UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Técnicas de Computação Natural para Segmentação de Imagens Médicas

Jackson Gomes de Souza

Orientador: Prof. Dr. José Alfredo Ferreira Costa

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Natal, RN, setembro de 2007

Livros Grátis

http://www.livrosgratis.com.br

Milhares de livros grátis para download.

Divisão de Serviços Técnicos

Catalogação da Publicação na Fonte. UFRN / Biblioteca Central Zila Mamede

Souza, Jackson Gomes de.

Técnicas de comunicação natural para segmentação de imagens médicas / Jackson Gomes de Souza. – Natal, RN, 2007.

103 f.

Orientador: José Alfredo Ferreira Costa.

Dissertação (Mestrado) – Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica.

1. Processamento de imagens digitais – Dissertação. 2. Segmentação de imagens médicas – Dissertação. 3. Otimização - enxame de partículas - Dissertação. 4. Computação natural – Dissertação. 5. Algoritmos genéticos – Dissertação. I. Costa, José Alfredo Ferreira. II. Universidade Federal do Rio Grande do Norte. III. Título.

RN/UF/BCZM CDU 621.397.331(043.3)

Técnicas de Computação Natural para Segmentação de Imagens Médicas

Jackson Gomes de Souza

Dissertação de Mestrado aprovada em 28 de setembro de 2007 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. José Alfredo Ferreira Costa (orientador) .....UFRN

Prof. Dr. Allan de Medeiros Martins.....UNP

Prof. Dr. Bruno Motta de Carvalho .....UFRN

A Deus, às três mulheres da minha vida (Mãe, Quesia e Corrinha) e aos amigos-vizinhos-irmãos dos Chalés Canaã (2005-2006).

Agradecimentos

A Deus, acima e primeiro de tudo, Criador de todas as coisas, do universo e de tudo que nele há, Senhor, Deus meu, Salvador e Redentor meu, sou eternamente grato pela saúde, disposição, pelo livramento, pela providência e por tantas outras coisas que me mantiveram são num momento em que eu tinha que ser mais de 2 ao mesmo tempo.

À minha família, Mãe, principalmente por ter me dado suporte e ter segurado a barra quando eu precisei ficar fora por um ano, à Quesia por ter cuidado da mãe e ser a melhor irmã do mundo, à Maria do Socorro "Corrinha", meu amor e coração e agradecimento sinceros, principalmente por ter estado comigo quando estávamos distante por mais de 2 mil km. Aos mesu avós, Roque e Levina, à Tia Francisca "Chiquinha", ao Tio Otanael "Thunna", e aos demais que apoiaram em oração e financeiramente. Ainda no que eu considero "família", agradeço à Parci e ao Fabiano "Thilfa", pelo carinho, cuidado e pela proteção em todo o tempo da graduação e, sei, ainda hoje.

Ao querido professor José Alfredo, pelo crédito a um tocantinense do qual ele nunca tinha ouvido falar e pela paciência e todo o suporte nos tempos de orientação à distância. Ao professor Alexandre Falcão, que mesmo não estando na lista oficial de orientação, até mesmo sem saber, foi um dos responsáveis pela minha motivação nos estudos de segmentação em imagens médicas.

Aos vizinhos, amigos e verdadeiros irmãos dos Chalés Canaã entre 2005 e 2006, sem ordem de preferência ou qualquer outra que eu me lembre agora, Carol e Raoni, Lia, Mariana, Simone, Ilka, Isabela, Aline, Estevão, Cassiano, Alexandre, vocês foram amigos certos nas horas incertas e irmãos para apoio dar em todo instante. Aos amigos da UFRN, DCA e LSA, Adriano, Raquel, Denis, Rafael, Danilo, Flavius e Sérgio.

A todos vocês, meu eterno Obrigado! Que Deus os retribua em dobro!

Resumo

Segmentação de imagens é um dos problemas de processamento de imagens que merece especial interesse da comunidade científica. Neste trabalho, são estudados métodos não-supervisionados para detecção de algomerados (*clustering*) e reconhecimento de padrões (*pattern recognition*) em segmentação de imagens médicas. Métodos baseados em técnicas de computação natural têm se mostrado bastante atrativos nestas tarefas e são estudados aqui como uma forma de verificar a sua aplicabilidade em segmentação de imagens médicas. Este trabalho trata de implementar os métodos GKA (Genetic K-means Algorithm), GFCMA (Genetic FCM Algorithm), PSOKA (Algoritmo de *clustering* baseado em PSO (*Particle Swarm Optimization*) e K-means) e PSOFCM (Algoritmo de *clustering* baseado em PSO e FCM (*Fuzzy C-Means*)). Além disso, como forma de avaliar os resultados fornecidos pelos algoritmos, são utilizados índices de validação de *clustering* como forma de medida quantitativa. Avaliações visuais e qualitativas também são realizadas, principalmente utilizando dados do sistema BrainWeb, um gerador de imagens do cérebro, como *ground truth.*

Palavras-chave: Processamento de Imagens Digitais; Segmentação de Imagens Médicas; Otimização por Enxame de Partículas; Computação Natural; Algoritmos Genéticos; K-means; Fuzzy C-means.

Abstract

Image segmentation is one of the image processing problems that deserves special attention from the scientific community. This work studies unsupervised methods to clustering and pattern recognition applicable to medical image segmentation. Natural Computing based methods have shown very attractive in such tasks and are studied here as a way to verify it's applicability in medical image segmentation. This work treats to implement the following methods: GKA (Genetic K-means Algorithm), GFCMA (Genetic FCM Algorithm), PSOKA (PSO and K-means based Clustering Algorithm) and PSOFCM (PSO and FCM based Clustering Algorithm). Besides, as a way to evaluate the results given by the algorithms, clustering validity indexes are used as quantitative measure. Visual and qualitative evaluations are realized also, mainly using data given by the BrainWeb brain simulator as ground truth.

Keywords: Digital Image Processing; Medical Image Segmentation; Particle Swarm Optimization; Natural Computing; Genetic Algorithms; K-means; Fuzzy C-means.

Sumário

RESUMO **7**

ABSTRACT **8**

SUMÁRIO ..... **I**

LISTA DE FIGURAS ..... **III**

LISTA DE TABELAS ..... **V**

CAPÍTULO 1 INTRODUÇÃO **1**

1.1 Organização do trabalho **4**

CAPÍTULO 2 COMPUTAÇÃO NATURAL **5**

2.1 Inspiração Biológica e Natural para a Computação **5**

2.1.1 Redes Neurais Artificiais 7

2.1.2 Algoritmos Genéticos 12

2.1.3 Inteligência de Enxames 15

2.1.4 Sistemas Imunológicos Artificiais 21

CAPÍTULO 3 DETECÇÃO DE AGRUPAMENTOS ATRAVÉS DE TÉCNICAS BASEADAS EM COMPUTAÇÃO NATURAL **24**

3.1 Detecção de Agrupamentos **24**

3.1.1 Algoritmo K-means 27

3.1.2 Soft Computing 28

3.1.3 Fuzzy Clustering 29

3.2 Detecção de Agrupamentos através de Algoritmos Genéticos **31**

3.2.1 Algoritmo K-Means Genético 31
3.2.2 Algoritmo Fuzzy C-Means Genético 36

3.3 Detecção de Agrupamentos através do PSO 36

3.4 Índices de Validação de *Clustering* 39
3.4.1 Índice Davies-Bouldin (DB) 40
3.4.2 Índice de Partição (SC) 40
3.4.3 Índice de Separação (S) 40
3.4.4 Índice de Xie e Beni (XB) 41

3.5 Resultados Experimentais e Discussões 41
3.5.1 Resultados dos algoritmos K-means e Fuzzy C-Means 42
3.5.2 *Clustering* baseado em Algoritmos Genéticos 44
3.5.3 Resultados dos algoritmos GKA e GFCMA 46
3.5.4 *Clustering* baseado em Otimização por Enxames de Partículas 55
3.5.5 Resultados dos algoritmos PSOKA e PSOFCM 56

CAPÍTULO 4 SEGMENTAÇÃO DE IMAGENS MÉDICAS UTILIZANDO MÉTODOS DE COMPUTAÇÃO NATURAL 66

4.1 Imageamento em Medicina 66
4.1.1 Tomografia computadorizada 66
4.1.2 Ressonância Nuclear Magnética 68

4.2 Escolha de Características de Imagens Médicas Para Segmentação de Imagens 68

4.3 Segmentação de Imagens Médicas usando Computação Natural 73

4.4 Resultados, Análises e Comparações 74
4.4.1 Algoritmos Genéticos aplicados à Segmentação de Imagens 75
4.4.2 PSO aplicado à Segmentação de Imagens 79

CAPÍTULO 5 CONSIDERAÇÕES FINAIS E CONCLUSÕES 84

REFERÊNCIAS BIBLIOGRÁFICAS 88

Lista de Figuras

Figura 1: Diversas abordagens utilizadas na computação natural 6
Figura 2: Modelo não-linear de um neurônio 8
Figura 3: Arquitetura de rede alimentada adiante com múltiplas camadas 10
Figura 4: Modelo das diferentes topologias de vizinhança; a) Topologia Estrela; b) Topologia Anel; c) Topologia Von Neumann 21
Figura 5: Taxonomia de métodos de clustering 25
Figura 6: Exemplos de estruturas de clusters: (a) clusters bem separados; (b) clusters superpostos 30
Figura 7: Pseudo-código do Algoritmo K-Means Genético Incremental 34
Figura 8: Pseudo-código do GKA 35
Figura 9: Pseudo-código do Algoritmo de Clustering baseado em PSO 38
Figura 10: Resultado de Clustering com GKA e GFCMA: (a) e (b) base de dados Iris: a) GKA; b) GFCMA; (c) e (d) base de dados Ruspini: c) GKA; d) GFCMA; (e) e (f) base de dados Wine: e) GKA e f) GFCMA 52
Figura 11: Exemplo de solução do problema de mínimos locais (GFCMA) 52
Figura 12: Resultado de Clustering com PSOKA e PSOFCM: (a) e (b) base de dados Iris: a) PSOKA; b) PSOFCM; (c) e (d) base de dados Ruspini: c) PSOKA; d) PSOFCM; (e) e (f) base de dados Wine: e) PSOKA e f) PSOFCM 61
Figura 13: Exemplo de solução do problema de ótimos locais (base de dados Iris, PSOKA) 62
Figura 14: Interface interativa do simulador de imagens do cérebro BrainWeb 69
Figura 15: Tela de pré-visualização do simulador BrainWeb 70
Figura 16: Exemplos de imagem discreta (a) e as diversas classes do conjunto de dados (b - k) 71
Figura 17: Exemplos de fatias de imagens nos modelos T1 (a), T2 (b) e PD (c) 72

Figura 18: Resultados de Segmentação de Imagem com GKA e GFCMA: imagem_intensidade: a) e b) GKA, c) e d) GFCMA; imagem_textura: e) e f) GKA, g) e h) GFCMA 77
Figura 18: Resultados de Segmentação de Imagem com GKA e GFCMA: imagem_textura: a) e b) GKA, c) e d) GFCMA; imagem_intensidade: e) e f) GKA, g) e h) GFCMA 82

Lista de Tabelas

Tabela 1: Resultados dos Algoritmos K-Means e Fuzzy C-Means (FCM): a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine..... 43
Tabela 2: Resultados dos Algoritmos GKA e GFCMA: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine..... 47
Tabela 3: Resultados dos Algoritmos GKA e GFCMA considerando taxa de erro mínima e máxima: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine..... 48
Tabela 4: Resultados dos Algoritmos GKA e GFCMA considerando valores mínimo e máximo dos índices de validação: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine..... 53
Tabela 2: Resultados dos Algoritmos PSOKA e PSOFCM: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine..... 57
Tabela 3: Resultados dos Algoritmos GKA e GFCMA considerando taxa de erro mínima e máxima: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine..... 58
Tabela 7: Resultados dos Algoritmos PSOKA e PSOFCM considerando valores mínimo e máximo dos índices de validação: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine..... 62
Tabela 6: Exemplos de números CT com relação a tecidos 67
Tabela 9: Resultados dos Algoritmos GKA e GFCMA: a) conjunto de dados imagem_textura; b) conjunto de dados imagem_intensidade..... 75
Tabela 10: Resultados dos Algoritmos GKA e GFCMA considerando taxa de erro mínima e máxima: a) conjunto de dados imagem_textura; b) conjunto de dados imagem_intensidade 76

Tabela 11: Resultados dos Algoritmos GKA e GFCMA considerando valores mínimo e máximo dos índices de validação: a) conjunto de dados imagem_intensidade; b) conjunto de dados imagem_textura 78
Tabela 12: Resultados dos Algoritmos PSOKA e PSOFCM: a) conjunto de dados imagem_textura; b) conjunto de dados imagem_intensidade 79
Tabela 13: Resultados dos Algoritmos PSOKA e PSOFCM considerando taxa de erro mínima e máxima: a) conjunto de dados imagem_textura; b) conjunto de dados imagem_intensidade 80
Tabela 14: Resultados dos Algoritmos PSOKA e PSOFCM considerando valores mínimo e máximo dos índices de validação: a) conjunto de dados imagem_intensidade; b) conjunto de dados imagem_textura 82

Capítulo 1
Introdução

A sociedade da qual fazemos parte, conhecida como "sociedade da informação" passa por um processo de adaptação ao contexto atual no que se relaciona à forma de abstração e processamento de informações. O conceito de "informação" é amplo, e da mesma forma pode-se afirmar que são também grandes as possibilidades de fontes de informação. Informações textuais, visuais e audíveis têm seu papel específico de acordo com o que se quer transmitir, entretanto, as informações visuais (por imagem ou vídeo) tendem a ter uma participação maior dado ao fato do ser humano tender à representação de informações através de símbolos visuais. Pode-se dizer também que a imagem, como uma representação virtual de uma realidade imaginada desperta o senso de imaginação, criatividade ou de fantasia.

Para a ciência da computação, trabalhar neste contexto é considerar a imagem não só como um conjunto de dados, mas também como um domínio que inclui semântica e significado que são dependentes, até mesmo, do indivíduo que trabalha com tal conjunto de dados. Atualmente, aplicações de processamento de imagens ultrapassam, com folga, o que foi pensado há anos quando esta ciência foi imaginada e suas bases foram desenvolvidas. Há processamento de imagem quando um sistema de navegação automática em um veículo automotor captura imagens de uma estrada através de um dispositivo de captura de imagens instalado na frente do veículo e avisa o motorista quando há presença de animais ou obstáculos na pista. Há processamento de imagem quando as inovações em televisão, em específico a TV digital, prometem recursos como a interação o telespectador com elementos e objetos na tela. Há processamento de imagens quando um radiologista ou profissional que trabalha com exames de ressonância magnética visualiza informações tridimensionais de um exame do cérebro. Estas são apenas algumas das aplicações de processamento de imagens, mas

ilustram a grande abrangência deste ramo da ciência da computação. Em grande parte das aplicações de processamento de imagem e vídeo é necessário separar elementos visuais uns dos outros, identificando-os como objetos individuais, com características e comportamentos próprios. Em geral, esta tarefa é realizada através de segmentação de imagens.

Técnicas de segmentação de imagens são baseadas em conceitos das disciplinas de Reconhecimento de Padrões (*Pattern Recognition*). De fato, a tarefa de segmentação de imagens significa a identificação de padrões em um conjunto de dados, que é representado por informações da imagem (os *pixels*, em função de intensidade de luz). Padrões ou classes na imagem representam informações válidas no contexto da imagem e, mais do que características e informações da imagem, associam à imagem conhecimento descoberto através das técnicas de reconhecimento de padrões. Conforme Jain (1999) a técnica de *Clustering* representa a classificação não-supervisionada de padrões (observações, itens de dados ou vetores de características) em grupos (*clusters*). Os algoritmos de *clustering* estão suscetíveis a problemas como os indicados por Frigui e Krishnapuram (1999):

a) determinação do número de clusters (que não é conhecido *a priori*);
b) independência de condições iniciais.

Ainda, como comenta Estivil-Castro (2002) os métodos de *clustering* sofrem de problemas de precisão; o autor aponta ainda para a necessidade constante de princípios indutivos como forma de avaliação dos algoritmos existentes.

Em Processamento Digital de Imagens, uma das etapas extremamente importantes, e fundamentais, para análise de imagens é a separação ou o isolamento de regiões de interesse em uma imagem, reduzindo, com isso, o espaço de dados que serão analisados. Segundo Gonzalez e Woods (2003) em imagens monocromáticas, os algoritmos de segmentação de imagens baseiam-se nas seguintes propriedades dos níveis de cinza da imagem: descontinuidade e similaridade.

Aplicações de segmentação de imagens contemplam inúmeras áreas de Computação Gráfica. No caso da Visão Computacional, um dos objetivos é fazer com que robôs consigam se locomover em um ambiente semi ou não-controlado, e realizar tarefas como encontrar e interagir com objetos específicos neste ambiente. Em Análise e Entendimento de Imagens, há aplicações como desenvolver mecanismos de busca eficientes em um banco de dados de imagens capazes de encontrar itens no banco de dados através de uma imagem de referência, detectando similaridades entre elas (área conhecida como *Content Based Image Retrieval*). Na Medicina, segmentação de imagens é aplicada no contexto de desenvolver sistemas que auxiliem profissionais e especialistas da área no diagnóstico de doenças ou identificação de, por exemplo, elementos na imagem que possam indicar a presença de câncer.

Ferramentas de diagnóstico por imagens são indispensáveis para a medicina atual. Como afirmam Pham et al (1998) Imagens de Ressonância Magnética (MRI), Tomografia Computadorizada (CT), Mamografia Digital e outras modalidades de imagem provêm maneiras efetivas para mapear, de forma não invasiva, a anatomia de determinada região do corpo de um indivíduo.

Assim, os algoritmos de segmentação de imagens representam um papel importante em aplicações biomédicas por imagem como a quantificação de volume de tecido, diagnóstico, localização de patologias e estudo de estruturas anatômicas.

Métodos de segmentação de imagens diferem bastante de acordo com a sua aplicação específica. Por exemplo, a segmentação do tecido cerebral apresenta requisitos e características diferentes da segmentação em imagens do pulmão. Mesmo com a grande existência de métodos para realizar tais tarefas de segmentação, nenhum deles é totalmente auto-suficiente nas tarefas para as quais se propõe realizar. Desta forma, para realizar uma tarefa mais genérica, há junções de métodos especializados em tarefas particulares. Assim, os métodos adaptativos e híbridos representam grande poder de qualidade nas soluções que fornecem.

Devido a características das imagens de MRI e CT, como a suavidade nas bordas e fronteiras, as técnicas de segmentação de imagens baseadas em *clustering Fuzzy* têm demonstrado que o processamento "suave" (*soft*) de dados de imagens, principalmente neste caso de imagens, pode fornecer resultados altamente consideráveis tanto em termos de qualidade quanto em consumo de recursos do computador e tempo de processamento.

Recentemente, técnicas evolucionárias, baseadas no conceito de Computação Natural, para segmentação de imagens baseadas em comportamento social acrescentam novos entendimentos ao processamento de dados e reconhecimento de padrões. A utilização de Algoritmos Genéticos e Otimização por Enxames de Partículas em tarefas de detecção de aglomerados e segmentação de imagens, em especial, é o foco principal do presente trabalho de pesquisa.

1.1 Organização do trabalho

Os capítulos do presente texto estão organizados da seguinte forma: o Capítulo 2 trata dos fundamentos de Computação Natural, como Redes Neurais Artificiais e Otimização por Enxame de Partículas. O Capítulo 3 apresenta conceitos de Reconhecimento de Padrões e Detecção de Aglomerados, focalizando métodos de computação suave, como os métodos baseados na Teoria de Conjuntos Fuzzy; apresenta também as definições da otimização através do método *Particle Swarm Optimization*; o Capítulo 5 apresenta as conclusões e considerações finais sobre o presente trabalho de dissertação.

Capítulo 2
Computação Natural

Computação inspirada na biologia vem sendo utilizada já há algum tempo, desde as décadas de 1960 e 1970 com a utilização dos algoritmos genéticos, inspirados na teoria da evolução e da seleção natural. As pesquisas recentes têm criado métodos que estendem as definições dos algoritmos inspirados na biologia e na natureza.

2.1 Inspiração Biológica e Natural para a Computação

Segundo Castro (2007) a Computação Natural é a versão computacional do processo de extrair idéias da natureza para desenvolver sistemas computacionais, ou utilizar materiais encontrados na natureza para realizar computação; e essa definição pode ser classificada da seguinte maneira:

a) **Computação inspirada pela natureza**. Os algoritmos desta categoria inspiram-se na natureza para desenvolver técnicas de solução de problemas: Redes Neurais Artificiais, Computação Evolucionária, Inteligência de Aglomerados (Enxames) e Sistemas Artificiais Imunes.

b) **Simulação e emulação da natureza por meios computacionais**. Esta categoria inclui métodos que formulam um processo artificial que tem o objetivo de identificar e simular (ou emular) padrões, formas, comportamentos e organismos nas formas em que são encontrados na natureza ou em indivíduos biológicos. Métodos nesta categoria incluem Geometria Fractal e Vida Artificial.

c) **Computação com materiais encontrados na Natureza**. Os métodos nesta categoria utilizam materiais encontrados na natureza para realizar computação e seu objetivo é utilizar estes materiais para substituir computadores baseados em silício.

Alguns dos métodos mais representativos são Computação baseada no DNA e Computação Quântica.

Por causa de toda esta motivação bio-inspirada, a Computação Natural é um campo de pesquisa que se torna um concentrador de conceitos e definições de outras ciências e profissionais como físicos, químicos, engenheiros, biólogos e cientistas da computação precisam trabalhar em conjunto de forma que consigam conciliar tais conceitos e alcançar os objetivos da computação. Esta integração entre diversos ramos da ciência pode ser resumida pela Figura 1.

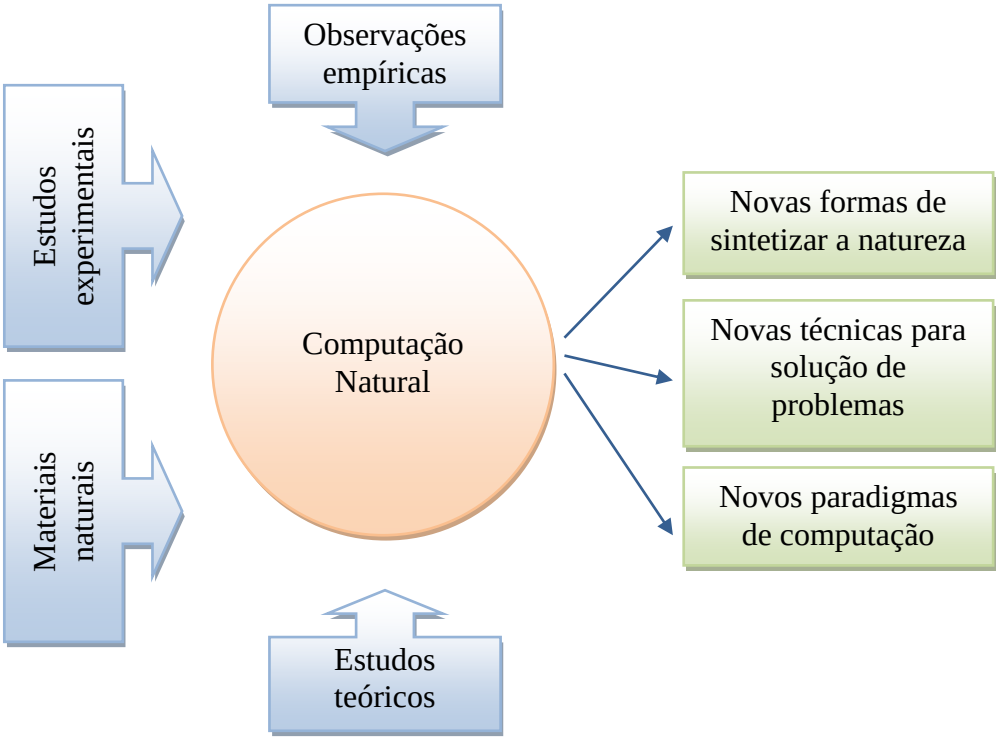


Figura 1: Diversas abordagens utilizadas na computação natural

Outro fator preponderante é que a Computação Natural lida com modelos que são baseados em simplificações da realidade. Estas simplificações são necessárias porque, na maioria dos métodos, um grande número de elementos faz parte da computação – como é o caso dos neurônios artificiais em Redes Neurais que, em conjunto, procuram simular o comportamento do cérebro humano. Além disso, uma visão simplificada de uma entidade natural também corresponde a um sistema, ou,

propriamente, a um subsistema, já que estes elementos, em conjunto, formam um sistema completo e de funcionamento bem mais complexo do que as partes que o compõem.

