

Universidade Federal de Alagoas
Instituto de Computação
Curso de Ciência da Computação



Dissertação de Mestrado

Uma Aplicação do Algoritmo Genético Baseado em Tipos Abstratos de Dados ao Problema de Separação Cega de Fontes

Valter Wellington Ramos Junior

valterwill@gmail.com

Orientadores:

Luis Claudius Coradine

Roberta Vilhena Vieira Lopes

Maceió, Março de 2012

Valter Wellington Ramos Junior

**Uma Aplicação do Algoritmo Genético Baseado em
Tipos Abstratos de Dados ao Problema de
Separação Cega de Fontes**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Curso de Mestrado em Modelagem Computacional de Conhecimento do Departamento de Tecnologia da Informação da Universidade Federal de Alagoas.

Orientadores:

Luis Claudius Coradine

Roberta Vilhena Vieira Lopes

Maceió, Março de 2012

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 – 1767

R175a Ramos Junior, Valter Wellington.
Uma aplicação do algoritmo genético baseado em tipos abstratos de dados ao problema de separação cega de fontes / Valter Wellington Ramos Junior. – 2020.
58 f. : il.

Orientador: Luis Claudius Coradine.
Co-orientadora: Roberta VilhenaVieira Lopes.
Dissertação (mestrado em Modelagem Computacional do Conhecimento) –
Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2012.

Bibliografia: f. 56-58.

1. Inteligência artificial. 2. Algoritmos genéticos. 3. Separação cega de fontes. I.
Título.

CDU: 004.421



UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Modelagem Computacional de Conhecimento
Avenida Lourival Melo Mota, Km 14, Bloco 09, Cidade Universitária
CEP 57.072-900 – Maceió – AL – Brasil
Telefone: (082) 3214-1364



Membros da Comissão Julgadora da Dissertação de Mestrado de Valter Wellington Ramos Junior, intitulada: “Uma Aplicação do Algoritmo Genético Baseado em Tipos Abstratos de Dados ao Problema de Separação Cega de Fontes”, apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas em 22 de março de 2012, às 15h00min, na sala de aula do Mestrado em Modelagem Computacional de Conhecimento.

COMISSÃO JULGADORA

Prof. Dr. Luis Cláudius Coradine

UFAL – Instituto de Computação
Orientador

Profa. Dra. Roberta Vilhena Vieira Lopes

UFAL – Instituto de Computação
Co-orientadora

Prof. Dr. Antônio Fernando de Souza Bezerra

UFAL – Faculdade de Medicina
Examinador

Prof. Dr. João Marcos Travassos Romano

Unicamp – Faculdade de Engenharia Elétrica e de Computação
Examinador

Prof. Dr. Aluizio Fausto Ribeiro Araújo

UFPE – Centro de Informática
Examinador

Maceió, março de 2012.

Resumo

A presente dissertação apresenta uma modelagem de um algoritmo genético, baseado em tipos abstratos de dados, aplicada a problemas de separação cega de fontes.

Nesse sentido, faz-se comparações através de abordagens utilizando o algoritmo genético de Holland e o Algoritmo Genético Baseado em Tipos Abstratos de Dados,

com o objetivo de mostrar a eficiência desse algoritmo na caracterização de uma solução combinada a problemas de otimização complexa, usando de maneira parcimoniosa os

recursos computacionais disponíveis. As aplicações em problemas de separação cega de fontes se concentram em duas vertentes, uma na recuperação de um sinal

eletrocardiográfico em meio ruidoso, a partir do processo de separação de fontes propriamente dito, e a outra na caracterização do filtro inverso, num problema de

desconvolução cega. Nessas duas abordagens, busca-se, através do processo evolutivo, ajustar os coeficientes dos sistemas ponderados a partir da consideração, como função

custo, de algumas técnicas de separação de fontes, como a Curtose, Informação Mútua, Negentropia e principalmente a combinação delas. Os resultados obtidos por meio de

simulações mostram um bom compromisso entre desempenho e custo computacional.

Palavras-chave: Inteligência Artificial, Algoritmo Genético, Separação Cega de Fontes.

Abstract

This dissertation presents a model of a genetic algorithm based on abstract data types, applied to problems of blind source separation. In this sense, it is through comparisons of approaches using genetic algorithm Holland and Genetic Algorithm Based on Abstract Data Types, in order to show the efficiency of this algorithm in the characterization of a combined solution of complex optimization problems, using so parsimonious computational resources. The problems in applications blind source separation are concentrated in two parts, one in the recovery of an electrocardiographic signal in noisy environment, from the source separation process itself, and the other in the characterization of the inverse filter, a problem of blind deconvolution . These two approaches, seeks to, through the evolutionary process, adjusting the weighted coefficients of the systems from consideration as the cost function, some techniques for the separation of sources, such as kurtosis, mutual information, negentropy and especially their combination. The results obtained by simulations show a good compromise between performance and computational cost.

Keywords: Artificial Intelligence, Genetic Algorithm, Blind Sources Separation.

Agradecimentos

Agradeço ao Senhor Jesus em primeiro lugar, por sempre ter me ajudado tanto.

Agradeço também a minha mãe e minha tia Dilza, pelo amor e carinho.

A minha esposa pela compreensão e dedicação na fase de desenvolvimento da dissertação.

Ao meu Orientador Luis Claudius Coradine e minha co-orientadora Roberta Vilhena Vieira Lopes, meus amigos, que me passaram conhecimentos e ensinamentos que levarei para toda a vida.

E aos meus amigos que conquistei no mestrado, Bruno Nascimento, Bruno Raphael, Helio, Leonardo, Andréa Vanderlei, Carlos Jean, Allan, Euzebio, Claudio e Cleo Brandão que sempre estiveram por perto me ajudando e me dando apoio.

Sumário

1	Introdução	1
1.1	O Processamento de Sinal	1
1.2	Motivação	2
1.3	Objetivo	3
1.4	Organização	4
2	Separação Cega de Fontes	5
2.1	Introdução	5
2.2	Histórico	5
2.3	Descrição Matemática	7
2.4	Aplicações	8
2.4.1	Separação de sinais de áudio - O cocktail party-problem	9
2.4.2	Processamento de sinais biomédicos	10
2.4.3	Telecomunicações - BSS e equalização cega de canais	10
2.4.4	Exploração Geofísica - Remote Sensing	12
2.5	Métodos Estatísticos	12
2.5.1	Entropia e Informação Mútua	13
2.5.2	CrITÉrios Baseados em Não-Gaussianidade - Negentropia,Curtose	14
2.6	Sumario	16
3	O Algoritmo Genético	17
3.1	O Algoritmo Genético de Holland	18
3.2	Variações do algoritmo de Holland	23
3.2.1	Representação do indivíduo	23
3.2.2	Representação da População	25
3.2.3	Função de Adaptação	25
3.2.4	Operação de seleção	26
3.2.5	Operação de Cruzamento	28
3.2.6	Operação de mutação	29
3.2.7	Probabilidade de Cruzamento e Mutação	30
3.3	O Algoritmo Genético Baseado em Tipos Abstratos de Dados	31
3.3.1	Introdução	31
3.3.2	Tipos Básicos do GAADT	32
3.3.3	Operadores Genéticos do GAADT	33
3.3.4	Ambiente do GAADT	35
3.3.5	Algoritmo do GAADT	36

4	Especificação dos algoritmos evolutivos para o problema de Separação Cega de Fontes	37
4.1	Introdução	37
4.2	Especificação do GAADT	37
4.2.1	Introdução	37
4.2.2	Tipos Básicos	38
4.2.3	Operadores Genéticos	40
5	Estruturação de um Sistema de Separação de Fontes para Recuperação de Sinais	41
5.1	Introdução	41
5.2	Tecnologias Utilizadas	41
5.3	O Sistema	42
5.3.1	Tela de Acesso ao sistema	42
5.3.2	Tela Inicial	42
5.4	Resultados	43
5.4.1	Cenário um	43
5.4.2	Cenário dois	44
5.4.3	Avaliação	46
6	Conclusão e Trabalhos Futuros	53
6.1	Introdução	53
6.2	Considerações Finais	53
6.3	Contribuições	54
6.4	Trabalhos Futuros	54

Lista de Figuras

2.1	Curvas da Curtose	15
3.1	O algoritmo genético R	18
3.2	A população P	19
3.3	Cálculo da função de adaptação de P	20
3.4	População auxiliar P' do algoritmo R	20
3.5	Operação de cruzamento de Holland	21
3.6	Operação de mutação de Holland	22
3.7	Operação de inversão de Holland	22
3.8	População gerada pelos operadores de cruzamento, mutação e inversão	23
3.9	Nova população	23
3.10	Mutação por troca	30
5.1	Tela de login	47
5.2	Tela de erro no login	47
5.3	Tela inicial do sistema	48
5.4	Tela do programa Algoritmo Genético	48
5.5	Tela de resultados	49
5.6	Tela de resultados, amostras	49
5.7	Tela de Gráficos	50
5.8	Tela do programa Configuração	50
5.9	Amostra de ECG obtida através do MIT-BIH Database	50
5.10	Arquivo 1 da amostra de ECG obtida através do MIT-BIH Database	50
5.11	Arquivo 2 da amostra de ECG obtida através do MIT-BIH Database	51

Lista de Códigos

Capítulo 1

Introdução

1.1 O Processamento de Sinal

O estudo do processamento de sinal Haykin (2000) é uma das áreas mais antigas e importantes da Engenharia Elétrica e tem servido de fonte de informação a diversos áreas de conhecimento, tais como: Geofísica, Medicina, Telecomunicações, etc.

Na Medicina o processamento de sinal tem servido para extrair informações dos sinais dos batimentos cardíacos, pulmonares e cerebrais, auxiliando no diagnóstico dos médicos, como também para remover sinais de ruídos que estão misturados ao sinal de referência e interferente no preciso diagnóstico.

Nos sinais de eletro cardiograma por exemplo, no caso o exame de eletro cardiograma fetal, o processamento de sinal, é empregado para separar os sinais do ECGF (Eletro cardiograma Fetal) dos sinais do ECGM (Eletro Cardiograma Materno), visto que no momento do exame de ECGF os sinais do batimento cardíaco da mãe interfere e é adicionado aos sinais do batimento cardíaco do bebê, dificultando a análise do médico, dentre outros fatores que influenciam no exame de ECGF descritos por [Lathauwer, 1997]:

- os movimentos mecânicos do coração podem introduzir variações espúrias na medida da frequência cardíaca fetal;
- a maioria dos sistemas baseados em Doppler calculam um número médio de variações da frequência cardíaca, não sendo possível medir a variação a cada batimento;
- as unidades de monitoramento são geralmente disponíveis até uma frequência cardíaca de 220 batimentos por minuto; no caso de frequência mais altas o número é dividido por dois resultando numa aparente frequência cardíaca normal quando, na verdade, o feto apresenta taquicardia.

Nas Telecomunicações, o processamento de sinais é utilizado entre outras aplicações na equalização de canais, onde se objetiva que a informação que é transmitida pelo emissor seja

recebida pelo receptor de forma fiel a original, eliminando as distorções ocorridas durante o processo de transmissão no canal.

Este tipo de problema utiliza uma operação de processamento de sinal chamada de Deconvolução, cujo o objetivo é reconstruir o sinal de entrada a partir do sinal de saída e do sistema, quando são conhecidos. No caso em que só se tem informação do sinal de saída e podendo haver informação das estatísticas da fonte, a operação utilizada é a Deconvolução cega, uma operação não-supervisionada que tem o objetivo de encontrar tanto o sinal de entrada quanto o sistema a partir do sinal de saída. Com a Deconvolução é possível obter a informação que é transmitida pelo emissor após passar pelo canal de transmissão.

No campo de estudos de processamento de sinais, um dos problemas de grande importância é a recuperação de sinais através da observação de misturas Jutten (2010). Nas estratégias clássicas, esse processo é realizado através da análise de informações sobre os sinais originais ou sobre o processo de mistura. Tal metodologia é referenciada como supervisionada, já que pressupõe o conhecimento de informações prévias sobre as fontes. A necessidade desse conhecimento torna-se uma limitação dessa estratégia, já que existem problemas em que não é possível obter tais informações. Além disso, há ainda, nesse modelo, hipóteses sobre os sinais e as misturas, que também restringe o seu espaço de aplicação. É nesse contexto, que se buscam criar estratégias que necessitem tão pouca informação quanto possível na recuperação dos sinais. Dessa metodologia não-supervisionada ou cega, se origina o problema de BSS (Separação Cega de Fontes, do inglês Blind Source Separation).

1.2 Motivação

A confiança atribuída a uma dada informação, sobre o processamento de sinais, especificamente utilizando uma estratégia não supervisionada, é diretamente proporcional ao volume de dados analisados na construção dessa informação. Nas estratégias clássicas, a tarefa de recuperar sinais de interesse (fontes) obtidos a partir da observação da misturas deles, é realizada levando em conta características relevantes desses sinais fontes, ou ainda, informações sobre o processo de mistura.

Um exemplo típico acontece nas situações em que as características espectrais dos sinais, misturados no tempo, são bem conhecidas. Neste caso, é possível separá-los a partir do emprego de filtros seletivos em frequência, por exemplo. Evidencia-se que há nesta estratégia uma hipótese tácita relativa à necessidade de que os sinais tenham espectros diferentes.

Neste simples exemplo vê-se a ilustração satisfatória de algumas restrições presentes nas estratégias supervisionadas, ou seja, aquelas que pressupõem a existência de informações sobre os sinais e sistemas envolvidos no processo de mistura. Inicialmente, a necessidade deste conhecimento já é uma considerável limitação em si, dado que existem aplicações nas quais tal exigência pode ser inatingível.

Uma forma de contornar essas dificuldades seria criar novas estratégias buscando sempre por um maior grau de generalidade, no sentido de se assumir tão pouca informação quanto possível sobre a geração dos dados a serem processados. Existem varias abordagens utilizadas no processo de separação cega das fontes, como por exemplo Analise de Componentes Independentes ICA (Independent Component Analysis), uma técnica em representação de dados que pode ser vista como uma extensão da clássica análise de componentes principais (PCA).

A utilização dessas abordagens foram baseados em métodos convencionais e receberam várias críticas devido à sua incapacidade em:

- fornecer uma explicação para as decisões tomadas pelo programa no tratamento de dados ambíguos e incompletos;
- manipular dados com diversas representações ao mesmo tempo;
- convergência prematura levando o resultado a ficar preso em mínimos locais

Essas críticas resultaram no entendimento de que a utilização de métodos baseados em Inteligência Artificial, mais especificamente computação evolutiva, auxiliaria no processo de tomada de decisão evitando assim que houve-se convergência prematura e o resultado ótimo não fosse alcançado.

1.3 Objetivo

A ideia da proposta da dissertação consiste em realizar comparações através de abordagens utilizando o algoritmo genético de Holland e o Algoritmo Genético Baseado em Tipos Abstratos de Dados, com o objetivo de mostrar a eficiência desse algoritmo na caracterização de uma solução combinada a problemas de otimização complexa, usando de maneira parcimoniosa os recursos computacionais disponíveis.

Para atingir tal objetivo propõe-se a implementação de um sistema para análise de sinais que utilize métodos estatísticos para ajustar os pesos do filtro através de processo evolutivo. Tal sistema se baseia no uso do algoritmo genético baseado em tipos abstratos de dados na etapa de treinamento e de um estimador de entropia baseado em estatísticas de ordem que:

- forneça uma explicação para as decisões tomadas pelo sistema no tratamento de dados ambíguos e incompletos;
- manipule dados com diferentes representações;
- gerencie a explosão combinatorial de todas as possíveis soluções geradas pela presença de dados ambíguos e incompletos;
- permita a análise do sinal de entrada e saída, para todos os métodos estatísticos utilizados.

1.4 Organização

O conteúdo da dissertação está organizada em cinco capítulos, e cada um deles é apresentado a seguir:

- Capítulo 2: São apresentados o histórico, a descrição do problema de separação cega de fontes e algumas das principais aplicações onde a BSS pode ser empregada.
- Capítulo 3: Neste capítulo são expostos o modelo de dois algoritmos evolutivos que foram empregados no tratamento de problemas de BSS. O algoritmo Genético de Holland e o Algoritmo Genético baseado em Tipos Abstratos de Dados.
- Capítulo 4: Neste capítulo, são apresentadas a estrutura do problema e a implementação proposta para a dissertação, utilizando os algoritmos descritos na capítulo 3 afim de avaliar a utilização das funções de custo separadas e em conjunto.
- Capítulo 5: Resume o tema proposto na dissertação, os experimentos e seus respectivos resultados.
- Capítulo 6: Neste capítulo são apresentadas as conclusões gerais da dissertação e as perspectivas para trabalhos futuros.

Capítulo 2

Separação Cega de Fontes

2.1 Introdução

Este capítulo retrata os aspectos teóricos sobre o problema de Separação Cega de Fontes (BSS, do inglês Blind Source Separation). Na introdução é exposta a descrição do problema, por conseguinte são apresentadas algumas aplicações em que a BSS é utilizada expondo seu modelo matemático e técnicas utilizadas.

O problema de separação cega de fontes trata da recuperação de sinais através das amostras de misturas de sinais que lhe deram origem.

O significado do termo "cega" em separação cega de fontes, é que não se tem conhecimento prévio, ou algum tipo de informação sobre sistema misturador. Visto que não se tem informações relevantes sobre o problema, ou que seria inviável ou até mesmo impossível recuperar ou utilizar algum tipo de sinal para ajustes do sistema misturador, esta técnica seria muito útil em problemas de caráter não-supervisionados.

2.2 Histórico

O trabalho de Héroult, Jutten e Ans (Héroult, Jutten e Ans, 1985) é considerado o marco inicial da BSS. A motivação de tal trabalho proveio de um problema biológico relacionado à codificação empregada pelo sistema nervoso central para a ativação muscular. Mais especificamente, levando em conta que um único sinal codifica duas informações relevantes (no caso, deslocamento e velocidade angular do movimento) e que o cérebro consegue realizar esta tarefa com sucesso, tentou-se obter um método computacional capaz de distinguir essas duas informações.

A despeito das limitações práticas do algoritmo desenvolvido (trataremos delas com mais detalhes na Seção 3.2.1), esse trabalho foi o primeiro a apontar para a necessidade da aplicação de estatísticas de ordem superior ao problema, o que, como veremos na Seção 2.5, mostrou-se fundamental na concepção de métodos eficientes de BSS. Além do mais, um dos alicerces do paradigma BSS/ICA, a modelagem algébrica dos sistemas misturador e separador, também foi

originalmente introduzido por esse mesmo trabalho. Apesar da importância das contribuições presentes em Héraul et al. (1985), foi somente no início da década de 1990 que a BSS passou a atrair uma atenção considerável dos pesquisadores, principalmente na Europa.

Um dos principais responsáveis por esse feito foi Pierre Comon, que, a partir dos resultados obtidos na década de 1950 por Darmois, formalizou a idéia da ICA e mostrou como a independência estatística se insere no problema de separação de fontes (Comon, 1994). Um outro pesquisador francês cuja contribuição foi fundamental para o desenvolvimento da BSS foi Jean-François Cardoso. Além dos estudos sobre o estimador de máxima verossimilhança em BSS (Cardoso, 1998a), Cardoso introduziu métodos tensoriais no problema (Cardoso e Souloumiac, 1993) e também desenvolveu o chamado gradiente relativo (Cardoso e Laheld, 1996), um método de otimização extremamente conhecido em BSS, que, como será visto na Seção 3.2.8, foi obtido independentemente por Amari (Amari, 1998), que o denominou de gradiente natural.

