforçar os testes feitos anteriormente, para o caso não supervisionado, e provar a capacidade de convergência das meta-heurísticas, apresentamos a seguir, na Tabela 4.1, os resultados obtidos com a aplicação da SIR. Vale lembrar que a SIR, como foi retratada no início desse capítulo, é uma medida logarítmica do erro resultante do processo de separação de sinais e, por esse motivo, quanto maior o valor resultante dessa operação, melhor terá sido a solução encontrada.

Método de otimização	Média	Desvio padrão	Menor índice	Maior índice
<b>Gradiente</b>	34,99	13,7	7,002	71,928
<b>ACO<sub>R</sub></b>	37,802	1,43	34,975	41,388
<b>SA</b>	11,557	4,579	3,405	23,504
<b>PSO</b>	40,192	7,699	27,644	58,699
<b>DE</b>	39,053	7,558	26,382	61,977

Tabela 4.1: Resultados da aplicação da SIR (cem simulações).

A partir desses dados, é possível observar que, com exceção do *simulated annealing*, as médias resultantes da aplicação das meta-heurísticas ao problema não supervisionado são muito próximas e com valores significativamente bons. Chama atenção o caso da ACO<sub>R</sub>, cujo desvio padrão da média dos resultados foi menor que os demais em proporções substanciais, provando sua estabilidade e precisão. Já o caso do *simulated annealing* precisa ser melhor investigado, pois a causa de seus resultados se encontrarem distantes da média geral obtida pelas demais meta-heurísticas pode se tratar de um simples ajuste de parâmetros, por exemplo.

## CAPÍTULO 5

### CONCLUSÕES E PERSPECTIVAS

#### 5.1 Considerações gerais

Neste trabalho, procuramos investigar a aplicabilidade de meta-heurísticas ao problema de separação de sinais, considerando o caso de modelos lineares de misturas. Para isso, o desempenho das meta-heurísticas *simulated annealing*, *particle swarm*, evolução diferencial e colônia de formigas para domínios contínuos foi testado em diferentes experimentos numéricos, o que nos permitiu comparar essas técnicas de otimização entre si e em relação ao desempenho do método baseado no uso do gradiente, buscando respeitar suas versões mais originais, sem considerar variações e/ou junções de meta-heurísticas propostas depois.

O estudo inicial de tais meta-heurísticas aplicadas em um problema de separação supervisionado nos permitiu verificar a consistência das implementações realizadas, além de ter sido de grande valia nos ajustes iniciais dos parâmetros de cada uma dessas meta-heurísticas. A importância desses ajustes refletiu diretamente nos testes que foram realizados em seguida, no cenário não supervisionado, no qual o ponto de partida para o ajuste dos parâmetros foi justamente o ajuste estabelecido no caso supervisionado.

Quanto à aplicação das meta-heurísticas ao problema de separação não supervisionado, pudemos constatar a eficácia desses métodos após realização dos experimentos numéricos. A aplicação do gradiente, clássico método de resolução dos problemas de separação de sinais estudados neste trabalho, foi uma forma de comprovar tal pertinência mediante comparação dos resultados obtidos por meio desse e dos outros métodos de otimização. Esses resultados foram analisados com base em um conjunto de gráficos, bem como, levando em consideração uma medida conhecida como relação sinal interferente, que mede, de certo modo, a proximidade entre as estimativas e as fontes originais.

Diante dos resultados obtidos, fica claro que as meta-heurísticas analisadas neste estudo podem ser consideradas como métodos efetivos quando se trata de sua aplicação em problemas de separação cega de fontes, nos casos de

modelos lineares. Devemos ressaltar, no entanto, a dificuldade de ajuste de parâmetros e os problemas de convergência ocasionados pela ambiguidade da técnica de separação utilizada.