2.1.1 Redes Neurais Artificiais

Esta abordagem é inspirada na definição matemática de um neurônio e no entendimento do comportamento do cérebro humano. Segundo Haykin (2001) uma rede neural artificial é um processador massivamente distribuído composto de simples unidades de processamento, os neurônios, que atuam, naturalmente, para armazenar conhecimento útil que é adquirido através de um processo de aprendizagem que leva a melhores resultados quando as unidades de processamento trabalham de uma forma interconectada. Ainda, conforme Haykin (2001) esse processo de aprendizagem, realizado através de um algoritmo de aprendizagem, se assemelha ao cérebro em dois aspectos:

a) O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem. Isso significa que a rede não atua em um ambiente desconhecido ou desordenado. Como será visto posteriormente, as Redes Neurais Artificiais encaixam-se em uma classificação de algoritmos que necessitam de um instrutor, um professor, que identifique e modele o domínio, ou o ambiente externo, apresente os dados à rede e avalie os resultados fornecidos, tornando-se, assim, parte ativa no processo de aprendizagem.

b) Forças de conexão entre neurônios, os pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

Algumas das propriedades úteis de redes neurais são:

1. Não linearidade
2. Mapeamento de Entrada-Saída
3. Adaptabilidade

4. Tolerância a falhas
5. Uniformidade de análise e projeto

O neurônio, citado anteriormente como componente principal e fundamental de uma rede neural, como no cérebro humano, pode ser definido sistematicamente conforme demonstra a Figura 2:

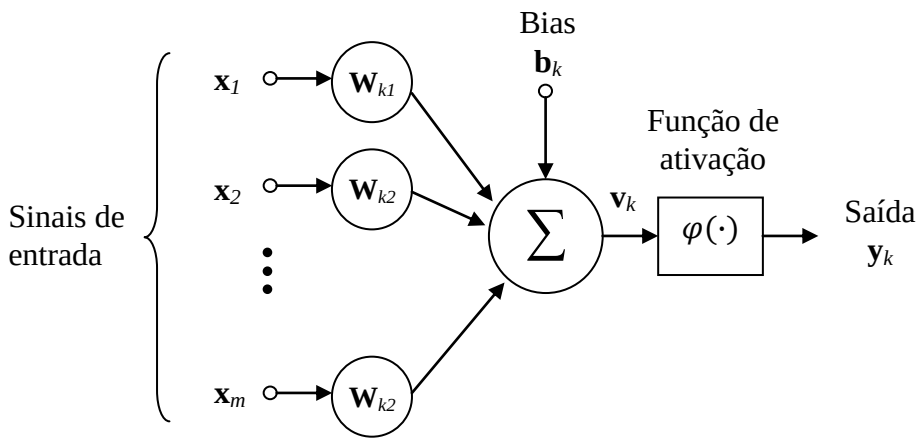


Figura 2: Modelo não-linear de um neurônio

Três elementos básicos deste modelo, que também é conhecido como *modelo de McCulloch-Pitts* em homenagem aos seus criadores, são:

1. As *sinapses*, caracterizadas pelos pesos W: um sinal $\mathbf{x}_j$ na entrada da sinapse *j* conectada ao neurônio *k* é multiplicado pelo peso sináptico $\mathbf{w}_{kj}$.

2. Um *somador* soma os sinais de entrada, ponderados pelos pesos sinápticos.

3. Uma *função de ativação*, também conhecida como *função restritiva*, limita o intervalo permissível de amplitude do sinal de saída a um valor finito.

O *bias*, sendo positivo, aumenta ou, sendo negativo, diminui a entrada da função de ativação.

As seguintes Equações (2.1), (2.2) e (2.3) fornecem suporte para a definição matemática de um neurônio:

$$u_k = \sum_{j=1}^{m} \left(w_{kj} \times x_j \right) \tag{2.1}$$

e

$$y_k = \varphi(u_k + b_k) \tag{2.2}$$

ou

$$\begin{aligned} v_k &= u_k + b_k \\ y_k &= \varphi(v_k) \end{aligned} \tag{2.3}$$

onde:

- $x_1, x_2, \ldots, x_m$ são os sinais de entrada;
- $w_{k1}, w_{k2}, \ldots, w_{km}$ são os pesos sinápticos do neurônio *k*;
- u_k é a saída do *combinador linear* (**sem** o *bias*);
- v_k é a saída do *combinador linear* (**com** o *bias*);
- b_k é o bias;
- $\varphi(\cdot)$ é a *função de ativação*; e
- y_k é o sinal de saída do neurônio.

O exemplo mais simples de uma função de ativação é a *Função de Limiar*, definida pela Equação 2.4:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \tag{2.4}$$

Outras funções de ativação são:

- Função Linear por Partes
- Função Sigmóide

Como a estrutura que se pretende representar é a de uma rede de neurônios, faz-se necessário, portanto, o conceito de arquiteturas de redes neurais. Como se pode perceber pelo conceito de rede, há conexões entre os diversos neurônios. O modelo estrutural de como esses neurônios estão conectados e, por que não dizer, como podem

se comunicar, é definido através de uma arquitetura de rede neural e do algoritmo de aprendizagem usado para treinar a rede. Há três classes de arquiteturas de redes neurais:

1) **Redes alimentadas adiante com camada única**

2) **Redes alimentadas diretamente com múltiplas camadas**

3) **Redes recorrentes**

Nos dois primeiros modelos de arquitetura, a idéia principal é de que os neurônios estão dispostos em camadas e que a saída de cada neurônio é utilizada como entrada para neurônios da camada imediatamente posterior. No caso das redes com camada única, há apenas os sinais de entrada (por muitas vezes chamados também de "camada de entrada") e um conjunto de neurônios que forma a "camada de saída". No caso das redes com múltiplas camadas, além das duas camadas do modelo anterior, há, entre elas, as chamadas "camadas ocultas" e, como se percebe, pode haver um número não determinado de camadas de neurônios. A Figura 3 ilustra o modelo de arquitetura com múltiplas camadas.

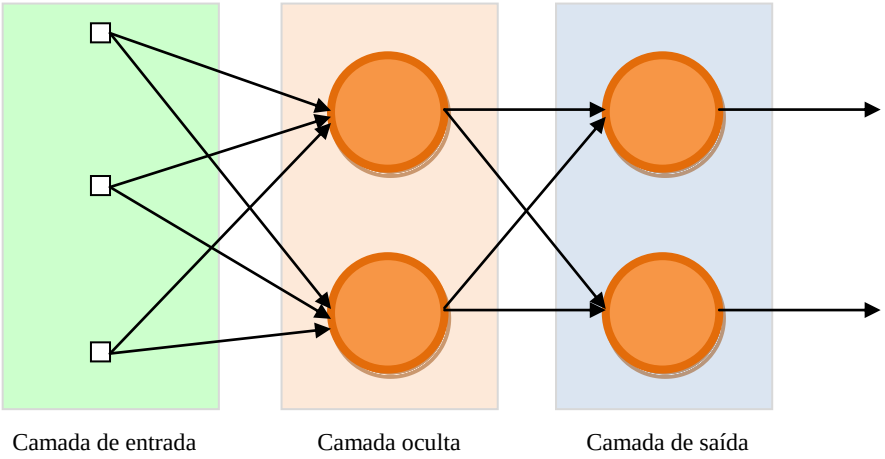


Figura 3: Arquitetura de rede alimentada adiante com múltiplas camadas

O modelo de redes recorrentes diferencia-se dos dois modelos anteriores no sentido de que conectado à saída do neurônio pode estar um laço de realimentação que permite que a saída de um neurônio sirva como uma nova entrada para ele próprio.

Enquanto o que foi apresentado até então é a definição do modelo de uma rede, resta saber como se dá, portanto, o aprendizado no modelo de redes neurais. Haykin (2001) apresenta uma definição baseada nas idéias originais de Mendel e McClaren, que em 1970 indicaram que a aprendizagem é:

> (...) um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre.

e, em suma, essa definição implica em que a rede neural é estimulada por um ambiente, sofre modificações para se adaptar a tal ambiente e, em conseqüência, responde a esse ambiente de uma maneira nova, possivelmente diferente da maneira anterior (o que é resultado da adaptação da rede neural). Haykin (2001), portanto, introduz o conceito de Aprendizagem por Correção de Erro da seguinte maneira: simplificando o modelo de neurônio artificial definido anteriormente, tem-se que a saída de um neurônio y_k e a resposta desejada d_k, uma informação que é indicada pelo supervisor (ou professor) são utilizadas para a definição do sinal de erro do neurônio $e_k(t)$ da seguinte forma (onde *t* é um instante de tempo):

$$e_k(t) = d_k(t) - y_k(t) \tag{2.5}$$

isso contribui para a definição de uma *função de custo* ou *índice de desempenho*:

$$\mathcal{E}(t) = \frac{1}{2} e_k^2(t) \tag{2.6}$$

Portanto, para se alcançar o aprendizado, o ideal é a minimização dessa função de custo.

Haykin (2001) ainda apresenta a *regra delta*, definida por Widrow e Hoff, em 1960, como: o ajuste feito em um peso sináptico de um neurônio é proporcional ao produto do sinal de erro pelo sinal de entrada da sinapse em questão. Isso leva à formulação das seguintes equações para atualização dos pesos sinápticos da rede:

$$\Delta w_{kj}(t) = \eta e_k(t) x_j(t) \tag{2.7}$$

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj}(t) \tag{2.8}$$

onde η é uma constante que define uma *taxa de aprendizado*.

Outros modelos de aprendizagem são:

- Aprendizagem Baseada em Memória
- Aprendizagem Hebbiana
- Aprendizagem Competitiva
- Aprendizagem de Boltzman

2.1.2 Algoritmos Genéticos

Algoritmos Evolucionários são baseados em noções da biologia, em especial, a teoria da evolução e da adaptação de organismos, para a realização de tarefas de otimização de problemas. Segundo Castro (2007) esta categoria indica a existência de uma população de indivíduos capazes de reprodução e sujeitos a variação genética seguida da seleção de novos indivíduos que mais se adaptem ao seu ambiente.

Há diversas variações do conceito de algoritmos evolucionários:

a) Algoritmos genéticos;
b) Estratégias de evolução;
c) Programação evolucionária; e
d) Programação genética.

entretanto, todas elas baseiam-se nos seguintes princípios:

a) *Uma população de indivíduos reproduzem-se e passam características para outras gerações (herança)*. Esse conceito determina que cada indivíduo da população, um *cromossomo*, carrega consigo uma solução em potencial para o problema de otimização em questão. A solução representa a *carga genética* do indivíduo, ou seja, os componentes do cromossomo, os *alelos*, e está codificada e estruturada de alguma forma.

Estes indivíduos são capazes de reprodução, ou seja, uma combinação entre dois indivíduos, e, durante este processo, gerações futuras carregam consigo características das gerações anteriores.

b) *Variação genética*. O mecanismo de reprodução de indivíduos gera modificações na carga genética dos indivíduos das populações seguintes. Um processo conhecido como mutação é o que permite que sejam exploradas novas soluções dentro do espaço de busca.

c) *Seleção natural*. O ambiente de existência dos indivíduos é competitivo, já que apenas um deles fornecerá uma solução mais adequada e útil para um determinado problema, portanto, é necessária uma forma de verificar o quanto um indivíduo está apto para participar do processo de geração de novos indivíduos, ou seja, para propagar sua *carga genética*. Esta avaliação é realizada por uma função de avaliação de desempenho, conhecida como *fitness.*

É importante ressaltar que algumas características de organismos vivos ou da forma como são encontrados na natureza não estão presentes na formulação dos métodos evolucionários. Conforme Bar-Cohen (2006) algumas destas características divergentes são as seguintes:

a) Na natureza, a ocorrência de variações climáticas e situações ambientais e mudanças em características nas espécies no decorrer do tempo são fundamentais para a verificação da aptidão dos organismos. Entretanto, isso não ocorre em algoritmos evolucionários, pois consideram que a função de avaliação é constante no tempo.

b) Na evolução natural, indivíduos de espécies distintas podem se confrontar em uma determinada situação de tal modo que apenas uma espécie sobreviva ou tenha melhor chance de sobrevivência. Nos algoritmos evolucionários não há a concepção de espécies diversificadas, ou seja, todos os indivíduos são da mesma espécie.

De forma bastante simplificada, tomando como base o trabalho de Krishna e Murty (1999) pode-se definir um algoritmo evolucionário como composto dos seguintes passos:

1. *Inicialização da população* ou *Geração da população inicial*. Geralmente, um processo aleatório é utilizado para a geração dos indivíduos da população inicial.

2. *Seleção*. Cromossomos de uma população anterior são selecionados para fazerem parte do processo de reprodução. Em geral, utiliza-se uma distribuição de probabilidade e a seleção é baseada no valor da função de avaliação de cada indivíduo.

3. *Mutação*. A solução codificada do indivíduo, os alelos, gerado na reprodução é misturada de tal forma a possibilitar que o algoritmo não fique preso a *ótimos locais*, mas que, através de um processo de exploração, possa estar mais próximo do *ótimo global*.

Este processo de geração de novos indivíduos e de conseqüente modificação da população é repetido diversas vezes, até que seja satisfeito um critério de parada.

Algumas das categorias das aplicações de algoritmos evolucionários são:

- Planejamento (ex.: roteamento, agendamento e empacotamento);
- Projeto (ex.: processamento de sinais);
- Simulação e identificação;
- Controle (ex. controle de planta); e
- Classificação (ex.: aprendizagem de máquina, reconhecimento de padrões e classificação).

Ainda, há outros campos de aplicação, como:

- Composição musical e artística;
- Eletrônica;

- Linguagem;
- Robótica;
- Engenharia;
- *Data mining* e descoberta de conhecimento;
- Indústria; e
- Processamento de sinais.

2.1.3 Inteligência de Enxames

Otimização por inteligência de enxames corresponde a métodos que têm se tornado alvo de pesquisas científicas recentes e Brabazon e O'Neill (2006) indicam que há duas variações deste modelo de enxames: um deles é inspirado no comportamento social de bandos de pássaros ou o modelo social de um grupo de pessoas; o outro é baseado no comportamento social de insetos, como formigas. Desta forma, o termo "inteligência de enxames" pode ter diversas definições. Castro (2007) apresenta algumas definições para "inteligência de enxames" encontradas na literatura:

> "Inteligência de enxames é uma propriedade de sistemas de agentes não-inteligentes com capacidades individuais limitadas que exibem comportamento coletivo inteligente" (White e Parurek, 1998).

> "[Inteligência de enxames] inclui qualquer tentativa de projetar algoritmos ou dispositivos distribuídos para solução de problemas inspirados no comportamento coletivo de insetos sociais e outras sociedades animais" (Bonabeau et al., 1999).

O método de Otimização por Colônias de Formigas (*Ant Colony Optimization – ACO)*, segundo Kennedy e Eberhart (2001), foi desenvolvido em 1997 quando Dorigo e colaboradores mostraram como o comportamento de formigas ao seguir o feromônio poderia ser utilizado para otimizar o Problema do Caixeiro Viajante. No presente trabalho, este método de otimização não será abordado em detalhes, indica-se que o leitor procure referência especializada no assunto, como Brabazon e O'Neill (2006).

Conforme Kenney e Eberhart (2001) *Particle Swam Optimization* (PSO) é um algoritmo estocástico baseado em população, modelado a partir da observação e simulação do comportamento social de bandos de pássaros. Embora bastante semelhante a métodos evolucionários, a diferença, segundo Omram (2004), está em que, em PSO, cada partícula individual beneficia-se do seu histórico, e não há tal mecanismo em métodos evolucionários.

A teoria que delineia o funcionamento do PSO está sob noções do Modelo de Cultura Adaptativa e de multidões de partículas e três princípios fundamentais são levados em consideração neste modelo de comportamento baseado em relacionamento social:

Avaliar: esta é uma das tendências mais características dos organismos vivos: avaliar um estímulo como positivo ou negativo. Neste sentido, o aprendizado é entendido como o resultado da avaliação que o organismo faz do meio no qual está inserido, se há condições que tornem o ambiente bom ou ruim diante de determinada situação.

Comparar: pela teoria social de Festinger (em 1954) pessoas utilizam experiências de outras pessoas como modo de aprendizado e mudança. No Modelo de Cultura Adaptativa os indivíduos são estimulados a se compararem com outros e realizarem medidas críticas que possam levar, ou não, à imitação do comportamento de um indivíduo de destaque dentro do grupo social.

Imitar: é a conseqüência lógica das duas características anteriores e é o que leva ao aprendizado do indivíduo dentro do grupo social. A imitação, por si, perpassa o conceito da repetição de ações. O aprendizado faz com que uma ação imitada possa ser realizada de uma forma diferente da original, e, ainda assim, levar a um resultado satisfatório.

Em métodos de otimização deve ser encontrado um ponto de equilíbrio entre exploração (a busca individual por uma solução melhor) e usufruto (fazer proveito do sucesso de outros indivíduos). Pouca exploração fará com que os indivíduos convirjam

nas primeiras soluções encontradas. Muito usufruto e nenhum indivíduo irá convergir (ou sempre continuarão a busca ou pararão quando encontrarem uma solução). Conforme Pomeroy (2003) este balanceamento também pode ser considerado do ponto de visa de individualidade e sociabilidade. Ou seja, são características desejáveis nos indivíduos a curiosidade e a busca individual pela melhor solução, mas também é interessante que os indivíduos aprendam com experiências passadas e com experiências de outros indivíduos. Esta não é uma noção estranha se considerarmos os indivíduos sendo seres humanos, já que esta é, realmente, uma das formas encontradas pela humanidade para a obtenção e transmissão do conhecimento para gerações futuras.

O algoritmo

Pela definição clássica do PSO, como indicam Kenney e Eberhart (2001), uma multidão de indivíduos (chamados partículas) voa através do espaço de busca; cada partícula representa uma solução em potencial. A posição de uma partícula é influenciada pela melhor posição visitada por ela própria (o que define um comportamento baseado em experiência ou memória) e a posição da melhor partícula na sua região de vizinhança, ou seja, considera as experiências dos indivíduos na sua vizinhança, princípio chamado de *lbest* (melhor local). Nesta avaliação, se a vizinhança representa toda a população, então a avaliação resulta no melhor global, chamado *gbest*. Conforme Omram (2004) o desempenho de cada partícula, ou seja, o quanto está perto do ótimo global, é medido através de uma função de aptidão que varia de acordo com o problema a ser otimizado.

Representação das partículas

Cada partícula é representada de acordo com as seguintes características:

x_i : a *posição atual* da partícula;

v_i : a *velocidade atual* da partícula;

y_i : a *melhor posição da partícula* (avaliação individual);

$\hat{y}_i$: a *melhor posição da vizinhança* da partícula.

Seja f uma função-objetivo, tal que a melhor posição de uma partícula individual, no passo de tempo *t*, é atualizada como:

$$y_i(t+1) = \begin{cases} y_i(t) & \text{se } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{se } f(x_i(t+1)) < f(y_i(t)) \end{cases} \tag{2.9}$$

No modelo *gbest* a melhor partícula é determinada a partir da avaliação de todos os indivíduos da população selecionando suas melhores posições individuais. Sendo a melhor posição denotada por $\hat{\mathbf{y}}$, então:

$$\hat{y}(t) \in \{y_0, y_1, ..., y_s\} = \min\{f(y_0(t)), f(y_1(t)), ..., f(y_s(t))\} \tag{2.10}$$

onde *s* denota o tamanho da multidão.

As partículas movimentam-se no espaço de acordo com a posição definida no modelo *gbest* e seguindo uma determinada velocidade. Esta velocidade, ou o tamanho do passo de atualização da posição da partícula, é específica para cada dimensão $j \in 1,...,\mathbf{N}_d$, onde $\mathbf{v}_{ij}$ representa o elemento *j* do vetor de velocidades da partícula *i*. A velocidade da partícula *i* é definida por:

$$\mathbf{v}_{ij}(t+1) = w\mathbf{v}_{ij}(t) + \mathbf{c}_1\mathbf{r}_{1j}(t)(\mathbf{y}_{ij}(t) - \mathbf{x}_{ij}(t)) + \mathbf{c}_2\mathbf{r}_{2j}(t)(\hat{\mathbf{y}}_j(t) - \mathbf{x}_{ij}(t)) \tag{2.11}$$

onde *w* é um peso de inércia, c_1 e c_2 são constantes de aceleração e $r_{1j}(t)$, $r_{2j}(t) \sim U(0,1)$. Aqui também são considerados três componentes principais:

a) Segundo Omram (2004) o termo *w*, o *peso de inércia*, funciona como uma memória de velocidades anteriores e controla o impacto destas velocidades

anteriores: um valor alto favorece a exploração, enquanto um valor pequeno leva a usufruir (o dilema *exploration-exploitation*);

b) O termo $y_i(t) - x_i$ é o *componente cognitivo*, que representa a experiência da própria partícula sobre a localização da melhor solução;

c) O termo $\hat{y}(t) - x_i(t)$ é o *componente social*, que representa a crença da multidão inteira a respeito de onde está a melhor solução.

Kenney e Eberhart (2001) indicam, ainda, que a velocidade máxima pode ser definida pelo usuário $V_{\max}$ e, geralmente, tem valor $\pm 4{,}0$.

Para garantir a convergência do algoritmo, a relação entre o peso de inércia e as constantes de aceleração deveria satisfazer a seguinte equação:

$$\frac{c_1 + c_2}{2} - 1 < w \tag{2.12}$$

e a posição x_i da partícula *i* é atualizada de acordo com a equação:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2.13}$$

Assim, o algoritmo atualiza a posição das partículas de acordo com as equações apresentadas e o processo é repetido até um determinado número de iterações, ou quando a atualização da velocidade for próxima de zero.

Em resumo, o algoritmo é dado como o seguinte:

```
1:   Para cada partícula i ∈ 1,...,s faça
2:          Inicialize aleatoriamente posições da partícula x_i
3:          Inicialize aleatoriamente (ou atribua zero) a velocidade
da partícula v_i
4:          y_i = x_i
5:   Fim-Para
6:   Repita
7:          Para cada partícula i ∈ 1,...,s faça
```

```
 8:                Avalie a aptidão da partícula i,  f(x_i)
 9:                Atualize y_i usando (Eq. 2.9)
10:                Atualize ŷ usando (Eq. 2.10)
11:                Para cada dimensão j ∈ 1,...,N_d faça
12:                      Atualize a velocidade usando (Eq. 2.11)
13:                Fim-Para
14:                Atualize a posição usando (Eq. 2.13)
15:           Fim-Para
16:    Até que seja satisfeito um critério de convergência
```

Listagem 1: Pseudocódigo para o PSO

Ótimos locais: o modelo *lbest*

Como afirma Omram (2004), neste modelo, uma multidão de partículas é dividida em vizinhanças que se sobrepõem. A *melhor partícula da vizinhança*, com posição $\hat{y}_i$, é determinada para cada vizinhança N_i. Os índices das partículas se sobrepõem ao redor de *s* e o tamanho da vizinhança é *l*. As equações de atualização são:

$$Ni = \{y_{i-l}(t), y_{i-l+1}(t),..., y_{i-1}(t), y_i(t), y_{i+1}(t),..., y_{i+l-1}(t), y_{i+l}(t)\} \tag{2.14}$$

$$\hat{y}_i(t+1) \in \{N_i \mid f(\hat{y}_i(t+1)) = \min\{f(y_i(t))\}, \forall y_i \in N_i\} \tag{2.15}$$

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) + c_2 r_{2j}(t)(\hat{y}_{ij}(t) - x_{ij}(t)) \tag{2.16}$$

Vizinhos representam o fator de colaboração social no PSO. Os vizinhos são geralmente determinados usando índices de partículas e topologias de vizinhança também podem ser usadas. Como já visto anteriormente, há uma situação quando *lbest* é o melhor de toda a multidão (i.e. a vizinhança é a multidão), neste caso $l = s$.

Modelos de Topologias de Vizinhança

Segundo Omram (2004) duas topologias comuns são:

a) **estrela** (ou *roda*) na qual todas as partículas da multidão estão conectadas (apenas) a um ponto central (*hub*);

b) **anel** (ou *círculo*) na qual as partículas estão organizadas na forma de um anel, ou seja, cada partícula possui um número de vizinhos à esquerda e à direita.

Um modelo proposto por Kennedy e Mendes, em 2002, como informa Omram (2004), apresenta uma topologia que segue uma topologia de *Von Neumann*, na qual as partículas estão conectadas na forma de uma grade e há quatro vizinhos para cada partícula: acima, abaixo, à esquerda e à direita.

A Figura 4 ilustra graficamente as diferentes topologias.

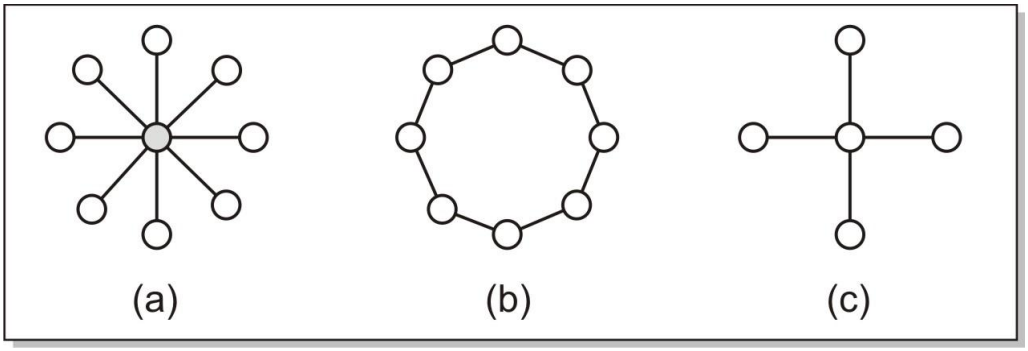


Figura 4: Modelo das diferentes topologias de vizinhança; a) Topologia Estrela; b) Topologia Anel; c) Topologia Von Neumann

Segundo Omram (2004) a escolha da topologia da vizinhança tem um efeito profundo na propagação da melhor solução encontrada pela multidão. Usando-se o modelo *gbest* a propagação é muito rápida, já que todas as partículas são afetadas pela melhor solução encontrada num tempo *t*. Por sua vez, nos modelos (b) e (c) a convergência será mais demorada porque serão necessários vários passos na propagação da melhor solução.