Podemos dizer que o trabalho de Bell e Sejnowski (Bell e Sejnowski, 1995) também foi crucial para a popularização da BSS, pois, além de estabelecer importantes ligações entre alguns estudos em codificação neural e a ICA, o algoritmo proposto por tais pesquisadores causou um certo furor na comunidade de processamento de sinais devido, sobretudo, à sua capacidade de separar um considerável número de fontes e à sua simplicidade de implementação, qualidades fundamentais em diversas aplicações das técnicas de BSS.

Assim, essa contribuição serviu para mostrar que, apesar da complexidade inerente ao problema de separação, seria sim possível desenvolver técnicas implementáveis em cenários práticos. Destacamos os trabalhos de três pesquisadores: Karhunen, Oja e Hyvärinen. Os trabalhos de Karhunen e Oja (Karhunen, Pajunen e Oja, 1998) permitiram interpretar a ICA como uma extensão não-linear da consagrada técnica de Análise de Componentes Principais (PCA, Principal Component Analysis). Tal abordagem teve um papel fundamental no entendimento da ICA como um tema relevante em análise de dados, ou, ainda, em análise multivariável. Já Hyvärinen contribuiu para o desenvolvimento de critérios baseados na maximização da não-gaussianidade (Hyvärinen, Karhunen e Oja, 2001), nos quais se baseia uma das técnicas mais utilizadas em separação de fontes, o FastICA.

Além das contribuições supracitadas, há ainda outros trabalhos da década de 1990 que foram importantes para a consolidação da BSS. Procuraremos nessa dissertação, especificamente no Capítulo 3, apresentar os aspectos básicos das principais abordagens desse período. Veremos que a grande maioria desses trabalhos foi desenvolvida levando em conta modelos 3 simplificados de sistemas misturadores, essencialmente lineares e instantâneos (adentraremos a descrição desses modelos na Seção 2.4). Já em um segundo período da BSS, cujo início se deu no final da década de 1990, as atenções dos pesquisadores voltaram-se para a extensão dos resultados previamente obtidos a casos mais complexos de sistemas misturadores, como, por exemplo, modelos não-lineares, dinâmicos e sub-determinados. Ainda nos dias atuais, essas vertentes correspondem aos principais temas de estudo em BSS.

Podemos também destacar outros dois assuntos que marcam o estágio atual dos estudos em

BSS. O primeiro deles refere-se à dissociação entre a BSS e a ICA, que pode ser constatada pelo crescente número de trabalhos na literatura que abordam a BSS por meio de outras metodologias, alternativas à ICA, e, também, pelo surgimento de outras aplicações para a ICA. A outra tendência na área remete à consolidação de diversas aplicações em BSS, principalmente em processamento de sinais biomédicos, que pode ser verificada pela expressiva quantidade de artigos que versam sobre essa questão nas duas últimas edições da conferência mais representativa da área (Puntonet e Prieto, 2004; Rosca, Erdogmus, Principe e Haykin, 2006).

2.3 Descrição Matemática

Após termos visto as principais aplicações da BSS, além de um breve histórico deste assunto, passaremos a descrever como o problema em questão pode ser resolvido. Um primeiro passo, que será dado na presente seção, é buscar modelos matemáticos capazes de expressar as ações dos sistemas misturador e separador, de acordo com a ideia ilustrada na Figura 2.1. De modo a obter uma descrição matemática do problema de BSS, considere que cada elemento do vetor $s(n) = [s_1(n)s_2(n) \dots s_N(n)]_T$ corresponde a um sinal fonte. Analogamente, representemos os sinais misturados através do vetor $x(n) = [x_1(n)x_2(n)\dots x_M(n)]_T$. Em sua forma mais geral, o processo de mistura das fontes pode ser representado pela seguinte expressão:

$$x(n) = \mathcal{F}(S(n), s(n-1)\dots s(n-L), n(n)), \quad (2.1)$$

onde o mapeamento $\mathcal{F}(\cdot)$ descreve a ação do sistema misturador, L corresponde ao número de amostras passadas levadas em conta no processo de mistura, ou seja, diz respeito à memória associada ao sistema, e o vetor de $n(n)$ denota o ruído associado às próprias fontes (ruído de fonte) e/ou aos sensores (ruído de sensor).

É importante salientar que a formulação apresentada (2.1) possui um caráter eminentemente didático, haja visto a inexistência de técnicas de BSS capazes de lidar com todos os efeitos representados nessa expressão (ruído, memória, não-linearidade, etc). Em geral, as técnicas desenvolvidas em BSS são direcionadas para casos particulares, mais simplificados, da formulação apresentada. Assim, de modo a auxiliar o leitor na identificação dos diversos casos presentes na literatura, apresentamos na sequência como é feita a classificação de um sistema misturador.

- **Sistemas Lineares e Não-Lineares** Um sistema misturador é dito linear se o mapeamento $\mathcal{F}(\cdot)$ atende ao princípio da superposição, ou seja, nas situações em que: para quaisquer constantes b_1 e b_2 , e vetores de sinais $s_1(n)$ e $s_2(n)$. Caso contrário, o sistema misturador é dito não-linear.
- **Sistemas Instantâneos e Convolutivos** Nas situações em que o processo de mistura depende de amostras passadas, ou seja, $L > 0$, o sistema misturador é dito convolutivo, ou

com memória. Por outro lado, nas situações em que $L = 0$, o sistema é dito instantâneo. Com Relação ao Número de Fontes e de Sensores Quando o número de sensores é maior que o número de fontes ($M > N$), tem-se o chamado caso sobre-determinado. Analogamente, o caso sub-determinado corresponde à situação em que ($M < N$). É interessante notar que os casos em que há sinais de ruído podem ser tratados como um caso específico de modelos sub-determinados, dado que é possível considerá-los como sendo fontes (Kofidis, 2001).

Já vimos que uma das características marcantes do problema de BSS é a falta de informação sobre o processo de mistura e sobre as fontes. Todavia, é fundamental ao menos um certo conhecimento sobre a estrutura do sistema misturador, pois, com base nessa informação, torna-se possível definir um sistema separador estruturalmente capaz de inverter a ação do processo de misturas (por exemplo, para inversão de misturas não-lineares, é imperativo o uso de uma estrutura de mesma natureza). Geralmente, esse tipo de conhecimento é obtido com base na aplicação de interesse. Como exemplo, é sabido que, em problemas de separação de sinais de áudio, o processo de mistura é notadamente convolutivo devido a efeitos de reverberação ou eco.

A maioria dos trabalhos em BSS abordam cenários com sistemas misturadores lineares, instantâneos e com o mesmo número de fontes e misturas. Nesta situação, o processo de mistura é descrito matematicamente (doravante omitiremos o índice temporal por questões de simplicidade) do seguinte modo:

$$x = As; \tag{2.2}$$

sendo que, nesta situação, a matriz A de dimensão $N \times N$ é chamada de matriz de mistura. Apesar de sua simplicidade, esta classe de modelos é válida em uma vasta quantidade de problemas de BSS (Hyvärinen, Karhunen e Oja, 2001). Ademais, é possível, neste caso, recuperar as fontes através da Análise de Componentes Independentes, técnica esta que será o assunto da próxima seção.

2.4 Aplicações

Esta seção trata de mostrar as principais aplicações em que seria útil a utilização de técnicas de BSS para resolução de problemas. Veremos que as aplicações são em diversas áreas e tipos diferentes de problemas, o que mostra que as técnicas são bem flexíveis e adaptáveis a diversas situações.

Além das aplicações que serão expostas: Separação de Sinais de Audio, Processamento de Sinais Biomedicos, Equalização cega de Canais, Exploração Geofisica e Extração de Imagens, há ainda outros problemas de BSS provenientes das mais diversas áreas como, por exemplo, ar-

ranjos de sensores químicos [Bermejo et al. (2006)] e cancelamento de reflexões [Hyvarinen et al. (2000)]. O leitor interessado em outras aplicações pode consultar as referências [Hyvarinen et al. (2000), Cichocki e Amari (2002), Puntonet e Prieto (2004); Rosca et al. (2006)].

2.4.1 Separação de sinais de áudio - O cocktail party-problem

O processo de cocktail-party effect [Arons (1992)] descreve a habilidade de focalizar a atenção em uma única voz dentre uma mistura de conversas e ruídos de fundo. Este efeito revela uma das habilidades surpreendentes do nosso cérebro que nos permite distinguir um som dentre vários outros. Existem vários tipos de problemas de separação de sinais de áudio onde a analogia ao que o cérebro humano executa nos leva a estudos para construir um modelo computacional similar ao do cérebro humano para resolve-los.

Essa observação motivou a pesquisa sobre o problema atualmente conhecido como o cocktail party problem. Um problema de separação de áudio normalmente é caracterizado por um ambiente ou uma sala, na qual existem pessoas conversando, e as vozes dessas pessoas são capturadas por um conjunto de microfones. Para simplificar, suponha que há em tal sala duas pessoas falando simultaneamente e que o som está sendo capturado por dois microfones (dois canais). Se nesse sistema é possível conhecer as condições de transmissão dos sinais desde sua fonte (a boca da pessoa que está falando) até o sensor (microfone), em alguns casos (sistemas não convolutivos e/ou não subparametrizados), através de cálculos não tão complexos, é possível recuperar as falas de cada uma das pessoas separadamente. Esse processo caracteriza uma separação não-cega de fontes, pois os princípios de mistura são conhecidos. No entanto, caso não seja possível conhecer previamente informações sobre a mistura, está caracterizado um problema de separação cega de fontes.

Um conjunto de aplicações nasce do cocktail party problem. Alguns exemplos:

- Análise de componentes independentes aplicada à separação de sinais de áudio.[Moreto, Fernando 2008]
- Algoritmos de Separação Cega de Sinais de Áudio no domínio da Frequencia em Ambientes Reverberantes: Estudos e Comparações.[Laport, Luis (2010)].
- Melhoramento do sinal de Voz por Inibição Lateral e Mascaramento Bineural.[Nascimento, James (2004)] .
- Métodos de Fatoração de Matrizes Não-Negativas para Separação de Sinais Musicais.[Alan, Tygel (2009)].
- Transcrição automática de sinais de áudio musicais determinando, em uma música, quais instrumentos e quais notas estão sendo tocados em determinado instante[Plumbley et al. (2002)];

- Processos de reconhecimento de voz [Arons e Schuster (1997)];
- Separação de vozes em uma sala de reuniões através da utilização de um conjunto de microfones espalhados pela sala: aqui há necessidade do uso de técnicas de supressão de eco de reverberação (eliminação da reverberação) [Huang et al. (2006)].

2.4.2 Processamento de sinais biomédicos

Em engenharia biomédica, é de grande interesse o desenvolvimento de métodos de aquisição de sinais que sejam não-invasivos e, ainda assim, confiáveis. O EEG (Eletroencefalograma) [Jung et al. (2000), Calhoun et al. (2003), Vigário (1988)] e o ECG (Eletrocardiograma) [Jung et al. (2000), Lathauwer et al. (2000), Barros (2002)] são dois exemplos bem conhecidos de técnicas que operam de acordo com esse princípio. Todavia, tal tarefa é de extrema complexidade, tendo em vista a impossibilidade de captar, por meio de sensores posicionados em uma determinada região do corpo humano, apenas os sinais de interesse para um determinado exame, principalmente devido à interferência de sinais gerados pelos mais diversos tipos de atividade fisiológica. Em suma, esses procedimentos são, geralmente, caracterizados por uma baixa relação sinal-ruído (SNR, Signal-to-Noise Ratio) ou sinal-interferente.

Uma estratégia frequentemente utilizada para diminuir a intensidade do ruído nas amostras obtidas fundamenta-se na repetição de diversas realizações do exame, de modo que seja possível levantar um comportamento médio dos dados de interesse. Apesar dos bons resultados atingidos, esse tipo de abordagem exige a execução de um elevado número de repetições, o que, em alguns casos, pode não ser um procedimento desejável, ou mesmo viável. Além disso, tal conduta pode causar fadiga nos indivíduos examinados, o que, por sua vez, acarreta alterações artificiais dos padrões obtidos, principalmente no monitoramento de sinais cerebrais.

O emprego de técnicas de BSS oferece uma alternativa eficiente a essa abordagem, posto que, nesse caso, a recuperação dos sinais de interesse se dá através de estágios sofisticados de processamento conduzidos posteriormente à captação dos dados, o que requer a realização de apenas um experimento. Além disso, a ausência de modelos capazes de determinar quais sinais fisiológicos interferentes são captados e, ademais, como eles se misturam, posiciona esse tipo de problema em uma condição favorável à aplicação dos métodos de BSS. Uma boa evidência dessa aplicabilidade pode ser comprovada pela expressiva quantidade de trabalhos de separação de sinais biomédicos, a tal ponto que pode-se dizer que, atualmente, essa área corresponde a um dos principais domínios de aplicações técnicas de BSS [Hyvarinen et al. (2000)].

2.4.3 Telecomunicações - BSS e equalização cega de canais

A aplicação da BSS em telecomunicações está fortemente relacionada a um tema de expressiva relevância em comunicações digitais: a equalização de canais. A seguir, faremos uma descrição sucinta deste assunto com o intuito de indicar as principais relações entre ambos os temas.

A ideia essencial de um sistema de comunicação é fazer com que a informação enviada por um transmissor possa ser obtida de maneira tão fiel ao original quanto possível por um receptor. Assim sendo, é primordial que o desenvolvimento de sistemas de comunicação leve em conta estratégias capazes de mitigar as distorções introduzidas pelo canal, elemento presente entre o transmissor e o receptor, na informação transmitida. Em uma das estratégias mais empregadas, a equalização de canal, utiliza-se um filtro (equalizador) no receptor de modo que este seja capaz de inverter a ação do canal. Em essência, o desenvolvimento de técnicas de equalização está intimamente relacionado à concepção de critérios que guiem o ajuste dos parâmetros livres do equalizador de modo que se obtenha uma boa estimativa do sinal transmitido.

Por exemplo, em um dos paradigmas mais conhecidos, adota-se como critério a minimização do erro quadrático médio entre a saída do equalizador e o sinal desejado, no caso, o sinal transmitido (Haykin, 1996).

No caso supracitado, chama a atenção o fato de que o critério adotado se apoia no conhecimento tanto do sinal recebido quanto de amostras do sinal transmitido. Essa necessidade caracteriza o paradigma de equalização supervisionada [Haykin (1996)].

Em contrapartida, os critérios presentes na equalização não-supervisionada (ou cega) utilizam, além dos sinais recebidos, apenas algumas informações estatísticas dos sinais transmitidos. Uma vantagem dessa estratégia em relação à supervisionada é a possibilidade de realizar o ajuste dos parâmetros concomitantemente com a transmissão dos dados. Por outro lado, a etapa de ajuste dos parâmetros no caso cego é significativamente mais complexa [Haykin (1994)].

Percebe-se então que a equalização cega busca recuperar o sinal transmitido, através de um filtro no receptor, valendo-se apenas de amostras da saída do canal. Observa-se que a equalização e a estratégia de BSS são muito similares. A diferença básica é que, originalmente, a equalização é definida em um cenário SISO (Single-Input Single-Output) e se baseia em filtragem temporal, ao passo que a BSS aborda sistemas MIMO (Multiple-Input Multiple-Output) e se fundamenta em filtragem espacial ou espaço-temporal. Ainda assim, é possível formular o problema de equalização cega de canais SISO como uma tarefa de BSS [Hyvarinen et al. (2000), Attux et al. (2006)].

No que tange o problema de equalização cega de canais MIMO, pode-se afirmar que, em um âmbito teórico, essa situação praticamente se confunde com a formulação da BSS.

Nesse contexto, merece destaque o trabalho de Cavalcante [Cavalcante (2004)], que tratou um tópico relacionado à equalização MIMO, a detecção multiusuário, a partir de uma abordagem fundamentada em BSS. A particularidade nessa situação é que se trata de um canal de múltiplo acesso, ou seja, a transmissão das informações enviadas por diferentes usuários se dá num mesmo canal. Esse compartilhamento de recursos é possível devido à implantação de um esquema de múltiplo acesso como, por exemplo, as conhecidas estratégias FDMA, TDMA e CDMA.

2.4.4 Exploração Geofísica - Remote Sensing

Remote sensing é um processo de aquisição de informações sobre um objeto ou fenômeno através da gravação ou uso de sensores em tempo real que não estão em contato próximo com esse objeto. Dentre alguns exemplos de sensores é possível citar: instrumentos de observação da Terra, satélites de coleta de dados meteorológicos, plataformas para observação atmosférica e oceanográfica.

São definidos dois tipos de sensores remotos [Levada (2006)]:

1. Sensores passivos que detectam energia natural (radiação) emitida ou refletida por um objeto ou área em observação. Luz solar refletida é uma das fontes mais comuns de radiação medida por sensores passivos. Alguns exemplos desse tipo de sensor são: fotografia com filme, infra-vermelho, radiômetros.
2. Sensores de coleta ativos que emitem energia para escanear objetos e áreas nas quais um sensor passivo é usado para detectar e medir a radiação refletida. O RADAR é um exemplo de sensor ativo cujo princípio básico é medir o tempo entre a emissão e o retorno do sinal estabelecendo a localização, tamanho e direção de um objeto.
3. Sensores remotos são comumente usados para monitorar desmatamento em áreas como a floresta Amazônica; monitorar os efeitos da mudança climática em áreas glaciais, regiões árticas ou antárticas; medir a profundidade de oceanos e costas. Pode-se citar alguns exemplos de sistemas sensoriais (não traduzidos para manter o significado): Landsat Thematic Mapper, Seasat, Heat Capacity Mapping Mission, Space Shuttle Imaging Radar, Large Format Camera, Advanced Very High Resolution Radiometer, Coastal Zone Scanner, and Thermal Infrared Multispectral Scanner.

Os sinais coletados através dos sensores descritos acima apresentam informações misturadas de várias fontes ou até mesmo interferências do mesmo sinal no tempo. As técnicas de separação cega de fontes podem então ser aplicadas diretamente nesses tipos de problemas, como, por exemplo, no uso de sensores hiperespectrais [Borges e Marçal (2007)]. Um dos principais problemas da aplicação desses sensores está relacionado ao número grande de dimensões, fato que aumenta muito a complexidade computacional e justifica a aplicação de técnicas de BSS. Tais técnicas são usadas com dois objetivos: redução da dimensão do sistema e, posteriormente, reconhecimento das imagens obtidas.

A idéia é preservar o maior conjunto de informações possível do conjunto de dados original para, em seguida, aplicar técnicas de classificação de imagens.

2.5 Métodos Estatísticos

Esta seção descreve os conceitos dos métodos estatísticos utilizados. São analisados os conceitos de entropia, informação mútua e negentropia.