A presença das ambiguidades inerentes à separação cega de fontes se mostrou um grande entrave na aplicação das meta-heurísticas, principalmente da evolução diferencial, por dificultar a busca por novas soluções viáveis. Isso acontece porque a combinação de vetores muito divergentes entre si, ou seja, de regiões ótimas diferentes, frequentemente, gera novas soluções que não pertencem a nenhuma dessas regiões. Isso significa que inúmeras vezes novas soluções são descartadas e, portanto não trazem melhorias ao processo de busca pela solução ótima. Esse problema foi contornado por meio da expansão da população de soluções para que houvesse mais vetores para serem combinados antes que essa população se dividisse completamente em regiões ótimas diferentes.

Outro problema observado diz respeito à definição dos parâmetros das meta-heurísticas investigadas, ainda que ajustes iniciais tenham sido possíveis com o caso supervisionado. Isso acontece porque cada tipo de problema pode exigir uma combinação de parâmetros diferente para ser resolvido. Como os ajustes são feitos, com raras exceções, por tentativa e erro, essa tarefa pode se tornar bastante onerosa, na mesma proporção em que cresce o número de parâmetros a serem definidos em uma meta-heurística.

Com relação ao desempenho dos métodos testados, podemos afirmar que, em sua maioria, esses métodos se mostraram efetivos e estáveis o suficiente para serem aconselhados como soluções de otimização mesmo quando a preocupação, quando da resolução de um problema, não recai necessariamente na presença de mínimos locais. Todavia, se tivermos de escolher apenas um dos métodos, devemos ponderar entre precisão, velocidade de convergência e dificuldade de implementação.

Nesse sentido, a otimização por colônia de formigas para domínios contínuos se mostrou consideravelmente precisa, porém de difícil implementação. Por outro lado, o *simulated annealing*, que não se mostrou tão preciso quanto os outros métodos, é o que apresenta menor dificuldade de implementação entre eles. Já as meta-heurísticas evolução diferencial e *particle swarm* podem ser consideradas como métodos de dificuldade moderada de implementação. Entretanto, enquanto esta precisa de pouco tempo para convergir, aquela

apresentou o maior número de iterações e o maior tempo médio gasto até a convergência em um ponto ótimo global.

Devemos ressaltar, no entanto, que, como o foco dos nossos estudos não se dirige a excelência dos códigos elaborados, é possível que algum ganho de tempo seja alcançado em termos de técnicas avançadas de programação. A medida de tempo em nossos estudos foi usada apenas como um reforço à base comparativa entre as diferentes técnicas de otimização, sabendo que foi usada a mesma técnica de programação para todas elas.

Finalmente, vale ressaltar que o presente estudo se trata de uma análise de meta-heurísticas de diversas origens, que formam um conjunto consideravelmente representativo no âmbito de otimização por meta-heurísticas. Assim, é razoável afirmar que esses métodos, de maneira geral, ou seja, mesmo aqueles que não foram avaliados em nosso trabalho são, possivelmente, igualmente aplicáveis ao problema de separação cega de fontes.

## 5.2 Perspectivas

As perspectivas para a continuidade deste trabalho recaem em quatro pontos principais:

- Como não houve confirmação da presença de mínimos locais nos problemas resolvidos em nosso trabalho, seria interessante continuar a busca por um problema que possua, comprovadamente, mínimos locais, onde as meta-heurísticas seriam novamente testadas;
- Pouco serviria a teoria se não pudesse ser colocada em prática. A aplicação das meta-heurísticas em problemas reais permitiria uma visão concreta do problema nessa perspectiva;
- Ainda que o conjunto de meta-heurísticas avaliadas em nosso trabalho seja consideravelmente representativo, a análise de outras meta-heurísticas ou de outras versões das meta-heurísticas já estudadas seria uma oportunidade de inserir, definitivamente, essas técnicas de otimização no domínio da separação de sinais.
- Como o ajuste de parâmetros foi uma das maiores dificuldades

encontradas no processo de implementação das meta-heurísticas, seria importante fazer uma análise de sensibilidade dos parâmetros. Tal análise seria completa se fosse feita não somente para problemas de separação de sinais, mas de um modo geral, da forma mais abrangente possível. Assim, seria possível entender quais os possíveis efeitos da perturbação dos parâmetros em termos de precisão e velocidade de convergência dos métodos, por exemplo. Seria também possível investigar se o resultado obtido com a aplicação do *simulated annealing* poderia ser melhorado com a alteração de seus parâmetros para algum valor ou conjunto de valores específico.