2.1.4 Sistemas Imunológicos Artificiais

Segundo Castro (2007) Sistemas Imunológicos Artificiais (*Artificial Immune Systems – AIS)* é uma terminologia para sistemas adaptativos, que surge na década de

1980, que extraem idéias e metáforas do sistema imunológico biológico para resolver problemas.

A idéia principal, conforme indica Castro (2007), é que, como todos os organismos vivos possuem a habilidade de resistir a doenças causadas por agentes, os patógenos, por exemplo, por vírus e bactérias, a primeira regra do sistema imunológico é proteger o corpo ou a estrutura do organismo vivo. As células do sistema imunológico são capazes de reconhecer padrões moleculares (o que é um tipo de assinatura molecular) que só está presente em patógenos. Uma vez que o patógeno é reconhecido pelo sistema imunológico, as células enviam sinais para outras células do sistema imunológico para lutar contra a doença.

O *framework de engenharia imunológica*, segundo Castro (2007), é um processo de engenharia composto pelos seguintes elementos:

a) uma representação dos componentes do sistema;
b) um conjunto de mecanismos para avaliar a interação de indivíduos com o ambiente e entre eles próprios. O ambiente é simulado por uma série de estímulos de entrada (ou padrões), uma ou mais funções de avaliação (*fitness*);
c) procedimentos de adaptação que governam a dinâmica do sistema, ou seja, como seu comportamento varia com o tempo.

Outros métodos em computação natural são (Castro, 2007):

- **Simulação da natureza**, que inclui métodos que formulam um processo sintético que tem o objetivo de criar padrões, formas, comportamentos e organismos que remontem à idéia da concepção natural da vida (*life-as-we-know-it*). A metodologia é imitar ou mimetizar fenômenos naturais e exemplos de métodos incluem Geometria Fractal e Vida Artificial.
- **Computação com materiais naturais**, que se utiliza de materiais naturais para realizar computação e seu objetivo é utilizar estes materiais para

substituir computadores baseados em silício. Alguns métodos são Computação por DNA e Computação Quântica.

Este capítulo encerra-se com a explanação de diversos métodos de computação natural e o que será visto nos capítulos seguinte é a aplicação destes métodos em tarefas de reconhecimento de padrões e detecção de aglomerados. Como os métodos apresentados são bastante inovadores, principalmente Sistemas Imunológicos Artificiais, Vida Artificial e Computação Quântica, indica-se que o leitor refira-se às respectivas referências para saber detalhes aprofundados, pois uma explicação em maior detalhe foge ao escopo deste trabalho.

Capítulo 3

Detecção de Agrupamentos através de técnicas baseadas em Computação Natural

Baseando-se em conceitos e fundamentos apresentados em capítulos anteriores, o capítulo atual apresenta informações técnicas e detalhadas sobre técnicas de detecção de agrupamentos utilizando métodos baseados em Computação Natural, em específico, dois métodos são selecionados: Algoritmos Genéticos e Otimização por Enxames de Partículas. Inicialmente serão apresentados conceitos sobre Reconhecimento de Padrões e Detecção de Agrupamentos e, posteriormente, estes conceitos serão aplicados à Computação Natural, trazendo à tona: Algoritmo K-means Genético e um Algoritmo de *Clustering* Baseado em Otimização por Enxames de Partículas.

3.1 Detecção de Agrupamentos

Segundo Jain e Dubes (1988) Detecção de Agrupamentos, ou *Clustering* (também conhecido como *Cluster Analysis* ou *Data Clustering*), é um tipo de classificação imposta a um conjunto finito de objetos, e o relacionamento entre tais objetos é representado através de uma matriz de proximidade, na qual linhas e colunas correspondem a objetos. O valor de um cruzamento linha-coluna, ou seja, um padrão, representa a proximidade entre os objetos relacionados e é definida por uma métrica de distância, que geralmente é a Distância Euclidiana. Segundo Jain e Dubes (1988), como esta é uma classificação não-supervisionada, esta matriz é o único dado de entrada para um algoritmo de *clustering*. Um conjunto bem definido de objetos representa um *cluster*. Um *cluster* é geralmente representado por um centro ou **centróide**, ou seja, um elemento mais representativo do *cluster*.

Segundo Jain et al (1999) pode-se proposta uma classificação é apresentada graficamente pela Figura 5 e os métodos são detalhados como:

a) **Exclusivos *versus* não-exclusivos**. Nos métodos exclusivos, uma classificação é uma partição no conjunto de objetos, e cada objeto pertence a, exatamente, um sub-conjunto, ou *cluster*. Nos não-exclusivos (ou superpostos), a classificação permite que um objeto esteja em vários conjuntos. *Fuzzy clustering* é um tipo desta classificação no qual é associado um grau de pertinência de padrão a um *cluster*.

b) **Intrínsecos *versus* extrínsecos**. Uma classificação intrínseca usa apenas a matriz de proximidade para avaliar o problema e realizar a classificação, por isso também pode ser chamada de classificação por aprendizado não-supervisionado. A extrínseca depende de um indivíduo, o professor, para determinar superfícies discriminantes que separam os objetos em categorias.

c) **Hierárquicos *versus* particionais**. Estes classificadores diferem na forma da estrutura dos dados, no primeiro, os dados são representados em seqüências em cascata, enquanto que no segundo há apenas uma partição.

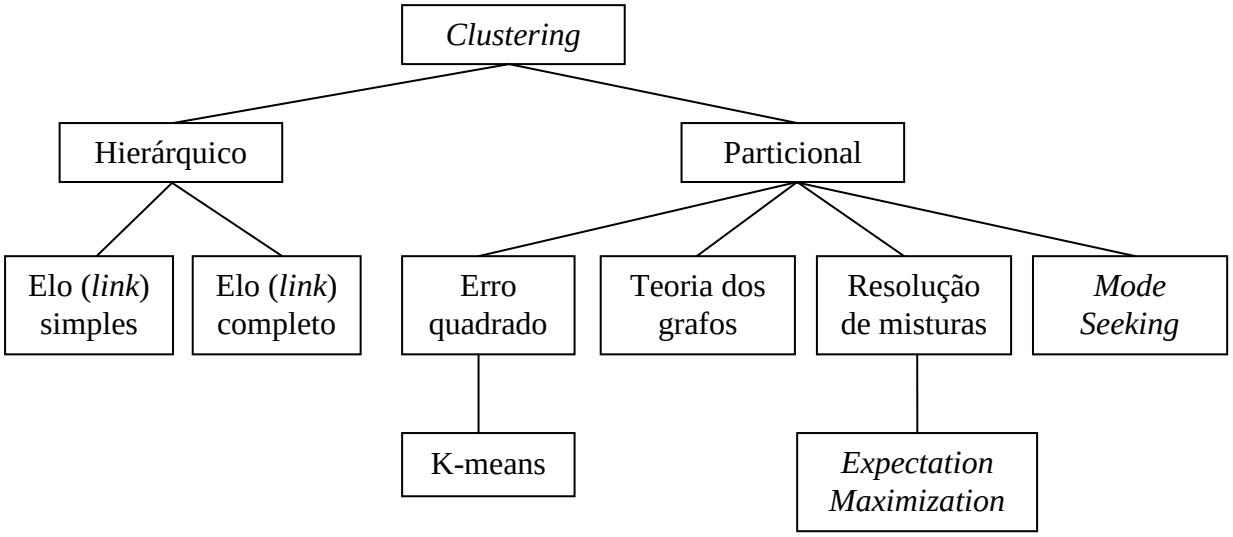


Figura 5: Taxonomia de métodos de clustering

Segundo definição de Jain et al (1999) os componentes de uma tarefa de *clustering* são:

a) *Representação do padrão* (extração e seleção de características) refere-se ao número de classes e tipo de características disponíveis para o algoritmo de *clustering*; *seleção de características* identifica os subconjuntos mais efetivo das características originais para ser usado na tarefa de *clustering* e *extração de características* é o processamento dos dados de entrada.

b) Uma *medida de distância* é usada para determinar a proximidade dos padrões e é, usualmente, definida através de uma função de distância em relação a um par de padrões. Uma simples, e possivelmente a mais utilizada, função de distância é a *Distância Euclidiana*, que é definida como:

$$d_2(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2 \right)^{1/2} = \|\mathbf{x}_i - \mathbf{x}_j\|^2 \tag{3.1}$$

c) *Clustering* é a tarefa de encontrar grupos (ou rotular dados) e pode ser *rígida* (quando os elementos apenas pertencem a um grupo) ou *fuzzy* (quando um elemento pertence a um grupo de acordo com um grau de pertinência).

d) *Abstração de dados* é uma fase opcional e extrai uma representação simples e compacta de um conjunto de dados e, no caso dos algoritmos em questão, os padrões mais representativos são escolhidos, ou seja, os centróides.

e) *Avaliação da saída* é o processo de avaliar o resultado do algoritmo. Técnicas de validação de *cluster* são métodos utilizados para avaliar, de forma quantitativa, o resultado do algoritmo de *clustering* de modo a encontrar o melhor particionamento dos dados. Conforme Omram et al (2006) técnicas de validação de *cluster* são também utilizadas em tarefas de *clustering* dinâmico, técnica esta que tenta resolver o problema de *clustering* realizando diversas avaliações, e posteriormente, avalia os resultados através de um índice de validação e seleciona os melhores centróides encontrados pelo algoritmo.

3.1.1 Algoritmo K-means

O Algoritmo K-means é o algoritmo particional mais utilizado, segundo Forgy (1965) seu objetivo é minimizar o funcional *J*, que representa a minimização da distância entre objetos (padrões) e *clusters*:

$$J_{\text{K-means}} = \sum_{k=1}^{K} \sum_{j \in S_k} d^2(x_j, c_k) \tag{3.2}$$

onde:

k é o número de *clusters* avaliados (espaço definido por S_k);

x_j é o padrão *j* avaliado em relação ao centróide c_k; e

$d^2(x_j, c_k)$ é a distância entre o padrão x_j e o centróide c_k.

O algoritmo inicia com *K* centróides (que podem ser definidos aleatoriamente ou baseados em alguma informação *a priori*). Então, iterativamente:

a) são calculadas as distâncias de todos os elementos do conjunto de dados em relação aos *K* centróides, os elementos de menor distância em relação aos centróides forma grupos (*clusters*); e

b) são atualizados os centróides em relação a cada *cluster*.

As funções de pertinência e de peso são definidas como:

$$u(c_k \mid x_j) = \begin{cases} 1 & \text{se } d^2(x_j, c_k) = \arg\min_k \{d^2(x_j, c_k)\} \\ 0 & \text{c.c.} \end{cases} \tag{3.3}$$

$$w(x_j) = 1 \tag{3.4}$$

Desta forma, segundo Hamerly e Elkan (2002), pela função de peso constante, todos os padrões têm igual importância.

Segundo Turi (2001) as principais vantagens deste algoritmo são:

a) é de fácil implementação; e

b) a complexidade é $O(N_p)$, tornando-o bastante aplicável a grandes conjuntos de dados.

E, segundo Davies (1997), suas principais desvantagens são:

a) é dependente dos dados;
b) é um algoritmo guloso (*greedy*) que depende de condições iniciais, que podem levá-lo à convergência a soluções sub-ótimas; e
c) o número de *clusters* (*K*) precisa ser informado pelo usuário.

3.1.2 Soft Computing

Computação suave (*soft computing*), como definido por Zadeh (1994), é um conjunto de metodologias que trabalham sinergicamente e provêm, de uma forma ou outra, capacidade de processamento flexível da informação para lidar com situações de ambigüidade encontradas em problemas da vida real. A computação deve se aproveitar desses conceitos para que possam ser explorados novos espaços de soluções que podem estar dentro de um conjunto de decisões baseadas em incertezas, mas que podem levar à exploração e tolerar aproximações de soluções. Este conceito é declarado em contraste com o de computação rígida (*hard computing*) no qual as considerações tradicionais levam à definição de que os resultados obtidos em computações devem ser precisos, certos e rigorosos.

Levando-se em consideração características de pensamento e forma de raciocínio dos seres humanos, as informações que são apresentadas em situações reais e as soluções que são fornecidas não são completamente certas. Quando somos apresentados a um problema no qual é muito alto o grau de certeza de uma solução (seja ela boa ou ruim, e poderíamos chamar isso de esperança ou expectativa) tendemos a nos retrair e aceitar a situação. Isto é, há menos possibilidade de que algo seja diferente do que o que pode ser uma grande certeza. Pelo contrário, se consideramos que a solução pode ser questionada, damos espaço a uma das características mais comuns e extremamente importantes no ser humano: a curiosidade, a partir da dúvida.

Exemplificando, considera-se o trabalho de um funcionário em um cartório, que precisa verificar assinaturas de punho em documentos; se ele pensar como um computador, nos seus métodos de lógica *booleana*, então, qualquer mínima variação na assinatura poderá levar à consideração de que a mesma está incorreta, invalidando, portanto, o documento. Entretanto, o mesmo já não acontecerá se a avaliação permite certo nível de desordem (considerando que o signatário, naquele instante, poderia estar nervoso e os riscos saíram trêmulos). Considerando estas características Zadeh conclui que o desafio é explorar a tolerância da imprecisão através de métodos computacionais que permitam uma solução aceitável a um custo menor.

3.1.3 Fuzzy Clustering

Se os *clusters* em um conjunto de dados estão bem divididos (Figura 6a), então os métodos exclusivos apresentam uma classificação bem sucedida. Mas se há ambigüidade ou incerteza sobre a classificação, como na estrutura da Figura 6b, então há superposição dos clusters. Mesmo que, visualmente, seja fácil para um ser humano perceber que ali há três *clusters*: o mais à esquerda, como representado pelo padrão x_i, o do centro, como representado por x_j e o mais à direita, como representado por x_k, a mesma tarefa, realizada de forma algorítmica bem facilmente não conseguirá chegar a este resultado. Como informam Jain e Dubes (1998) estes *clusters* possuem fronteiras *fuzzy*.

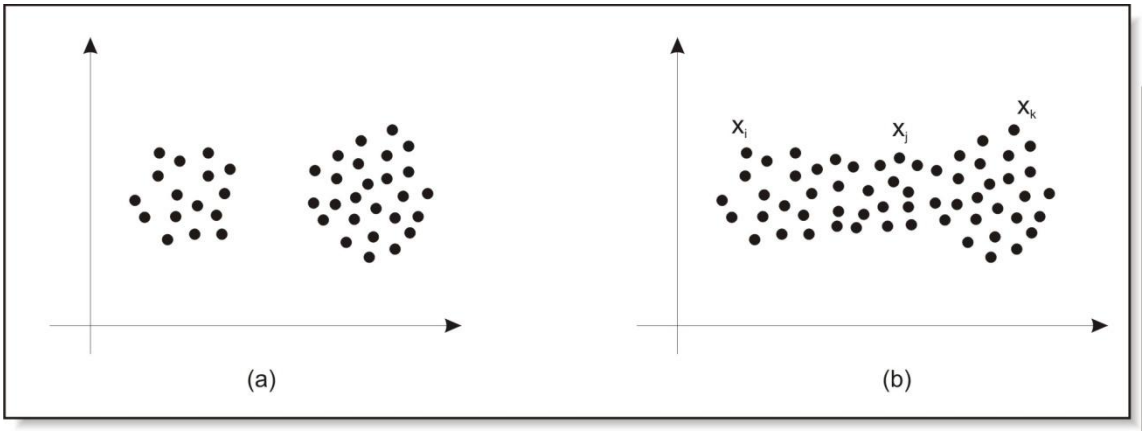


Figura 6: Exemplos de estruturas de clusters: (a) clusters bem separados; (b) clusters superpostos

Segundo Zadeh (1965) a Teoria dos Conjuntos *Fuzzy* permite que um objeto pertença a um conjunto com um determinado grau de pertinência, que é um valor no intervalo [0, 1]. Esta pequena, mas fundamental modificação no conceito de *clustering* rígido (*crispy*), tornou capaz uma generalização dos algoritmos de partição, como o K-means, e a criação de uma versão *fuzzy*, do mesmo, o algoritmo *Fuzzy C-means*.

Fuzzy C-means

O algoritmo (abreviado como FCM) é definido por Bezdek et al (1987) como: seja $c \geq 2$ um inteiro; seja $X = \{x_1,...,x_n\}$ um conjunto finito de dados contendo pelo menos $c < n$ pontos distintos; e seja R^{cn} o conjunto de todas as matrizes reais $c \times n$. Uma partição do conjunto X é representada por uma matriz $U = [u_{ik}] \in R^{cn}$ cujos elementos satisfazem:

$$u_{ik} \in [0,1], \qquad 1 \leq i \leq c;\ 1 \leq k \leq n \tag{3.5a}$$

$$\sum_{i=1}^{c} u_{ik} = 1, \qquad 1 \leq k \leq n \tag{3.5b}$$

$$\sum_{k=1}^{n} u_{ik} > 0, \qquad 1 \leq i \leq c \tag{3.5c}$$

sendo v_i o centróide do *cluster i* (elemento representativo). Assim, as partições e os centróides são escolhidos a partir da minimização do seguinte funcional:

$$J_{\mathrm{m}}(U,v)=\sum_{k=1}^{n}\sum_{i=1}^{c}\left(\mu_{ik}\right)^{m}\left|x_k-v_i\right|^2 \tag{3.6}$$

onde: $1\leq m'<\infty$ (*fuzzyficador*) e $|\cdot|$ é uma medida de distância.

Ainda, as seguintes condições são necessárias:

$$v_i=\frac{\sum_{i=1}^{n}\left(u_{ik}\right)^{m}x_k}{\sum_{k=1}^{n}\left(u_{ik}\right)^{m}}, \quad \text{para todo } i \tag{3.7a}$$

e, para cada *k* tal que $d_{ik}=\left|x_k-v_i\right|^2>0$ para todo *i*, então

$$u_{ik}=\frac{1}{\sum_{i=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{1}{m-1}}}, \quad \text{para todo } i \tag{3.7b}$$

Segundo Hamerly e Elkan (2003) o algoritmo FCM fornece melhores resultados do que o K-Means.

3.2 Detecção de Agrupamentos através de Algoritmos Genéticos

Como uma extensão aos modelos de algoritmos de particionamento apresentados anteriormente, a saber K-Means e FCM, esta seção dedica-se ao estudo dos métodos de particionamento baseados nestes algoritmos mas levando em consideração conceitos de Computação Natural, em específico, a utilização dos Algoritmos Genéticos.

3.2.1 Algoritmo K-Means Genético

O Algoritmo K-Means Genético (ou *Genetic K-means Algorithm – GKA*), criado por Krishna e Murty (1999) e estendido por Lu et al (2004) para a criação do *Algoritmo K-Means Genético Rápido* (*Fast Genetic K-Means Algorithm - FGKA*), estabelece um critério de avaliação baseado na minimização da *Variação Total Intracluster* (ou *Total*

Within Cluster Variation –TWCV*)*, uma função objetivo definida como: dado **X**, o conjunto de *N* padrões, e X_{nd} a *d-ésima* característica de um padrão *Xn*, G_k o *k*-ésimo *cluster* e Z_k o número de padrões em G_k, a TWCV é definida como:

$$TWCV = \sum_{n=1}^{N}\sum_{d=1}^{D} {X_{nd}}^2 - \sum_{k=1}^{K}\frac{1}{Z_k}\sum_{d=1}^{D} {SF_{kd}}^2 \tag{3.8}$$

onde SF_{kd} é a soma das *d*-ésimas características de todos os padrões em G_k. Segundo Krishna e Murty (1999) a TWCV também é conhecida como uma medida de erro quadrático. Conforme indicam Omram et al (2006) a função objetivo, portanto, tenta minimizar a TWCV, encontrando o particionamento que possui centróides que atendem aos conceitos de:

a) Compactação: padrões de um *cluster* são similares entre si e diferentes de padrões de outros *clusteres*; e
b) Separação: os centróides dos *clusters* são bem separados (considerando a medida da Distância Euclidiana).

Um outro método de *clustering* através de Algoritmos Genéticos definido por Bandyopadhyay e Maulik (2002) utiliza, como função objetivo, o índice Davies-Boudin, que é uma função da razão entre a soma da dispersão intra-*cluster* e a separação entre-*cluster*:

$$S_{i,q} = \left(\frac{1}{|C_i|}\sum_{x\in C_i}\{\|x - z_i\|_2^q\}\right)^{1/q} \tag{3.9}$$

onde z_i é o centróide do *cluster* C_i e é definido como $z = \frac{1}{n_i}\sum_{x\in C_i} x$, n_i é o número de padrões no *cluster* C_i. A função usada para calcular a distância entre padrões e centróides é a Distância de Minkowski. Outra abordagem para a definição de uma função objetivo é apresentada por Cowgill e Harvey (1999), o Critério de Razão de Variação (ou *Variance Ratio Criterion – VRC*) que é definido como:

$$VRC = \frac{trace\ B/(k-1)}{trace\ W/(n-k)} \tag{3.10}$$

onde *n* e *k* são o número total de objetos e o número de *clusters* na partição; ***B*** e ***W*** são as matrizes de covariância, respectivamente, de dispersão entre-*clusters* e dispersão intra-cluster.

Desconsiderando-se a função objetivo os principais aspectos do GKA, segundo Krishna e Murty (1999) e Lu et al (2004), são:

1. **Codificação**. Refere-se ao processo da codificação da solução (o cromossomo). Uma maneira de fazer isso é a codificação de *strings* na qual para *Z* soluções (partições), representadas por *strings* de tamanho *N*, cada elemento da *string*, um alelo, contém o número de um *cluster*. O conjunto de cromossomos define a população.
2. **Inicialização**. A população inicial P_0 é definida aleatoriamente, da seguinte forma:cada alelo recebe (é inicializado) o número de um *cluster*. A próxima população P_{i+1} é definida em termos da *seleção*, da *mutação* e do *operador K-means*.
3. **Seleção**. Cromossomos de uma população anterior são escolhidos aleatoriamente de acordo com a seguinte distribuição:

$$P_z = \frac{F(S_z)}{\sum_{z=1}^{Z} F(S_z)}\ (z = 1, \dots, Z) \tag{3.11}$$

onde $F(S_z)$ é o valor da função de desempenho da solução Sz, que é o valor de TWCV, como visto na Eq. 4.8.

4. **Mutação**. O operador de mutação troca o valor de um alelo dependendo das distâncias dos centróides do *cluster* do respectivo padrão e de acordo com a seguinte distribuição:

$$p_k = \frac{1.5 \times dmax(X_n) - d(X_n, c_k) + 0.5}{\sum_{k=1}^{K}(1.5 \times dmax(X_n) - d(X_n, c_k) + 0.5)} \tag{3.12}$$

onde $d(X_n, c_k)$ é a Distância Euclidiana entre o padrão X_n e o centróide c_k do k-ésimo *cluster*, e $dmax(X_n) = \max_k\{d(X_n, c_k)\}$.

5. **O Operador K-Means** (*K-Means Operator – KMO*). Este operador é usado para dar agilidade ao processo de convergência e está relacionado com um passo do algoritmo K-means "clássico". Dado um cromossomo, cada alelo é modificado de forma a estar mais próximo de seu centróide.

Lu et al (2004) propuseram um método chamado *Algoritmo K-Means Genético Incremental* (*Incremental Genetic K-Means Algorithm* – IGKA) que herda as vantagens do FGKA, incluindo a convergência a um ótimo global. O IGKA calcula o valor de TWCV e realiza o processo de *clustering* incrementalmente. A Figura 7 ilustra o pseudo-código do IGKA.

```
1: P[0] = Initialize();
2: BEST_TWCV = +Inf;
3: i = 1;
4: while (~Terminate()) {
5:    P[i] = Selection(P[i]);
6:    P[i] = Mutation(P[i]);
7:    P[i] = KMeans(P[i]);
8:    TWCV = Fitness(P[i]);
9:    if (TWCV < BEST_TWCV) {
10:         BEST_TWCV = TWCV;
11:   }
12:   i = i + 1;
13: }
```

Figura 7: Pseudo-código do Algoritmo K-Means Genético Incremental

O critério de parada pode ser, por exemplo, um número máximo de gerações ou um valor definido pelo usuário para o limite de TWCV ou de qualquer outra definição para a função de desempenho (*fitness*).

Uma outra abordagem proposta por Bandyopadhyay e Maulik (2002) é chamada *Algoritmo Genético K-Means* (*K-Means Genetic Algorithm – KGA*) e inclui uma pequena modificação em relação ao que foi apresentado anteriormente: o *operador de crossover* é adicionado ao algoritmo e ele é um processo probabilístico que troca

informações entre dois pares de cromossomos pais para gerar novos cromossomos [descendentes]. A função objetivo é baseada no erro quadrado médio e é definida como:

$$\mathcal{M} = \sum_{i=1}^{K} \sum_{\forall \mathbf{x}_j \in C_i} \|\mathbf{x}_j - \mathbf{z}_i\| \tag{3.13}$$

onde $\|\mathbf{x}_j - \mathbf{z}_i\|$ é representado por uma função de distância, que é aplicada sobre $\mathbf{x}_j$, o padrão de entrada, e $\mathbf{z}_i$, um centróide. O objetivo é maximizar a função de avaliação de desempenho, que é definida como $1/\mathcal{M}$.