2.5.1 Entropia e Informação Mútua

Segundo a teoria de Shannon [Shannon (1948)], a entropia é uma medida da incerteza de uma variável aleatória. Matematicamente, pode-se definir, para um vetor de v.a contínuas x , sua entropia diferencial $H(x)$ como:

$$H(x) \triangleq -E \left\{ \ln [p_x(x)] \right\} = - \int_{-\infty}^{\infty} p_x(x) \cdot \ln(p_x(x)) dx \quad (2.3)$$

Outra definição importante é a da entropia condicional, que mede a incerteza do vetor de variáveis aleatórias x condicionada a um outro vetor y . A incerteza remanescente em x após observar y é dada por:

$$H(x|y) \triangleq -E \left\{ \ln [p_{x|y}(x|y)] \right\} = - \int_{-\infty}^{\infty} p_{x|y}(x|y) \cdot \ln(p_{x|y}(x|y)) dx dy \quad (2.4)$$

Em que:

$$p_{x,y}(x,y) = p_{x|y}(x|y)p_y(y) \quad (2.5)$$

é a função densidade de probabilidade conjunta de x e y . Então, é válida a seguinte relação [Cover e Thomas (1991)]:

$$I(x,y) = H(x) - H(x|y) \quad (2.6)$$

e tem-se a definição de informação mútua, que é a diferença entre as entropias do vetor de v. a. x e a incerteza que se tem após a observação de x por y . Em resumo, essa grandeza é capaz de medir a informação adquirida sobre x através da observação de y . A informação mútua assume valor zero se e somente se x e y são independentes. Isto significa que pode-se usar a informação mútua como uma medida da independência estatística. A informação mútua pode também ser definida entre componentes de um único vetor de variáveis aleatórias y [Haykin (1998)]:

$$I(y) = -H(y) + \sum_{i=1}^k H(y_i) \quad (2.7)$$

Essa expressão mostra que minimizar a informação mútua entre os componentes do vetor y é, como esperado, tornar a entropia de y o mais próximo possível da soma de suas entropias

marginais.

2.5.1.1 Divergência de Kullback-Leibler

Outra forma de definir a informação mútua é através da Divergência de Kullback-Leibler (KLD, do inglês Kullback-Leibler Divergence), que permite mensurar a similaridade entre funções estritamente positivas [Kullback and Leibler (1951)]. Esse tipo de medida é usualmente empregado para a comparação entre duas funções densidade de probabilidade. Matematicamente:

$$D(p_x(x) || g_x(x)) \triangleq \int_{-\infty}^{\infty} p_x(x) \cdot \ln \left[\frac{p_x(x)}{g_x(x)} \right] dx \quad (2.8)$$

onde

$p(x)$ e $g(x)$ são duas funções densidade de probabilidade estritamente positivas. $D(\cdot || \cdot)$ é a divergência de Kullback-Leibler.

Usando a KLD, pode-se escrever a informação mútua como:

$$I(x, y) = \int_{-\infty}^{\infty} p_{x,y}(x, y) \cdot \ln \left[\frac{p_{x,y}(x, y)}{p_x(x)p_y(y)} \right] dx dy = D(p_{x,y}(x, y) || p_x(x)p_y(y)) \quad (2.9)$$

2.5.2 Critérios Baseados em Não-Gaussianidade - Negentropia, Curtose

As medidas de não-gaussianidade permitem quantificar a proximidade de uma variável aleatória qualquer em relação a uma variável aleatória gaussiana.

2.5.2.1 Negentropia

A Negentropia é uma medida de gaussianidade de uma variável aleatória. Para defini-la, considera-se uma variável aleatória y e uma variável aleatória y Gaussiano com distribuição gaussiana e com média e matriz covariância iguais às do vetor y . Matematicamente, a negentropia é então dada por:

$$N_G(y) \triangleq H(y^{Gaussiano}) - H(y) \quad (2.10)$$

Como a entropia de uma variável gaussiana é maior que a de qualquer outra variável [Picinbono e Barret (1990)], pode-se dizer que a negentropia é uma medida sempre não negativa.

Reescrevendo (2.25) com o auxílio da divergência de Kullback-Leibler, tem-se:

$$N_G(y) \triangleq H(p_y(y)p_{y^{Gaussiano}}(y^{Gaussiano})) \quad (2.11)$$

Esse é um estimador ótimo cuja finalidade é medir a não-gaussianidade dos sinais [Picinbono e Barret (1990)]. Nesse caso, o objetivo é maximizar a Negentropia para que os sinais estimados não sejam gaussianos.

Na prática, há uma certa dificuldade na utilização da negentropia ao problema de BSS devido à necessidade de estimação de entropia como mostrado na equação (2.25). Por esse motivo, geralmente, faz-se necessária a utilização de uma aproximação baseada nos chamados momentos polinomiais [Hyvarinen (1999)], dada por:

$$N_G(y) = \alpha(E\{G_y\} - E\{G_{y^{Gaussiano}}\})^2 \quad (2.12)$$

onde:

- $G(\cdot)$ é uma função não-linear não-quadrática;
- α é uma constante;
- $y^{Gaussiano}$ é uma variável aleatória gaussiana de média zero e variância unitária.

2.5.2.2 Curtose

A Curtose é o grau de achatamento da distribuição, ou o quanto uma curva de frequência será achatada em relação a uma curva normal de referência.

Teremos, portanto, no tocante às situações de Curtose de um conjunto, as seguintes possibilidades:

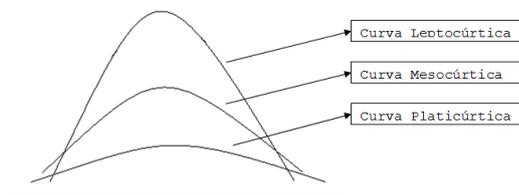


Figura 2.1: Curvas da Curtose

Logo, como vemos acima, uma curva (um conjunto) poderá ser, quanto à sua Curtose:

- Mesocúrtica: ou de curtose média! Será essa a nossa Curva Normal;
- Platicúrtica: é a curva mais achatada;
- Leptocúrtica: é a curva mais afilada!

A curtose é o cumulante de quarta ordem de uma variável aleatória [Papoulis (1993); Picinbono (1993)]. Matematicamente, para uma variável x , a curtose é dada por:

$$\frac{\mu^4}{\mu_2^2} = \frac{E(X - E(X))^4}{[E(X - E(X))^2]^2} \quad (2.13)$$

onde

$E(.)$ é o operador de Esperança matemática e μ é uma constante.

A curtose permite classificar uma função densidade de probabilidade em relação à uma função densidade de probabilidade gaussiana:

- Distribuição gaussiana: $K_{fxg} = 0$
- Distribuição sub-gaussiana: $K_{fxg} < 0$
- Distribuição super-gaussiana: $K_{fxg} > 0$

Conclui-se então que, ao maximizar o módulo da curtose num problema de separação de fontes, o sistema está, na realidade, diminuindo a gaussianidade dos sinais estimados. Há uma série de trabalhos que mostram a aplicabilidade dessa idéia em BSS [Zarzoso e Nandi (1998), Matsuoka et al. (2000), Papadias (1993), Papadias (2000), Sharma e Paliwala (2006)].

2.6 Sumario

Neste capítulo, nós vimos o problema de separação cega de fontes, inicialmente o seu histórico, por conseguinte seu modelo matemático. Foram apresentadas aplicações em diversos assuntos o que mostrou a versatilidade do uso das técnicas de BSS na resolução de problemas. Em seguida discutiu-se métodos baseados em estatísticas de segunda ordem aliados a BSS.

Capítulo 3

O Algoritmo Genético

Neste capítulo veremos a fundamentação da técnica de algoritmo evolucionário proposta por Rojas (2006), para a criação de um algoritmo genético canônico para inversão cega de canais lineares, o algoritmo genético de Holland, bem como uma outra técnica, o GAADT - algoritmo genético baseado em tipos abstratos de dados, utilizado para comparar a qualidade e eficiência dos resultados obtidos, como também é o objeto de estudo da presente tese .

A técnica de algoritmo genético procura utilizar o princípio evolucionário encontrado na natureza para encontrar soluções para problemas algorítmicos, baseando-se no modelo natural da sobrevivência do mais apto. Por esta razão, são categorizados dentro do contexto da inteligência artificial como algoritmos de busca cega que se utilizam ainda de uma abordagem de metáforas entre os conceitos da biologia e dos algoritmos de busca. Estas metáforas são exibidas a seguir, juntamente com uma possível interpretação para seu significado.

- *Indivíduo* - possível solução para o problema.
- *Cromossomo* - representação da estrutura de dados que constitui o indivíduo.
- *População* - conjunto de indivíduos, isto é, o conjunto de possíveis soluções.
- *Adaptação do indivíduo* - grau de satisfação do indivíduo para um dado problema.
- *Operadores genéticos* - operadores que atuam no espaço de estados do problema.
- *Seleção* - critério para determinar quais indivíduos serão reutilizados na busca de uma solução para o problema.

Agora será feita uma apresentação detalhada da técnica priorizando os pontos importantes a este trabalho, bem como uma demonstração formal de sua convergência.

3.1 O Algoritmo Genético de Holland

Na década de setenta, um estudante de doutorado em Ciência da Computação da Universidade de Michigan chamado John H. Holland, tentava desenvolver um método computacional que se prestasse para abordar fenômenos gerados por sistemas adaptativos complexos (LOPES, 2001 apud HOLLAND, 1975). Os fenômenos gerados por estes sistemas são aqueles cujos resultados dependem das interações não lineares entre vários agentes adaptativos. Por exemplo, a seca do nordeste brasileiro é um fenômeno deste tipo.

No decorrer do seu trabalho, Holland percebeu que existia uma nítida semelhança entre os fenômenos que estudava e o processo de evolução das espécies, pois assim como à interação entre os agentes adaptativos determinava o resultado dos fenômenos investigados por ele, a interação entre os fatores ambientais determinava a próxima população de uma determinada região. Com base nesta constatação, Holland propôs o modelo de algoritmo genético denominado de R , exibido na figura 3.1 e que procura simular o modelo evolutivo de Darwin na busca por soluções adequadas ao problema em questão.

```

início do procedimento R
   $t \leftarrow 0$ 
  gera  $P$ 
  calcule o valor da adaptação dos indivíduos de  $P$ 
  enquanto (condição de parada não for satisfeita) faça
     $t \leftarrow t + 1$ 
    SELEÇÃO( $P, P', h$ )
    CRUZAMENTO( $P', P'', p_c$ )
    MUTAÇÃO( $P', P'', p_m$ )
    INVERSÃO( $P', P'', p_i$ )
    SUBSTITUIÇÃO( $P, P'', k$ )
  fim do enquanto
fim do procedimento R

```

Figura 3.1: O algoritmo genético R

As variáveis deste algoritmo representam respectivamente o número de gerações t , a população na geração atual P , o número de indivíduos selecionados para gerar descendentes em cada iteração h , o número de novos indivíduos que devem fazer parte da próxima população k , a população dos indivíduos que irão gerar novos indivíduos P' , a população dos novos indivíduos gerados P'' , a probabilidade de cruzamento p_c , a probabilidade de mutação p_m e a probabilidade de inversão p_i . As funções *SELEÇÃO*, *CRUZAMENTO*, *MUTAÇÃO*, *INVERSÃO* e *SUBSTITUIÇÃO* são as operações responsáveis pela evolução da população trabalhada pelo algoritmo genético.

Os dados manipulados pelo algoritmo R representam a população e os indivíduos, sendo os últimos correspondentes a vetores binários de tamanho m , enquanto a população é representada por uma matriz de $n \times m$ onde cada linha da matriz representa um indivíduo. Uma população e seus indivíduos podem ser definidos da seguinte maneira:

Definição 3.1.1 Indivíduo - Um indivíduo i de uma população P é uma cadeia $\langle b_1, b_2, \dots, b_m \rangle$, tal que, $\forall j \in \{1, 2, \dots, m\} b_j \in \{0, 1\}$

Definição 3.1.2 População - Uma população P é uma matriz $\langle \langle b_{11}, \dots, b_{1m} \rangle, \dots, \langle b_{n1}, \dots, b_{nm} \rangle \rangle$, onde cada linha $\forall j \in \{1, 2, \dots, n\}$ representa um indivíduo da população.

Inicialmente, o algoritmo genético R inicializa a variável t com valor zero e a variável P com uma população construída aleatoriamente. Por exemplo, considere que P foi inicializado com a população mostrada na figura 3.2, composta por quatro cadeias de tamanho oito.

0	1	1	1	0	1	0	1
1	0	1	1	1	0	1	0
0	1	1	0	1	0	1	0
1	1	0	0	0	1	1	0

Figura 3.2: A população P

Em seguida, o algoritmo calcula o valor da função de adaptação (*fitness*) dos indivíduos dessa população. A adaptação de um indivíduo é definida pela função objetiva e pela intenção do problema. A função objetiva do problema é definida por $\hat{o} : D \rightarrow \mathbb{R}^+$ e recebe um elemento do domínio do problema D e retorna a medida do grau de satisfação do elemento ao problema em questão. A intenção é responsável por informar se o resultado desejado é o elemento com maior ou menor grau de satisfação.

Dessa forma se a intenção do problema for encontrar o elemento com maior grau de satisfação, então a função de adaptação f do algoritmo será igual a \hat{o} , do contrário a função de adaptação f do algoritmo será igual a $-\hat{o}$. Essa convenção é necessária uma vez que o algoritmo genético está sempre em busca do indivíduo mais adaptado.

Definição 3.1.3 Adaptação - função de adaptação f de um indivíduo é dada por uma função do tipo $f : I \rightarrow \mathbb{R}$ de forma que,

$$f(i) = \begin{cases} \hat{o}(\text{decod}(i)) & \text{se a intenção for encontrar o indivíduo mais adaptado.} \\ -\hat{o}(\text{decod}(i)) & \text{se a intenção for encontrar o indivíduo menos adaptado.} \end{cases} \quad (3.1)$$

onde I é o conjunto de todos os vetores binários de tamanho m considerados para o problema, decod é uma função que recebe um indivíduo e retorna o elemento do domínio do problema que este indivíduo representa.

Por exemplo, tome os indivíduos da população P anteriormente apresentada e suponha que representem a codificação binária dos números inteiros pertencentes ao intervalo $[0, 255]$. Suponha também que a função objetiva do problema está à procura do número inteiro cuja codificação em binário contenha a maior ocorrência da subcadeia “01”. Neste caso a função adaptação seria dada pelo número de ocorrências da subcadeia “01” em cada indivíduo. A figura 3.3 mostra o valor da adaptação dos indivíduos da população apresentada na figura 3.2.

Encerrado o cálculo da função de adaptação para todos os indivíduos de P , o algoritmo verifica se alguma condição de parada já foi satisfeita. Se sim ele encerra sua execução. Caso contrário, ele inicia seu laço enquanto-faça. A satisfação de sua condição de parada indica um resultado aceitável para o problema, podendo ser encontrar um valor de adaptação x ou atingir a j -ésima iteração.

0	1	1	1	0	1	0	1	$f(\langle 0,1,1,1,0,1,0,1 \rangle) = 3$
1	0	1	1	1	0	1	0	$f(\langle 1,0,1,1,1,0,1,0 \rangle) = 2$
0	1	1	0	1	0	1	0	$f(\langle 0,1,1,0,1,0,1,0 \rangle) = 3$
1	1	0	0	0	1	1	0	$f(\langle 1,1,0,0,0,1,1,0 \rangle) = 1$

Figura 3.3: Cálculo da função de adaptação de P

O laço enquanto-faça do algoritmo R compreende o processo evolutivo propriamente dito. A cada execução do laço, t é incrementado e a população alterada pela sucessiva ação dos operadores genéticos de cruzamento, mutação e inversão sobre os indivíduos da população P' , a qual é formada por indivíduos selecionados pela função de seleção aplicada a P .

A função de seleção recebe a população atual P e o parâmetro h , retornando uma população de tamanho h e formada pelos indivíduos mais adaptados de P .

Definição 3.1.4 Seleção - Caracteriza uma função de seleção uma função do tipo $P \times \mathbb{R}^+ \rightarrow P$, de forma que:

$$\begin{aligned}
 \text{seleção}(\langle \langle x_{11}, \dots, x_{1m} \rangle, \langle x_{n1}, \dots, x_{nm} \rangle \rangle, h) = & \{ \langle \langle y_{11}, \dots, y_{1m} \rangle, \dots, \langle y_{a1}, \dots, y_{am} \rangle \rangle / \\
 & (\forall j \in \{1, \dots, a\} \exists 1 \in \{1, \dots, n\} \langle y_{j1}, \dots, y_{jm} \rangle = \langle x_{11}, \dots, x_{1m} \rangle) \wedge \\
 & \exists \langle \langle z_{11}, \dots, z_{1m} \rangle, \dots, \langle z_{b1}, \dots, z_{bm} \rangle \rangle \in P (\forall j \in \{1, \dots, b\} \exists 1 \in \\
 & \{1, \dots, n\} \langle z_{j1}, \dots, z_{jm} \rangle = \langle x_{11}, \dots, x_{1m} \rangle) \wedge \\
 & (\forall j \in \{1, \dots, b\} \neg \exists 1 \in \{1, \dots, a\} \langle z_{j1}, \dots, z_{jm} \rangle = \langle y_{11}, \dots, y_{1m} \rangle) \wedge \\
 & \forall u \in \{1, \dots, a\} \forall v \in \{1, \dots, b\} f(\langle y_{u1}, \dots, y_{um} \rangle) \leq f(\langle z_{v1}, \dots, z_{vm} \rangle) \}
 \end{aligned} \tag{3.2}$$

Para entender melhor isto, tome como exemplo a população apresentada na figura 3.2 e considere $h = 2$. Neste caso a população auxiliar, P' , gerada, será formada pelos indivíduos $\langle 0, 1, 1, 1, 0, 1, 0, 1 \rangle$ e $\langle 0, 1, 1, 0, 1, 0, 1, 0 \rangle$.

População P'							
0	1	1	1	0	1	0	1
0	1	1	0	1	0	1	0

Figura 3.4: População auxiliar P' do algoritmo R

Estudemos agora a função de cruzamento. Esta é uma função do tipo um-ponto que recebe como entrada um par de indivíduos da população P' junto a um número aleatório r e retorna o par de indivíduos fornecidos caso $r < p_c$, ou um novo par de indivíduos caso $r \geq p_c$, tal que o novo par seja a combinação dos vetores dos indivíduos fornecidos, tal qual a definição a seguir.

Definição 3.1.5 Cruzamento - A função cruzamento é toda aquela do tipo um-ponto em que um-ponto : $I \times I \times \mathbb{R}^+ \rightarrow I \times I$, na qual:

$$\text{um-ponto}(\langle x_1, \dots, x_m \rangle, \langle y_1, \dots, y_m \rangle, r) = \begin{cases} (\langle x_1, \dots, x_m \rangle, \langle y_1, \dots, y_m \rangle) & \text{se } r < p_c \\ (\langle x_1, \dots, x_z, y_{z+1}, \dots, y_m \rangle, \langle y_1, \dots, y_z, x_{z+1}, \dots, x_m \rangle) & \text{se } r \geq p_c \end{cases} \quad (3.3)$$

onde I é o conjunto de todos os vetores binários de tamanho m considerados para o problema e z um número inteiro pertinente ao conjunto $\{1, 2, \dots, m\}$ escolhido de forma aleatória.

Como exemplo, observe o cruzamento dos indivíduos apresentados no exemplo anterior $\langle 0, 1, 1, 1, 0, 1, 0, 1 \rangle$ e $\langle 0, 1, 1, 0, 1, 0, 1, 0 \rangle$ conforme o ilustrado na figura 3.5, considerando $r = 4$.