## REFERÊNCIAS

- [1] R. R. F. Attux. *Novos paradigmas para equalização e identificação de canais baseados em estruturas não-lineares e algoritmos evolutivos*. Tese de Doutorado. Universidade Estadual de Campinas. Brasil, 2005.
- [2] V. P. Sinha. *Symetries and groups in signal processing: an introduction*. *Signals and Communication Technologies*. Springer, 2010.
- [3] Z. M. Hussain, A. Z. Sadik, P. O'Shea. *Digital signal processing: an introduction with MATLAB and applications*. Springer-Verlag, 2011.
- [4] A. Ukil. *Intelligent systems and signal processing in power engineering*. *Power Systems*, Springer-Verlag, 2007.
- [5] J. Hammond, P. White. Signals and systems. *In: Handbook of signal processing in acoustics*. Volume 1. Springer, 2008.
- [6] W. J. Witteman. *Detection and signal processing: technical realization*. *Advanced Microelectronics*, Springer, 2006.
- [7] B. Meddins. *Introduction to digital signal processing*. *Essential Electronics Series*, Newnes, 2000.
- [8] A. V. Oppenheim, R. W. Schaffer, J. R. Buck. *Discrete-time signal processing*. 2ª Edição. Prentice-Hall, 1999.
- [9] E. C. Ifeachor, B. W. Jervis. *Digital signal processing, a practical approach*. 2ª Edição. Prentice Hall, 2001.
- [10] P. A. Lynn, J. G. Fuerst. *Introductory digital signal processing with computer applications*. 2ª Edição. John Wiley & Sons, 1999.
- [11] L. R. Rabiner, B. Gold. *Theory and application of digital signal processing*. Prentice-Hall, 1975.
- [12] D. Anastassiou. Genomic signal processing. *IEEE Signal Processing Magazine*, 18(4): p. 8-20, 2001.
- [13] A. K. Takahata, E. Z. Nadalin, R. Ferrari, L. T. Duarte, R. Suyama, R. R. Lopes, J. M. T. Romano, and M. Tygel. Unsupervised processing of geophysical signals: a review of some key aspects of blind deconvolution and blind source separation. *IEEE Signal Processing Magazine*, 29(4):p. 27-35, 2012.
- [14] B. P. Lathi. *Signal processing and linear systems*. Oxford University Press, 2001.