Assim sendo, o algoritmo, com as modificações propostas, é definido como mostra o pseudo-código a seguir:

```
1: P = InitializePopulation();
2: i = 1;
3: while (~Terminate()) {
4:   for (i = 1 to POPULATION_SIZE) {
5:      PerformClustering();
6:      CalculateNewCenters();
7:      TWCV = Fitness(P);
8:      if (TWCV < BEST_TWCV) {
9:        BEST_TWCV = TWCV;
10:     }
11:     if (~Terminate()) {
12:       P = Selection(P);
13:       P = CrossOver(P)
14:       P = Mutation(P);
15:     }
16:     i = i + 1;
17:   }
```

Figura 8: Pseudo-código do GKA

As funções **`PerformClustering()`** e **`CalculateNewCenters()`** estão relacionadas com um passo do algoritmo K-means, como indicado anteriormente, e elas atuam, respectivamente, para:

1. Calcular as distâncias dos padrões aos centróides e para cada padrão mais próximo de um centróide, indicar que o padrão pertence àquele centróide (isso define uma classe, um *cluster*);

2. Calcular a média de cada *cluster* e usar a média definir a posição dos novos centróides.

3.2.2 Algoritmo Fuzzy C-Means Genético

O Algoritmo Fuzzy C-Means Genético (GFCMA) é uma variação do Algoritmo K-means Genético que utiliza o algoritmo Fuzzy C-Means (FCM) como operador base para as funções **`PerformClustering()`** e **`CalculateNewCenters()`**. Como visto anteriormente, o algoritmo FCM apresenta melhores resultados do que o algoritmo K-means e faz parte dos objetivos deste trabalho avaliar a eficácia prática desta afirmação, ou seja, verificar de forma quantitativa e qualitativa se a utilização do FCM pode apresentar melhores resultados quando se trata de segmentação de imagens médicas.

3.3 Detecção de Agrupamentos através do PSO

Enquanto a seção 4.2 tratou da utilização do método de detecção de agrupamentos através de algoritmos genéticos baseados no método K-Means, esta seção apresenta as definições de detecção de agrupamentos através do método de otimização através de enxames de partículas (PSO).

Diferentes abordagens procuram implementar um método baseado em PSO para a tarefa de *clustering*, como: Omram (2004), Omram et al (2006) e Cohen e Castro (2006), estes últimos definem um algoritmo que, embora apresente bons resultados, não segue as definições originais do PSO, a saber: cada partícula corresponde a um protótipo de *cluster* e, assim, codifica apenas parte da solução e todo o enxame de partículas é necessário para realizar o *clustering*. Embora esse método seja considerado na literatura como *clustering* baseado em PSO, uma avaliação mais detalhada indica que isso não procede, já que no PSO, cada partícula carrega consigo uma solução para o problema (Omram, 2004) e não apenas parte da solução. Portanto, para definição de um algoritmo de *clustering* baseado em PSO seguimos a definição de Omram (2004) e Omram et al (2006).

Um Algotitmo de *Clustering* baseado em PSO pode ser definido, segundo Omram (2004) e Omram et al (2006) como: no contexto de *clustering* de dados, uma única partícula representa um conjunto de K centróides, em outras palavras, cada partícula representa uma solução para o problema de *clustering*, portanto, o enxame representa um conjunto de soluções candidatas para *clustering* de dados. A qualidade da solução de uma partícula pode ser medida conforme a seguinte equação:

$$f(\mathbf{x}_i, \mathbf{Z}_i) = w_1 d_{max}(\mathbf{Z}_i, \mathbf{x}_i) + w_2(\mathbf{Z}_{max} - d_{min}(\mathbf{x}_i)) \tag{3.14}$$

onde $\mathbf{Z}$ é o conjunto de dados (os padrões de entrada); $\mathbf{Z}_{max}$ é o valor máximo no conjunto de dados; $\mathbf{Z}_i$ é uma matriz que representa a associação de padrões aos *clusters* da partícula i e cada elemento $z_{i,k,p}$ possui 1 (um) se o padrão p pertence ao centróide k da partícula i; e possui (zero) c.c.; w_1 e w_2 são constantes definidas pelo usuário; $d_{min}(\mathbf{x}_i)$ é a Distância Euclidiana mínima entre qualquer par de *clusters*; e d_{max} é a máxima Distância Euclidiana média entre partículas e seus *clusters*:

$$d_{max}(\mathbf{x}_i, \mathbf{Z}_i) = \max_{k=1,\ldots,K} \left\{ \sum_{\forall z_p \in C_{i,k}} d(z_p, m_{i,k})/n_{i,k} \right\} \tag{3.15}$$

onde $n_{i,k}$ é o número de padrões que pertencem ao *cluster* k (ou $C_{i,k}$) da partícula i. Segundo Omram (2004) esta definição da função de avaliação de desempenho minimiza a distância entre padrões e os centróides de seus *clusters*, e maximiza a distância entre qualquer par de *clusters*. Outra forma de calcular o desempenho de uma partícula é através de índices de validação, como indicam Omram et al (2006), e um deles é o índice de Dunn, que identifica *clusters* compactos e bem separados. O pseudo-código do Algoritmo de *Clustering* baseado em PSO é definido como:

```
1: K; // o número de centróides
2: N; // o número de partículas
3: P; // o número de padrões [de entrada]
4: Data; // os padrões de entrada (i.e. o conjunto de dados)
5: Z = [,]; // a matriz de associação partícula-cluster-padrão
6: X = []; // o conjunto de N partículas
7: Pi = []; // o conjunto das melhores posições das partículas
```

```
8: Pg = +inf; // a melhor posição [entre todas as partículas]
9: V = []; // a velocidade das partículas
10:
11:   for i = 1 to N // para cada partícula i
12:    X(i) = InitializeParticle();
13:    V(i) = InitializeVelocity(X(i));
14:   End
15:
16:   for t = 1 to T_MAX do
17:    for i = 1 to N do
18:     // realiza clustering
19:     for p = 1 to P do
20:      d = dist(Z(p), X(i));
21:      dmin = min(d);
22:      Z(i,p) = dmin;
23:     end
24:
25:     q = ParticleQuality(X(i),Z(i));
26:     Pi(i) = FindParticleBest(Pi(i), q);
27:     Pg = FindGlobalBest(X(i), Pi(i), Pg, q);
28:    end
29:
30:    for i = 1 to N do
31:     V(i) = UpdateVelocity(V(i), X(i), Pi(i), Pg);
32:    end
33:
34:   end
```

Figura 9: Pseudo-código do Algoritmo de Clustering baseado em PSO

As funções que definem o pseudo-código são as seguintes:

a) **`InitializeParticle()`** e **`InitializeVelocity()`** inicializam, aleatoriamente, a posição das partículas e as velocidades iniciais das partículas, respectivamente.

b) **`Evaluate()`** representa a função objetivo do problema.

c) **`FindParticleBest()`** encontra a melhor posição da partícula, considerando as melhores posições anteriores (i.e. a memória da partícula) (Eq. 2.10).

d) **`ParticleQuality()`** avalia a qualidade da solução da partícula utilizando, por exemplo, um índice de validação.

e) **`FindGlobalBest()`** encontra a melhor posição considerando todo o enxame de partículas.

f) **`UpdateVelocity()`** representa a atualização da velocidade da partícula (Eq. 2.11)

g) **`UpdatePosition()`** representa a modificação da posição da partícula (Eq. 2.13).

Como demonstrado na Figura 9, a tarefa de *clustering* é realizada entre as linhas 18 e 23. É bastante simples e perceptível que o que há entre essas linhas é o mesmo conceito da definição do "operador K-means" no GKA, ou seja, correspondem a um passo do algoritmo K-means:

1. Calcular distâncias entre padrões e centróides e associar padrões aos centróides mais próximos;
2. Calcular a média entre os padrões de cada centróide e reajustar os centróides.

As principais considerações em relação ao algoritmo apresentado na Figura 9 são:

a) O critério de término do algoritmo é baseado em um valor definido pelo usuário que representa o máximo de iterações (T_MAX);

b) O passo que avalia a proximidade de um padrão aos centróides de uma partícula (linhas 18 a 22) é usado como informação para a medida da qualidade da partícula (linha 23).

3.4 Índices de Validação de *Clustering*

Tanto os métodos baseados em Algoritmos Genéticos quanto PSO requerem que seja feita uma análise da qualidade da solução obtida. Como visto, diversos autores procuram formas diferentes de definir a função objetivo a ser otimizada. Neste trabalho a função objetivo é sempre baseada no resultado obtido de um Índice de Validação de *Clustering*. A literatura apresenta diversos exemplos de índices de validação, mas para efeitos práticos, vamos basear nossas considerações nos índices de validação a seguir.

3.4.1 Índice Davies-Bouldin (DB)

Conforme Silva et. al. (2003) o Índice Davies-Bouldin pode usado para avaliação da partição dos dados e é definido como:

$$\frac{1}{c}\sum_{k=1}^{c}\max_{c\neq k}\left\{\frac{S_c(Q_k)+S_c(Q_l)}{d_{ce}(Q_k,Q_l)}\right\} \tag{3.16}$$

onde c é o número de agrupamentos, $S_c(Q_k)$ representa a distância intra-agrupamento, $d_{ce}(Q_k, Q_l)$ representa a distância entre os agrupamentos Q_k e Q_l. $S_c(Q_k)$ é dado por

$$S_c(Q_k)=\left\{\frac{1}{N_K}\sum_{j=1}^{N_k}\left|x_j-c_k\right|^q\right\}^{1/q} \tag{3.17}$$

onde $x_j \in Q_k$, N_k é o número de elementos no agrupamento Q_k,

$$c_k=\frac{1}{N_k}\sum_{x_i\in Q_k}x_i \tag{3.18}$$

e $d_{ce}(Q_k, Q_l)$ é dado por

$$d_{ce}(Q_k,Q_l)=\left\{\sum_{k=1}^{d}\left|c_{ki}-c_{kj}\right|^p\right\}^{1/p} \tag{3.19}$$

3.4.2 Índice de Partição (SC)

É a razão da soma entre quão compactos e separados são os *clusters*. O objetivo é minimizar este índice:

$$SC(c)=\sum_{i=1}^{c}\frac{\sum_{j=1}^{N}(\mu_{ij})^m\left\|x_j-v_i\right\|^2}{N_i\sum_{k=1}^{c}\|v_k-v_i\|^2} \tag{3.20}$$

3.4.3 Índice de Separação (S)

Inversamente ao Índice de Partição, o índice de separação valida a partição através de uma medida separação de distância mínima. O objetivo é minimizar este índice:

$$S(c)=\frac{\sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij})^m\left\|x_j-v_i\right\|^2}{Nmin_{i,k}\|v_k-v_i\|^2} \tag{3.21}$$

3.4.4 Índice de Xie e Beni (XB)

Quantifica a razão da variância total intra-cluster e a separação entre os *clusters*; o objetivo é minimizar este índice:

$$XB(c) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij})^{m}\left\|x_j - v_i\right\|^2}{N\min_{ij}\left\|x_j - v_i\right\|^2} \tag{3.22}$$

As implementações, para Matlab, dos índices SC, S, e XB são fornecidas pelo *Fuzzy Clustering Toolbox*[1], enquanto o índice DB é fornecido pelo *Cluster Validation Toolbox*[2].

3.5 Resultados Experimentais e Discussões

O esquema de codificação dos dois algoritmos (Genético e baseado em PSO) pode é bastante similar:

a) a solução é codificada como: cada padrão armazena (ou contém) o número do centróide ou do *cluster* relacionado. No KGA a codificação é realizada pelos cromossomos (e seus respectivos alelos) e no método baseado no PSO há a posição da partícula;
b) ambos consideram que cada membro da população (cromossomo ou partícula) carrega consigo uma solução em potencial para o problema;
c) toda a população de soluções (a população, em si, no KGA; e o enxame de partículas no método baseado no PSO) trabalha em conjunto para alcançar melhores resultados na tarefa de *clustering*.

O método baseado em PSO pode ser considerado como de custo mais alto, em termos computacionais, já que as estruturas de dados são direcionadas para o conceito de que cada partícula mantém sua memória das suas posições anteriores.

[1] O Fuzzy Clustering Toolbox é desenvolvido por Janos Aboni.
[2] O Cluster Validation Toolbox é desenvolvido por Kaijun Wang.

Os algoritmos de *clustering* foram implementados em *Matlab*, versão 2006a. O código-fonte completo da implementação está nos Apêndices, o que será apresentado a seguir corresponde a detalhes que se fazem necessários para esclarecer, de forma prática, a aplicação dos conceitos apresentados nas seções anteriores.

Os experimentos são realizados em um *hardware* com a seguinte configuração:

- processador Intel Core 2 Duo, 1.6 GHz, 2 MB cache L2
- 2 GB RAM DDR 667MHz

Os conjuntos de dados escolhidos para testes são:

- **Ruspini**: com 75 padrões, em duas colunas de dados (75 x 2), com quatro classes facilmente separáveis, inclusive a olho nu.
- **Wine**: com 178 padrões, em 13 colunas de dados (178 x 13), com três classes.
- **Iris**: com 150 padrões de entrada, em 4 colunas de dados (150 x 4), com três classes.

Para que pudessem ser obtidos os resultados da análise do erro de classificação dos algoritmos, cada base de dados possui uma coluna adicional que indica, para cada padrão, o rótulo (número) da classe associada.

3.5.1 Resultados dos algoritmos K-means e Fuzzy C-Means

Para efeitos de comparação dos resultados encontrados com os métodos estudados neste trabalho, são apresentados, a seguir, os resultados encontrados a partir da utilização dos algoritmos K-Means e Fuzzy C-Means. Para que os resultados pudessem representar uma amostragem mais adequada, foram realizados 50 testes, e a Tabela 1 apresenta, para cada base de dados (Iris, Ruspini, Wine), o valor mínimo, médio e máximo encontrado para cada índice de validação estudado (DB, SC, S, XB) e a taxa de erro.

Tabela 1: Resultados dos Algoritmos K-Means e Fuzzy C-Means (FCM): a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine

(a)

Índice	**K-Means**			**FCM**
	Min	Méd.	Max	-
DB	0,61212	0,69081	0,77991	0,62613
SC	0,46308	0,48372	0,51758	0,62798
S	0,00446	0,00465	0,00497	0,00638
XB	3,41458	4,47178	4,93836	3,97634
Erro (%)	11,33	19,60	42,67	10,67

(b)

Índice	**K-Means**			**FCM**
	Min	Méd.	Max	-
DB	0,29046	0,44717	0,87690	0,33632
SC	0,27625	0,35350	0,51807	0,36330
S	0,00407	0,00665	0,01296	0,00541
XB	2,88926	7,18313	8,57401	6,04515
Erro (%)	0,00	30,43	100,00	0,00

(c)

Índice	**K-Means**			**FCM**
	Min	Méd.	Max	-
DB	1,10551	1,26082	1,57878	1,30418
SC	0,95495	0,97880	1,25538	1,62948
S	0,00664	0,00682	0,00859	0,01197
XB	1,90714	1,96253	2,17519	0,97245
Erro (%)	2,81	6,19	47,75	5,06

Pelo que é possível observar algumas considerações podem ser apresentadas:

- K-Means, devido ao seu modo de funcionamento (principalmente a inicialização) é mais sujeito a fatores de aleatoriedade, enquanto FCM demonstra-se estável.
- Pelos valores mínimos, K-Means apresenta melhores resultados do que FCM, entretanto, pela média, o mesmo só acontece para os índices DB, SC e S da base de dados Wine e o índice S da base de dados Iris.

- Como a base de dados Ruspini é a mais simples das três utilizadas para análise de resultados, a taxa de erro encontrada pelo FCM é nula; o mesmo acontece para o mínimo de K-Means.
- Pela análise geral dos resultados, pode-se afirmar que FCM apresenta os melhores resultados para os índices estudados e quanto à taxa de erro.

Esta seção apresentou resultados numéricos dos índices de validação de classificação obtidos pelos algoritmos K-Means e FCM. As próximas seções apresentam resultados encontrados pelos algoritmos GKA, GFCMA, PSOKA e PSOFCM. Posteriormente, será apresentado um comparativo resumido de desempenho destes algoritmos.

3.5.2 *Clustering* baseado em Algoritmos Genéticos

Esta seção apresenta detalhes da implementação do método de *clustering* baseado em Algoritmos Genéticos e em dois algoritmos clássicos vistos anteriormente: K-means e Fuzzy C-Means.

O Algoritmo K-means Genético (GKA) é implementado através da seguinte função:

```
function [gkaresult] = gka(params)
% Genetic K-Means (Clustering) Algorithm
```

Ou seja:

- *Parâmetros de Entrada:*
  a) **np** : o tamanho da população (quantidade de indivíduos ou cromossomos);
  b) **data** : o conjunto de dados; uma matriz no formato (linhas x colunas);
  c) **k** : o número de centróides (ou o número de *clusters*);
  d) **gmax** : o número máximo de gerações (utilizado como critério de parada do algoritmo);

e) **mp** : probabilidade de mutação utilizada no operador de mutação do algoritmo genético;

f) **vi** : o índice de validação a ser utilizado;

- *Saída:*

a) **clustering_time** : o tempo de execução total (em segundos);

b) **s** : o melhor indivíduo (ou cromossomo); um vetor com quantidade de elementos correspondente à quantidade de linhas do conjunto de dados;

c) **c** : os centróides correspondentes ao melhor indivíduo;

d) **fitness** : o valor de desempenho do melhor indivíduo;

e) **fitness_history** : o histórico de *fitness* durante as gerações.

A função objetivo é definida em tempo de chamada da função. Como visto anteriormente, a função objetivo pode ser independente do algoritmo de *clustering*, entretanto, mas a eficácia do método de avaliação da solução tem relação direta com o resultado geral do algoritmo genético. A implementação realizada considera que a função objetivo é baseada diretamente no índice de validação utilizado, indicado através de um dos parâmetros da função. Assim sendo, o índice de validação será utilizado para guiar o algoritmo de *clustering* a encontrar a melhor solução. Ainda assim, o algoritmo calcula, independentemente do índice de validação escolhido para definir a função objetivo, o valor de cada índice de validação em estudo neste trabalho (DB, SC, S, e XB) e a taxa de erro de classificação da solução de cada indivíduo da população (cromossomo).

O Algoritmo Fuzzy C-Means Genético (GFCMA) é implementado através da seguinte função:

```
function [gfcmaresult] = gfcma(params)
% Genetic Fuzzy C-Means (Clustering) Algorithm
```

Os parâmetros de entrada e saída assemelham-se aos da função **gka()**, diferenciando-se apenas no seguinte:

- Parâmetro de entrada **m**: o *fuzzyficador*, com valor padrão = 2;

- Saída:
  a) **s** : diferencia-se do valor de saída s da função **gka()**, devido ao fato de que o algoritmo Fuzzy C-means utiliza uma matriz com valores de pertinência de cada padrão em relação a cada aglomerado (classe). Desta forma, **s** é uma matriz padrões x centróides;
  b) **scrisp**: contém uma representação rígida do valor **s**, ou seja, os valores *defuzzyficados*. O processo de *defuzzyficação*, como indicam Pham e Prince (1999), consiste em encontrar, para cada padrão, o maior valor de pertinência em relação a cada centróide.

3.5.3 Resultados dos algoritmos GKA e GFCMA

A seção 4.5.1 apresentou resultados dos algoritmos K-means e FCM, esta seção, por sua vez, apresenta os resultados dos algoritmos GKA e GFCMA. A metodologia de testes considera:

- Bases de dados: Iris, Ruspini e Wine;
- Índices de validação: DB, SC, S e XB.
- O que será avaliado dos resultados obtidos (avaliação dos resultados)
  - O resultado dos algoritmos em relação aos valores mínimo, médio e máximo obtidos pela função objetivo (advindos dos índices de validação) e a taxa de erro geral;
  - A taxa de erro mínima e máxima e quais conjuntos de parâmetros; e
  - Os valores mínimo e máximo para cada índice de validação e quais os conjuntos de parâmetros.

Enquanto os algoritmos testados anteriormente não permitem muitas variações de parâmetros, os algoritmos baseados em Algoritmos Genéticos permitem diversos arranjos de parâmetros. Os experimentos procuram avaliar:

a) Qual a contribuição de cada algoritmo (K-means e Fuzzy C-means) para o processo de *clustering* com Algoritmos Genéticos;

b) Qual a importância dos parâmetros para o algoritmo genético: número de indivíduos na população (np), quantidade máxima de iterações (gmax), taxa ou probabilidade de mutação (mp);
c) Qual a importância do parâmetro de índice de validação e sua contribuição no resultado final.

Os parâmetros e seus valores são:

- NP: 5; 20; 50;
- MP: 0,025; 0,5; 0,85;
- Índice de Validação: DB; SC; S; XB;
- GMAX: 5; 10; 100.

A Tabela 2 apresenta os resultados obtidos considerando os algoritmos GKA e GFCMA, as bases de dados Iris, Ruspini e Wine e os valores mínimo, médio e máximo obtidos para cada índice de validação (DB, SC, S, XB) e a taxa de erro.

Tabela 2: Resultados dos Algoritmos GKA e GFCMA: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine

(a)

Índice	**GKA**			**GFCMA**		
	Min.	**Méd.**	**Max.**	**Min.**	**Méd.**	**Max.**
DB	0,58931	0,61651	0,66188	0,45831	0,62613	0,64908
SC	0,43933	0,45049	0,45839	0,62725	0,62947	0,63466
S	0,00389	0,00416	0,00458	0,00630	0,00638	0,00644
XB	2,50475	2,58308	2,68649	1,35521	1,63055	1,85673
Erro (%)	10,67	32,39	68,00	10,00	15,42	55,33

(b)

Índice	**GKA**			**GFCMA**		
	Min.	**Méd.**	**Max.**	**Min.**	**Méd.**	**Max.**
DB	0,29046	0,29046	0,29046	0,29046	0,32035	0,32237
SC	0,27625	0,27625	0,27625	0,36330	0,36332	0,36339
S	0,00407	0,00407	0,00407	0,00540	0,00541	0,00542
XB	2,81341	2,90996	3,04789	0,77498	1,28165	1,95288
Erro (%)	0,00	11,56	100,00	0,00	7,19	76,00

(c)

Índice	GKA			GFCMA		
	Min.	**Méd.**	**Max.**	**Min.**	**Méd.**	**Max.**
DB	1,10055	1,10605	1,29697	0,84352	1,11319	1,30337
SC	0,96569	0,96961	0,97382	1,62937	2,41962	5,31974
S	0,00670	0,00674	0,00680	0,01197	0,01913	0,04760
XB	1,52923	1,58911	1,63309	0,60118	0,60694	5,31974
Erro (%)	3,37	9,11	23,03	5,06	17,10	53,37

Analisando os resultados, pode-se afirmar que, em relação à base de dados Iris, o algoritmo GKA é melhor que o GFCMA nos índices SC e S, enquanto que, considerando a taxa de erro, o algoritmo GFMCA apresentou valores menores. Em relação à base de dados Ruspini, o algoritmo GKA é melhor nos índices SC, SC e S, enquanto o algoritmo GFCMA é melhor no índice XB e em relação à taxa de erro. Em relação à base de dados Wine, o algoritmo GKA é melhor em todos os índices – embora GFCMA tenha menor valor mínimo – e, também, em relação à taxa de erro. Portanto, pode-se afirmar que, em relação aos valores mínimo, máximo e médio e taxa de erro, o algoritmo GKA apresentou melhores resultados.

Em relação à função objetivo, ainda que o menor valor possa indicar uma possível solução ótima, é interessante notar também que a taxa de erro de classificação também deve ser considerada para, após os resultados, eleger a melhor solução. É importante notar que a taxa de erro não interfere no funcionamento do algoritmo, sendo apenas uma informação gerada após encontrar a melhor solução. Desta forma, considerando a taxa de erro mínima e máxima, a Tabela 3 apresenta os resultados obtidos e os parâmetros utilizados pelos algoritmos para encontrá-los, inclusive, indicando o índice que direcionou o algoritmo para a menor e maior taxa de erro.