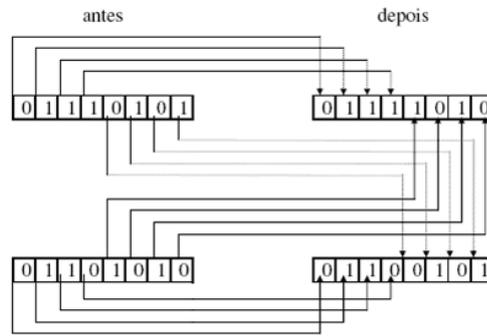


Figura 3.5: Operação de cruzamento de Holland

A mutação é outro dos operadores propostos por Holland para seu algoritmo R . Constitui-se de uma função complemento que recebe um indivíduo da população P' e um número aleatório r , e retorna o indivíduo fornecido se $r > p_m$, ou um novo indivíduo se $r \leq p_m$. Para o último caso, o novo vetor é obtido a partir de alterações feitas no indivíduo fornecido.

Definição 3.1.6 Mutação - O operador de mutação é definido como sendo uma função complemento do tipo, complemento : $I \times \mathbb{R}^+ \rightarrow I$, onde:

$$\text{complemento}(\langle x_1, \dots, x_m \rangle, r) = \begin{cases} \langle x_1, \dots, x_m \rangle & \text{se } r > p_m \\ \langle x_1, \dots, \bar{x}_{z_1}, \dots, \bar{x}_{z_2}, \dots, \bar{x}_{z_u}, \dots, x_m \rangle & \text{se } r \leq p_m \end{cases} \quad (3.4)$$

onde I é o conjunto de todos os vetores binários de tamanho m considerados para o problema, $\{z_1, z_2, \dots, z_u\}$ um subconjunto de $\{1, 2, \dots, m\}$ escolhido ao acaso e tal que $z_1 < z_2 < \dots < z_u$, e com os caracteres $\bar{x}_{z_1}, \bar{x}_{z_2}, \dots, \bar{x}_{z_u}$ representando o complemento binário do caractere que ocupa esta mesma posição no indivíduo $\langle x_1, \dots, x_m \rangle$.

Por exemplo, a figura 3.6 ilustra a ação da operação de mutação aplicada ao indivíduo $\langle 0, 1, 1, 1, 0, 1, 0, 1 \rangle$ do exemplo da definição 3.1.4, considerando $u = 2, z_1 = 3$ e $z_2 = 7$.

Continuando o estudo dos operadores, tratemos agora da operação de inversão. Esta constitui uma função inverte que recebe um indivíduo da população P' e um número aleatório r , e retorna o indivíduo fornecido se $r > p_i$, ou um novo indivíduo se $r \leq p_i$, novo indivíduo este, formado pela alteração da ordem dos caracteres que ocupam posições vizinhas no indivíduo fornecido. A operação de inversão é definida por:

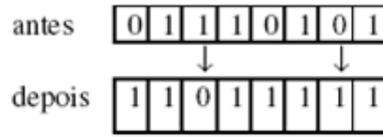


Figura 3.6: Operação de mutação de Holland

Definição 3.1.7 Inversão - É definida por uma função *inverte* do tipo $\text{inverte} : I \times \mathbb{R}^+ \rightarrow I$, de forma que:

$$\text{inverte}(\langle x_1, \dots, x_m \rangle, r) = \begin{cases} \langle x_1, \dots, x_m \rangle & \text{se } r > p_i \\ \langle x_1, \dots, x_{z_1-2}, x_{z_1-1}, x_{z_2}, x_{z_2-1}, x_{z_2-2}, \dots, x_{z_1}, x_{z_2+1}, \dots, x_m \rangle & \text{se } r \leq p_i \end{cases} \quad (3.5)$$

onde I é o conjunto de todos os vetores binários de tamanho m considerados para o problema, e $\{z_1, z_2\}$ um subconjunto de $\langle x_1, \dots, x_m \rangle$ escolhido aleatoriamente com $\{z_1 < z_2\}$.

Para facilitar a compreensão, tome o seguinte exemplo da ação de operação de inversão sobre o indivíduo $\langle 0, 1, 1, 0, 1, 0, 1, 0 \rangle$ do exemplo da definição 3.1.4, considerando $z_1 = 3$ e $z_2 = 7$.

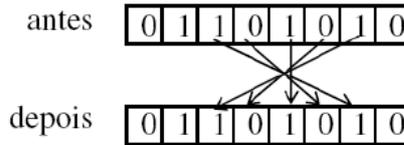


Figura 3.7: Operação de inversão de Holland

Em seguida, o algoritmo genético formará uma nova população aplicando a operação de substituição. A operação de substituição é uma função *substitui* que recebe a população atual P , a população dos indivíduos novos P'' e o parâmetro k , e retorna uma nova população formada pelos indivíduos mais adaptados da população P'' e pelos $n - k$ indivíduos mais adaptados da população P .

Definição 3.1.8 Substituição - É uma função de substituição toda aquela definida pelo tipo $\text{substitui} : P \times P \times \mathbb{R}^+ \rightarrow P$, onde:

$$\begin{aligned} \text{substitui}(\langle \langle a_{11}, \dots, a_{1m} \rangle, \dots, \langle a_{n1}, \dots, a_{nm} \rangle \rangle, \langle \langle b_{11}, \dots, b_{1m} \rangle, \dots, \langle b_{n1}, \dots, b_{nm} \rangle \rangle, k) = \\ \{ \langle \langle c_{11}, \dots, c_{1m} \rangle, \dots, \langle c_{n1}, \dots, c_{nm} \rangle \rangle / \\ (\forall u \in \{1, \dots, k\} \exists v \in \{1, \dots, n\} \langle c_{u1}, \dots, c_{um} \rangle = \langle a_{v1}, \dots, a_{vm} \rangle) \wedge \\ (\forall u \in \{k, \dots, (n-k)\} \exists v \in \{1, \dots, n\} \langle c_{u1}, \dots, c_{um} \rangle = \langle b_{v1}, \dots, b_{vm} \rangle) \wedge \\ (\exists \langle \langle d_{11}, \dots, d_{1m} \rangle, \dots, \langle d_{(n-k)1}, \dots, d_{(n-k)m} \rangle \rangle \in P \forall u \in \{1, \dots, (n-k)\} \\ \exists v \in \{1, \dots, n\} \langle d_{u1}, \dots, d_{um} \rangle = \langle a_{v1}, \dots, a_{vm} \rangle) \wedge \\ (\exists \langle \langle e_{11}, \dots, e_{1m} \rangle, \dots, \langle e_{k1}, \dots, e_{km} \rangle \rangle \in P \forall u \in \{1, \dots, k\} \\ \exists v \in \{1, \dots, n\} \langle e_{u1}, \dots, e_{um} \rangle = \langle b_{v1}, \dots, b_{vm} \rangle) \wedge \\ (\forall u \in \{1, \dots, (n-k)\} \exists v \in \{1, \dots, k\} f(\langle d_{u1}, \dots, d_{um} \rangle) \leq f(\langle c_{v1}, \dots, c_{vm} \rangle)) \wedge \\ (\forall u \in \{1, \dots, k\} \exists v \in \{k, \dots, (n-k)\} f(\langle e_{u1}, \dots, e_{um} \rangle) \leq f(\langle c_{v1}, \dots, c_{vm} \rangle)) \} \end{aligned} \quad (3.6)$$

Como exemplo dessa definição, seja a figura 3.9 representando a população formada pela ação da função de substituição sobre a população da figura 3.3 e a população da figura 3.8 para $k = 2$.

Holland encarava seu algoritmo como um algoritmo de busca, onde os indivíduos nada mais eram do que amostras do espaço de busca e os operadores como sendo mecanismos capazes de direcionar o algoritmo para melhores amostras, utilizando-se da preservação das características presentes nos indivíduos mais adaptados obtidas pela operação de cruzamento e da alteração de determinadas características presentes em tais indivíduos através das operações de mutação e de inversão. Holland também propôs um modelo formal do comportamento de seu algoritmo R , o qual denominou de teoria dos *schemata*, a qual será estudada agora.

0	1	1	1	1	0	1	0	$f(\langle 0,1,1,1,1,0,1,0 \rangle) = 2$
0	1	1	0	0	1	0	1	$f(\langle 0,1,1,0,0,1,0,1 \rangle) = 3$
1	1	0	1	1	1	1	1	$f(\langle 1,1,0,1,1,1,1,1 \rangle) = 1$
1	1	1	0	1	0	1	1	$f(\langle 1,1,1,0,1,0,1,1 \rangle) = 2$

Figura 3.8: População gerada pelos operadores de cruzamento, mutação e inversão

0	1	1	1	0	1	0	1	$f(\langle 0,1,1,1,0,1,0,1 \rangle) = 3$
0	1	1	0	0	1	0	1	$f(\langle 0,1,1,0,0,1,0,1 \rangle) = 3$
0	1	1	0	1	0	1	0	$f(\langle 0,1,1,0,1,0,1,0 \rangle) = 3$
1	1	1	0	1	0	1	1	$f(\langle 1,1,1,0,1,0,1,1 \rangle) = 2$

Figura 3.9: Nova população

3.2 Variações do algoritmo de Holland

A percepção evolutiva de um problema apresentada por Holland em seu algoritmo, tem servido de base em pesquisas na busca por soluções abstratas para problemas de alta complexidade.

Soluções iniciais utilizando o modelo de algoritmo proposto por Holland na abordagem de problemas de alta complexidade como otimização de funções, aprendizagem, planejamento estratégico e vários outros, eram limitadas de forma a se adaptar as restrições impostas pela representação do indivíduo e da população, pelo comportamento dos operadores genéticos e pelos efeitos das funções de seleção e substituição. O sucesso destas implementações, instigou seus usuários à construção de novos modelos a partir daquele proposto por Holland, de forma a adaptar o algoritmo para o problema em foco. Este capítulo apresenta algumas das variações que podem ser encontradas na literatura e consideradas interessantes para a proposta deste trabalho.

3.2.1 Representação do indivíduo

A representação do indivíduo em vetor binário obriga o usuário de AG a discretizar o resultado de seu problema, podendo isto acarretar em distorções na visão do algoritmo em relação ao problema e gerar o que é conhecido como indivíduo monstro. Este indivíduo, apesar de pertencer

ao conjunto dos vetores binários de tamanho m propostos para o problema, não representa um possível resultado para o problema, ou, direciona o sistema evolucionário a indivíduos que não apresentam o maior grau de adaptação.

McMahon (1991) adota a matriz booleana para representação do cromossomo ao tentar aplicar o algoritmo genético na solução de problemas, onde o relacionamento existente entre os elementos do domínio do problema deve ser considerado na construção do resultado, tal relacionamento é expresso através da associação das linhas e colunas desta matriz.

Definição 3.2.1 Matriz Booleana *Um cromossomo é uma matriz quadrática $\vec{i}_{j \times j}$, tal que, $\forall j_1, j_2 \in \{1, 2, \dots, j\} ((i_{j_1, j_2} = 0 \Rightarrow \text{não existe um relacionamento entre o elemento do domínio } j_1 \text{ e } j_2) \vee ((i_{j_1, j_2} = 1 \Rightarrow \text{existe um relacionamento entre o elemento do domínio } j_1 \text{ e } j_2))$*

Em uma outra solução proposta para este problema, está a de Antonisse de representar o indivíduo por cadeias alfanuméricas de comprimento fixo com e sem repetições, as quais aumentam o poder de representação e impedem que os operadores genéticos aplicados acabem por gerar indivíduos monstros, resolvendo em definitivo o problema para resultados que podem ser discretizados.

Definição 3.2.2 Vetor Alfanumérico com Repetição *Um cromossomo é um vetor $\vec{i}_1^m = \langle i_1, i_2, \dots, i_m \rangle$, tal que $(\forall j \in \{1, 2, \dots, m\}, i_j \in \{0, 1, \dots, 9\} \cup \{a, b, \dots, z\}) \wedge (\exists j_1, j_2 \in \{1, 2, \dots, m\} i_{j_1} = i_{j_2})$*

Definição 3.2.3 Vetor Alfanumérico sem Repetição *Um cromossomo é um vetor $\vec{i}_1^m = \langle i_1, i_2, \dots, i_m \rangle$, tal que $(\forall j \in \{1, 2, \dots, m\}, i_j \in \{0, 1, \dots, 9\} \cup \{a, b, \dots, z\}) \wedge (\forall j_1, j_2 \in \{1, 2, \dots, m\} i_{j_1} \neq i_{j_2})$*

Após alguns experimentos com estas representações Antonisse conclui que ela é incapaz de capturar a essência do resultado do problema, e passa a sugerir que o cromossomo seja representado por um vetor derivado de uma gramática.

Definição 3.2.4 Vetor Derivado de Uma Gramática *Um cromossomo derivado de uma gramática é um vetor $\vec{i}_1^m = \langle i_1, i_2, \dots, i_m \rangle$ tal que existe uma gramática G cujas regras de produção derivam a palavra i_1, i_2, \dots, i_m . Doravante o símbolo C_v será usado para representar o tipo cromossomo, cuja estrutura v é citada acima. Se v não for especificado significa que todos os tipos de representação para o cromossomo podem ser adotados.*

Não trataremos aqui de problemas para o qual o resultado não possa ser discretizado, como é o caso de problemas do domínio $\mathbb{R} \rightarrow \mathbb{R}$ dos números reais, que gera indivíduos monstros tanto para a representação de Holland com a de Fox e Antonisse devido à impossibilidade de se mapear em os valores da cadeia fixa do domínio, já que a representação decimal de um número real é infinita. Informações sobre os procedimentos para este caso podem ser encontrados nos materiais 3.2 constam na bibliografia deste trabalho.

3.2.2 Representação da População

A população segundo Holland é representada por um vetor de tamanho n , tal que cada linha desta matriz contém um cromossomo.

Michalewicz (1996) propõem a representação da população por uma lista de n -úplas contendo um cromossomo e dois valores inteiros, um para o parâmetro idade e outro para o parâmetro tempo de vida. Quando o cromossomo é gerado o valor do seu parâmetro idade é inicializado com o valor inteiro zero e cada vez que o laço enquanto-faça é executado, o valor do parâmetro idade é incrementado de uma unidade.

O valor do seu parâmetro tempo de vida é inicializado por uma das seguintes funções:

- proporcional $vida(i) = \min(\text{MinLT} + \eta \times \frac{f(i)}{\text{AvgFit}}, \text{MaxLT})$
- linear $vida(i) = \text{MinLT} + 2 \times \eta \times \frac{f(i) - |\text{MinFit}|}{|\text{MaxFit}| - |\text{MinFit}|}$
- bilinear $vida(i) = \begin{cases} \text{MinLT} + \eta \times \frac{f(i) - \text{MinFit}}{\text{AvgFit} - \text{MinFit}} & \text{se } \text{AvgFit} \geq f(i) \\ \frac{1}{2} \times (\text{MinLT} + \text{MaxLT}) + \eta \times \frac{f(i) - \text{AvgFit}}{\text{MaxFit} - \text{AvgFit}} & \text{se } \text{AvgFit} < f(i) \end{cases}$

onde MinLT é o tempo de vida mínimo permitido, MaxLT é o tempo de vida máximo permitido, η é o tempo de vida médio, $f(i)$ é o valor da adaptação do cromossomo i , AvgFit é o valor da adaptação média da população, MinFit é o valor da adaptação do cromossomo menos adaptado da população e MaxFit é o valor da adaptação do cromossomo mais adaptado da população.

Definição 3.2.5 *Lista* Uma população é uma lista $[i] = [(i_1, idade_1, vida_1), \dots, (i_n, idade_n, vida_n)]$, tal que, $\forall j \in \{1, 2, \dots, n\} \ i_j \in C_v$

Por conversão o símbolo P_{v_u} será usado para denotar o tipo população com representação v cujos cromossomos são representado pelo tipo u . Quando um dos índices não for declarado significa que a variável associada a ele pode apresentar qualquer uma das representações existentes.

3.2.3 Função de Adaptação

Uma função de adaptação que reflete o valor "real" do cromossomo para o problema nem sempre serve para guiar a evolução da população do algoritmo genético para outra população mais adaptada. Por isto, algumas metodologias capazes de auxiliar o usuário a construir uma função de adaptação adequada ao problema foram desenvolvidas. Entre as metodologias propostas estão os resultados da pesquisa de Cramer (1985) que propõe a decomposição da função de adaptação em submetas; de Goldberg (1989) que sugere uma função de adaptação cujo objetivo é penalizar os cromossomos que apresentam informações indesejáveis, ou que utiliza técnicas de aproximação de função para substituir a função de adaptação de um problema por uma versão

simplificada da mesma; e de Spears (1989) que adota um método aceitável para a otimização de expressões lógicas booleanas. Destas metodologias, a mais difundida na comunidade de computação evolucionária é a baseada no conceito de penalidade $f_{pen} : C_u \rightarrow \mathbb{R}$, cuja aplicação tornou-se comum depois da publicação do trabalho de J. T. Richardson (1989) que descreve um conjunto de regras para se construir funções de adaptação segundo esta metodologia.

Alguns exemplos de função de adaptação baseados no conceito de penalidade serão narrados a seguir.

A. Homaifar (1994) propõem que f_{pen} seja determinada pela análise da família de intervalos que limitam cada restrição do problema do seguinte modo $f_{pen}(i) = f(i) + \sum_{j_1=1}^w \sum_{j_2=1}^m R_{j_1, j_2}(i)$, onde f é a função de adaptação "real" do problema, w é o número de níveis de violação associado a cada posição do cromossomo, $R_{j_1, j_2}(i)$ é o coeficiente de penalidade do nível de violação j_1 associado à posição j_2 do cromossomo i .

Houck (1994) sugerem que f_{pen} seja calculada dinamicamente em função do número de iteração t pela seguinte formula $f_{pen}(i) = f(i) + (\gamma + t)^\alpha \sum_{j=1}^m R_j^\beta(i)$, onde f é a função de adaptação "real" do problema, $R_j(i)$ é o coeficiente de penalidade de violação das restrições associadas à posição j do cromossomo i , onde γ , α e β são constantes.

Skolnick (1993) descrevem uma função de adaptação baseada na penalidade dinâmica e heurística dos cromossomos, tal que $f_{pen}(i) = f(i) + r_1 \times \sum_{j=1}^m R_j(i) + \lambda(t, i)$, onde f é a função de adaptação "real" do problema, r_1 é uma constante, $R_j(i)$ é o coeficiente de violação das restrições associadas à posição j do cromossomo i , e $\lambda(t; i)$ é uma função heurística que retorna zero para o cromossomo i que não viola qualquer restrição, caso contrário ela retorna um valor diferente de zero e crescente em função do parâmetro t .

3.2.4 Operação de seleção

Definição 3.2.6 Elitista A operação de seleção elitista é uma função elitista do seguinte tipo: $elitista : P_{Vetor_u} \times \mathbb{N}^* \rightarrow P_{Vetor_u}$ $elitista(\vec{x}_1^n, h) = (\vec{x}_1^n(\vec{x}_1^n, 1, 1), n + 1 - h, n)$

Baker (1985) aconselha a adoção da seleção de rank linear ou não, a qual se baseia na utilização de uma função definida pelo usuário para determinar a probabilidade do cromossomo que ocupa a posição j da população p estar presente na população \check{p} .