- [15] D. B. Rowe. *Multivariate Bayesian statistics: models for source separation and signal unmixing*. Chapman and Hall/CRC, 2003.
- [16] L. T. Duarte. *Um estudo sobre separação cega de fontes e contribuições ao caso de misturas não-lineares*. Dissertação de Mestrado. Universidade Estadual de Campinas. Brasil, 2006.
- [17] J. M. T. Romano, R. de F. Attux, C. C. Cavalcante, R. Suyama. *Unsupervised signal processing: channel equalization and source separation*. CRC Press, 2011.
- [18] J. Héroult, C. Jutten, B. Ans, B. Détection de grandeurs primitives dans un message composité par une architecture de calcul neuromimétique en apprentissage non supervisé. *In: Actes du Xème Colloque*. GRETSI, p.1017-1022. França, 1985.
- [19] A. Peterson, J. N. Knight, M. J. Kirby, C. W. Anderson, M. H. Thaut. Feature selection and blind source separation in an EEG-based brain-computer interface. *EURASIP, Journal on Applied Signal Processing* 2005:19, p. 3128-3140, 2005.
- [20] L. D. Lathauwer, B. D. Moor, J. Vandewalle. Fetal electrocardiogram extraction by blind source subspace separation. *IEEE Transactions on Biomedical Engineering*, 47 (5), p. 567-572, 2000.
- [21] A. K. Barros. Extracting the fetal heart rate variability using a frequency tracking algorithm. *Neurocomputing*, 49, p. 279-288, 2002.
- [22] T. P. Makeig, S. Lee, T. W. McKeown, M. Brown, G. Bell, A. J. et al. Independent component analysis of biomedical signals. *In: Proceedings of the Second International Workshop on Independent Component Analysis and Blind Signal Separation*, ICA 2000, p. 633-644, 2000.
- [23] A. Hyvärinen, J. Karhunen, E. Oja. *Independent component analysis*. John Wiley & Sons, 2001.
- [24] R. R. F. Attux, A. Neves, L. T. Duarte, R. Suyama, C. C. M. Junqueira, L. E. P. et al. Rangel. *In: The Relationships between Blind Equalization and Blind Source Separation - Part II: Relationships*. Journal of Communication and Information Systems, 22(1), p. 53-61, 2007.
- [25] C. C. Cavalcante. *Sobre separação cega de fontes: proposições e análise de estratégias para processamento multi-usuário*. Tese de Doutorado, Universidade Estadual de Campinas, Brasil, 2004.
- [26] A. D. Back, A. S. Weigend. A first application of independent component analysis to extracting structure from stock returns. *In: Journal on Neural Systems*, 8(4):p. 473-484, 1997.

- [27] K. Kiviluoto, E. Oja. Independent component analysis for parallel financial time series. *In: Proceedings International Conference on Neural Information Processing (ICONIP'98)*, volume 2, p. 895-898, Japão, 1998.
- [28] P. Bofill. Undetermined blind source separation using sparse representations. *Signal Processing*, 81, p. 2353-2362, 2001.
- [29] A. M. Djafari. *A Bayesian approach to source separation*. In: Proceedings of the 19th International Workshop on Bayesian and Maximum Entropy Methods, 1999.
- [30] D. Guilhon, E. Medeiros, A. K. Barros. *ECG data compression by Independent Component Analysis*. In Proceedings of the IEEE Workshop on Machine Learning for Signal Processing, 2005.
- [31] V. Sanchez-Poblador, E. Monte-Moreno, J. Solé-Casal. ICA as a preprocessing technique for classification. *In: Proceedings of the Fifth International Workshop on Independent Component Analysis and Blind Signal Separation*. ICA 2004, p. 1165 – 1172, 2004.
- [32] P. Comon. Independent Component Analysis, a new concept. *In: Signal Processing*, 36 (6), p. 287-314, 1994.
- [33] J. V. Stone. *Independent component analysis: a tutorial introduction*. Massachusetts Institute of Technology, 2004.
- [34] J. F. Cardoso, A. Souloumiac. Blind beamforming for non-Gaussian signals. *IEE Proceedings Radar and Signal Processing*, 140 (6), p. 362-370, 1993.
- [35] A. J. Bell, T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.
- [36] A. Amari, A. Cichocki, H. H. Yang. A new learning algorithm for blind signal separation. In: *Advances in neural information processing systems 8*. MIT Press, p. 757–763, 1996.
- [37] S. Amari. Natural gradient works efficiently in learning. *Neural computation*, 10, 251–276, 1998.
- [38] A. Cichocki, S. Amari. *Adaptive blind signal and image processing: learning algorithms and applications*. John Wiley and Sons, 2002.
- [39] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling. *Numerical recipes*. Cambridge University Press, 1989
- [40] D. Pham, F. Vrins. Local minima of information-theoretic criteria in Blind Source Separation. *IEEE Signal Processing Letters*, 12 (11), p. 788-791, 2005.