Tabela 3: Resultados dos Algoritmos GKA e GFCMA considerando taxa de erro mínima e máxima: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine

(a)

	GKA					FCM				
	Erro	**NP**	**MP**	**Gmax**	**Índice**	**Erro**	**NP**	**MP**	**Gmax**	**Índice**
Min	10,67	5	0,500	5	S	10	5	0,025	5	XB
Max	68	50	0,025	5	XB	55,33	50	0,850	5	XB

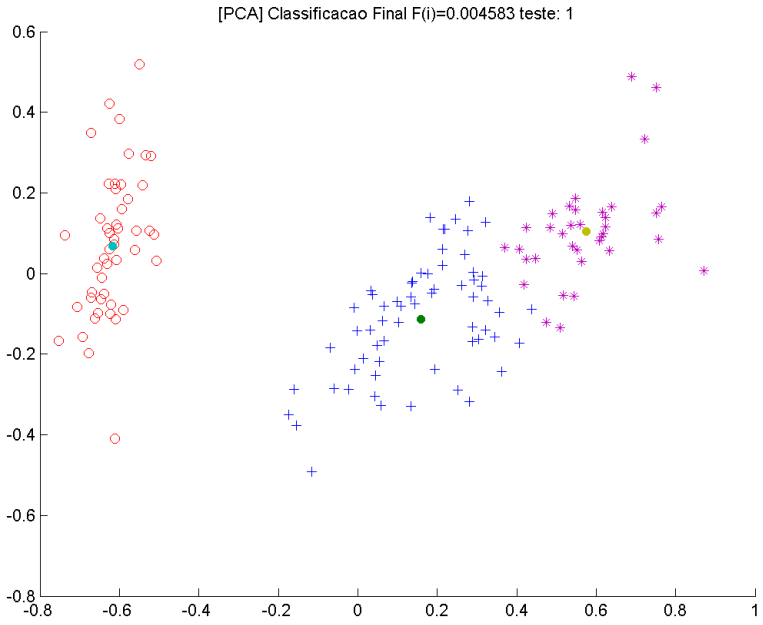
(b)

	GKA					FCM				
	Erro	**NP**	**MP**	**Gmax**	**Índice**	**Erro**	**NP**	**MP**	**Gmax**	**Índice**
Min	0	5	0,025	5	S	0	5	0,025	5	S
Max	100	50	0,500	100	XB	76	50	0,025	10	XB

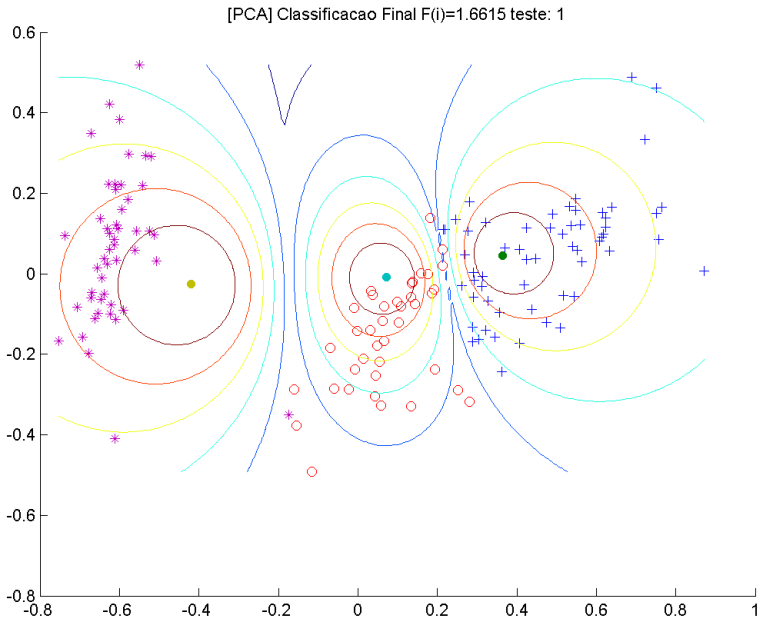
(c)

	GKA					FCM				
	Erro	**NP**	**MP**	**Gmax**	**Índice**	**Erro**	**NP**	**MP**	**Gmax**	**Índice**
Min	3,37	50	0,025	10	SC	5,06	20	0,85	10	S
Max	23,03	20	0,85	100	XB	53,37	5	0,025	5	XB

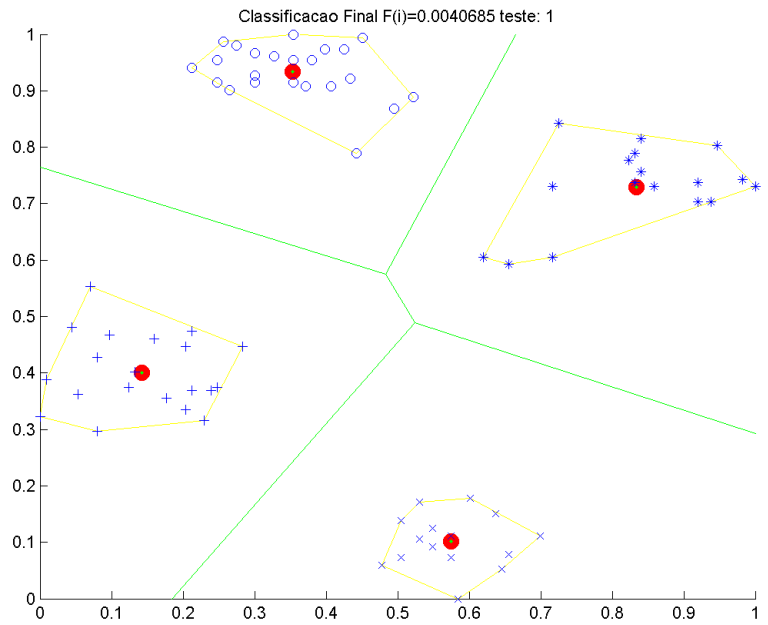
Através de análise dos resultados, pode-se afirmar que em relação à base de dados Wine, GKA apresentou melhores resultados, enquanto que GFCMA apresentou melhores resultados nas bases de dados Iris e Ruspini. Os parâmetros dos algoritmos indicam também que, no geral, uma menor taxa de erro pode ser obtida com uma baixa quantidade de indivíduos na população (NP = 5), baixa taxa de mutação (MP = 0,025) e a quantidade máxima de iterações também pode ser baixa (GMAX = 5). Interessante notar que, em relação aos índices de validação, o índice XB apresentou maior taxa de erro nos dois algoritmos. A Figura 10 apresenta uma imagem que ilustra o melhor resultado para cada base de dados.



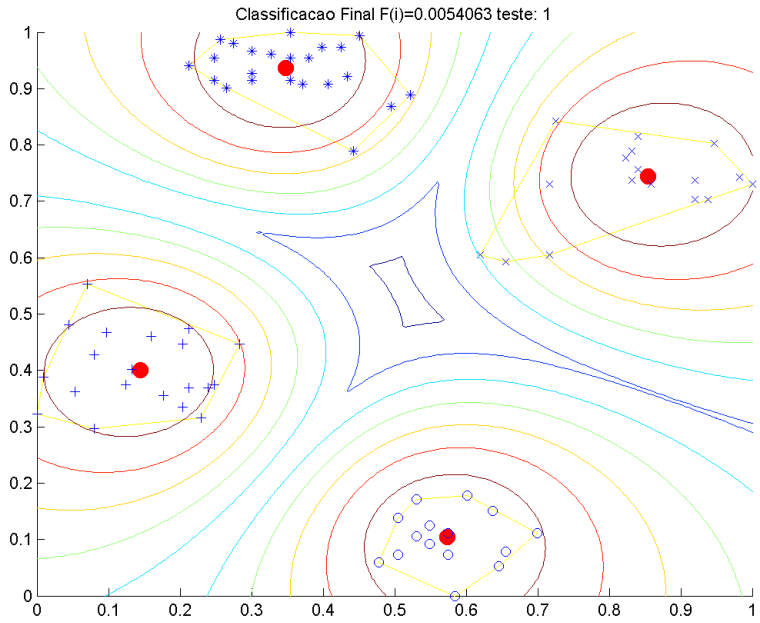
(a)


[PCA] Classificacao Final F(i)=1.6615 teste: 1
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

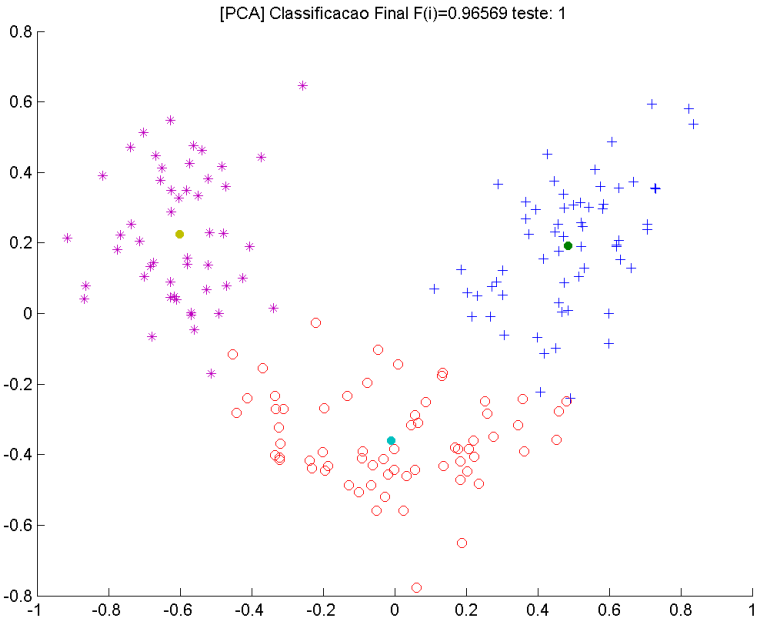
(b)


Classificacao Final F(i)=0.0040685 teste: 1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

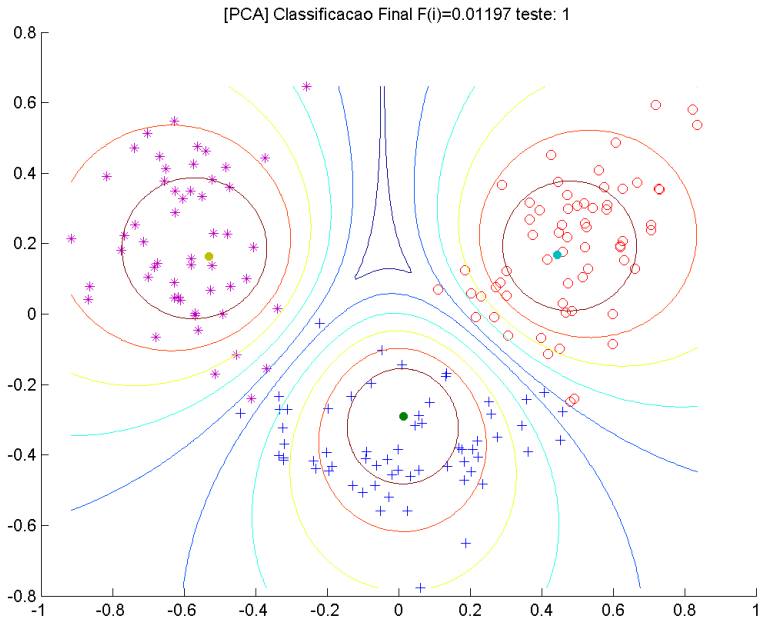
(c)


Classificacao Final F(i)=0.0054063 teste: 1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(d)


[PCA] Classificacao Final F(i)=0.96569 teste: 1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

(e)



(f)

Figura 10: Resultado de Clustering com GKA e GFCMA: (a) e (b) base de dados Iris: a) GKA; b) GFCMA; (c) e (d) base de dados Ruspini: c) GKA; d) GFCMA; (e) e (f) base de dados Wine: e) GKA e f) GFCMA

A Figura 11 ilustra como o algoritmo GFCMA contorna o problema de mínimos locais.

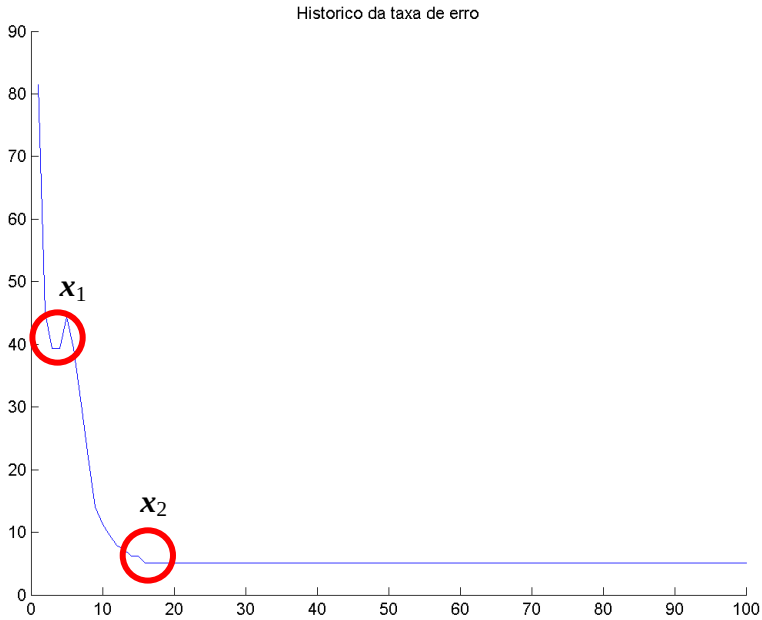


Figura 11: Exemplo de solução do problema de mínimos locais (GFCMA)

A Tabela 4 apresenta os resultados obtidos considerando-se o valor mínimo e máximo para cada índice de validação (sendo utilizado na função objetivo) e os parâmetros utilizados pelos algoritmos para encontrá-los.

Tabela 4: Resultados dos Algoritmos GKA e GFCMA considerando valores mínimo e máximo dos índices de validação: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine

(a)

	GKA					**GFCMA**				
	Valor	**Erro**	**NP**	**MP**	**Gmax**	**Valor**	**Erro**	**NP**	**MP**	**Gmax**
DB-min	0,58931	12,00	5	0,025	5	0,45831	32,67	5	0,500	10
DB-max	0,66188	56,00	50	0,850	100	0,64908	18,67	50	0,500	5
SC-min	0,43933	49,33	5	0,025	5	0,62725	11,33	5	0,025	100
SC-max	0,45839	10,67	50	0,500	100	0,63466	11,33	50	0,850	5
S-min	0,00389	49,33	5	0,025	5	0,0063	11,33	20	0,025	5
S-max	0,00458	10,67	50	0,500	100	0,00644	11,33	50	0,500	5
XB-min	2,50475	68,00	5	0,025	5	1,35521	37,33	5	0,025	10
XB-max	2,68649	22,67	50	0,850	100	1,85673	24,67	50	0,850	100

(b)

	GKA					**GFCMA**				
	Valor	**Erro**	**NP**	**MP**	**Gmax**	**Valor**	**Erro**	**NP**	**MP**	**Gmax**
DB-min	0,00407	0,00	20	0,025	5	0,00540	0,00	50	0,500	5
DB-max	0,00407	0,00	5	0,025	100	0,00542	0,00	20	0,250	10
SC-min	0,27625	0,00	20	0,025	5	0,29046	0,00	20	0,850	10
SC-max	0,27625	0,00	50	0,025	100	0,32237	0,00	50	0,025	5
S-min	0,29046	0,00	50	0,025	5	0,36330	0,00	20	0,500	100
S-max	0,29046	0,00	50	0,025	100	0,36339	0,00	5	0,850	5
XB-min	2,81341	100	20	0,500	5	0,77498	8,00	5	0,850	5
XB-max	3,04789	2,67	20	0,025	100	1,95288	1,33	50	0,500	10

(c)

	GKA					**GFCMA**				
	Valor	**Erro**	**NP**	**MP**	**Gmax**	**Valor**	**Erro**	**NP**	**MP**	**Gmax**
DB-min	0,00670	6,18	5	0,025	10	0,01197	43,82	5	0,850	5
DB-max	1,10605	5,06	5	0,500	100	1,62937	6,18	5	0,850	10
SC-min	0,00674	3,37	50	0,025	10	0,01827	5,06	50	0,025	100
SC-max	1,29697	5,06	5	0,500	5	2,62530	16,29	20	0,500	5
S-min	0,00680	4,49	50	0,850	10	0,03417	5,06	50	0,025	100
S-max	1,29697	5,62	50	0,850	5	5,31974	16,29	5	0,500	5
XB-min	1,52923	23,03	20	0,850	5	0,60118	32,02	20	0,500	5
XB-max	1,63309	22,47	20	0,025	100	5,31974	12,92	20	0,025	10

A Tabela 4 apresenta um cruzamento de resultados, ou seja, apresenta os resultados dos índices de validação, mínimo e máximo, as configurações dos parâmetros dos algoritmos e a taxa de erro encontrada. Um fato curioso é que na grande maioria das situações, o valor mínimo do índice de validação não leva ao menor erro de classificação, mas a um erro maior do que o erro obtido pelo máximo do índice de validação. Pela análise dos resultados, não foi possível chegar a uma conclusão sobre essa situação. Em relação ao algoritmo GKA, a maior parte dos melhores resultados pelo índice de validação são configurações de parâmetros que indicam uma quantidade de indivíduos média (NP = [5, 20]), taxa de mutação baixa (MP = 0,025) e uma baixa quantidade de gerações (GMAX = 5). Quanto ao algoritmo GFCMA, as melhores configurações indicam quantidade média de indivíduos (NP = [5, 20]), taxa de mutação média (MP = 0,5) e quantidade de gerações média-alta (GMAX=[10, 100]). Estes resultados indicam que, no geral, o algoritmo GFCMA precisa de mais gerações para convergir para um ótimo.

Avaliando os resultados obtidos em relação aos índices de validação e à taxa de erro, pela média, é possível afirmar que os algoritmos GKA e GFCMA são superiores que K-means e FCM. Entretanto, considerando a diferença, é também possível afirmar que a melhora obtida é pouco considerável:

- no caso da base de dados Iris, GKA é superior ao K-means em todos os índices de validação, mas, quanto à taxa de erro, apenas a mínima é melhor: 10,67 contra 11,33; na média, K-means obteve taxa de erro menor. GFCMA e FCM têm os mesmos resultados para os índices DB, SC e S, GFCMA é melhor no índice XB e no mínimo da taxa de erro, entretanto, na média da taxa de erro, FCM possui melhor resultado: 10,67 contra 15,42;
- no caso da base de dados Ruspini, a melhora do GKA é mais perceptivel em relação ao K-Means, tanto nos índices de validação quanto na taxa de erro. GFCMA também é superior ao FCM;
- no caso da base de dados Wine, a superioridade do GKA em relação ao K-means é mais evidente apenas taxa de erro, já que os valores obtidos nos índices de validação são bastante próximos, principalmente considerando os mínimos.

GFCMA é superior ao FCM nos índices DB e S, FCM é melhor nos índices SC e XB e também na taxa de erro.

3.5.4 *Clustering* baseado em Otimização por Enxames de Partículas

Esta seção apresenta detalhes da implementação do método de *clustering* baseado em Otimização por Enxames de partículas e em dois algoritmos clássicos vistos anteriormente: K-means e Fuzzy C-Means.

Tanto o Algoritmo PSO K-means (PSOKA) quanto o Algoritmo PSO Fuzzy C-Means (PSOFCM) são implementos através da seguinte função (simplificada, apenas para efeito ilustrativo):

```
function [psoresult] = pso(params)
% PSOKA e PSOFCM
```

Ou seja:

- *Parâmetros de Entrada:*
  a) **algorithm** : o algoritmo de *clustering* ('fcm' ou 'kma');
  b) **p** : o tamanho da população (quantidade de partículas);
  c) **data** : o conjunto de dados; uma matriz no formato [*linhas*, *colunas*];
  d) **k** : o número de centróides (ou o número de *clusters*);
  e) **tmax** : o número máximo de iterações;
  f) **thist** : o número de iterações a serem consideradas no critério de parada do algoritmo: uma verificação de modificações em thist épocas de iteração;
  g) **vmax** : valor máximo da velocidade de atualização da partícula (padrão = 2);
  h) **w** : parâmetro regulador da equação de atualização de velocidade (padrão = 0,1);
  i) **c1** e **c2** : constantes usadas na equação de atualização de velocidade (padrão c1 = c2 = 1,49); e
  j) **índice** : o índice de validação utilizado para definir a função objetivo.

- *Saída:*
  a) **clustering_**time : o tempo de execução total (em segundos);
  b) **Yp** : as melhores posições das partículas;
  c) **X** : as posições finais das partículas (ao término do algoritmo);
  d) **Yg** : a melhor posição dentre as partículas;
  e) **Xgi** : o índice da melhor partícula;
  f) **Fg** : o melhor valor da função objetivo;
  g) **Fg_hist** : os valores da função objetivo em cada iteração;
  h) **Z** : uma matriz contendo associações de padrões e centróides (conforme as partículas); para definir o valor de Z, no caso do algoritmo ser FCM, é utilizado o método de *defuzzificação* utilizado no algoritmo GFCMA.

3.5.5 Resultados dos algoritmos PSOKA e PSOFCM

Esta seção apresenta os resultados dos algoritmos PSOKA e PSOFCM. A metodologia de testes considera as mesmas definições dos testes com os algoritmos GKA e GFCMA. Os algoritmos baseados em PSO permitem diversos arranjos de parâmetros. Os experimentos procuram avaliar:

a) Qual a contribuição de cada algoritmo (K-means e Fuzzy C-means) para o processo de *clustering* com PSO;
b) Qual a importância dos parâmetros para o PSO: número de partículas (p), velocidade máxima (vmax), tempo máximo considerado no critério de parada (thist);
c) Qual a importância do parâmetro de índice de validação e sua contribuição no resultado final.

Os parâmetros e seus valores são:

- P: 5; 10; 30;
- VMAX: 0,2; 0,9;
- THIST: 10; 50;
- Índice de Validação: DB; SC; S; XB;

A Tabela 2 apresenta os resultados obtidos considerando os algoritmos PSOKA e PSOFCM, as bases de dados Iris, Ruspini e Wine e os valores mínimo, médio e máximo obtidos para cada índice de validação (DB, SC, S, XB) e a taxa de erro.

Tabela 5: Resultados dos Algoritmos PSOKA e PSOFCM: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine

(a)

Índice	**PSOKA**			**PSOFCM**		
	Min.	**Méd.**	**Max.**	**Min.**	**Méd.**	**Max.**
DB	0,27045	0,36780	0,52796	0,57841	0,62335	0,62613
SC	0,39613	0,47915	0,54635	0,62484	0,62749	0,62796
S	0,00365	0,00421	0,00516	0,00637	0,00637	0,00638
XB	1,19575	1,52560	2,06278	1,21399	1,37726	1,68562
Erro (%)	6,00	65,97	100,00	10,67	15,75	35,33

(b)

Índice	**PSOKA**			**PSOFCM**		
	Min.	**Méd.**	**Max.**	**Min.**	**Méd.**	**Max.**
DB	0,00471	0,00489	0,29046	0,00538	0,00540	0,00541
SC	0,29046	0,30640	0,33533	0,29046	0,30042	0,32035
S	0,00832	0,71131	1,68594	0,36314	0,36328	0,36330
XB	0,18273	1,01053	1,75728	0,80723	1,44156	3,27777
Erro (%)	0,00	22,25	100,00	0,00	9,81	92,00

(c)

Índice	**PSOKA**			**PSOFCM**		
	Min.	**Méd.**	**Max.**	**Min.**	**Méd.**	**Max.**
DB	0,23707	0,69798	1,02811	0,86679	1,10988	1,30337
SC	0,79707	0,95203	1,22435	1,62937	1,62937	1,62937
S	0,00479	0,00672	0,00870	0,01197	0,01197	0,01197
XB	1,19477	1,36998	1,54367	0,59022	0,60331	0,61588
Erro (%)	5,06	31,25	72,47	5,06	13,80	49,44

Analisando os resultados, pode-se afirmar que, em relação à base de dados Iris, o algoritmo PSOKA é melhor que o PSOFCM nos índices DB, SC e S. Em relação à taxa de erro, considerando-se a média, o algoritmo PSOFCM apresentou melhores resultados. Em relação à base de dados Ruspini, o algoritmo PSOKA é melhor nos índices DB e XB, enquanto o algoritmo PSOFCM é melhor nos índices SC e S e em

relação à taxa de erro. Quanto à base de dados Wine, o algoritmo PSOKA é melhor nos índices DB, SC e S, enquanto PSOFCM é melhor no índice XB e na taxa de erro. Portanto, pode-se afirmar que, em relação aos valores mínimo, máximo e médio e taxa de erro, embora o algoritmo PSOKA tenha sido superior em relação aos índices de validação, em relação à taxa de erro quem apresentou melhores resultados foi o algoritmo PSOFCM.

Considerando a taxa de erro mínima e máxima, a Tabela 3 apresenta os resultados obtidos e os parâmetros utilizados pelos algoritmos para encontrá-los, inclusive, indicando o índice que direcionou o algoritmo para a menor e maior taxa de erro.