Definição 3.2.7 Rank Linear A operação de seleção rank linear é uma função rank do seguinte tipo: $rank : C_v \rightarrow \mathbb{N}^*$ $rank(i_j) = r_1 \times f(i_j) + r_2$

onde r_1 e r_2 são constantes reais cujos valores não dependem do problema.

Definição 3.2.8 Rank Não Linear A operação de seleção rank não linear é uma função power do seguinte tipo: $power : C_v \rightarrow \mathbb{N}^*$ $power(i_j) = f(i_j)_1^r$

onde r_1 é uma constante real cujo valor depende do problema. E. (1990) apresenta a seleção de Boltzmann que seleciona um cromossomo a partir de dois cromossomos i_1 e i_2 pela avaliação do grau de entropia da busca realizada pelo algoritmo genético.

Definição 3.2.9 Boltzmann A operação de seleção Boltzmann é uma função boltzmann do seguinte tipo: $boltzmann : C_v \times C_v \rightarrow C_v$ $boltzmann(i_1, i_2) = \begin{cases} i_1 & \text{se } \frac{1}{1+e^{\circ}} \leq \frac{1}{2} \\ i_2 & \text{caso contrário} \end{cases}$

$$\text{onde } \circ = \frac{f(i_1) - f(i_2)}{\text{Temperatura}}.$$

Spears (1989) propõe a seleção de Monte Carlo, segundo a qual a relevância das informações codificadas em um cromossomo é diretamente proporcional ao valor da sua adaptação. Esta operação começa com a construção de uma roleta, que nada mais é do que um vetor, cujas células só podem ser ocupadas pelos cromossomos da população p . O número de células da roleta, ocupadas por um cromossomo i da população p , é calculado pela seguinte fórmula: $roleta(i_1) = \frac{f(i_1)}{\sum_{i_2 \in p} f(i_2)} \times 360^\circ$. A seleção do cromossomo é feita gerando-se aleatoriamente um número natural j que pertence ao conjunto $1, 2, \dots, w$, onde w é o número de regiões da roleta. Se a região j da roleta for ocupada por um cromossomo, então este será o cromossomo selecionado, senão gera-se aleatoriamente um outro número j até que uma região ocupada seja obtida.

Definição 3.2.10 Monte Carlo A operação de seleção Monte Carlo é uma função roleta do seguinte tipo: $roleta : V \times \mathbb{N}^* \rightarrow C_v$ $roleta(\vec{v}_1^w, j) = \begin{cases} v_r & \text{se } v_j \neq \vec{\pm} \\ roleta(\vec{v}_1^w, random(w)) & \text{caso contrário} \end{cases}$

onde $\vec{\pm}$ é o conteúdo de uma posição da roleta vazia e $random(w)$ é uma função de geração de números aleatórios pertencente ao intervalo $[1, w]$. Brindle (1981) sugere a seleção torneio, para permitir que a extinção das informações codificados nos cromossomos de baixa adaptação seja gradativa.

A seleção torneio pode ser assim descrita: dado um conjunto de cromossomos da população atual p escolhidos aleatoriamente, selecione o cromossomo mais adaptado deste conjunto.

Definição 3.2.11 Torneio A operação de seleção de torneio é uma função torneio do seguinte tipo

$$\begin{aligned} \text{torneio} : P_{u_v} \times \mathbb{N}^* &\rightarrow C_v \\ \text{torneio}(x, w) &= \text{maisAdaptado}(\text{selecione}(x, w)) \end{aligned}$$

onde maisAdaptado é a função que recebe um conjunto de cromossomo do tipo C_v e retorna um dos cromossomos mais adaptados do conjunto fornecido, e selecione é uma função que recebe uma população do tipo P_{u_v} e retorna um conjunto de cromossomos do tipo C_v com tamanho w formado por cromossomos da população fornecida escolhidos aleatoriamente.

3.2.5 Operação de Cruzamento

A operação de cruzamento de Holland recebe um par de cromossomos da população \check{p} , um ponto de corte j , com $j \in \{1, \dots, m\}$, e um número aleatório r_1 , com $r_2 \in [1, 100]$, e retorna o par de cromossomos fornecidos, se r_1 for maior que a probabilidade de cruzamento $prob_c$; caso contrário esta operação retorna um novo par de cromossomos formados pela combinação dos vetores dos cromossomos fornecidos.

Porém antes de apresentar a definição da operação de cruzamento de Holland, será apresentada a definição da função adição vetorial utilizada nas definições das operações de cruzamento que trabalham sobre cromossomos representados por um vetor de números.

A função soma vetorial recebe dois vetores de mesma dimensão e um número natural j , para indicar a célula dos cromossomos fornecidos que está sendo avaliada em cada momento, e retorna um vetor cujas informações contidas nas suas células é o resultado de uma avaliação das informações existente nestas mesmas células nos vetores fornecidos. Quando a função soma vetorial é chamada, o valor passado para o parâmetro j é "1".

Definição 3.2.12 Soma Vetorial A soma vetorial é uma função $\vec{+}$ do seguinte tipo:

$$\vec{+} : \vec{V} \times \vec{V} \times \mathbb{N}^* \rightarrow \vec{V}$$

$$\vec{+}(\vec{x}_1^w, \vec{y}_1^w, j) = \begin{cases} \vec{x}_1^w & \text{se } j > w \\ \vec{+}(\langle \dots, x_{y-1}, x_j + y_j, x_{j+1}, \dots \rangle, \vec{y}_1^w, j+1) & \text{se } j \leq w, x_j \neq \vec{+} \text{ e } y_j \neq \vec{+} \\ \vec{+}(\vec{x}_1^w, \vec{y}_1^w, j) & \text{se } j \leq w \text{ e } y_j = \vec{+} \\ \vec{+}(\langle \dots, x_{y-1}, y_j, x_{j+1}, \dots \rangle, \vec{y}_1^w, j+1) & \text{se } j \leq w, x_j = \vec{+} \text{ e } y_j \neq \vec{+} \end{cases}$$

Definição 3.2.13 Um Ponto de Corte A operação de cruzamento de um ponto de corte é uma função um do seguinte tipo: $um : C_v \times C_v \times \mathbb{N}^* \times \mathbb{R}^+ \rightarrow C_v \times C_v$

$$um(\vec{x}_1^m, \vec{y}_1^m, j, r_1) = \begin{cases} (\vec{x}_1^m, \vec{y}_1^m) & \text{se } r_1 > prob_c \\ (\vec{+}(\vec{x}_1^m, 1, j), (\vec{y}_1^m, j+1, m), 1), \\ \vec{+}(\vec{y}_1^m, 1, j), (\vec{x}_1^m, j+1, m), 1) & \text{se } r_1 \leq prob_c \end{cases}$$

onde v pode ser um dos seguintes tipos de representação de cromossomo: vetor binário, vetor alfanumérico com e sem repetição.

K. S. Tag (1996) propõe a operação de cruzamento de vários pontos de corte que recebe dois cromossomos, um conjunto de pontos de corte $\{j_1, \dots, j_w\}$ com $j_1, \dots, j_w \in \{1, \dots, m\}$ e $j_1 < \dots < j_w$, e um número aleatório r_1 , com $r_1 \in [1, 100]$; e retorna dois cromossomos formados por aproximadamente $\frac{j_w+1}{2}$ pedaços alternados de cada um dos cromossomos fornecidos, se r_1 é menor ou igual à probabilidade de cruzamento, caso contrário ela retorna os cromossomos fornecidos.

Definição 3.2.14 Vários Pontos de Corte A operação de cruzamento de vários pontos de corte é uma função vários do seguinte tipo: $vrios : C_v \times C_v \times \mathbb{P}(\mathbb{N})^+ \times \mathbb{R}^+ \rightarrow C_v \times C_v$

$$vrios(\vec{x}_1^m, \vec{y}_1^m, \{j_1, \dots, j_w\}, r_1) = \begin{cases} (\vec{x}_1^m, \vec{y}_1^m) & \text{se } r_1 > prob_c \\ (\vec{\uparrow}(\vec{x}_1^m, 1, j_1), (\vec{\uparrow}(\vec{y}_1^m, j_1 + 1, j_2), \dots)), & \\ (\vec{\uparrow}(\vec{y}_1^m, 1, j_1), (\vec{\uparrow}(\vec{x}_1^m, j_1 + 1, j_2), \dots))) & \text{se } r_1 \leq prob_c \end{cases}$$

onde v pode ser um dos seguintes tipos de representação de cromossomo: vetor binário, vetor alfanumérico com e vetor alfanumérico sem repetição, e $\mathbb{P}(X)$ é o conjunto das partes do conjunto X .

Syswerda (1989) sugere a operação de cruzamento proporcional, baseada na constatação de que uma combinação rápida das informações codificadas nos cromossomos fornecidos para o cruzamento de vários pontos de corte pode ser obtida quando a distância entre os pontos de corte considerados for equitativa, ou seja, o conjunto de pontos de corte $\{j_1, \dots, j_w\}$ é tal que: $j_1 = 1; j_w = m; e \sum_{a \in \text{Impar}} j_a - j_{a-1} = \sum_{b \in \text{Impar}} j_b - j_{b-1}$, onde Impar é o conjunto formado por todos os pontos de corte de índice ímpar com exceção de j_1 e j_w se w for ímpar, e Par é o conjunto formado por todos os pontos de corte de índice par com exceção de j_w se w for par.

Alleles (1985) adotam o operador de cruzamento parcialmente mapeado, para ser aplicado a problemas de otimização combinatorial, que recebe dois cromossomos, uma relação mapa, que relaciona as informações contidas na mesma posição dos cromossomos fornecidos, dois pontos de corte j_1 e j_2 , com $j_1 \in \{1, \dots, m\}$ e $j_2 \in \{j_1 + 1, \dots, m\}$, e um número aleatório r_1 , com $r_1 \in [1, 100]$; e que retorna dois cromossomos formados pelas informações presentes em um dos cromossomos fornecidos nas posições j_1, \dots, j_2 , e por preservar tanto quanto possível a ordem e as posições das informações contidas no outro cromossomo fornecido, se r_1 for menor ou igual à probabilidade de cruzamento, senão esta operação retorna os cromossomos fornecidos.

Para tanto a especificação deste operador é dividida em duas fases. Na primeira fase dois cromossomos são formados pela permuta das informações compreendidas entre as posições j_1 e j_2 dos cromossomos fornecidos e na segunda as informações repetidas são removidas.

Definição 3.2.15 Complemento A operação de permutação das informações contidas entres dois cromossomos submetidos ao operador de cruzamento parcialmene mapeado é realizada pela função comp do seguinte tipo: $comp : C_u \times C_u \times \mathbb{N}^* \times \mathbb{N}^* \rightarrow C_v \times C_v$

$$comp(\vec{x}_1^m, \vec{y}_1^m, j_1, j_2) = \begin{pmatrix} (\vec{\uparrow}(\vec{x}_1^m, 1, j_1 - 1), (\vec{\uparrow}(\vec{y}_1^m, j_1, j_2)), (\vec{\uparrow}(\vec{x}_1^m, j_2 + 1, m), 1), 1), \\ (\vec{\uparrow}(\vec{y}_1^m, 1, j_1 - 1), (\vec{\uparrow}(\vec{x}_1^m, j_1, j_2)), (\vec{\uparrow}(\vec{y}_1^m, j_2 + 1, m), 1), 1) \end{pmatrix}$$

onde u é o tipo vetor alfanumérico sem repetição e v é o tipo vetor alfanumérico com repetição.

3.2.6 Operação de mutação

Mutação é a operação responsável pelo surgimento de características nos indivíduos da população atual que anteriormente inexistiam. O comportamento da operação depende da forma de representação do indivíduo e do problema modelado pelo algoritmo genético.

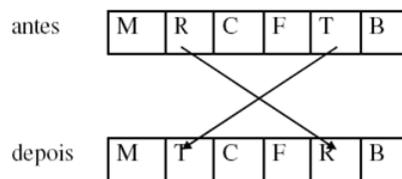


Figura 3.10: Mutao por troca

Mostraremos agora as operaes de mutao por complemento e por troca, as quais se aplicam respectivamente a indivduos representados com cadeias de caracteres alfanumricos com e sem repetio.

3.2.6.1 Mutao por complemento

operao trata-se na verdade de uma extenso daquela proposta por Holland. Esta recebe um indivduo $u = u_1u_2 \dots u_m$ e um alfabeto U , no qual, os elementos so indexados pelo conjunto dos nmeros naturais, e retorna um indivduo tal que $v = \langle v_1 \dots v_{x_1-1} \dots \mathbf{v_{x_1}} \dots v_{x_1+1} \dots v_{x_2-1} \dots \mathbf{v_{x_2}} \dots v_{x_2+1} \dots v_{x_m} \rangle$, onde os caracteres das posies x_z com $1 \leq z \leq m$ e que constam em negrito nesta cadeia, so ocupados pelo caractere que ocupa a posio θ do alfabeto, sendo θ um inteiro que somado a x_z  igual  cardinalidade do alfabeto U .

Por exemplo, seja a mutao por complemento aplicada ao indivduo $ABAA$ de uma funo que  dada por $f : \{A, B, C\}^* \rightarrow \mathcal{R}$ na posio quatro, obtm-se o indivduo $ABAC$, j que A  complemento de C e B  complemento de si mesmo.

3.2.6.2 Operao de mutao por troca

A mutao por troca recebe um indivduo $a = a_1a_2 \dots a_m$ e retorna um indivduo b tal que $b = \{a_1 \dots a_{x_1-1} \mathbf{a_{x_2}} a_{x_1+1} \dots a_{x_2-1} \mathbf{a_{x_1}} a_{x_2+1}\}$, onde x_1 e x_2 so nmeros pertencentes ao conjunto $\{1, 2, \dots, m\}$ selecionados aleatoriamente.

Como exemplo, observe na figura 3.10 a construo de um novo indivduo a partir da aplicao da mutao por troca no indivduo $MRCFTB$ pertencente  populao do problema do caixeiro viajante, considerando $x_1 = 2$ e $x_2 = 5$.

3.2.7 Probabilidade de Cruzamento e Mutao

A freqncia com que os operadores genticos de cruzamento e mutao so aplicados sobre os cromossomos da populao atual pode acelerar ou retardar a convergncia da populao manipulada pelo algoritmo gentico. Sendo a definio do valor da probabilidade de cruzamento e mutao uma dos tpicos investigados por alguns pesquisadores Booker (1987) Bramlette (1991) Davis (1985) Davis (1989).

Uma das primeiras abordagens apresentadas para atribuir dinamicamente a probabilidade dos operadores genticos de cruzamento e mutao foi variar linearmente $prob_c$ e $prob_m$ du-

rante a execução do algoritmo genético, tal que a cada iteração $prob_c$ é decrementada de uma quantidade r_1 e $prob_m$ é incrementada de uma quantidade r_2 Davis (1985) Syswerda (1989), com $r_1, r_2 \in \mathbb{R}^+$. Booker [16] sugere que somente a probabilidade de cruzamento deve ter seu valor atribuído dinamicamente em função da adaptação média da população.

Outra abordagem de atribuição dinâmica para a probabilidade de cruzamento e mutação foi proposta por Davis (1985). Segundo a qual a probabilidade de cada operador recebe um incremento de uma quantidade $r_1, r_1 \in \mathbb{R}^+$, toda vez que sua ação gera um cromossomo mais adaptado do que algum cromossomo da população atual, e um decremento de uma quantidade r_1 caso contrário.

Existe ainda um conjunto de trabalhos baseados na atribuição dinâmica da operação de mutação, que é decrementada por um fator exponencial durante a execução do algoritmo genético Michalewicz (1991) Bramlette (1991).

3.3 O Algoritmo Genético Baseado em Tipos Abstratos de Dados

3.3.1 Introdução

O modelo de Algoritmo Genético Baseado em Tipos Abstratos de Dados (GAADT), utilizado neste trabalho, foi originalmente proposto por Vieira LOPES (2003). No GAADT o processo de evolução ocorre a nível epistemológico do ambiente, definido como uma estrutura, sendo a população um de seus elementos. A estrutura do ambiente deve conter um conjunto de modelos para todas as entidades que o compõe, bem como as relações entre estas entidades. Segundo este modelo, as mudanças ambientais representam um marco para o início de um novo período de evolução durante o qual os habitantes da população do ambiente atual irão sofrer a ação dos operadores genéticos com o intuito de construir uma nova população formada somente por habitantes adaptados as novas condições ambientais vigentes. Neste momento, o ambiente atinge um período de equilíbrio evolutivo, durante o qual a sua população não evolui. O período de equilíbrio evolutivo é finalizado quando ocorre uma alteração ambiental, dando início a um novo ciclo de um período de evolução seguido por um período de equilíbrio evolutivo.

O resultado do problema para o ambiente atual é o habitante da população mais adaptado do período de equilíbrio evolutivo atingindo para este ambiente. As propriedades que diferem o GAADT dos outros modelos de algoritmo genético são:

- Representação dos habitantes da população estratificada nos tipos abstratos: base, gene e cromossomo. Os tipos abstratos base, gene e população não apresentam limites de tamanho, vínculos a um alfabeto pré-definido. A cada tipo abstrato do GAADT está associado um conjunto de regras de formação para impedir a geração de habitantes que não correspondem a um resultado para o problema a ser resolvido;

- Representação da população por um conjunto, evitando assim que os habitantes promissores sejam descartados, e a convergência prematura resultante da construção de populações formadas com várias cópias de um mesmo habitante;
- Garantia de que as características mais adaptadas ao ambiente dos habitantes pais sejam passadas para os cromossomos filhos durante a execução da operação de cruzamento; e
- Evita a perda das características adaptadas ao ambiente presente nos habitantes não adaptados ao ambiente pela ação da operação de mutação.

Na sua Tese de Doutorado Vieira LOPES (2003) propôs a Teoria de Processos Evolutivos utilizando os conceitos de monotonicidade e ponto fixo. Uma vez que as outras teorias desenvolvidas para mostrar a convergência de algoritmos genéticos: estavam comprometidas com a representação; não consideravam a ocorrência; a existência de vários resultados. Com a Teoria de Processos Evolutivos Vieira LOPES (2003) provou que o GAADT é um algoritmo convergente. Na seção 3.3.2 do presente capítulo, será apresentado o GAADT através dos seus tipos abstratos, das definidas de suas operações genéticas e da descrição do seu ambiente.

3.3.2 Tipos Básicos do GAADT

Neste trabalho, os cromossomos serão representados por seu material genético, o qual têm nas bases suas unidades elementares de formação. A única exigência para o tipo abstrato base do GAADT é que ele tenha pelo menos uma base b_λ , denominado de base-inócua, que será usado na construção do gene-inócua g_λ e do cromossomo-inócua c_λ . As constantes do GAADT base-inócua, gene-inócua e cromossomo-inócua deverão ter seu valor definido no momento da instanciação do GAADT a um dado problema.

Definição 3.3.1 (*Base*) - O tipo abstrato base B é o conjunto de todas as unidades genéticas elementares que podem ser usadas na formação do material genético dos cromossomos de uma população tal que $\exists b_\lambda \in B$.

Os elementos da base se agrupam em seqüência para formar os genes. As seqüências de bases que representam um gene válido para a construção do cromossomo devem satisfazer as restrições estabelecidas pelo conjunto de *Axiomas de Formação de Genes (AFG)*, o qual deverá ser definido para cada problema de acordo com a semântica atribuída ao tipo abstrato gene.