- [41] F. Vrins, M. Verleysen. Information theoretic versus cumulant-based contrasts for multimodal source separation. *IEEE Signal Processing Letters*, 12 (3), p. 190-193, 2005.
- [42] B. V. D. Veen. *Introduction to the theory of operational research*. Springer-Verlag, 1967.
- [43] R. Storn. Differential evolution research – Trends and open questions. In: *Advances in differential evolution*, Springer, p. 1 - 33, 2008.
- [44] F. Glover, G. A. Kochenberger, Editors. *Handbook of metaheuristics*. Kluwer Academic Publishers, 2003.
- [45] R. C. P. Silva. *Um estudo sobre a autoadaptação de parâmetros na evolução diferencial*. Monografia, Universidade Federal de Ouro Preto (Instituto de Ciências Exatas), Brasil, 2010.
- [46] T. Back. *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.
- [47] R. Storn, K. Price. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, p. 341 - 359, 1997.
- [48] E. Bonabeau, M. Dorigo, G. Theraulaz. *Swarm intelligence: from natural to artificial systems*. Oxford University Press, 1999.
- [49] S. Binitha, S. S. Sathya. A survey of bio inspired optimization algorithms. *International Journal of Soft Computing and Engineering (IJSCE)* ISSN, p. 2231 - 2307, Volume 2, 2012.
- [50] A. G. Nikolaev; S. H. Jacobson. Simulated Annealing. In: *Handbook of Metaheuristics*. Springer, 2ª Edição, p. 1- 40, 2010.
- [51] J. R. Koza. *Genetic Programming: on the programming of computers by means of natural selection*. The MIT Press, 1992.
- [52] H. G. Beyer, H. P. Schwefel. Evolution strategies. *Natural Computing*. Volume 1, p. 3-52, 2002.
- [53] J.H. Holland. Genetic algorithms and the optimal allocation of trials. *Comput.* Volume 2 , p.88-105, 1973.
- [54] K. V. Price, R. M. Storn, J. A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer, 2005.
- [55] Dorigo M. Ottimizzazione. *Apprendimento automatico, ed algoritmi basati su metafora naturale (Optimization, learning and natural algorithms)*. Ph.D.Thesis (in Italian), Politecnico di Milano, Itália, 1992.

- [56] K. Socha, M. Dorigo. Ant colony optimization for continuous domains. *European Journal of Operational Research* 185 (2008), p. 1155–1173, Elsevier, 2008.
- [57] M. Kong, P. Tian. A Direct application of ant colony optimization to function optimization problem in continuous domains. *In: Ant Colony Optimization and Swarm Intelligence. 5th International Workshop, ANTS 2006 Proceedings*, 4 – 7, Springer, p. 324 - 331, 2006.
- [58] G. Bilchev, I. Parmee. The ant colony metaphor for searching continuous design spaces. *In: AISB Workshop on Evolutionary Computing*, Springer-Verlag, p. 25 - 39, 1995.
- [59] G. Bilchev, I. Parmee. Constrained optimization with an ant colony search model. *In: II International Conference on Adaptive Computing in Engineering Design and Control*, p. 26 - 38, 1996.
- [60] M. Mathur, S. Karale, S. Priye, V. Jayaraman, B. Kulkarni. Ant colony approach to continuous function optimization. *Industrial Engineering Chemistry Research* 39(10), p. 3814 - 3822, 2000.
- [61] M. Wodrich, G. Bilchev. Cooperative distributed search: the ant's way. *Control & Cybernetics* 3, p. 413 - 446, 1997.
- [62] J. Dreoo, P. Siarry. A new ant colony algorithm using the heterarchical concept aimed at optimization of multim minima continuous functions. *In: Third International Workshop, ANTS 2002. LNCS 2463*, Springer, p. 216 - 221, 2002.
- [63] J. Dreoo, P. Siarry. Continuous interacting ant colony algorithm based on dense heterarchy. *Future Generation Computer Systems* 20(5), p. 841 - 856, 2004.
- [64] K. Socha. Aco for continuous and mixed variable optimization. *In ANTS 2004. LNCS 3172*, Springer, p. 25 - 36, 2004.
- [65] M. Dorigo, T. Stützle. The ant colony optimization metaheuristic: algorithms, applications and advances. *In: Handbook of metaheuristics*. Kluwer Academic Publishers, p. 251 – 286, 2003.
- [66] L. N. de Castro. Fundamentals of natural computing: an overview. *Physics of Life Reviews* 4 (2007), p. 1 – 36, Elsevier, 2007.
- [67] M. Dorigo, V. Maniezzo, A. Coloni. The Ant system: an autocatalytic optimizing process. *Technical Report 91-016 Revised*, Dipartimento di Elettronica, Politecnico di Milano, Itália, 1991.
- [68] M. Dorigo, V. Maniezzo, A. Coloni. Positive feedback as a search strategy. *Technical Report 91-016*, Dipartimento di Elettronica, Politecnico di Milano, Itália, 1991.