Tabela 6: Resultados dos Algoritmos GKA e GFCMA considerando taxa de erro mínima e máxima: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine

(a)

	PSOKA					PSOFCM				
	Erro	**P**	**VMAX**	**THIST**	**Índice**	**Erro**	**P**	**VMAX**	**THIST**	**Índice**
Min	6,00	5	0,2	10	SC	10,67	5	0,9	10	S
Max	100,00	10	0,9	50	SC	35,33	5	0,9	10	XB

(b)

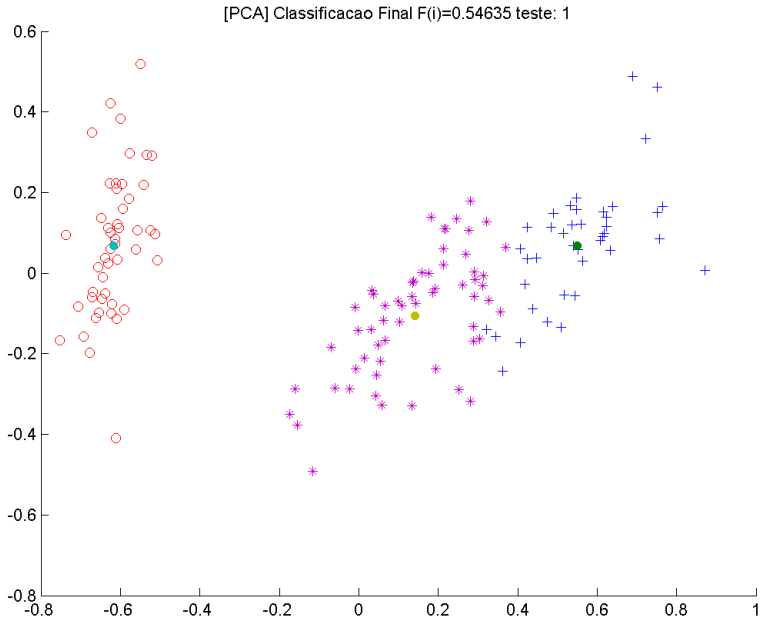
	PSOKA					PSOFCM				
	Erro	**P**	**VMAX**	**THIST**	**Índice**	**Erro**	**P**	**VMAX**	**THIST**	**Índice**
Min	0,00	10	0,2	10	S	0,00	30	0,9	10	S
Max	100,00	10	0,9	10	SC	92,00	30	0,2	50	XB

(c)

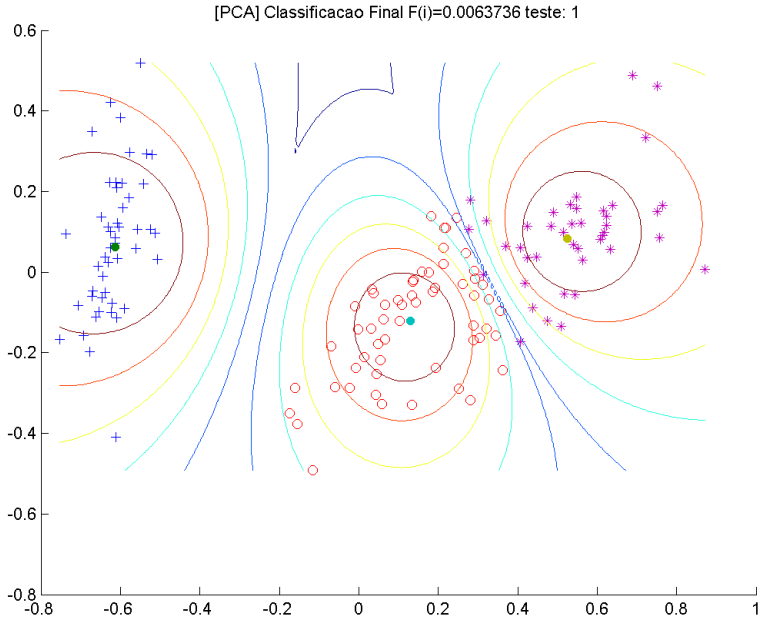
	PSOKA					PSOFCM				
	Erro	**P**	**VMAX**	**THIST**	**Índice**	**Erro**	**P**	**VMAX**	**THIST**	**Índice**
Min	5,06	5	0,2	10	S	5,06	5	0,2	10	S
Max	72,47	30	0,9	10	DB	49,44	5	0,2	50	XB

Através de análise dos resultados, pode-se afirmar que PSOFCM foi superior ao PSOKA em todas as bases de dados e, no caso da base de dados Iris, considera-se que seja melhor também pois, embora a taxa de erro mínima seja maior, a taxa de erro máxima é 1/3 da obtida pelo algoritmo PSOKA. Os parâmetros dos algoritmos indicam também que, no geral, uma menor taxa de erro pode ser obtida com uma baixa quantidade de indivíduos na população (P = 5), baixo valor para o parâmetro

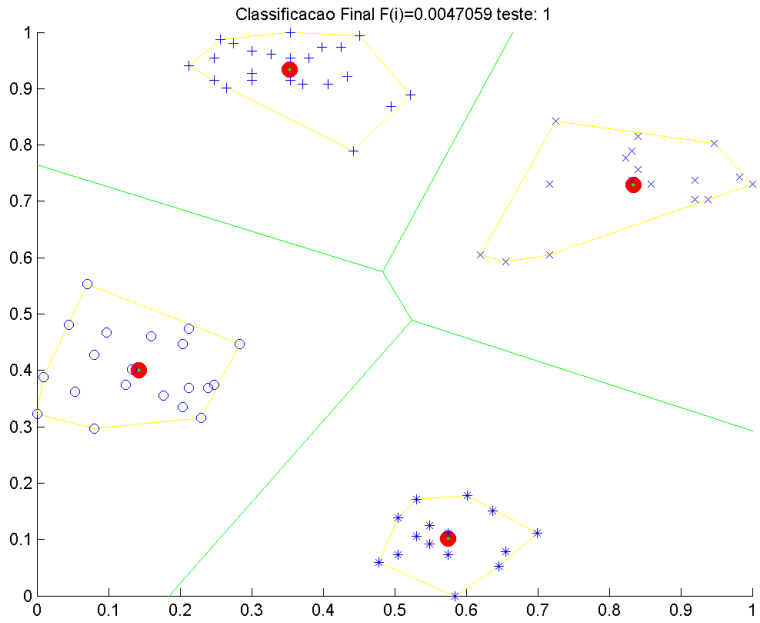
controlador da velocidade de atualização das partículas (VMAX = 0,2) e o valor do parâmetro do critério de parada também é baixo (THIST = 10). Interessante notar que, em relação aos índices de validação, o índice XB apresentou maior taxa de erro nos dois algoritmos, enquanto o índice S obteve melhores resultados. A Figura 12 ilustra melhor estes resultados.



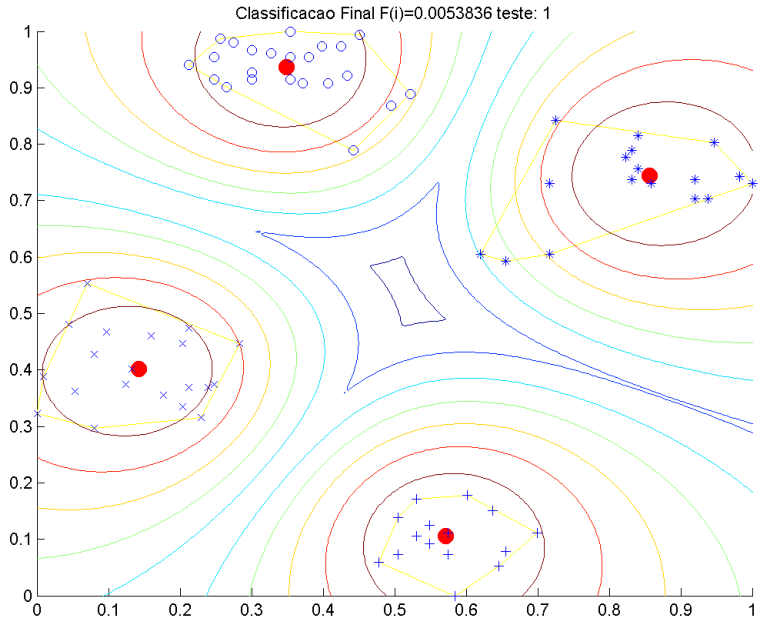
(a)



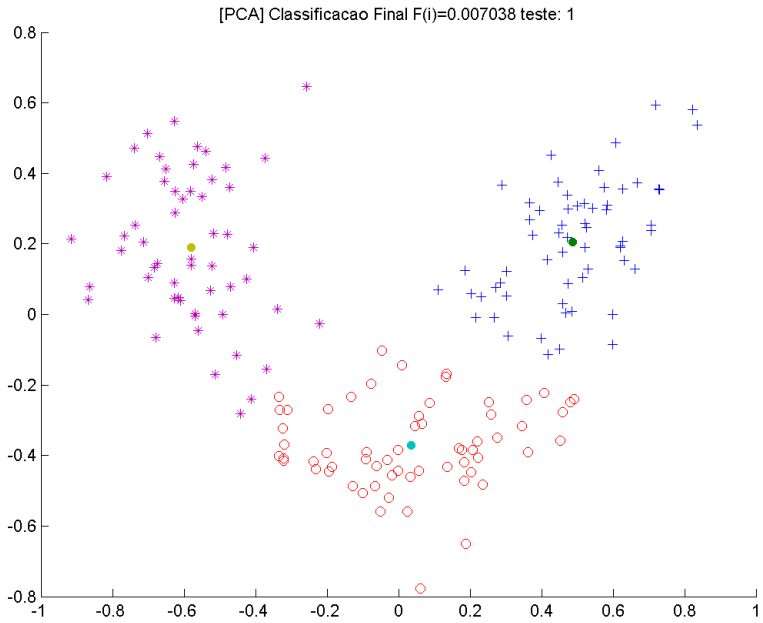
(b)


Classificacao Final F(i)=0.0047059 teste: 1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

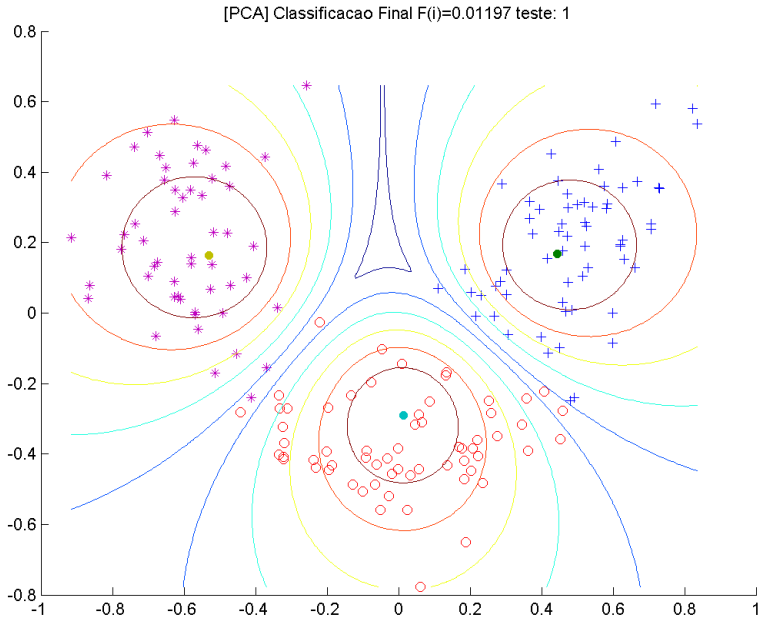
(c)


Classificacao Final F(i)=0.0053836 teste: 1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(d)



(e)



(f)

Figura 12: Resultado de Clustering com PSOKA e PSOFCM: (a) e (b) base de dados Iris: a) PSOKA; b) PSOFCM; (c) e (d) base de dados Ruspini: c) PSOKA; d) PSOFCM; (e) e (f) base de dados Wine: e) PSOKA e f) PSOFCM

Em relação à base de dados Iris, a Figura 13 ilustra como o algoritmo baseado em PSO, nesse caso o PSOKA, contorna o problema de ótimos locais.

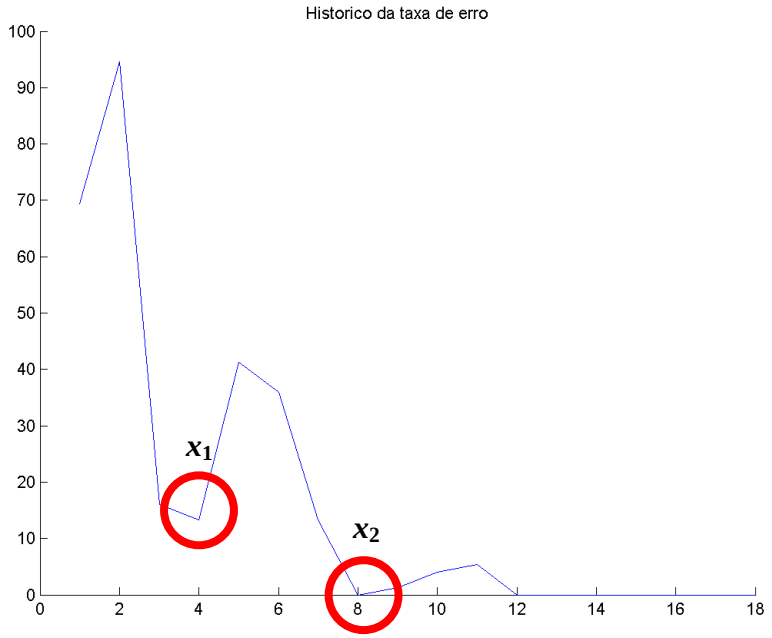


Figura 13: Exemplo de solução do problema de ótimos locais (base de dados Iris, PSOKA)

A Tabela 7 apresenta os resultados obtidos considerando-se o valor mínimo e máximo para cada índice de validação (sendo utilizado na função objetivo) e os parâmetros utilizados pelos algoritmos para encontrá-los.

Tabela 7: Resultados dos Algoritmos PSOKA e PSOFCM considerando valores mínimo e máximo dos índices de validação: a) base de dados Iris; b) base de dados Ruspini; c) base de dados Wine

(a)

	PSOKA					PSOFCM				
	Valor	**Erro**	**P**	**VMAX**	**THIST**	**Valor**	**Erro**	**P**	**VMAX**	**THIST**
DB-min	0,48089	66,00	30	0,9	50	0,62613	29,33	30	0,9	50
DB-max	1,92507	30,67	5	0,2	50	0,62796	10,67	10	0,2	50
SC-min	0,00516	100,00	30	0,2	50	0,00637	11,33	30	0,2	50
SC-max	2,06278	6,00	5	0,9	10	0,62796	10,67	10	0,9	50
S-min	0,36734	100,00	10	0,2	50	0,00637	10,67	5	0,9	10
S-max	0,47038	40,00	5	0,9	50	1,52661	10,67	10	0,9	50

XB-min	0,00365	30,67	5	0,9	10	0,57841	25,33	30	0,9	50
XB-max	0,54315	38,00	30	0,2	10	1,68562	30,67	5	0,2	10

(b)

	PSOKA					**PSOFCM**				
	Valor	**Erro**	**P**	**VMAX**	**THIST**	**Valor**	**Erro**	**P**	**VMAX**	**THIST**
DB-min	0,18273	74,67	30	0,9	50	0,29046	0,00	10	0,2	10
DB-max	0,30338	73,33	10	0,9	10	0,32035	0,00	5	0,2	50
SC-min	0,30073	0,00	10	0,9	50	0,36314	0,00	5	0,2	50
SC-max	0,50594	100,00	10	0,9	10	0,36330	0,00	30	0,2	50
S-min	0,00471	0,00	10	0,2	10	0,00538	0,00	30	0,9	10
S-max	0,00832	20,00	10	0,9	50	0,00541	0,00	10	0,2	50
XB-min	1,15200	42,67	10	0,9	50	0,80723	76,00	30	0,2	10
XB-max	1,75728	93,33	10	0,2	10	3,27777	6,66	5	0,2	10

(c)

	PSOKA					**PSOFCM**				
	Valor	**Erro**	**P**	**VMAX**	**THIST**	**Valor**	**Erro**	**P**	**VMAX**	**THIST**
DB-min	0,23707	72,47	10	0,9	50	0,86679	39,33	30	0,2	50
DB-max	1,02811	12,36	5	0,2	10	1,30337	6,18	5	0,9	50
SC-min	0,79707	51,12	30	0,9	50	1,62937	5,06	5	0,2	10
SC-max	1,22435	6,74	5	0,9	10	1,62937	5,06	30	0,2	50
S-min	0,00479	53,37	30	0,9	50	0,01197	5,06	5	0,2	10
S-max	0,00870	6,74	5	0,9	10	0,01197	5,06	30	0,2	50
XB-min	1,19477	71,35	10	0,9	50	0,59022	20,79	30	0,2	10
XB-max	1,54367	43,26	10	0,2	10	0,61588	37,64	5	0,2	10

Como feito anteriormente para métodos baseados em Algoritmos Genéticos, A Tabela 7 apresenta um cruzamento de resultados, ou seja, apresenta os resultados dos índices de validação, mínimo e máximo, as configurações dos parâmetros dos algoritmos e a taxa de erro encontrada. De modo geral, as mesmas considerações feitas anteriormente são mantidas, inclusive a consideração sobre a taxa de erro e o valor mínimo dos índices de validação. Tanto em relação ao algoritmo PSOKA quanto ao PSOFCM, os melhores resultados obtidos indicam configurações de parâmetros com quantidade de indivíduos média (P = 10), parâmetro regulador de velocidade com valor baixo (VMAX = 0,2) e o parâmetro para o critério de parada não representa influência nos resultados.

Avaliando os resultados obtidos em relação aos índices de validação e à taxa de erro, pela média, é possível afirmar que os algoritmos GKA e GFCMA obtiveram melhores resultados em todas as bases de dados. Entretanto, considerando a diferença, é também possível afirmar que a melhora obtida é pouco considerável:

- no caso da base de dados Iris, PSOKA é superior ao K-means em todos os índices, entretanto, quanto à taxa de erro, K-means é superior, já que a taxa de erro média e máxima são melhores que as encontradas pelo PSOKA. PSOFCM é superior ao FCM em todos os índices, e, quanto à taxa de erro, encontrou o mesmo valor para o mínimo, mas, na média, FCM é um pouco superior;
- no caso da base de dados Ruspini, PSOKA é superior ao K-means nos índices DB, SC e XB, e também em relação à taxa de erro. PSOFCM também é superior ao FCM, embora a taxa de erro média um pouco alta em relação à mínima;
- no caso da base de dados Wine, PSOKA é superior ao K-means em todos os índices, mas quanto à taxa de erro, K-means é um pouco superior. PSOFCM é superior ao FCM nos índices de validação, mas possui taxa de erro média um pouco superior à obtida pelo FCM (note-se que a taxa de erro mínima é a mesma).

Considerando os resultados obtidos nos testes realizados, pode-se chegar a algumas conclusões que irão ajudar a nortear as configurações dos testes de segmentação de imagens médicas, no próximo Capítulo:

a) o índice XB apresentou resultados inferiores, considerando-se tanto o valor do índice de validação quanto a taxa de erro;
b) quantidade pequena a média de soluções paralelas é suficiente: para GKA e GFCMA, 5 <= NP <= 20; para PSOKA e PSOFCM, P = 10;
c) para GKA e GFCMA:
   - baixa taxa de mutação (MP = 0,025);
   - baixa quantidade de gerações (GMAX = 10)

d) para PSOKA e PSOFCM:
   - parâmetro regulador de velocidade com valor baixo (VMAX = 0,2)
   - o parâmetro para o critério de parada não representa influência nos resultados.

Este capítulo tratou de apresentar os conceitos de detecção de aglomerados com técnicas de computação natural e uma análise bastante detalhada dos resultados guiará os testes realizados no próximo capítulo, onde será tratada a tarefa de segmentação de imagens médicas.

Capítulo 4

Segmentação de Imagens Médicas Utilizando Métodos de Computação Natural

Os capítulos anteriores trataram de apresentar informações que ajudaram o leitor a se fundamentar nos conceitos de Computação Natural e nas técnicas de Detecção de Agrupamentos baseadas neste ramo da Computação Científica. O capítulo em questão representa o foco principal do trabalho, a saber, a aplicação das técnicas de Computação Natural na tarefa de Segmentação de Imagens Médicas. Inicialmente serão apresentados conceitos e fundamentos sobre imageamento em medicina e, posteriormente, os resultados obtidos com a utilização dos algoritmos GKA, GFCMA, PSOKA e PSOFCM.

4.1 Imageamento em Medicina

Segundo Oliveira Filho (2006) o objetivo do imageamento médico é a avaliação médica de tecidos e funções orgânicas do corpo humano por meio de imagens; trabalhando com informações quantitativas das imagens, no domínio tridimensional, com vistas a obter a máxima informação daquelas imagens e diagnosticar doenças o mais cedo possível, com maior precisão.

Os exames e as formas de aquisição das imagens podem ser assim discriminados:

4.1.1 Tomografia computadorizada

Na Tomografia Computadorizada (*Computer Tomography* – CT) os fótons são coletados por um cristal cintilador ou uma fotomultiplicadora, que convertem a energia

incidente em corrente elétrica, proporcional à energia dos fótons de raio-X incidentes. Conforme Oliveira Filho (2006) os seguintes conceitos resumem definições deste tipo de exame:

a) uma fatia de corte, com 1 mm de expessura, é imageada; tais imagens possuem 512 x 512 pixels e, tipicamente, a dimensão do voxel é de 0,5 x 0,5 x 1 mm;

b) o brilho, a escala de cinza de cada *pixel*, representa o coeficiente de atenuação linear média do tecido do *voxel* correspondente. Este número, chamado também de número CT, pode ter valores exemplificados como indica a Tabela 8:

Tabela 8: Exemplos de números CT com relação a tecidos

Tecido	**Número CT**
Ar	-1000
Pulmão	-900 a -400
Gordura	-110 a -65
Água	0
Rim	30
Sangue normal	35 a 55
Sangue coagulado	80
Músculo	40 a 60
Fígado	50 a 85
Ossos	130 a 250

A CT também permite o imageamento 3D em condições iguais ou superiores às de Ressonância Magnética; e, como indicam Pham et al (1998) é mais adequado para o imageamento de tecidos como ossos e tumores ósseos.

4.1.2 Ressonância Nuclear Magnética

Como afirma Oliveira Filho (2006) o imageamento por ressonância magnética nuclear tem se tornado o método não-evasivo mais poderoso para o diagnóstico e pesquisa médica; e, segundo Pham et al (1998) é o método mais indicado para exames do cérebro por causa de sua habilidade em fornecer :

a) imagens que combinam alta resolução (ox *voxels* possuem dimensão de 1 mm^3);
b) excelente contraste de tecidos; e
c) uma alta razão sinal-ruído.

Ainda, como indicam Pham et al (1998), é possível a aquisição de imagens de múltiplos canais, as quais fornecem informações adicionais que permitem uma melhor distinção das diferentes estruturas anatômicas.

Outros métodos para aquisição de imagens médicas são:

a) Tomografia por Emissão de Prótons;
b) Tomografia Computadorizada por Emissão de Fóton Único;
c) Ultrassom;
d) Multimodal, entre outros.

4.2 Escolha de Características de Imagens Médicas Para Segmentação de Imagens

Os experimentos realizados neste trabalho utilizando dados sintéticos fornecidos pelo sistema BrainWeb (2007), que gera um modelo anatômico do cérebro, simulando efeitos da realização de um exame MRI convencional. Há dois modos para geração de dados de imagens:

a) Normal (cérebro sem doenças)
b) Anormal (cérebro com doenças)

O simulador BrainWeb apresenta uma interface com o usuário que permite a definição de diversos parâmetros para a obtenção dos dados de imagens (Figura 14):

a) Modalidade: T1, T2 e PD

b) Largura da fatia de imagem (*slice*): 1, 2, 5, 7 e 9 mm

c) Ruído: 0, 2, 3, 5, 7 e 9 %

d) Intensidade: 0, 20 e 40 %

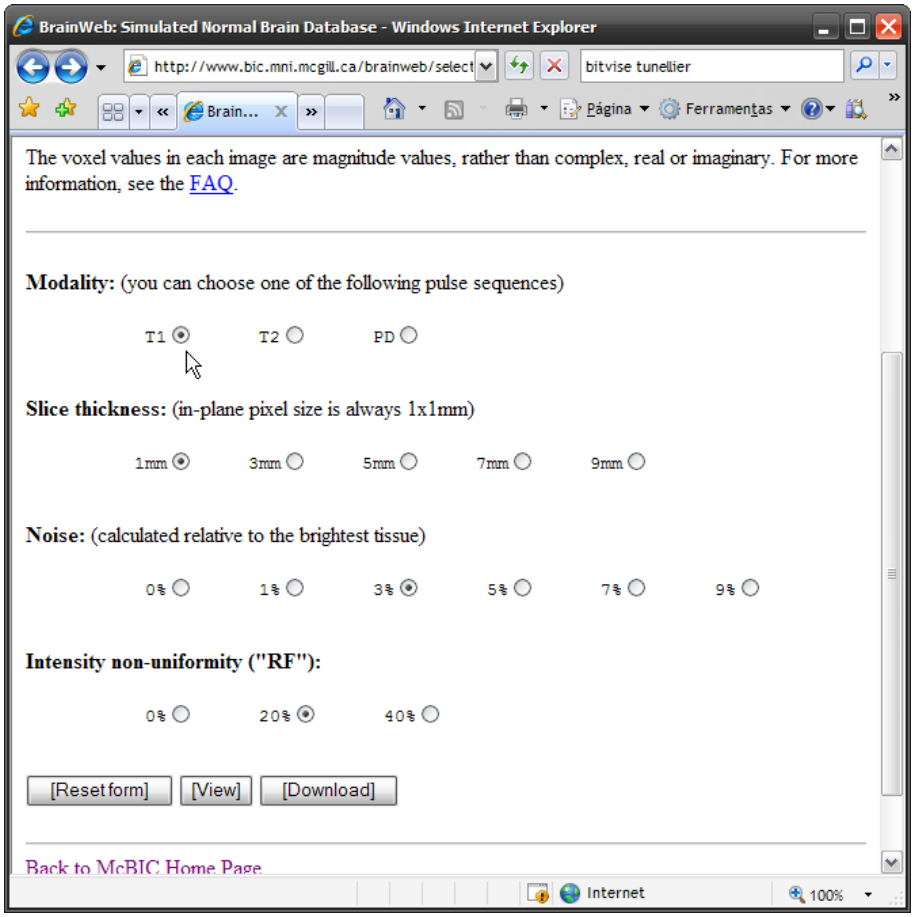


Figura 14: Interface interativa do simulador de imagens do cérebro BrainWeb

A próxima tela do simulador é a de pré-visualização das fatias de corte do volume gerado e permite, entre outras opções, configurar o esquema de cores utilizado na pré-visualização e o fator de *zoom*. A Figura 15 apresenta esta tela de pré-visualização com a configuração padrão.