Definição 3.3.2 (*Gene*) - O tipo abstrato gene G é um conjunto de todas as seqüências formadas pelos elementos do tipo abstrato base construídos conforme descrito na Definição 3.3.1 que pertence ao conjunto *AFG*.

Os genes são agrupados em conjuntos para formar os cromossomos. O conjunto de genes $\{g_1, g_2, \dots, g_n\}$ válido para compor uma população tem que obedecer as regras estabelecidas

pelo conjunto de *Axiomas de Formação de Cromossomos (AFC)*, o qual deverá ser definido para cada problema de acordo com a semântica adotada para o cromossomo.

Definição 3.3.3 (*Cromossomo*) - *O tipo abstrato cromossomo C é um conjunto de todos os conjuntos de genes construídos conforme descrito na Definição 3.3.2 que obedece às condições estabelecidas pelo AFC.*

Os cromossomos são agrupados em conjuntos para formar uma população. Esta representação para a população irá garantir a imparcialidade na avaliação dos cromossomos que compõem a população, já que cada cromossomo só poderá ocorrer uma vez na população avaliada pelo GAADT em cada momento.

Definição 3.3.4 (*População*) - *O tipo abstrato população P é o conjunto de todos os um conjunto de cromossomos construídos conforme descrito na Definição 3.3.3.*

3.3.3 Operadores Genéticos do GAADT

O GAADT trabalha com os operadores genéticos de cruzamento e mutação. O operador genético de cruzamento caracteriza-se por combinar os genes de dois cromossomos para formar outros cromossomos formados somente pelos genes dominantes dos cromossomos pais. Enquanto que o operador genético de mutação caracteriza-se por alterar o conjunto de genes de um cromossomo para formar outro cromossomo tendo o cuidado de preservar cinquenta por cento dos genes do cromossomo original. Em ambos os operadores os novos cromossomos devem ter grau de adaptação for maior ou igual ao grau de adaptação de seus ancestrais.

Considere dois genes g_1 e g_2 , que expressem uma mesma característica com diferentes fenótipos, diz-se que um gene g_1 é mais adaptado ao ambiente do que o gene g_2 , se o grau de adaptação do gene g_1 for superior ou igual ao grau de adaptação do gene g_2 . No GAADT, o grau de adaptação de um gene é calculado pela função *grau* e a relação de que dois genes expressão uma mesma característica é obtido com o predicado *mesma*.

Definição 3.3.5 (*Grâu*) *O grau de adaptação de um gene é uma função grau do seguinte tipo: grau : $G \rightarrow K$ tal que, a cada gene g , $g \in G$, é associado um único número k , $k \in K$ (K é um corpo ordenado¹), chamado de grau(g) e que reflete, segundo a interpretação adotada para o problema, uma estratificação comparativa entre a adaptação dos genes.*

Por convenção, o grau de adaptação do gene-inócuo é igual ao elemento neutro do corpo K para a operação de adição, logo o grau de adaptação do gene-inócuo menor do que o grau de adaptação dos outros genes pertencentes ao tipo abstrato G .

¹Corpo ordenado - é uma estrutura algébrica, com duas operações, sem divisores próprios de zero e munido de uma ordem. Ex: $\langle \mathbb{R}, \leq, +, \times, 0, 1 \rangle$

O gene dominante para uma dada característica é o gene que corresponde à expressão fenotípica desta característica mais adaptada a configuração atual do ambiente. A função *domi* foi definida para informar o gene dominante de um par de genes, se os genes fornecidos não expressem uma mesma característica a função *domi* retornando o g_λ , caso contrário a função *domi* retorna o gene fornecido mais adaptado o ambiente atual.

Definição 3.3.6 (*Dominante*) - O gene dominante é uma função *domi* do seguinte tipo:

$$\begin{aligned} \text{domi} : G \times G &\rightarrow G \\ \text{domi}(g_1, g_2) &= \begin{cases} g_\lambda & \text{se } (g_1, g_2) \notin \text{mesma}, \\ g_1 & \text{se } (g_1, g_2) \in \text{mesma} \wedge \text{grau}(g_1) \geq \text{grau}(g_2), \\ g_2 & \text{se } (g_1, g_2) \in \text{mesma} \wedge \text{grau}(g_1) < \text{grau}(g_2). \end{cases} \end{aligned}$$

A produção de novos cromossomos durante o processo evolutivo de uma população serve para direcionar a busca por cromossomos mais adaptados através da transmissão das características de maior grau de adaptação presentes nos cromossomos da população atual. A adaptação de um cromossomo é dada pela função *adapt*.

Definição 3.3.7 (*Adaptação*) - A adaptação de um cromossomo é uma função *adapt* do seguinte tipo:

$$\begin{aligned} \text{adapt} : C &\rightarrow K \\ \text{adapt}(c) &= \sum_{g \in c} \Theta_{c,g} \times \text{grau}(g) \end{aligned}$$

onde $\Theta_{c,g}$ é o peso com o qual o gene g contribui para a adaptação do cromossomo c .

A operação de cruzamento recebe dois cromossomos pais, aptos a cruzarem, e retorna uma população de todos os cromossomos filhos formados somente com genes dominantes dos cromossomos fornecidos. No GAADT, a seleção dos cromossomos aptos a cruzarem é obtida com a função *sel* e o conjunto dos genes dominantes para cada característica presente nos cromossomos pais é construído com a função de fecundação *fec*.

A função de seleção recebe uma população P_1 e retorna a subpopulação de P_1 formada pelos cromossomos que satisfazem um requisito do problema r , descrito por uma fórmula em lógica de primeira ordem, o qual indica quando um dado cromossomo é considerado apto a cruzar.

Definição 3.3.8 (*Seleção*) - A seleção dos cromossomos que satisfazem um predicado r é uma função *sel* do seguinte tipo:

$$\begin{aligned} \text{sel} : (P) \times (P) &\rightarrow (P) \\ \text{sel}(P_1, r) &= P_1 \cap r. \end{aligned}$$

A função fecundação recebe dois cromossomos e retorna o conjunto de genes dominantes entre todos os genes dos cromossomos fornecidos.

Definição 3.3.9 (*Fecundação*) - A fecundação é uma função *fec* do seguinte tipo:

$$\begin{aligned} \text{fec} : C \times C &\rightarrow (G) \\ \text{fec}(c_1, c_2) &= \{g \mid \forall g_1 \in c_1 \forall g_2 \in c_2 (g = \text{domi}(g_1, g_2))\} \end{aligned}$$

Considere que o conjunto de cromossomos aptos da população atual P_1 é a população $P_{1_{aptos}}$. A partir da população $P_{1_{aptos}}$ obtem-se os conjuntos $MACHO$ e $FEMEA$, formados da seguinte forma: $MACHO = sel(P_{1_{aptos}}, M)$ e $FEMEA = sel(P_{1_{aptos}}, F)$, onde M e F são dois predicados sobre o tipo população pertencentes ao conjunto de requisitos do ambiente Rq , escritos em uma linguagem de primeira ordem.

Note que, dependendo da especificação dos requisitos do ambiente M e F , a reprodução gerada pelo GAADT pode ser sexuada quando $M \cap F = \emptyset$, assexuada quando $M = F$, e mista quando $M \cap F \neq \emptyset$ e $M \neq F$.

O operador genético de mutação, definido para o GAADT, é composto pelas função de troca, que remove um conjunto de genes do cromossomo de origem e lhe adiciona outro conjunto de genes.

Definição 3.3.10 (*Troca*) - A troca é uma função troc do seguinte tipo:

$$troc : C \times (G) \times (G) \rightarrow C$$

$$troc(c, G_1, G_2) = \begin{cases} (c \cup G_1) - G_2 & \text{se } c \cup G_1 \in AFC \wedge (c \cup G_1) - G_2 \in AFC, \\ c \cup G_1 & \text{se } c \cup G_1 \in AFC \wedge (c \cup G_1) - G_2 \notin AFC, \\ c - G_2 & \text{se } c \cup G_1 \notin AFC \wedge c - G_2 \in AFC, \\ c & \text{se } c \cup G_1 \notin AFC \wedge c - G_2 \notin AFC. \end{cases}$$

Definição 3.3.11 (*Mutação*) - A mutação é um predicado $mut \subseteq (P)$, tal que:

$$mut(c_1) = \{c_2 | \exists G_1, G_2 : (G) ((\#G_1 \leq \#c_1 \div 2) \wedge (\#G_2 \leq \#c_1 \div 2) \wedge (troc(c_1, G_1, G_2) = c_2) \wedge (adapt(c_2) \geq adapt(c_1)))\}$$

3.3.4 Ambiente do GAADT

Um algoritmo genético opera sobre populações de cromossomos que evoluem de acordo com as características de um ambiente A . Um ambiente A é uma 8-tupla $\langle P, (P), Rq, AFG, AGC, Tx, \Sigma, P_0 \rangle$, onde:

- P é a população,
- (P) é o conjunto potência de P ,
- Rq é o conjunto dos requisitos (características expressas através de fórmulas numa linguagem de primeira ordem) do problema que influenciam a genealogia da população P ,
- AFG é o conjunto de axiomas de formação dos genes dos cromossomos da população P ,
- AFC é o conjunto de axiomas de formação dos cromossomos da população P e
- Tx é o conjunto de pares de cromossomos (x, y) , onde x é um cromossomo construído a partir do cromossomo y , pela ação da operação de cruzamento ou mutação, registrando

desta forma a genealogia dos cromossomos pertencentes às populações geradas pelo GAADT durante a sua execução,

- Σ é o conjunto de operadores genealógicos que atuam sobre a população P ,
- P_0 é uma sub-população pertencente a (P) , chamada de população inicial, com no mínimo um cromossomo.

3.3.5 Algoritmo do GAADT

O GAADT é uma função $GAADT$ que recebe a população P_0 e, depois de submetê-la à simulação de um processo evolutivo, devolve uma população P_t . Os cromossomos da população P_t são os cromossomos das populações P_0, P_1, \dots, P_{t-1} que melhor satisfazem os requisitos do problema Rq . Diz-se então que a população P_t evoluiu da população P_0 .

A preservação e morte dos cromossomos da população atual P_t trabalhada pelo GAADT é orientada por um predicado unário denominado de p_{corte} , tal predicado pertencente ao conjunto de requisitos do problema Rq , que atua sobre os cromossomos de P_t . Os cromossomos que satisfazem o predicado p_{corte} irão fazer parte da população P_{t+1} , enquanto os outros cromossomos da população P_t irão morrer. Os cromossomos mortos podem ser recuperados através da taxonomia Tx dos cromossomos da população atual para evitar que eles apareçam novamente nas próximas iterações da função $GAADT$. Esta restrição atende ao entendimento do processo de evolução darwinista, que não contempla a possibilidade de uma espécie extinta voltar a aparecer num outro momento futuro.

Os critérios de parada adotados pela função $GAADT$ são: a) o número máximo de iterações desejadas, b) valor da adaptação dos cromossomos considerado satisfatório para o resultado do problema em análise. Estes critérios também fazem parte do conjunto de requisitos do problema Rq .

Definição 3.3.12 ($GAADT$) - *O GAADT é uma função GAADT do seguinte tipo:*

$$GAADT : A \rightarrow A$$

$$GAADT(P_t) = \begin{cases} P_{otm} & \text{se } P_{otm} = \{c | \forall c : P_t(adapt(c) \geq k)\} \neq \emptyset, \\ P_{t+1} & \text{se } t + 2 = T, \\ GAADT(P_{t+1}) & \text{caso contrário.} \end{cases}$$

onde $P_{t+1} = cruz(a, b) \cup mut(c) \cup p_{corte}(P_t)$ com e $a, b, c \in P_t$, P_0 é a população inicial considerada, $k \in K$ é um valor imposto pelo ambiente A , como critério de aceitação de cromossomos em P_t que satisfazem o problema e $T \in \mathbb{N}$ é um número dado como critério de satisfação do número de iterações.

Apresentado o que é o algoritmo genético, como funciona e os teoremas que o suportam.

Capítulo 4

Especificação dos algoritmos evolutivos para o problema de Separação Cega de Fontes

4.1 Introdução

A presente dissertação foi baseada em problema proposto por Rojas (2006), onde foi concebido um sistema que permitia que os coeficientes do filtro fossem ajustados através de processo evolutivo. O processo evolutivo foi proposto utilizando como algoritmo o algoritmo genético de Holland. Neste trabalho foi definido o modelo do GAADT para fins de comparação, utilizando os métodos estatísticos não só de maneira disjunta como no trabalho original, mas optou-se em utilizar também de forma conjunta para verificar se ocorrem melhorias na qualidade dos resultados obtidos.

4.2 Especificação do GAADT

4.2.1 Introdução

O GAADT foi proposto com o objetivo de ser aplicado a diferentes problemas, a qualidade dos resultados encontrados depende da especificação dos tipos abstratos manipulados por este algoritmo, da adequação dos operadores genéticos e das restrições impostas pelo ambiente. Portanto, o GAADT, quando instanciado para um problema particular, deve preservar todas as suas propriedades fundamentais. Em Vieira LOPES (2003) é apresentada uma metodologia para modelar os requisitos do problema, baseada em 7 passos:

1. Examinar o problema, com o objetivo de determinar qual a natureza da solução requerida no sentido de orientar a escolha metafórica dos tipos básicos: base, gene e cromossomo;
2. Identificar o gene inócuo g_λ e a população inicial P_0 ;

3. Definir as funções grau de adaptação do gene *grau* e peso do gene no cromossomo Θ , e a relação *atributoRelevante*;
4. Definir os elementos de $Rq(F, M \text{ e } r)$ usados pela função *cruz* de acordo com o problema;
5. Definir os elementos de $Rq(p_{corte}, t \text{ e } k)$ utilizados pela função *GAADT*;
6. Construir o algoritmo;
7. Verificar os eventuais ajustes relativos à verificação dos tipos manipulados pelas funções, relações e operações manipuladas pela função *GAADT*, e a modelagem do problema através da definição dos componentes do ambiente.

O restante da seção descreve como o GAADT pode ser aplicado e refinado ao problema de separação cega de fontes a partir das informações propostas por Rojas (2006) para criação de um algoritmo genético canônico para inversão cega de canais lineares.

4.2.2 Tipos Básicos

O tipo base para a construção do vetor de pesos para o sinal de entrada do sistema é o conjunto B_{Sinal} formado por números aleatórios que pertencem ao domínio $[-1,1]$ para cada sinal de entrada e o conjunto B_{λ} contendo os pesos inócuos.

Definição 4.2.1 (*Base*) - O tipo abstrato base $B = B_{Sinal} \cup B_{\lambda}$.

Os elementos do conjunto B_{Sinal} são pares (*peso, metodoEstatistico*), onde *peso* pertence ao intervalo $[-1,1]$ e

$metodoEstatistico \in \{Curtose, InformaoMutua, Negentropia, Curtose + informaoMutua, Curtose + Negentropia, InformaoMuta + Negentropia, Curtose + InformaoMuta + Negentropia\}$.

Os elementos do conjunto B_{Sinal} contém valores a partir dos quais será possível mensurar o sinal original a partir da minimização dos valores extratidos dos metodos estatisticos utilizados.

No processo de formação dos genes é passado por referência o número de coeficientes que serão utilizados, estes coeficientes formam o conjunto B_{Sinal} na interpretação adotada para os elementos.

Os elementos do conjunto B_{λ} são pares (*peso, metodoEstatistico*), onde $peso \in \{\lambda\}$ e $metodoEstatistico \in \{Curtose, InformaoMutua, Negentropia, Curtose + informaoMutua, Curtose + Negentropia, InformaoMuta + Negentropia, Curtose + InformaoMuta + Negentropia\}$.

Assim a característica (gene) relevante para o problema tratado nessa dissertação são os pares de pesos consecutivos de uma mesma configuração de sinal.

Definição 4.2.2 (*Gene*) - O tipo abstrato gene G é um conjunto de todas as seqüência $\langle b_1, b_2, \dots, b_n \rangle$ formadas pelos elementos do tipo abstrato base conforme descrito na Definição 4.2.1 que satisfaz o conjunto AFG .

Os axiomas do conjunto AFG estabelecem que:

- para todo gene $g = \langle b_1, b_2, \dots, b_n \rangle$, os elementos de b_1 á b_n devem conter o mesmo metodo estatístico, ou seja, $afg_1 = \forall g = \langle b_1, b_2, \dots, b_n \rangle \in G (\text{metodoEstatistico}_1 = \text{metodoEstatistico}_2, \dots, \text{metodoEstatistico}_1 = \text{metodoEstatistico}_n)$;

O conjunto gene inócuo (G_λ) é formado por todas os pares $g_\lambda = \langle b_1, b_2 \rangle$ tais que $peso_1 = peso_2 = \lambda$. Os elementos deste conjunto serão representados por g_λ . Deve-se observar que os elementos do conjunto G_λ satisfazem o $AFG = \{afg_1\}$ como exigido pelo GAADT.

O cromossomo é instanciado como o conjunto de pesos para cada configuração do sinal que conduzem a inferência da aproximação ao sinal original.

Definição 4.2.3 (*Cromossomo*) - O tipo abstrato cromossomo C é um conjunto de todos os conjuntos de genes construídos conforme descrito na Definição 4.2.2 que obedece às condições estabelecidas pelo AFC.

Assim, o conjunto de AFC é especificado como:

- para uma dada configuração o número de metodos estatísticos tem que ser igual a sete, ou seja, $afc_1 = (\forall c \in C (\forall g_i = \langle b_{1i}, b_{2i}, \dots, b_{ni} \rangle \in c (\text{tamanho}(\{g_j | \forall g_j = \langle b_{1j}, b_{2j}, \dots, b_{ni} \rangle \in c - \{g_i\}\}) \leq 1)))$;

O conjunto cromossomo inócuo, denotado por C_λ , é formado por todos os conjuntos de genes inócuos, que satisfazem as restrições do AFC. Os elementos deste conjunto serão representados por c_λ .

A população é formada pelo conjunto de cromossomos que justificam os pesos extraídos através da minimização obtida pelos metodos estatísticos.

Definição 4.2.4 (*População*) - O tipo abstrato população P é o conjunto de todos os um conjunto de cromossomos construídos conforme descrito na Definição 4.2.3.

4.2.3 Operadores Genéticos

Assim como a especificação dos tipos abstratos dados: base, gene, cromossomo e população, preservaram as exigências contidas na definição do GAADT. A especificação das funções e relações necessárias para o cálculo da função GAADT devem atender a todas as pré-condições de sua definição original apresentada no Capítulo 3.3. Nesta seção será apresentada a definição somente das funções e relações cuja especificação para o presente problema é mais concreta do que a definição original, ficando subentendido que as funções e relações que não forem redefinidas nesta seção irão preservar sua definição original.

O grau de adaptação do gene deve considerar para cada configuração cadastrada o método estatístico ao qual vai ser submetido, bem com o número de coeficientes empregados no processo. Logo para calcular o grau de adaptação do gene é indentificado o método e exposto ao cálculo da função de fitness correspondente.

Definição 4.2.5 (*Grau*) *O grau de adaptação de um gene é uma função grau do seguinte tipo:*

$$\text{grau} : G \rightarrow \mathbb{R} \quad \text{adapt}(g) = \sum_{a \in \mathbb{I}} \text{padrao}(g, a)$$

onde \mathbb{I} contém o nome de todos métodos estatísticos cadastrados no sistema.