- [69] M. Dorigo, V. Maniezzo, A. Colorni. The Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics - PartB*, 26(1), p. 29 - 41, 1996.
- [70] K. Socha, C. Blum. Ant colony optimization. *In: Metaheuristic procedures for training neural networks*. Springer, 2006.
- [71] D. Henderson, S. H. Jacobson, A. W. Johnson. The theory and practice of simulated annealing. *In: Handbook of metaheuristics*. Kluwer Academic Publishers, p. 287 – 321, 2003.
- [72] W. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller. *Equation of state calculations by fast computing machines*. *Journal of Chemical Physics*, vol. 21, p. 1087 – 1092, 1953.
- [73] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi. Optimization by simulated annealing. *Science, New Series*, Vol. 220, No. 4598, p. 671-680, 1983.
- [74] V. Cerny. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of optimization theory and applications*, Vol. 45, No. 1, p. 41 – 51, 1985.
- [75] C. M. Tan, editor. *Simulated annealing*. In-teh, 2008.
- [76] H. A. e Oliveira Junior, L. Ingber, A. Petraglia, M. Rembold Petraglia, M. A. S. Machado. Stochastic global optimization and its applications with fuzzy adaptive simulated annealing. *Springer-Verlag*, 2012.
- [77] E. Aarts, J. Korst, W. Michiels. Simulated annealing. *In: Search methodologies: introductory tutorials in optimization and decision support techniques*. II Edição, Springer, p. 187 – 210, 2014.
- [78] H. A. De Araujo. *Algoritmo simulated annealing: uma nova abordagem*. Dissertação de mestrado, Universidade Federal de Santa Catarina, Brasil, 2001.
- [79] J. Kennedy, R. Eberhart. Particle Swarm Optimization. *In: Proceedings of IEEE International Conference on Neural Networks IV*, p. 1942–1948, 1995.
- [80] C. W. Reynolds. Flocks, herds, and schools: a distributed behavioral model. *Computer Graphics*, 21(4), p. 25 – 34, 1987.
- [81] F. Heppner, U. Grenander. A stochastic nonlinear model for coordinate bird flocks. *In The ubiquity of chaos*, p. 233-238, 1990.
- [82] M. Clerc, J. Kennedy. The particle swarm – explosion, stability, and convergence in a multidimensional complex space. *In: IEEE Transactions on Evolutionary Computation*, VOL. 6, NO. 1, p.58 – 73, 2002.

- [83] Q. Guo-dong, L. Zhang-hua. Application of particle swarm algorithm in the optimal allocation of regional water resources based on immune evolutionary algorithm. *J. Shanghai Jiaotong University and Springer-Verlag*, 18(5), p. 634-640, 2013.
- [84] Y. Shi, R. Eberhart. A Modified Particle Swarm Optimizer. *In: Proceedings of the IEEE International Conference on Evolutionary Computation*, p. 69 – 73, 1998.