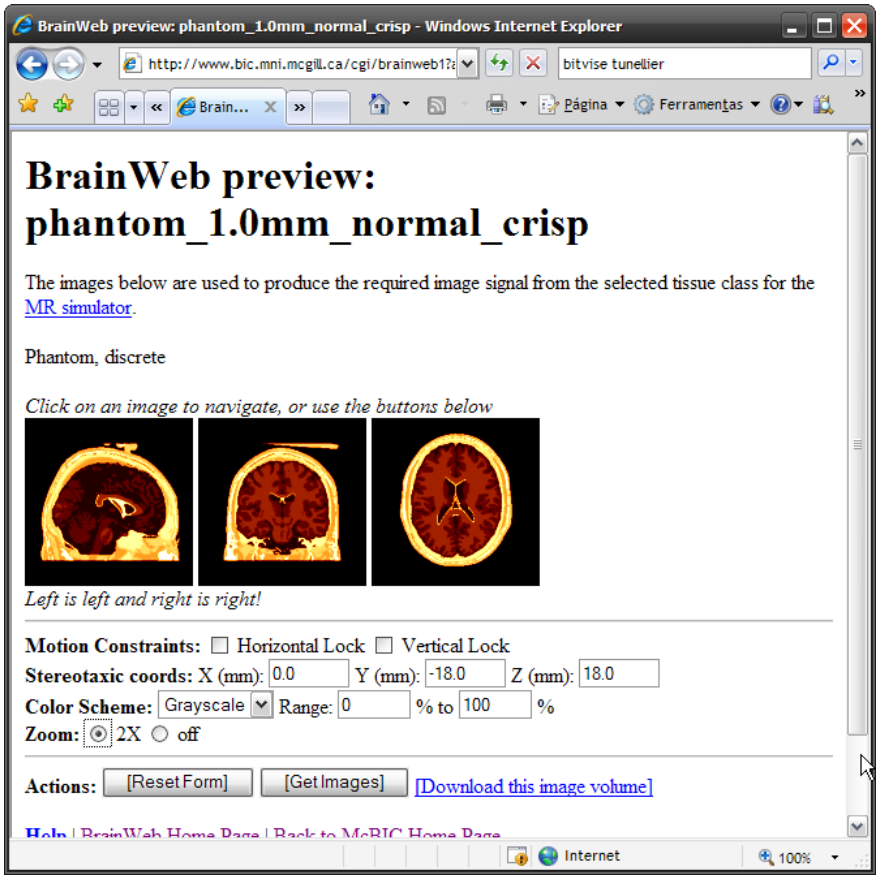


Figura 15: Tela de pré-visualização do simulador BrainWeb

Os dados são fornecidos na forma de um volume de imagens, ou seja, um conjunto de dados [*x*,*y*,*z*], com *x* e *y* correspondendo à largura e à altura da imagem e *z* correspondendo ao índice da imagem no conjunto de fatias de imagens (*slices*). Além da natureza dos modelos dos dados de imagem e das configurações demonstradas anteriormente, o sistema fornece dados da seguinte forma:

a) Modelo Discreto: que fornece uma espécie de imagem rotulada, ou seja, cada pixel possuindo como valor o número da classe à qual pertence;

b) Modelo *Fuzzy*: que fornece um conjunto de dados segmentado na forma *fuzzy*, ou seja, um pixel podendo pertencer a mais de uma classe e, desta forma, contém 10 conjuntos de dados, um para cada classe.

As classes existentes são 10, a saber:

1. Plano de fundo (*background*);
2. CSF (fluido cérebro-espinhal) (*csf*);
3. Matéria cinzenta (*gray matter*);

4. Matéria branca (*white matter*);
5. Gordura (*fat*);
6. Músculo / Pele (*músculo*);
7. Pele (*pele*);
8. Crânio / osso (*skull*);
9. Matéria Glial (*glial matter*);
10. Material Conectivo (*connective*).

A Figura 16 apresenta algumas imagens para ilustração das classes do conjunto de dados.

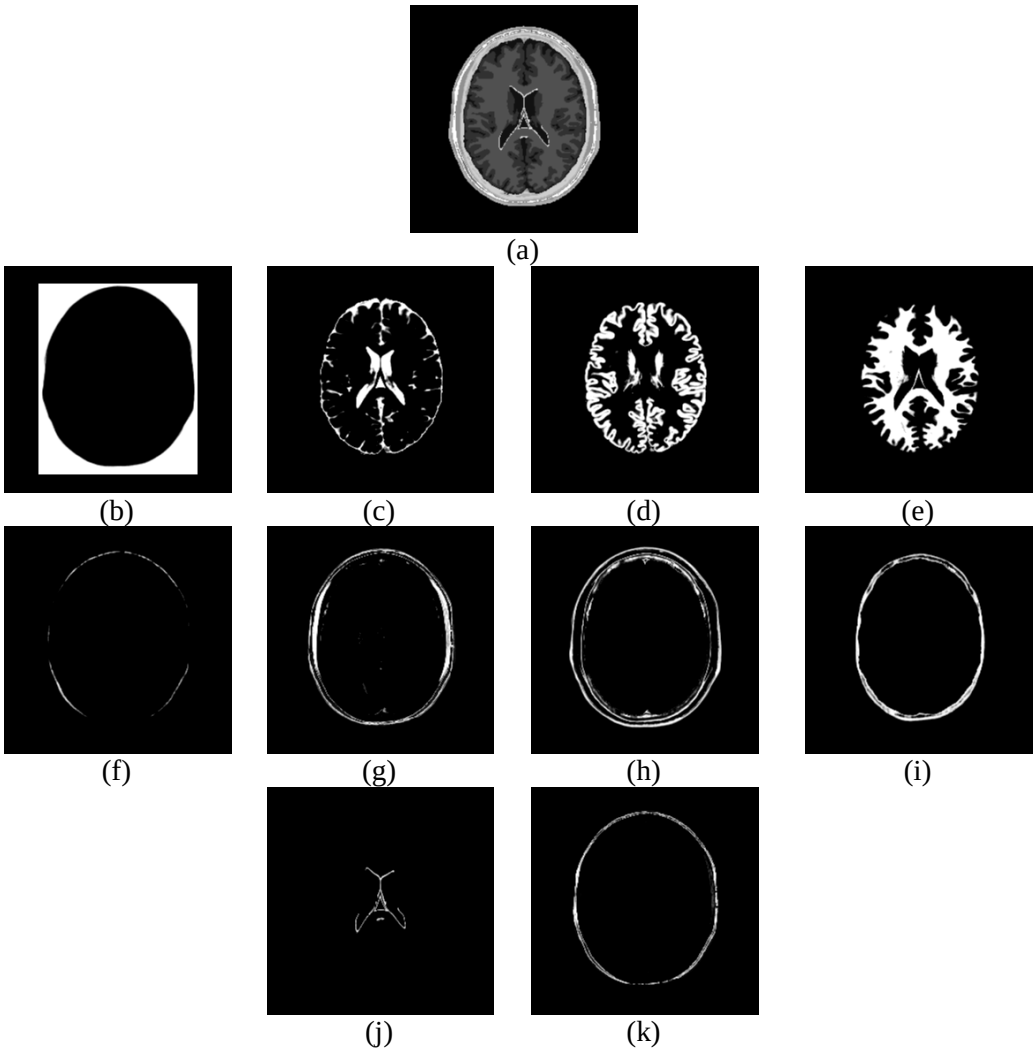
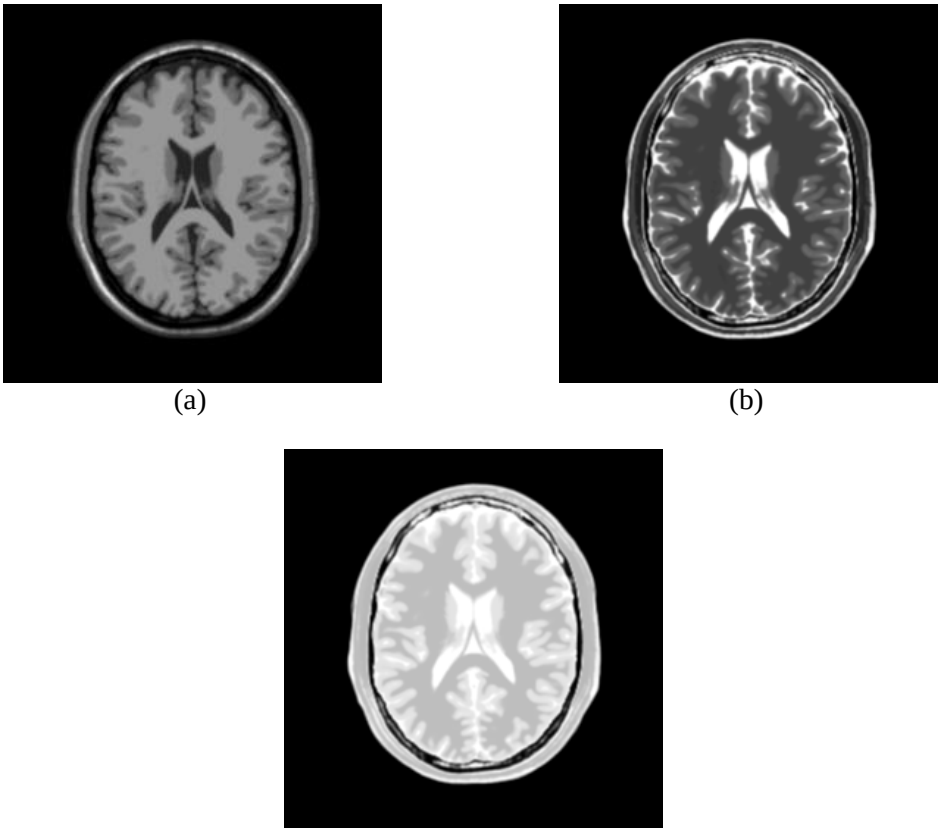


Figura 16: Exemplos de imagem discreta (a) e as diversas classes do conjunto de dados (b - k)

A simulação do BrainWeb, como visto, é realizada de acordo com diversos parâmetros, um deles é a modalidade da simulação (T1, T2 e PD). A Figura 17 apresenta exemplos de fatias de imagem no modelo discreto que ilustram as diferenças de intensidade entre esses modelos.



(a) (b) (c)

Figura 17: Exemplos de fatias de imagens nos modelos T1 (a), T2 (b) e PD (c)

As imagens utilizadas para testes contêm os valores de intensidade dependendo da modalidade do conjunto de dados (T1, T2 e PD) e estão no intervalo [0, 1000], portanto, antes da imagem passar pelo algoritmo de segmentação de imagens, ela é normalizada no intervalo [0, 1]. Por fim, nos testes com os algoritmos de segmentação de imagens é utilizado o seguinte arranjo de configurações:

a) Modalidade: T1;
b) Espessura da fatia: 1mm;
c) Ruído: 0%; e
d) RF: 0%.

4.3 Segmentação de Imagens Médicas usando Computação Natural

Como já demonstrado nos capítulos anteriores, a tarefa de Segmentação de Imagens pode ser equiparada à classificação (ou *clustering*) não-supervisionada. Seguindo a metodologia do Capítulo anterior, esta seção apresenta resultados de experimentos e discussões a respeito de testes e avaliações da utilização de métodos de computação natural (Algoritmos Genéticos e PSO). As imagens utilizadas para teste não são reais, como que obtidas de exames médicos, ou seja, são simuladas pelo simulador BrainWeb, como indicado na seção 5.2.

Os testes são realizados seguindo uma seqüência de procedimentos:

1. leitura da imagem e armazenamento em uma matriz [*linhas*, *colunas*];
2. conversão dos dados da imagem para uma matriz no formato [*linhas* * *colunas*, 1] gerando o conjunto de dados de teste;
3. normalização dos dados de teste no intervalo [0, 1];
4. definição de parâmetros para o algoritmo de segmentação de imagens;
5. segmentação da imagem utilizando o conjunto de dados de teste;
6. análise do resultado da segmentação.

Os volumes de imagens fornecidos pelo BrainWeb estão em um formato chamado MINC. Para ler os volumes de dados e, posteriormente, apresentar as imagens aos algoritmos de segmentação e visualizá-las, foram utilizadas as funções do *toolbox* para Matlab MNI-EMMA desenvolvido por Morales-Aguilera (2007).

Os experimentos são realizados com os seguintes algoritmos para segmentação de imagens:

- Algoritmo K-means Genético (GKA)
- Algoritmo Genético com Fuzzy C-Means (GFCMA)
- Algoritmo de Segmentação através de PSO e K-means (PSOKA)
- Algoritmo de Segmentação através de PSO e Fuzzy C-means (PSOFCM)

Em cada utilização dos algoritmos citados, são realizadas validações da segmentação através de índices de validação. O objetivo dos experimentos é encontrar o melhor conjunto de parâmetros para os algoritmos que fornece melhor resultado, ou seja, otimiza as funções-objetivo, de acordo com cada índice de validação. Como visto anteriormente, na seção sobre características das imagens médicas, não é necessário realizar procedimentos de pré-processamento ou transformações de dados além da normalização dos dados no intervalo [0, 1]. As imagens de testes, que são em escala de cinza, reduzem a necessidade dos testes serem realizados com imagens coloridas e não requer que sejam feitas análises e transformações do espaço de cores da imagem.

Alguns experimentos requerem pré-processamento adicional dos dados de entrada no sentido de adicionar certas estatísticas: média, desvio-padrão, variância, co-variância e coeficiente de correlação. Estas medidas estatísticas são calculadas considerando uma "janela" no conjunto de dados com o objetivo de que essas medidas auxiliem na segmentação de imagens indicando padrões que estão mais próximos em relação a uma vizinhança. O objetivo destas colunas adicionais no conjunto de dados é verificar se as mesmas contribuem para que o algoritmo de segmentação de imagens considere características como textura para obtenção de melhores resultados.

O BrainWeb fornece o *ground-truth* para a segmentação, ou seja, os resultados desejados para um algoritmo de segmentação são fornecidos através do *modelo discreto*. Desta forma, o modelo discreto é utilizado para obtenção da taxa de erro dos algoritmos. Conforme o que foi feito no Capítulo anterior, esta informação não é utilizada para guiar os algoritmos, mas apenas como métrica e avaliação do desempenho de forma diferente do valor da função objetivo.

4.4 Resultados, Análises e Comparações

Esta seção apresenta os resultados encontrados com os experimentos realizados. Como visto no capítulo anterior, algumas conclusões levaram à definição de certos parâmetros para os algoritmos, facilitando assim a análise dos resultados. Há dois conjuntos de dados para os algoritmos de segmentação: um que considera as estatísticas

e a textura de uma janela de tamanho 5 x 5, chamado conjunto imagem_textura; e outro que considera apenas os valores de intensidade dos pixels da imagem, chamado imagem_intensidade.

4.4.1 Algoritmos Genéticos aplicados à Segmentação de Imagens

A Tabela 9 apresenta os resultados obtidos considerando os algoritmos GKA e GFCMA, os conjuntos de dados imagem_intensidade e imagem_textura e os valores mínimo, médio e máximo obtidos para cada índice de validação (DB, SC, S, XB) e a taxa de erro.

Tabela 9: Resultados dos Algoritmos GKA e GFCMA: a) conjunto de dados imagem_textura; b) conjunto de dados imagem_intensidade

(a)

Índice	**GKA**			**GFCMA**		
	Min.	**Méd.**	**Max.**	**Min.**	**Méd.**	**Max.**
DB	0,31476	0,33616	0,35755	0,33990	0,35202	0,36414
SC	0,16075	0,16075	0,16075	0,16075	0,28419	0,29732
S	0,00001	0,00001	0,00001	0,00001	0,00001	0,00001
XB	169,59364	202,84558	236,09751	84,10645	92,94704	101,78762
Erro (%)	75,19	85,17	90,80	55,66	61,42	70,80

(b)

Índice	**GKA**			**GFCMA**		
	Min.	**Méd.**	**Max.**	**Min.**	**Méd.**	**Max.**
DB	0,68072	0,68072	0,68072	0,71655	0,73323	0,74991
SC	0,20654	0,20654	0,20654	0,64346	0,64346	0,64346
S	0,00001	0,00001	0,00001	0,00003	0,00003	0,00003
XB	10,19488	10,19488	10,19488	1,17901	1,24838	1,31775
Erro (%)	84,96	87,68	91,68	67,67	69,49	71,27

Analisando os resultados, pode-se afirmar que, tanto em relação ao conjunto de dados imagem_textura quanto ao conjunto de dados imagem_intensidade, o algoritmo PSOKA é melhor que o PSOFCM nos índices DB, SC e S; e PSOFCM é melhor no índice XB e na taxa de erro. Vale notar também que dentre os dois conjuntos, o que proporcionou menor taxa de erro, em ambos os algoritmos, foi o imagem_textura.

Considerando a taxa de erro mínima e máxima, a Tabela 10 apresenta os resultados obtidos e os parâmetros utilizados pelos algoritmos para encontrá-los, inclusive, indicando o índice que direcionou o algoritmo para a menor e maior taxa de erro.

Tabela 10: Resultados dos Algoritmos GKA e GFCMA considerando taxa de erro mínima e máxima: a) conjunto de dados imagem_textura; b) conjunto de dados imagem_intensidade

(a)

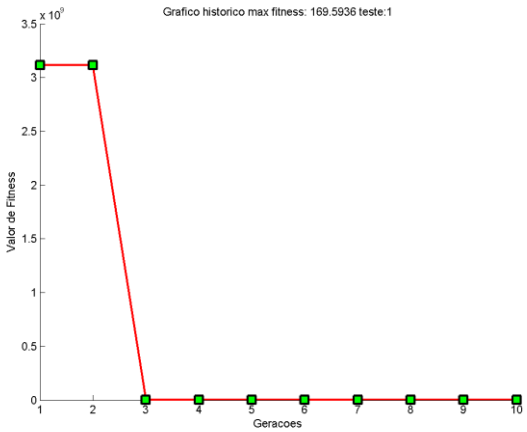
	GKA					GFCMA				
	Erro	**P**	**VMAX**	**THIST**	**Índice**	**Erro**	**P**	**VMAX**	**THIST**	**Índice**
Min	75,19	5	0,025	10	XB	55,66	20	0,025	10	XB
Max	90,80	5	0,025	10	DB	70,80	5	0,025	10	SC

(b)

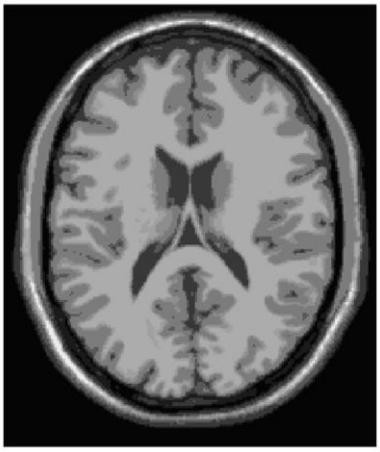
	GKA					GFCMA				
	Erro	**P**	**VMAX**	**THIST**	**Índice**	**Erro**	**P**	**VMAX**	**THIST**	**Índice**
Min	84,96	20	0,025	10	S	67,67	5	0,025	10	S
Max	91,68	5	0,025	10	SC	71,27	20	0,025	10	DB

Através de análise dos resultados, pode-se afirmar que GFCMA foi superior ao GKA em todos os conjuntos de dados. Os parâmetros dos algoritmos indicam também que, para o conjunto de dados imagem_textura, a menor taxa de erro é obtida pelo índice XB, enquanto que para o conjunto de dados imagem_intensidade, o índice S é o que fornece melhores resultados. A Figura 18 ilustra melhor estes resultados.

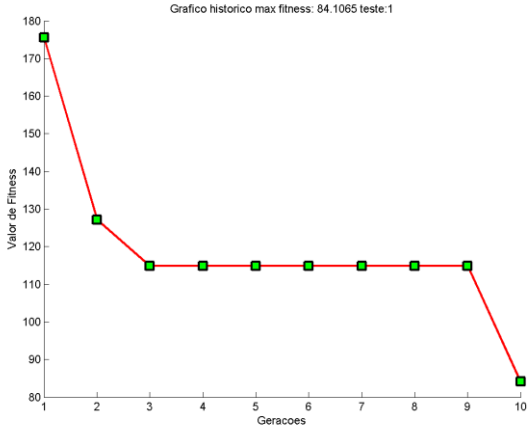
(a)



(b)

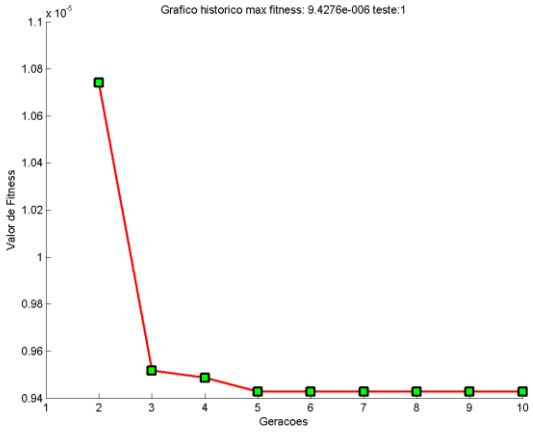


(c)



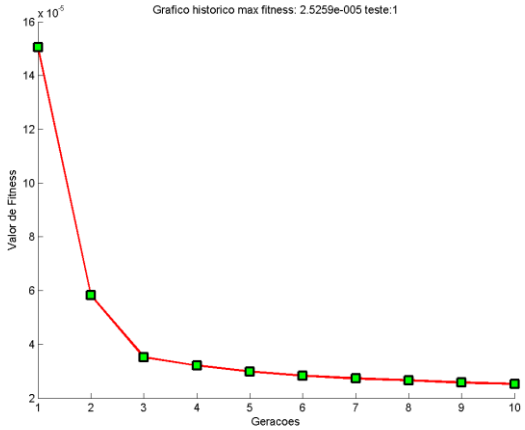
(d)

(e)



(f)

(g)



(h)

Figura 18: Resultados de Segmentação de Imagem com GKA e GFCMA: imagem_intensidade: a) e b) GKA, c) e d) GFCMA; imagem_textura: e) e f) GKA, g) e h) GFCMA

A Tabela 7 apresenta os resultados obtidos considerando-se o valor mínimo e máximo para cada índice de validação e os parâmetros utilizados pelos algoritmos para encontrá-los.

Tabela 11: Resultados dos Algoritmos GKA e GFCMA considerando valores mínimo e máximo dos índices de validação: a) conjunto de dados imagem_intensidade; b) conjunto de dados imagem_textura

(a)

	GKA					GFCMA				
	Valor	**Erro**	**NP**	**MP**	**GMAX**	**Valor**	**Erro**	**NP**	**MP**	**GMAX**
DB-min	0,31476	88,89	20	0,025	10	0,33990	66,14	5	0,025	10
DB-max	0,35755	90,80	5	0,025	10	0,36414	65,32	20	0,025	10
SC-min	0,16075	85,29	5	0,025	10	0,16075	58,63	20	0,025	10
SC-max	0,16075	85,29	20	0,025	10	0,29732	70,8	5	0,025	10
S-min	0,00001	85,29	20	0,025	10	0,00001	57,97	20	0,025	10
S-max	0,00001	85,29	5	0,025	10	0,00001	57,97	5	0,025	10
XB-min	169,59364	75,19	5	0,025	10	84,10645	55,66	20	0,025	10
XB-max	236,09751	85,28	20	0,025	10	101,78762	58,91	5	0,025	10

(b)

	GKA					GFCMA				
	Valor	**Erro**	**NP**	**MP**	**GMAX**	**Valor**	**Erro**	**NP**	**MP**	**GMAX**
DB-min	0,68072	84,96	20	0,025	10	0,71655	71,27	20	0,025	10
DB-max	0,68072	84,96	5	0,025	10	0,74991	67,96	5	0,025	10
SC-min	0,20654	91,68	20	0,025	10	0,64346	71,27	20	0,025	10
SC-max	0,20654	91,68	5	0,025	10	0,64346	71,27	5	0,025	10
S-min	0,00001	84,96	20	0,025	10	0,00003	67,67	5	0,025	10
S-max	0,00001	84,96	5	0,025	10	0,00003	67,67	20	0,025	10
XB-min	10,19488	89,12	20	0,025	10	1,17901	70,46	5	0,025	10
XB-max	10,19488	89,12	5	0,025	10	1,31775	68,40	20	0,025	10

A Tabela 11 apresenta os resultados dos índices de validação, mínimo e máximo, as configurações dos parâmetros dos algoritmos e a taxa de erro encontrada. Considerando-se que os valores para os parâmetros MP e GMAX são, respectivamente, 0,025 e 10, é correto afirmar que, para os dois conjuntos de dados, ambos os algoritmos possuem comportamento semelhante: o valor mínimo e máximo de cada índice de validação é encontrado, de forma alternada, tanto por NP = 5 quanto por NP = 20, ou

seja, 50% para cada valor; e ainda ocorre de terem sido encontrados valores idênticos. Assim, é interessante notar a influência destes parâmetros na taxa de erro: NP = 5 obteve melhor resultado apenas no índice XB, desta forma é conclusivo afirmar que é mais adequado utilizar NP = 20.

4.4.2 PSO aplicado à Segmentação de Imagens

A Tabela 12 apresenta os resultados obtidos considerando os algoritmos PSOKA e PSOFCM, os conjuntos de dados imagem_intensidade e imagem_textura e os valores mínimo, médio e máximo obtidos para cada índice de validação (DB, SC, S, XB) e a taxa de erro.

Tabela 12: Resultados dos Algoritmos PSOKA e PSOFCM: a) conjunto de dados imagem_textura; b) conjunto de dados imagem_intensidade

(a)

Índice	PSOKA			PSOFCM		
	Min.	**Méd.**	**Max.**	**Min.**	**Méd.**	**Max.**
DB	0,56977	0,63134	0,69290	0,67515	0,69406	0,71297
SC	0,25217	0,25217	0,25217	0,54645	0,55021	0,55397
S	0,00001	0,00001	0,00001	0,00002	0,00002	0,00002
XB	1,85475	1,85475	1,85475	1,03556	1,08944	1,14332
Erro (%)	45,18	72,28	90,06	57,78	66,60	72,69

(b)

Índice	PSOKA			PSOFCM		
	Min.	**Méd.**	**Max.**	**Min.**	**Méd.**	**Max.**
DB	0,29822	0,32084	0,34345	0,34237	0,34627	0,35017
SC	0,12357	0,15516	0,18674	0,12357	0,27962	0,29527
S	0,00001	0,00001	0,00001	0,00001	0,00001	0,00001
XB	210,67416	235,63937	260,60458	81,46051	83,40448	85,34844
Erro (%)	60,57	76,13	88,45	65,87	70,00	72,86

Tanto relação ao conjunto de dados imagem_textura quanto ao conjunto de dados imagem_intensidade, o algoritmo PSOKA é melhor que o PSOFCM nos índices DB, SC e S; e PSOFCM é melhor no índice XB e na taxa de erro. Dentre os dois

conjuntos de dados, o que proporcionou menor taxa de erro, em ambos os algoritmos, foi o imagem_textura.