A especificação do GAADT para a mensuração dos pesos para aproximação do padrão do sinal original, trabalha com os operadores genéticos de cruzamento e mutação. A reprodução por cruzamento será assexuada, ou seja, será permitido que todo par de cromossomos da população atual cruzem entre si. O processo de cruzamento foi definido da seguinte forma: dados dois cromossomos, verifica-se gene a gene quais os mais aptos ao meio, cruzando os genes com as melhores características. A Mutação foi definida da seguinte forma: dado um cromossomo verifica-se qual gene que tem a característica que menos influi no meio ambiente, encontrado este gene é submetido ao processo de mutação ao qual o peso deste gene é modificado de forma gradual afim de se obter um maior grau de adaptação.

O parâmetro p_{corte} é a adaptação média da população atual. O critério de parada por valor de adaptação aceitável para o cromossomo mais adaptado não foi usado. O critério de parada número de iterações desejadas considerado neste trabalho foi 120 gerações.

Capítulo 5

Estruturação de um Sistema de Separação de Fontes para Recuperação de Sinais

5.1 Introdução

BSS é um problema clássico em processamento de sinal, ela está relacionada com a recuperação de um conjunto de sinais da fonte utilizando amostras das misturas dos sinais originais, onde não se têm um conhecimento preciso das misturas e fontes. A recuperação deste conjunto de fontes é feita através de estimativas, e utiliza o critério de independência entre os componentes para maximizar a semelhança entre o resultado e o valor esperado. Como citado na 4.1 a presente dissertação foi baseada em problema proposto por [Rojas, F., Solé-Casals, J., Monte-Moreno, E., Puntonet, C.G., Prieto, A., 2006], onde foi concebido um sistema que permitia que os coeficientes do filtro fossem ajustados através de processo evolutivo, utilizando como função de ativação métodos estatísticos. Na presente dissertação foi desenvolvido um sistema web com parâmetros configuráveis. O sistema possui a implementação dos algoritmos genéticos de Holland e Baseado e Tipos Abstratos de Dados - GAADT. Em ambas implementações utilizam como funções de fitness os métodos estatísticos tanto na forma disjunta quanto em conjunto, totalizando sete funções. Nas seções a seguir será exibido o sistema desenvolvido, mostrado as tecnologias utilizadas bem como suas características.

5.2 Tecnologias Utilizadas

Para concepção do sistema, foi implementado uma aplicação web através da linguagem de programação Python utilizando o framework Django para desenvolvimento web desta aplicação. O banco de dados escolhido foi o mysql pela simplicidade da aplicação. A geração dos

graficos apartir dos resultados utilizaram a biblioteca Matplotlib do Python.

5.3 O Sistema

Nesta seção serão detalhadas as áreas do sistema, para melhor entendimento do leitor.

5.3.1 Tela de Acesso ao sistema

A primeira tela ao qual o usuario do sistema se depara é a tela de acesso 5.1, onde é requisitado um nome de usuario e sua respectiva senha. Ao digitar os dados requeridos será redirecionado automaticamente, caso os dados estejam corretos, o contrario será exibida a tela de erro 5.2, onde ficará esperando que o usuario retifique as informações.

5.3.2 Tela Inicial

Após ter efetuado o acesso 5.1, será exibida a tela inicial do sistema, onde o usuario encontrará varios modulos, tanto para configuração do sistema quanto para cadastro dos parametros do sistema proposto na tese. Na figura 5.3 podemos ver três modulos que são: Programas, Autenticação e Sistema.

5.3.2.1 Modulo Programas

No modulo Programas estão agupados os programas Algoritmo genético e Resultados. Na subsecoes a seguir serão explicados com detalhe a utilização de cada um.

5.3.2.2 Algoritmo Genético

Neste programa 5.4 estão listadas todas as configurações que se encontram cadastradas no sistema. Ao usuario é dada a opção de selecionar apenas uma configuração, feito isto o proximo passo é clicar no botão rodar. Quando o botão rodar é pressionado é enviada uma requisição via parametros GET, que invoca a class Algoritmo Genético passando por referencia a configuração selecionada, e apartir deste ponto é instanciada a configuração e é selecionado o algoritmo correspondente apartir da configuração. O algoritmo escolhido é então executado e a cada geração o algoritmo salva as informações referentes a adaptação e tempo de execução em banco de dados. Ao final da execução o algoritmo direciona o sistema para a tela de resultados exibindo os resultados referentes a amosra corrente.

5.3.2.3 Aplicações

Nesta tela são listadas todas as configurações cadastradas no sistema na forma de links como mostrado na figura 5.5, e ao clicar sobre um desses links são carregadas todas as amostras. Para cada amostra são exibidos links para diversos tipos de gráficos como mostrado na figura 5.6. Ao clicar no link para gerar os gráficos, será plotado o gráfico correspondente como na figura 5.7

5.3.2.4 Módulo Sistema

Neste módulo encontramos os programas para cadastro de configuração que será utilizada na execução do algoritmo genético como também para cadastro de resultados.

5.3.2.5 Configuração

Nesta programação temos o cadastro efetivo das configurações que serão utilizadas na execução do algoritmo genético. Vemos na figura 5.8 os parâmetros que são preenchidos pelo usuário para interagir com o algoritmo.

5.4 Resultados

Nesta seção vamos expor os resultados obtidos através do sistema.

Foram executadas simulações em dois cenários distintos, no primeiro cenário foi introduzido o sinal de entrada de ECG para avaliar a eficácia do algoritmo proposto na dissertação em comparação com o algoritmo genético de Holland utilizado para separar o sinal do ruído, no segundo cenário foi empregado na caracterização do filtro inverso, num problema de desconvolução cega.

Ambos os casos dos cenários, Separação do Sinal de ECG e do ruído e do problema do filtro inverso, foram parte integrante de artigos aceitos e publicados nos anais dos congressos de Valter Ramos (2010), Valter Ramos (2011a) e Valter Ramos (2011b). Onde Valter Ramos (2011a) foi publicado também no IEEE explore.

5.4.1 Cenário um

No cenário de avaliação do algoritmo foi utilizado como sinal de entrada uma amostra de sinal de ECG obtido através da base de dados do MIT-BIH distribuída gratuitamente pela PhysioNet. A base de dados do MIT-BIH contém 48 registros de exame ECG com duração de trinta minutos e amostrados a uma frequência de 360 Hz a uma resolução de $4,88 \mu\text{V}$ sobre uma faixa de variação de 10 mV. A amostra utilizada foi a de número 100.

A seguir serão apresentados os resultados obtidos nesses casos.

A amostra da base de dados do MIT-BIH vem dividida em dois arquivos, um contendo as informações de indexação do posicionamento de ocorrência de cada onda (começo, pico e final) como também o tempo em milissegundos em que a onda ocorre, pode-se observar na figura 5.10. O segundo arquivo contém o posicionamento e os pontos utilizados para plotagem das ondas, onde pode-se observar na figura 5.11.

Quando se inicia o processo é adicionado ruído branco ao sinal de ECG, com isso obtemos o sinal original e o sinal com ruído o qual iremos aplicar o algoritmo afim de remover o ruído através de processo evolucionário, testificando a eficácia da técnica proposta.

O sinal com ruído é então submetido ao algoritmo que após a realização de 1000 gerações com o algoritmo de Holland e a convergência do GAADT após a estagnação por 10 gerações consecutivas, mostrou que foi possível obter boas estimativas que se aproximaram muito da fonte original, o qual pode ser observado na tabela 5.4, que mostra a acurácia que cada algoritmo com configurações diferentes alcançaram.

5.4.2 Cenário dois

5.4.2.1 Problema da DeConvolação

Supõe-se que a entrada do sistema $S = s(t)$ é uma fonte desconhecida não Gaussiana independente e processos identicamente distribuídos, e que o subsistema h é um filtro desconhecido linear inverso. A fim de estimar $s(t)$ pela observação a partir da saída do sistema é necessário calcular cegamente a estrutura inversa composto de um filtro linear w . Portanto, sejam s e e são os vectores de dimensão infinita, cuja as e -nesimas entradas são $s(t)$ ou $e(t)$, respectivamente. A matriz de entrada-saída desconhecida pode ser escrita como:

$$e = Hs; \quad (5.1)$$

Where:

$$H = \begin{pmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & h(t+1) & h(t) & h(t-1) & \dots \\ \dots & h(t+2) & h(t+1) & h(1) & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \quad (5.2)$$

A matriz H é não-singular, desde que o filtro h é inversível, ou seja, satisfaz $(h^{-1}(t) * h(t) = h(t) * h^{-1}(t) = \delta(t))$, onde $\delta(t)$ é o impulso Dirac. A dimensão infinita dos vectores e da matriz é devido à falta da ordem do filtro. Se o h é um filtro impulso de resposta finito (FIR) filtro de ordem N_h , então a dimensão da matriz pode ser reduzida ao tamanho N_h .

5.4.2.2 Estudo de caso

O estudo de caso do cenário dois considera a aplicação da técnica proposta na dissertação para o problema da inversão cega do canal linear.

Ambos algoritmos (GAADT e Holland) implementados aqui usam uma população de 100 cromossomos. O cromossomo aqui é apresentado como um vector binário de tamanho n , com cada conjunto de posições Y que codificam um coeficiente de polinómio pertencente ao intervalo $[-1, 1]$.

O algoritmo genético implementado é o seguinte:

A. Propriedades do Algoritmo

- Tamanho da população: 100 indivíduos.
- Probabilidade de crossover: 0,8.
- Probabilidade de mutação: 0,075.
- Condição de parada do GA: Chegar a 100 gerações ou 5 gerações seguidas obrigando a aumentar a adaptação média da população.

B. A aplicação da presente GA usa o adaptativo sete, sendo que alguns deles são combinações de outras funções. As funções correspondem aos seguintes resultados:

- func1: Curtose;
- func2: Informação Mútua;
- func3: neguentropia;
- func4: Curtose + Informação Mútua;
- func5: Curtose + neguentropia;
- func6: Informação Mútua + neguentropia;
- func7: Curtose + + neguentropia Informação Mútua.

As definições de funções de custo utilizados neste trabalho pode ser encontrada em Rojas (2006).

C. População Inicial

Uma população inicial é gerada a partir de uma matriz de coeficientes aleatórios do filtro, retornando uma matriz que reflecte um conjunto de sinais, em que:

- Cada indivíduo tem coeficientes 5, 10, 15 ou 20.
- Os coeficientes são representados por números aleatórios flutuantes (variando de -1 a 1).

Durante a execução do algoritmo, os coeficientes são submetidos as funções de custo (func1, func2, func3, func4, func5, func6, func7) como também aos diferentes operadores genéticos de seleção, cruzamento e mutação definidos no capítulo 3.

A ideia neste tipo de problema é obter através da matriz de mistura aplicado a um filtro o sinal original. Com a execução do algoritmo proposto na dissertação assim como o algoritmo de holland, os pesos do filtro vão sendo ajustados através do processo evolutivo, onde as funções de custo tem papel fundamental em maximizar a independencia entre as componentes do vetor. Neste cenário avaliamos além da acuracia do resultado o custo computacional da tecnica.

5.4.3 Avaliação

Após feita uma analise através dos resultados obtidos verificamos que o algoritmo GAADT obteve o resultado com uma porcentagem de aproximação maior que o algoritmo genético de Holland, em termos de adaptação, como também convergiu em um otimo tempo de processamento. Observando as tabelas 5.4 e 5.5 verificamos que a função que obteve o melhor resultado foi a função 7 que é a utilização em conjunto dos três métodos estatísticos(kurtose + Informação Mutua + Negentropia). Pode-se também observar que a utilização de operadores genéticos diferenciados influênciam no custo computacional. Através desta verificação podemos ver que o método da informação mutua obteve os melhores resultados utilizado de forma disjunta, como também a Curtose que obteve um resultado significativo. Em relação ao numero de coeficientes utilizados na composição do algoritmo , vemos que é diretamente proporcional a qualidade dos resultados, todavia pode-se levar a um maior custo computacional devido ao aumento no tempo de processamento por conta da complexidade dos calculos. Disso temos base para inferir que a utilização em conjunto dos métodos estatísticos obtêm um ganho significativo na qualidade dos resultados, assim aumentando ainda mais a independencia entre os componentes.

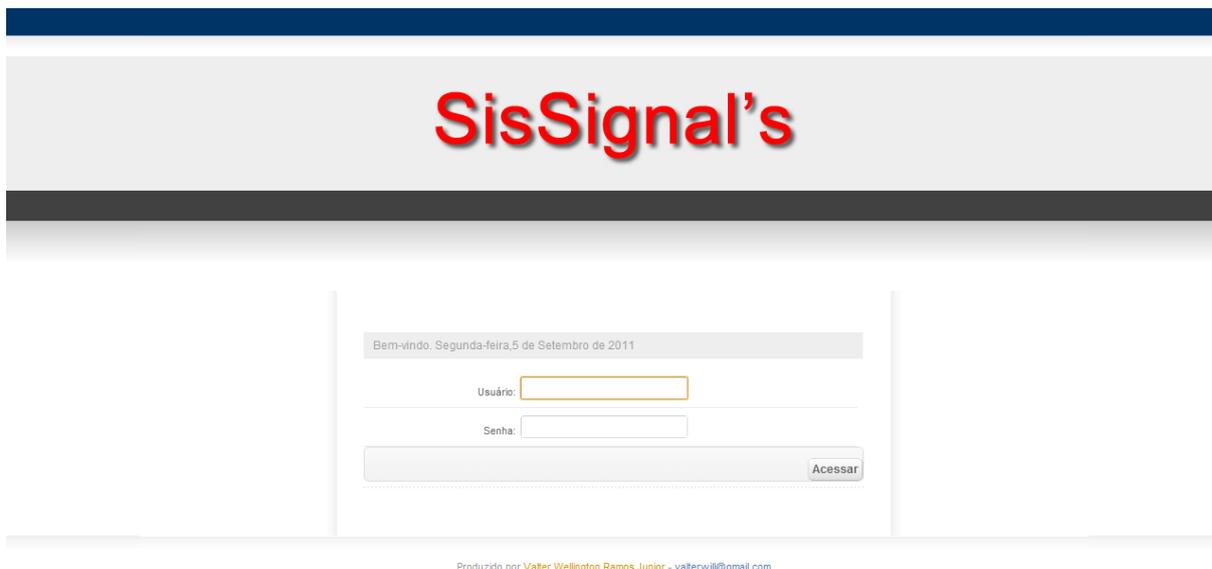


Figura 5.1: Tela de login



Figura 5.2: Tela de erro no login

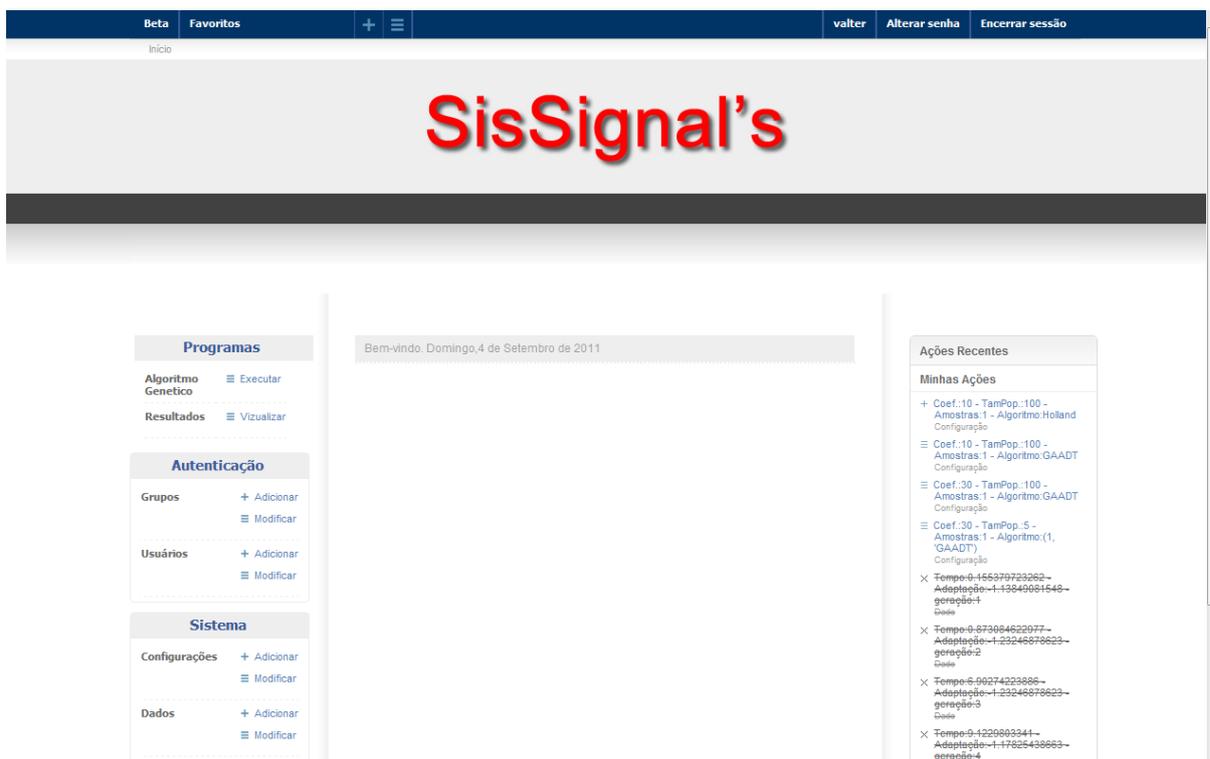


Figura 5.3: Tela inicial do sistema



Figura 5.4: Tela do programa Algoritmo Genético

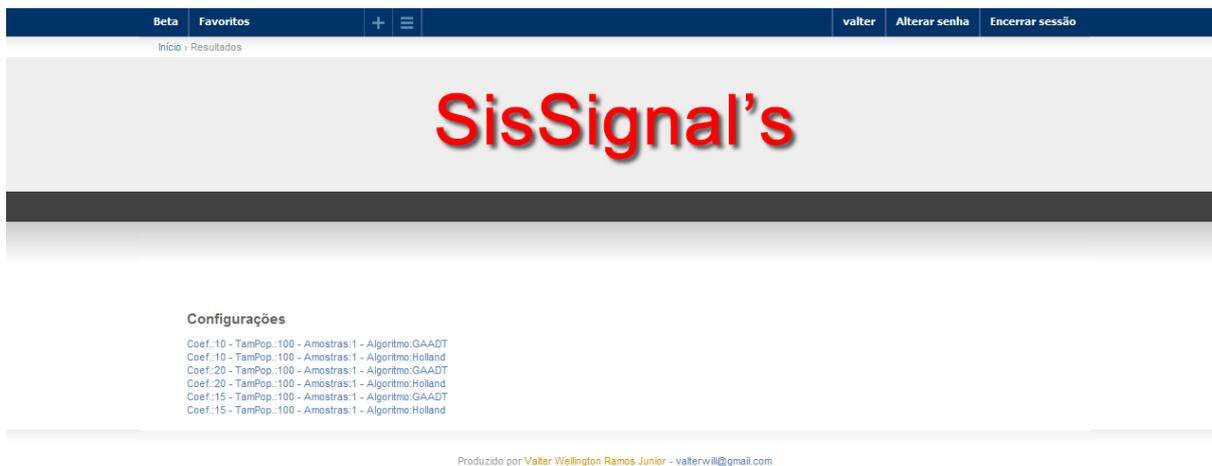


Figura 5.5: Tela de resultados

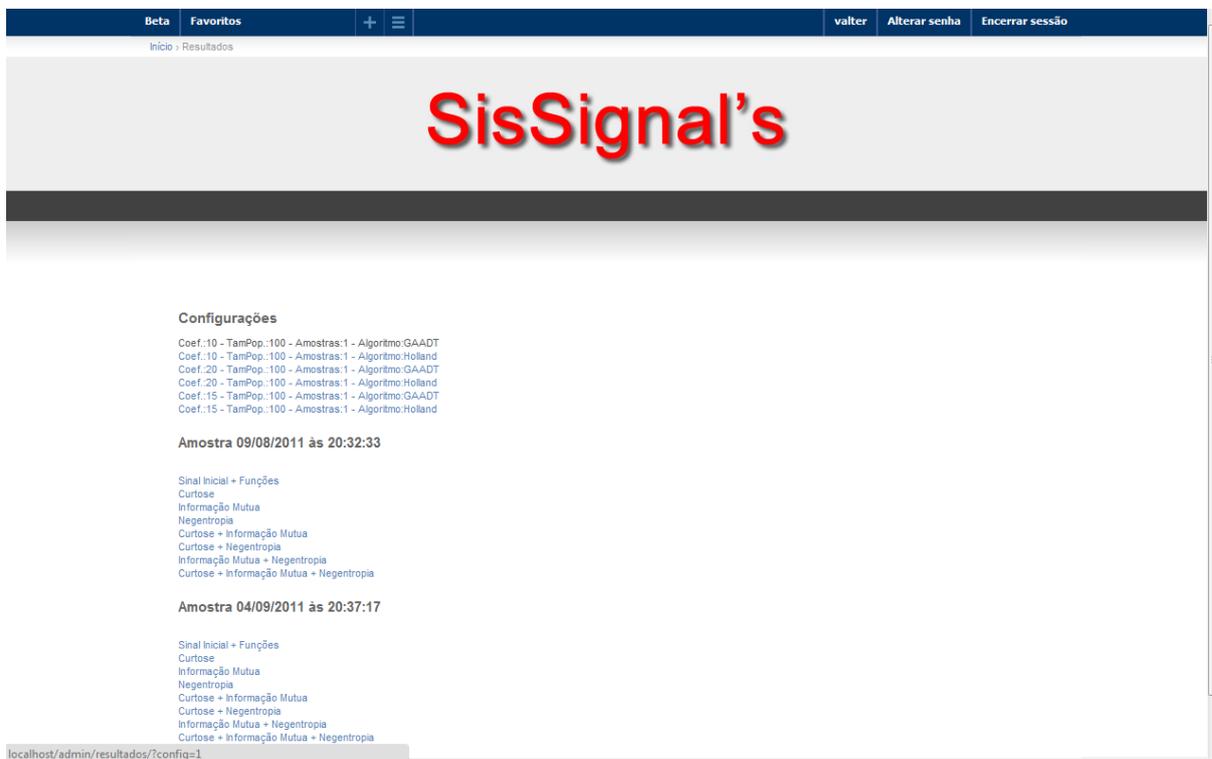


Figura 5.6: Tela de resultados, amostras

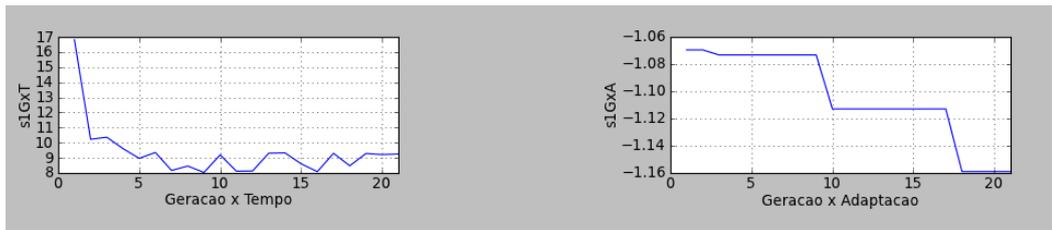


Figura 5.7: Tela de Gráficos

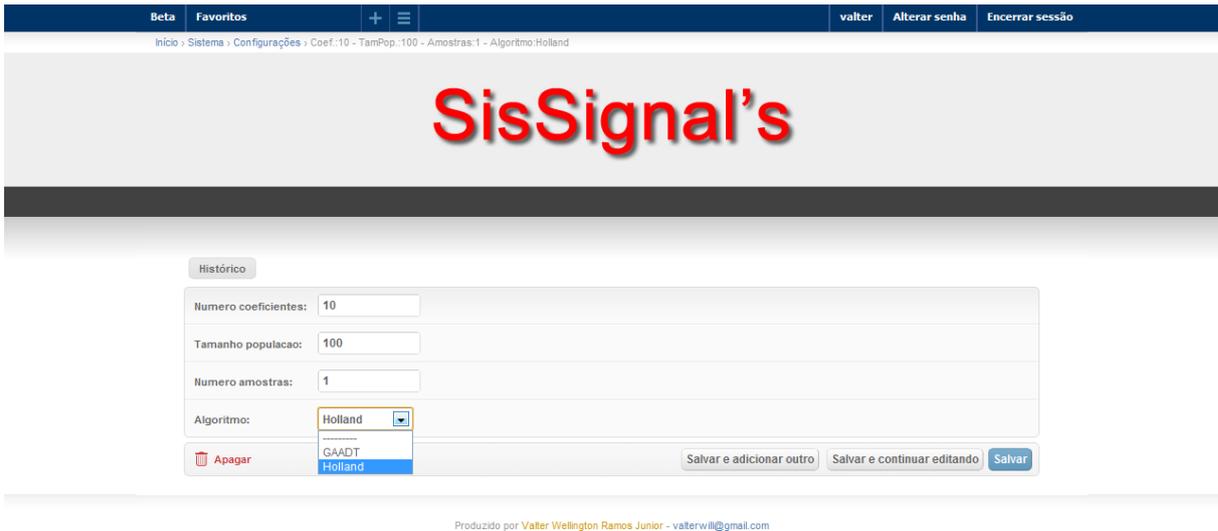


Figura 5.8: Tela do programa Configuração



Figura 5.9: Amostra de ECG obtida através do MIT-BIH Database

0:00.819	295	(0	0	0
0:00.861	310	p	0	0	0
0:00.917	330)	0	0	0
0:00.972	350	(0	0	1
0:01.025	369	N	0	0	0
0:01.069	385)	0	0	1
0:01.219	439	(0	0	2
0:01.300	468	t	0	0	4
0:01.433	516	t	0	0	4
0:01.489	536)	0	0	2

Figura 5.10: Arquivo 1 da amostra de ECG obtida através do MIT-BIH Database

995:1011
 995:1011
 995:1011
 995:1011
 995:1011
 1000:1008
 997:1008
 995:1007
 994:1007
 992:1009

Figura 5.11: Arquivo 2 da amostra de ECG obtida através do MIT-BIH Database

Indice	Operador
1	Elitista
2	Rank Linear
3	Rank Não Linear
4	Boltzmann
5	Monte Carlo
6	Torneio

Tabela 5.1: Operador Seleção

Indice	Operador
1	Soma Vetorial
2	Um Ponto de Corte
3	Varios Pontos de Corte
4	Por Complemento

Tabela 5.2: Operador Cruzamento

Indice	Operador
1	Por Complemento
2	Por Troca

Tabela 5.3: Operador Mutação

T(s)	Adapt	Func	Alg.	Coef.	OP Sel	OP Cruz	OP Mut	Acuracia
16.1459	0.0376	1	GAADT	20	-	-	-	96.2400%
14.1327	0.0539	1	GAADT	15	-	-	-	94.6100%
12.2590	0.1279	1	GAADT	10	-	-	-	87.2100%
10.0045	0.1841	1	GAADT	5	-	-	-	81.5900%
17.0853	0.0853	1	Holland	15	6	2	1	91.4735%
13.0491	0.0491	2	Holland	20	4	1	1	95.0935%
19.0399	0.0399	3	Holland	20	1	3	1	96.0091%
16.1562	0.1562	4	Holland	10	3	3	1	84.3838%
17.1540	0.1540	5	Holland	10	2	3	1	84.5959%
17.1630	0.1630	6	Holland	5	1	2	2	83.6960%
12.0451	0.0451	7	Holland	20	2	2	2	95.4942%
15.0589	0.0589	1	Holland	20	2	2	2	94.1086%
17.1395	0.1395	2	Holland	10	1	4	1	86.0454%
10.2386	0.2386	3	Holland	5	3	2	2	76.1412%
20.2961	0.2961	4	Holland	5	3	2	1	70.3932%
12.2567	0.2567	5	Holland	10	3	3	1	74.3273%
17.0996	0.0996	6	Holland	15	4	2	1	90.0353%
15.0539	0.0539	7	Holland	20	5	1	1	94.6119%

Tabela 5.4: Resultados Separação Cega de Fontes ECG

T(s)	Adapt	Func	Alg.	Coef.	OP Sel	OP Cruz	OP Mut	Acuracia
15.2519	0.0359	1	GAADT	20	-	-	-	96.4060%
12.1638	0.0429	1	GAADT	15	-	-	-	95.7100%
11.2087	0.1637	1	GAADT	10	-	-	-	83.6300%
10.2113	0.1731	1	GAADT	5	-	-	-	82.6900%
19.2796	0.2796	2	Holland	10	3	3	2	72.0380%
12.2516	0.2516	3	Holland	5	1	4	2	74.8429%
10.2780	0.2780	4	Holland	5	1	2	2	72.1954%
13.1156	0.1156	5	Holland	10	3	2	1	88.4360%
17.1426	0.1426	6	Holland	15	3	1	2	85.7408%
12.0760	0.0760	7	Holland	20	5	2	2	92.3993%
18.2965	0.2965	1	Holland	5	3	2	2	70.3495%
16.1013	0.1013	1	Holland	15	6	4	1	89.8721%
17.2152	0.2152	2	Holland	10	6	4	1	78.4771%
16.1485	0.1485	3	Holland	15	6	3	1	85.1536%
20.2975	0.2975	4	Holland	5	5	2	1	70.2470%
17.2228	0.2228	5	Holland	10	4	2	2	77.7183%
12.2298	0.2298	6	Holland	10	2	1	2	77.0165%
20.0846	0.0846	7	Holland	15	5	3	2	91.5401%

Tabela 5.5: Resultados Deconvolução Cega

Capítulo 6

Conclusão e Trabalhos Futuros

6.1 Introdução

Neste capítulo serão apresentadas a conclusão deste trabalho e algumas sugestões de trabalhos futuros.

6.2 Considerações Finais

A importância da busca por novas abordagens computacionais é indiscutível, quer para resolver problemas considerados insolúveis pelas abordagens usuais, quer para resolver problemas já resolvidos por algumas abordagens de maneira insatisfatória. Diante desse contexto, encontram-se várias propostas de sistemas que integram técnicas evolutivas com métodos estatísticos.

O sistema evolutivo SisSignals, desenvolvido neste trabalho, tem como propósito resolver o problema de separação cega de fontes (BSS - Blind Source Separation).

A grande dificuldade de desenvolver uma abordagem que resolva problemas desse tipo é o cuidado que se deve ter para minimizar a convergência prematura que leva os resultados a ficarem presos em mínimos locais. Por esse motivo, a maioria das abordagens desenvolvidas para resolver problemas de BSS não se preocupa em justificar a qualidade de suas soluções.

A justificativa da solução de um problema de BSS pode até não ser necessária em alguns domínios de aplicação, tais como no domínio das ciências exatas. Mas no domínio das ciências biológicas, a justificativa da solução de um problema assume um papel de vital importância. Pois é através da análise da justificativa do sistema, que o usuário será capaz de encontrar os erros cometidos durante a fase de entrada de dados.

Neste trabalho procurou-se lidar com a convergência prematura e com a necessidade de gerar uma justificativa para a solução de um problema de BSS.

O SisSignals na atual versão apresenta um inconveniente para o usuário que não está familiarizado com as técnicas de inteligência artificial. O inconveniente percebido pelo usuário está relacionado com a escolha dos parâmetros do algoritmo genético que devem ser usados para resolver um determinado problema. A escolha dos parâmetros do algoritmo genético pressupõe que o usuário entenda o funcionamento do algoritmo genético de Holland ou do algoritmo genético baseado em tipos abstratos de dados. Dessa forma o SisSignals restringe o perfil do seu usuário as pessoas que entendem o funcionamento do algoritmo genético.

Apesar deste inconveniente, a análise obtida pelo SisSignals está respaldada sob as premissas filosóficas da separação cega de fontes, já que este sistema adota critério de independência estatística entre as fontes. O sinal de saída gerado pelo SisSignals é resultante de um processo evolutivo onde a cada geração os pesos do filtro são ajustados através de um estimador de entropia baseado em estatística de ordem de forma a minimizar a dependência entre as fontes que compõem o sinal.

A proposta deste trabalho foi avaliar o desempenho e a eficácia do algoritmo genético baseado em tipos abstratos de dados - GAADT, aplicado ao problema de separação cega de fontes.

Com base nos testes efetuados em comparação com o problema original proposto por Rojas (2006) verificamos que a utilização do GAADT obteve ganhos tanto com o custo computacional em relação ao tempo de processamento e consumo do processador e memória, quanto a qualidade dos resultados obtidos. Foi detectado também que a utilização dos métodos em conjunto ajuda muito a aumentar a qualidade dos resultados.

6.3 Contribuições

Como contribuições para o meio acadêmico temos:

- Criação de um sistema para evolutivo configurável para separação cega de fontes;
- Iniciativa de combinar métodos estatísticos;
- Validação do GAADT para problema de otimização combinatorial;
- Definição de uma arquitetura extensível para transformações;

6.4 Trabalhos Futuros

Como trabalhos futuros sugerimos:

- aplicar o SisSignals a outros tipos de problemas de BSS, que tenham uma função de custo a otimizar e que trabalhem com objetos facilmente representáveis em uma matriz que relacione estes objetos aos seus atributos;

- implementar no SisSignals outros algoritmos evolutivos ou bioinspirados , bem como outros métodos estatísticos como função de custo, para verificar a eficácia destes algoritmos ao problema de BSS.

Referências Bibliográficas

- A. Homaifar, S. H. Y. L. . X. Q. (1994), 'Constrained optimization via genetic algorithms', *Morgan Kaufmann* .
- Alleles, D. E. G. . R. L. (1985), 'loci and the tsp, in j. j. grefenstette, editor, proceedings of the first international conference on genetic algorithms', *Lawrence Erlbaum Associates* .
- Baker, J. E. (1985), 'Adaptive selection methods for genetic algorithms, in j. j. grefenstette, editor, proceedings of the first international conference on genetic algorithms, pages 101-111, lawrence erlbaum associates', *Proceedings of the International Conference on Genetic Algorithms* .
- Booker, L. (1987), 'Improving search in genetic algorithms', *Genetic Algorithms and Simulated Annealing, Pitman* .
- Bramlette, M. F. (1991), 'Initialization, mutation and selection methods in genetic algorithms for function optimization, in r. k. belew and l. b. booker, editor, proceedings of the fourth international conference on genetic algorithms', *Morgan Kaufmann* .
- Brindle, A. (1981), 'Genetic algorithms for function optimization', *Complex Systems* .
- Cichoki, A. & Amari, S. (2002), 'Learning algorithms and applications', *John Wiley & Sons* .
- Cramer, N. L. (1985), 'A representation for the adaptive generation of simple sequential programs, in , editor, proceedings of the first international conference on genetic algorithms, lawrence erlbaum associates', *J. J. Grefenstette* .
- Davis, L. (1985), 'Job shop scheduling with genetic algorithms', *Lawrence Erlbaum Associates* .
- Davis, L. (1989), 'Adapting operator probabilities in genetic algorithms', *Morgan Kaufmann* .
- Django.org (2005-2011), 'Django'. Disponível em: <http://www.djangoproject.com>, último acesso em 03 de setembro de 2011.

- E., G. D. (1990), 'A note on boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing.', *Complex Systems* .
- Goldberg (1989), 'Genetic algorithms in search, optimization and machine learning.', *Addison-Wesley* .
- Haykin, S. (2000), 'Unsupervised adaptive filtering', *Blind source Separation. John Wiley & Sons* **1**.
- Houck, J. A. J. . C. R. (1994), 'On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with gas', *Proceedings of the International Conference on Evolutionary Computation* .
- J. T. Richardson, M. R. Hilliard, M. R. P. . G. E. L. (1989), 'Some guidelines for genetic algorithms with penalty functions', *Morgan Kaufmann* .
- Jutten, P. C. . C. (2010), 'Handbook of blind source separation: Independent component analysis and applications', *Academic Press* .
- K. S. Tag, K. F. Man, S. K. . Q. H. (1996), 'Genetic algorithms and their applications, signal processing magazine', *Complex Systems* .
- LOPES, R. V. V. (2003), Um algoritmo genético baseado em tipos abstratos de dados e sua especificação em z, (tese de doutorado em informática), Universidade Federal de Pernambuco, Recife - PE.
- McMahon, B. R. F. . M. B. (1991), 'Genetic operators for sequencing problems, proceedings of the foundations of genetic algorithms', *Morgan Kaufmann* .
- Michalewicz, C. Z. J. . Z. (1991), 'An experimental comparison of binary an floating point representations in genetic algorithms', *Proceedings of the International Conference on Genetic Algorithms* .
- Michalewicz, Z. (1996), 'Genetic algorithms + data structures = evolution programs', *Springer-Verlag* .
- Python.org (1999-2011), 'Python'. Disponível em: <http://www.python.org>, último acesso em 03 de setembro de 2011.
- Rojas, F. . P. (2003), 'Evolutionary algorithm using mutual information for independent component analysis', *IWANN 2003* .
- Rojas, F. . S.-C. (2006), 'A canonical genetic algorithm for blind inversion of linear channels', *ICA 2006* .

- Skolnick, D. P. . M. M. (1993), 'Using genetic algorithms in engineering design optimization with nonlinear constraints', *Proceedings of the International Conference on Genetic Algorithms* .
- Spears, K. A. D. J. . W. M. (1989), 'Using genetic algorithms to solve np-complete problems', *Morgan Kaufmann* .
- Syswerda, G. (1989), 'Uniform crossover in genetic algorithms', *Morgan Kaufmann* .
- Valter Ramos, Bruno Raphael, L. C. . R. L. (2010), 'A modified evolutionary algorithm approach for blind channel', *CSNDSP 2010* .
- Valter Ramos, Bruno Raphael, L. C. . R. L. (2011a), 'A new evolutionary algorithm for blind linear channel inversion', *IWSSIP 2011* .
- Valter Ramos, Luis Coradine, A. L. . R. L. (2011b), 'An application of algorithm based on abstract data types to the problem of separation of signals of the ecg', *JICSIT 2011* .
- Xiang-Yan Zeng, Yen-Wei Chen, Z. N. . G. Y. (2000), 'Signal separation by independent component analysis based on a genetic algorithm', *Proceedings of ICSP2000* .