Considerando a taxa de erro mínima e máxima, a Tabela 14 apresenta os resultados obtidos e os parâmetros utilizados pelos algoritmos para encontrá-los, inclusive, indicando o índice que direcionou o algoritmo para a menor e maior taxa de erro.

Tabela 13: Resultados dos Algoritmos PSOKA e PSOFCM considerando taxa de erro mínima e máxima: a) conjunto de dados imagem_textura; b) conjunto de dados imagem_intensidade

(a)

	PSOKA					**PSOFCM**				
	Erro	**P**	**VMAX**	**THIST**	**Índice**	**Erro**	**P**	**VMAX**	**THIST**	**Índice**
Min	45,18	10	0,2	10	XB	57,78	10	0,2	10	S
Max	90,06	10	0,2	50	SC	72,69	10	0,200	50	DB

(b)

	PSOKA					**PSOFCM**				
	Erro	**P**	**VMAX**	**THIST**	**Índice**	**Erro**	**P**	**VMAX**	**THIST**	**Índice**
Min	60,57	10	0,2	10	XB	65,87	10	0,2	50	SC
Max	88,45	10	0,2	10	SC	72,86	10	0,200	10	DB

Através de análise dos resultados, pode-se afirmar que PSOFCM foi superior ao PSOKA em todos os conjuntos de dados, principalmente considerando a média da taxa de erro, embora PSOKA tenha encontrado um valor mínimo de 45,18% para imagem_textura. Os parâmetros dos algoritmos indicam também que, a menor taxa de erro para PSOKA é obtida pelo índice XB, e para PSOFCM, índice S, no conjunto de dados imagem_textura, enquanto que para o conjunto de dados imagem_intensidade, o índice SC é o que fornece melhores resultados. A Figura 18 ilustra melhor estes resultados.

(a)

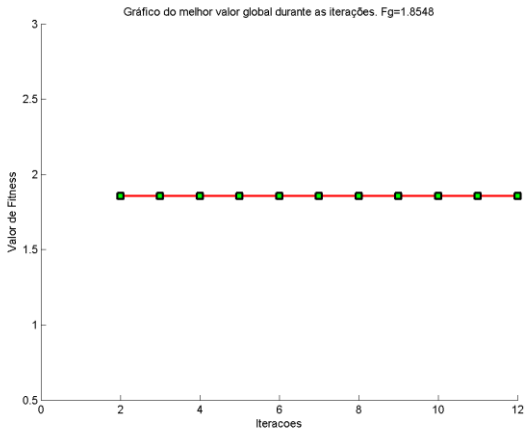


Gráfico do melhor valor global durante as iterações. Fg=1.8548
Valor de Fitness
Iteracoes

(b)

(c)

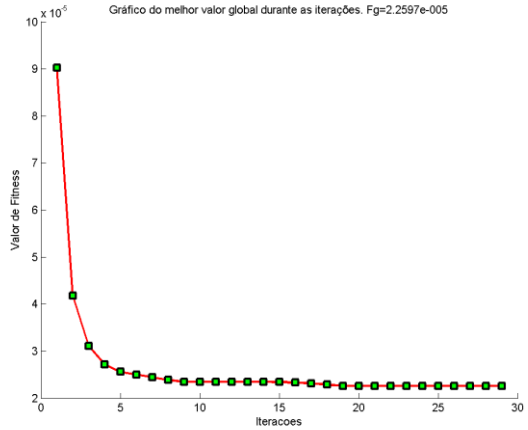
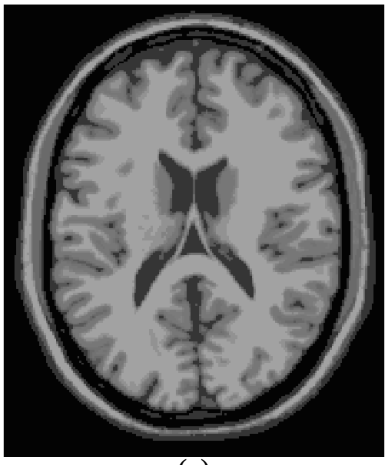


Gráfico do melhor valor global durante as iterações. Fg=2.2597e-005
x 10^-5
Valor de Fitness
Iteracoes

(d)



(e)

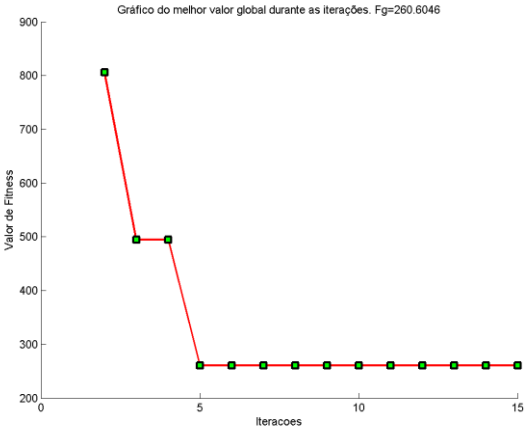
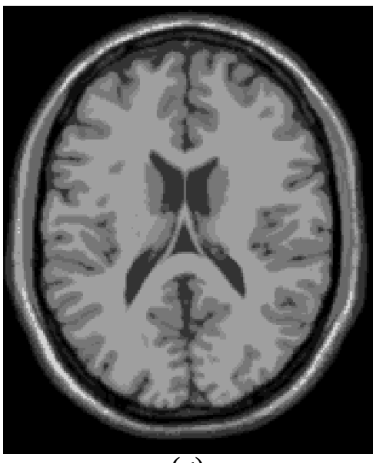
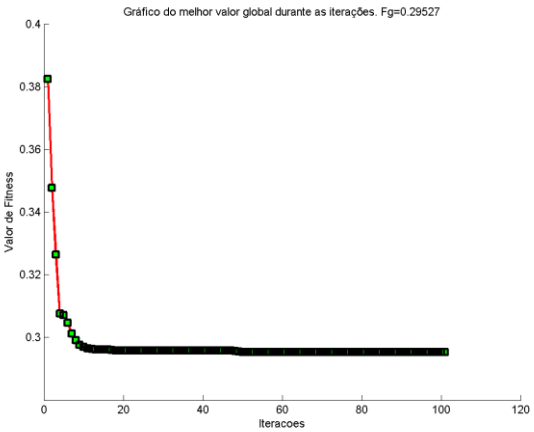


Gráfico do melhor valor global durante as iterações. Fg=260.6046
Valor de Fitness
Iteracoes

(f)



(g)



(h)

Figura 19: Resultados de Segmentação de Imagem com GKA e GFCMA: imagem_textura: a) e b) GKA, c) e d) GFCMA; imagem_intensidade: e) e f) GKA, g) e h) GFCMA

A Tabela 14apresenta os resultados obtidos considerando-se o valor mínimo e máximo para cada índice de validação e os parâmetros utilizados pelos algoritmos para encontrá-los.

Tabela 14: Resultados dos Algoritmos PSOKA e PSOFCM considerando valores mínimo e máximo dos índices de validação: a) conjunto de dados imagem_intensidade; b) conjunto de dados imagem_textura

(a)

	PSOKA					**PSOFCM**				
	Valor	**Erro**	**N**	**VMAX**	**THIST**	**Valor**	**Erro**	**N**	**VMAX**	**THIST**
DB-min	0,56977	57,41	10	0,2	50	0,67515	58,34	10	0,2	10
DB-max	0,69290	70,27	10	0,2	10	0,71297	72,69	10	0,2	50
SC-min	0,25217	90,06	10	0,2	10	0,54645	68,35	10	0,2	10
SC-max	0,25217	90,06	10	0,2	50	0,55397	67,85	10	0,2	50
S-min	0,00001	90,06	10	0,2	10	0,00002	57,78	10	0,2	10
S-max	0,00001	90,06	10	0,2	50	0,00002	64,55	10	0,2	50
XB-min	1,85475	45,18	10	0,2	10	1,03556	68,71	10	0,2	10
XB-max	1,85475	45,18	10	0,2	50	1,14332	71,52	10	0,2	50

(b)

	PSOKA					**PSOFCM**				
	Valor	**Erro**	**N**	**VMAX**	**THIST**	**Valor**	**Erro**	**N**	**VMAX**	**THIST**
DB-min	0,29822	72,01	10	0,2	50	0,34237	71,13	10	0,2	50

DB-max	0,34345	71,69	10	0,2	10	0,35017	72,86	10	0,2	10
SC-min	0,12357	71,61	10	0,2	50	0,12357	70,63	10	0,2	10
SC-max	0,18674	88,45	10	0,2	10	0,29527	54,87	10	0,2	50
S-min	0,00001	88,45	10	0,2	10	0,00001	70,28	10	0,2	10
S-max	0,00001	88,45	10	0,2	50	0,00001	66,02	10	0,2	50
XB-min	210,67416	67,79	10	0,2	50	81,46051	72,52	10	0,2	50
XB-max	260,60458	60,57	10	0,2	10	85,34844	70,7	10	0,2	10

Considerando-se que os valores para os parâmetros P e VMAX são, respectivamente, 10 e 0,2, é correto afirmar que, de modo geral, é mais adequado THIST = 50, embora PSOFCM tenha obtido melhor resultado no conjunto de dados imagem_textura quando THIST = 10.

Avaliando os resultados obtidos em relação aos índices de validação e à taxa de erro, pela média, é possível afirmar que:

- no caso do conjunto de dados imagem_textura, GKA e GFCMA são superiores nos índices DB, S e SC, enquanto que PSOKA e PSOFCM são superiores no índice XB e na taxa de erro;
- no caso do conjunto de dados imagem_intensidade, GKA e GFCMA são superiores apenas no índice XB, enquanto PSOKA e PSOFCM são superiores nos índices DB, SC e S e na taxa de erro.

Conclui-se este Capítulo sobre segmentação de imagens utilizando Algoritmos Genéticos e PSO afirmando que ambos os algoritmos apresentaram resultados consideráveis nos testes realizados, entretanto, considerando a taxa de erro, que é obtida a partir do erro de classificação que leva em consideração a melhor solução encontrada pelo algoritmo e os dados (pixels) já rotulados, PSO apresentou melhores resultados.

Capítulo 5

Considerações Finais e Conclusões

Métodos de computação natural, embora não sejam novos, como Redes Neurais Artificiais e Algoritmos Genéticos, estão se tornando cada vez mais utilizados em diversas aplicações científicas ou não. O trabalho de pesquisadores na área de computação natural, empenhando-se em organizar os métodos que se encaixem nesta área é bastante válido, uma vez que permite o desenvolvimento de bases sólidas para o desenvolvimento de novos métodos e para o melhoramento dos já existentes. Os trabalhos desenvolvidos nesta área, em andamento ou concluídos, demonstram o quanto é válida a afirmação de que o homem aprende a resolver problemas e a melhorar soluções existentes a partir de si próprio e da observação do seu eu e do que está ao seu redor, como a natureza e outros seres considerados, muitas vezes, inferiores por causa de seu intelecto limitado, mas que acabam por se tornar nossa inspiração e, por muitas vezes, razão de ser. Neste sentido, faz-se necessário agradecer também aos pesquisadores das disciplinas de Filosofia, Psiquiatria e outras que se prestam ao estudo do comportamento do homem consigo mesmo e em sociedade.

Durante o tempo de desenvolvimento do presente trabalho de dissertação de mestrado, foi possível observar a grande quantidade de aplicações de técnicas de computação natural na solução de problemas complexos. Entre essas técnicas, destaque se faz necessário para a mais nova delas, Sistemas Imunes Artificiais, que, considerando a ciência de Reconhecimento de Padrões, pode indicar o desenvolvimento de algoritmos para mineração de dados e segmentação de imagens. Faz-se importante citar este fato, considerando também que trabalhos oriundos da presente dissertação podem explorar ainda mais as técnicas que aqui foram apenas mencionadas ou citadas. Além disso, levando-se em consideração a natureza estocástica dos processos de resolução de

problemas de otimização, foi despertado o interesse de desenvolvimento de metodologias que utilizem sistemas multi-agentes para, por exemplo:

- segmentação de imagens de forma coordenada e cooperativa utilizando sistemas multi-agentes;
- mecanismos para comportamento inteligente em agentes de jogos de computador;
- sistemas de simulação de comportamento de mercado, atuando na área de *business intelligence*.

Voltando para o assunto do trabalho de pesquisa em questão, a metodologia para a avaliação dos resultados foi crucial para o bom desenvolvimento do trabalho. Como visto a metodologia de avaliação dos resultados obtidos pelos algoritmos foi baseada na seguinte definição:

- realizar testes de detecção de aglomerados com as bases de dados Iris, Ruspini e Wine;
- realizar testes de segmentação de imagens com os dados de imagens MRI simulados pelo BrainWeb e utilizar um conjunto de dados com informações de textura e outro apenas com as informações de intensidades dos pixels;
- analisar os resultados em relação a:
  a) valor mínimo e máximo para cada índice de validação;
  b) valor mínimo e máximo para a taxa de erro;
  c) configurações de parâmetros utilizados para encontrar cada solução considerada como ótima.

A utilização dos índices de validação também serviu como forma bastante objetiva para avaliação dos resultados. Como continuação do presente trabalho, conforme o surgimento de novos índices de validação é interessante atualizar os testes realizados e verificar o desempenho dos algoritmos e os novos resultados.

Considerando a tarefa de detecção de aglomerados, mesmo sendo um conjunto de dados bastante simples, o conjunto de dados de Ruspini, se mostrou bastante adequado para a realização dos experimentos, levando-se em conta o pequeno número de padrões e dimensões, tal característica deste conjunto foi imprescindível para a implementação dos algoritmos, uma vez que eram necessárias constantes execuções de código passo-a-passo para verificação do real comportamento do algoritmo. Neste ponto, o ambiente de programação do Matlab 2006b apresentou-se como uma ferramenta poderosa para o desenvolvimento de algoritmos. O conjunto de ferramentas de visualização de dados, como gráficos e ferramentas para análise de imagem, foi de grande ajuda no desenvolvimento dos algoritmos, pois, considerando que outras linguagens e outros ambientes de programação, como Java e C++, possam apresentar resultados melhores, desenvolver os algoritmos nestes ambiente demandaria em bem mais tempo de programação. O *Fuzzy Clustering Toolbox* e suas implementações rápidas dos algoritmos K-means e FCM e os índices de validação também contribuíram enormemente. Além disso, o *Cluster Validation Toolbox* também foi fundamental para a obtenção dos resultados em relação à taxa de erro dos algoritmos, tanto na detecção de aglomerados quanto na segmentação de imagens.

Os experimentos realizados em detecção de aglomerados demonstraram que tanto os métodos baseados em Algoritmos Genéticos quanto os baseados em Otimização por Enxame de Partículas apresentaram resultados consideráveis. Os algoritmos foram comparados com suas implementações convencionais (K-means e Fuzzy C-means) e, como visto, foram bastante superiores. Estas comparações foram realizadas, mesmo levando em consideração que os métodos baseados em Algoritmos Genéticos e PSO executam avaliações de diversas soluções em paralelo. Inclusive, é interessante comentar que o modo de trabalho dos métodos baseados em computação natural, como parte do contexto de otimização de problemas, está estritamente relacionado com a utilização de múltiplas soluções buscando, com isso, resolver um problema de modo cooperativo. Nos Algoritmos Genéticos, os membros da população relacionam-se de maneira tal a trocar informações sobre as soluções que cada um tenha encontrado e procuram, assim, gerar novos indivíduos, em novas gerações, que sejam

melhores que seus genitores. No PSO, os membros do enxame, as partículas, carregam consigo uma solução individual para o problema e procuram conhecer as soluções dos demais membros da vizinhança para, então, melhorar sua própria solução, de forma que todos os membros, ao final, tenham encontrado soluções relativamente boas.

A tarefa de segmentação de imagens não constitui um processo trivial; considerando o domínio de imagens médicas é altamente importante a opinião de um especialista do domínio em relação aos resultados apresentados pelos experimentos. Como os dados utilizados para testes foram fornecidos pelo sistema BrainWeb, a forma de avaliação dos resultados foi baseada nas classes que tal sistema apresenta. Desta forma, mesmo apresentando resultados adequados, seria interessante expandir o trabalho para possibilitar sua aplicação a imagens reais – de exames de MRI e CT.

A utilização dos métodos *fuzzy* embutidos como algoritmos de detecção de aglomerados nos métodos de Algoritmos Genéticos e Enxames de Partículas permitiram a verificação da execução de tais algoritmos com métodos mais avançados de detecção de aglomerados. O que se pôde verificar, até no que concerne à segmentação de imagens, é que é visível a superioridade do algoritmo FCM em termos de maior proximidade de um valor ótimo global, embora em relação ao tempo de processamento essa superioridade não seja comprovada. Sobretudo, como os métodos implementados estão baseados em otimização computacional, foi possível verificar que tanto os algoritmos baseados em K-means quanto os baseados em FCM apresentaram resultados consideráveis e semelhantes, já que este é o objetivo dos métodos: encontrar um valor ótimo que minimize a função objetivo. Entretanto, como o presente trabalho não se desenvolve até as tarefas de registro e classificação de imagem, ainda são necessários mais subsídios para poder afirmar se o algoritmo FCM é superior ao K-means, nos termos dos métodos de computação natural, ou em quais casos um seja mais adequado que o outro.

Referências Bibliográficas

Guillermo N. Abras and Virginia L. Ballarin. A Weighted K-means Algorithm applied to Brain Tissue Classification. Journal of Computer Science & Technology. vol. 5. No. 3. October, 2005. Disponível online em: <http://journal.info.unlp.edu.ar/journal/journal15/papers/JCST-Oct05-2.pdf >. Último acesso em: 27 de abril de 2007.

Mohamed N. Ahmed. **Novel Image Segmentation And Registration Algorithms For The Study Of Brain Structure And Function**. Doctor Thesis. Faculty of the Graduate School of the University of Louisville. Program in Computer Science and Engineering. Louisville, Kentucky. 1997.

M. P. Albuquerque, I. A. Esquef, A. R. Mello. 2004. Image thresholding using Tsallis entropy. Pattern Recognition Letters. Vol. 25, no. 9, p. 1059 – 1065, Julho, 2004. DOI= http://dx.doi.org/10.1016/j.patrec.2004.03.003.

S. Bandyopadhyay, U. Maulik. Genetic clustering for automatic evolution of clusters and application to image classification. Pattern Recognition (35), Elsevier, 2002, pp. 1197-1208.

Yoseph Bar-Cohen. Biomimetics: biologically inspired technologies. California, Estados Unidos: Taylor & Francis, 2006.

James C. Bezdek, Richard J. Hathaway, Michael J. Sabin, William T. Tucker. **Convergence Theory for Fuzzy c-Means: Counterexamples and Repairs**. IEEE Trans. Syst., Man, Cybern., vol. SMC-17, no. 5, pp. 897—877, September/October, 1987.

Anthony Brabazon and Michael O'Neill. Biologically inspired algorithms for financial modeling. Springer Natural Computing Series. Berlin, Alemanha: Springer-Verlag, 2006.

BrainWeb. *BrainWeb: Simulated Brain Database*. Disponível online em: <http://www.bic.mni.mcgill.ca/brainweb/>. Último acesso em: 27 de abril de 2007.

Leandro N. de Castro. *Fundamentals of natural computing: an overview*. Physics of Life Reviews, Elsevier. Vol. 4. Issue 1. March 2007. pp. 1-36.

S. C. M. Cohen, and L. N. de Castro. Data Clustering with Particle Swarms. 2006 IEEE Congress on Evolutionary Computations, Canada, July 2006, pp. 1792-1798.

E. Davies. Machine Vision: Theory, Algorithms, Practicalities. Academic Press, 2 ed., 1997.

E. Forgy. Cluster Analysis of Multivariate Data: Efficiency versus Interoperability of Classification. Biometrics, 21, 1965, pp. 768—769.

H. Frigui, R. Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. IEEE Trans Pattern Anal. Mach. Intell. 21 (5), 1999, pp. 450–465.

Rafael C. Gonzalez, Richard E. Woods. **Processamento de Imagens Digitais**. Tradução: Roberto Marcondes César Junior e Luciano da Fontoura Costa. 1ª reimpressão. São Paulo: Editora Edgard Blücher, 2003.

G. Hamerly, C. Elkan. Alternatives to the K-means Algorithm that Find Better Clusterings. In Proceedings of the ACM Conference on Information and Knowledge Management (CIKM-2002), 2002, pp. 600—607.

Simon Haykin. Redes neurais: princípios e prática. Trad. Paulo Martins Engel. 2 ed. Porto Alegre: Bookman, 2001.

Bernd Jähne. **Digital Image Processing**. 5 ed. Berlim: Springer-Verlag, 2002.

Anil K. Jain, Richard C. Dubes. **Algorithms for Clustering Data**. New Jersey: Prentice Hall, 1988.

A. K. Jain, M.N. Murty, P. J. Flynn. **Data clustering: a review**. ACM Comput. Surv., ACM Press, 1999, 31, 264-323.

Keith A. Johnson. *Neuroimaging Primer*. Harvard Medical School. Disponível online: < http://www.med.harvard.edu/AANLIB/hms1.html>. Último acesso em: 27 de abril de 2007.

James Kennedy and Russel C. Eberhart (com Yuhui Shi). Swarm intelligence: collective, adaptive. The Morgan Kaufmann Series in Evolutionary Computation. New York: Morgan Kaufmann, 2001.

K. Krishna and M. Narasimba Murty, Genetic K-Means Algorithm. IEEE Trans. Systems, Man, and Cybernetics – Part B: Cybernetics. vol. 29. no. 3. 1999. pp. 433-439.

Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. J. Brown. FGKA: A Fast Genetic K-means Clustering Algorithm. 2004 ACM Symposium on Applied Computing, March 2004, pp. 622-623.

D. Martin et. al. *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*. Proc. 8th Int'l Conf. Computer Vision. Vol 2. July, 2001. pp. 416 – 423. <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>

Sushmita Mitra, Tinku Acharys. **Data mining: multimedia, soft computing and bioinformatics**. New Jersey: John Wiley & Sons, 2003.

Frank R. Morales-Aguilera. Frank MORALE'S Home Page, MNI-EMMA. Disponível em: <http://www.bic.mni.mcgill.ca/users/fmorales/>. Último acesso: 26/06/2007.

Kepler S. Oliveira Filho. **Fundamentos de Radiodiagnóstico por Imagem**. Instituto de Física da UFRGS. Disponível on-line em: < http://www.if.ufrgs.br/ast/med/imagens/index.htm>. Último acesso em: 19/06/2006.

M. G. Omram, A. Salman, and A. P. Engelbrecht, Dynamic clustering using particle swarm optimization with application in image segmentation, Pattern Anal. Applic. (8), Springer-Verlag, London, 2006, pp. 332-344.

Dzung L. Pham, Chenyang Xu, Jerry L. Prince. **A Survey of Current Methods in Medical Image Segmentation**. Department of Electrical and Computer Engineering. The Johns Hopkins University, Baltimore MD 21218. Technical Report JHU/ECE 99-01. Annual Review of Biomedical Engineering. January. 1998.

Dzung L. Pham e Jerry L. Prince. Adaptive Fuzzy Segmentation of Magnetic Resonance Images. IEEE Transactions on Medical Imaging, vol. 18, n. 9, setembro/1999, pp. 737–752.

Mehmet Sezgin, Bülent Sankur. **Survey over image thresholding techniques and quantitative performance evalutation**. *Journal of Eletronic Imaging.* vol. 13, no. 1, p. 146 – 165, Janeiro, 2004.

Mhemet Sezgin, Bülent Sankur. Survey over image thresholding techniques and quantitative performance evalutation. Journal of Eletronic Imaging. vol. 13, no. 1, 2004, p. 146 – 165.

Marcos A. S. da Silva, Antônio M. V. Monteiro, and José S. de. Medeiros. Segmentação Semi-Automática de Dados Geo-espaciais Multivariados com Mapas Auto-Organizáveis. WORCAP, INPE, 2003.

R. H. Turi. Clustering-Based Colour Image Segmentation. PhD Thesis. Monash University, Australia, 2001.

L. A. Zadeh. Fuzzy sets. Information and Control, 8, 1965, pp. 338—353.

L. A. Zadeh. Fuzzy logic, neural networks, and soft computing. Communications of the ACM, 37, 1994, pp. 77—84.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Administração
Baixar livros de Agronomia
Baixar livros de Arquitetura
Baixar livros de Artes
Baixar livros de Astronomia
Baixar livros de Biologia Geral
Baixar livros de Ciência da Computação
Baixar livros de Ciência da Informação
Baixar livros de Ciência Política
Baixar livros de Ciências da Saúde
Baixar livros de Comunicação
Baixar livros do Conselho Nacional de Educação - CNE
Baixar livros de Defesa civil
Baixar livros de Direito
Baixar livros de Direitos humanos
Baixar livros de Economia
Baixar livros de Economia Doméstica
Baixar livros de Educação
Baixar livros de Educação - Trânsito
Baixar livros de Educação Física
Baixar livros de Engenharia Aeroespacial
Baixar livros de Farmácia
Baixar livros de Filosofia
Baixar livros de Física
Baixar livros de Geociências
Baixar livros de Geografia
Baixar livros de História
Baixar livros de Línguas

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo