

COMPARAÇÃO ENTRE AS TÉCNICAS ENXAME DE PARTÍCULAS E ALGORITMO GENÉTICO NA OTIMIZAÇÃO DE UM SISTEMA DE RECONHECIMENTO DE VOZ

AMANDA ABELARDO VIANA BESERRA*, GINALBER L.O. SERRA*, WASHINGTON LUÍS SANTOS SILVA*

**Instituto Federal de Educação Ciência e Tecnologia do Maranhão*
Av. Getúlio Vargas, nº 04 - Monte Castelo
São Luís, Maranhão, Brasil 65030-005

Emails: amandaabelardo@hotmail.com, ginalber@ifma.edu.br, washington.wlss@ifma.edu.br

Abstract— This paper proposes a comparison of optimization techniques using Particle Swarm and Genetic Algorithm and a fuzzy inference system for automatic recognition of numerical commands voice. In addition preprocessing the speech signal with mel-frequency cepstral coefficients, we use the discrete cosine transform (DCT) to generate a two-dimensional temporal array used as input to a system of fuzzy implication to generate the pattern of the words to be recognized.

Keywords— Fuzzy Systems, Particle Swarm, Genetic Algorithm, Discrete Cosine Transform, Speech Recognition.

Resumo— Este artigo propõe a comparação da otimização entre as técnicas Enxame de Partículas e Algoritmo Genético de um sistema de inferência fuzzy para reconhecimento automático de comandos numéricos de voz. Além do pré-processamento do sinal de voz com coeficientes mel-cepstrais, utiliza-se a Transformada Cosseno Discreta (TCD) para gerar uma matriz temporal bidimensional, utilizada como entrada de um sistema de implicação fuzzy para gerar o padrão das palavras a serem reconhecidas.

Keywords— Sistemas Fuzzy, Enxame de Partículas, Algoritmo Genético, Transformada Cosseno Discreta, Reconhecimento de Voz.

1 Introdução

Nas últimas décadas algoritmos bioinspirados baseados em populações e meta heurísticas vêm sendo utilizados para resolver problemas de busca e otimização em vários domínios de problemas, para os quais soluções robustas são difíceis ou impossíveis de se encontrar usando abordagens clássicas. Esses algoritmos são inspirados em mecanismos biológicos da evolução, tais como o Enxame de Partículas e o Algoritmo Genético, técnicas que serão comparadas neste artigo. O Enxame de Partículas, ou do original Particle Swarm Optimization (PSO) é um método de busca heurística relativamente recente, que é inspirado pelo comportamento colaborativo e individual de populações biológicas (da Luz, 2008). Este método foi introduzido por James Kennedy e Russel Elberhart em 1995 (Kennedy and Eberhart, n.d.) para tratar problemas no domínio contínuo. Uma teoria sócio-cognitiva está por trás da PSO. Cada indivíduo de uma população possui sua própria experiência e é capaz de estimar a qualidade da mesma. Como os indivíduos são sociais, eles também possuem conhecimentos sobre como seus vizinhos se comportam. Esses dois tipos de informações correspondem à aprendizagem individual (cognitiva) e à transmissão cultural (social), respectivamente.

A probabilidade de um determinado indivíduo decidir será uma função de seu conhecimento adquirido e do conhecimento de alguns de vizinhos (Serapião, 2009). A partir desse princípio é definido um espaço de busca, em que as partículas têm sua velocidade e posição próprias, que a cada interação é atualizada levando-se em conta uma série de variáveis individuais e coletivas. O algoritmo Genético (AG) por sua vez é baseado na evolução natural e assim como acontece no ambiente, em um AG existe um grupo de indivíduos que competem entre si, para garantirem a própria sobrevivência e para assegurar que as melhores características sejam passadas adiante através da prole. Nessa analogia, cada indivíduo do AG é, de fato, uma solução candidata para o problema em questão, que por sua vez, faz o papel do próprio meio ambiente, já que estabelece os critérios que permitem avaliar se um indivíduo/solução é adaptado ou não (da Motta Salles Barreto, 2003).

1.1 Formulação do Problema

Na comparação feita neste trabalho utiliza-se um sistema de reconhecimento de voz que através de técnicas de otimização bioinspiradas tenta-se diminuir a confusão intrínseca dos modelos gerados para representar a voz. O objetivo deste trabalho é identificar qual dos algoritmos bioinspirados tem melhor desempenho na otimização dos modelos treinados que serão utilizados na etapa de reconhecimento.

2 Metodologia

Nesta proposta, um sinal de voz é codificado em uma matriz temporal bidimensional com o objetivo de modelar o sinal de voz com o número reduzido de parâmetros. Após a codificação das matrizes temporais bidimensionais dos modelos que gerarão o padrão, são calculadas as médias e as variâncias de cada modelo gerando assim, uma nova matriz temporal bidimensional de parâmetros gaussianos para representar o padrão a ser utilizado no reconhecimento. Essas matrizes temporais bidimensionais representativas dos padrões serão utilizadas para gerar a base de regras de um sistema de inferência fuzzy Mamdani. Essa base de regras será otimizada através do Enxame de Partículas e Algoritmo Genético, cujo objetivo é maximizar o desempenho do sistema proposto para o reconhecimento. Este artigo utiliza como padrão para o reconhecimento, locuções portuguesa do Brasil dos dígitos '0', '1', '2', '3', '4', '5', '6', '7', '8', '9'. A Transformada Cosseno Discreta (TCD) (Ahmed and Rao, 1974; Zhou, 2009) é utilizada para codificar os sinais de voz após o pré-processamento, pois o uso da TCD em compressão de dados e classificação de padrões aumentou em anos recentes, principalmente, devido ao fato de seu desempenho aproximarse dos resultados obtidos pela transformada de Karhunen-Loève que é considerada ótima para uma variedade de critérios, tais como, o erro médio quadrático de truncamento e a entropia (Fu, 1968; Hua and Liu, 1998; Effros et al., 2004). Este artigo explora o potencial do Enxame de Partículas, da TCD e de Sistema de Inferência Fuzzy na modelagem de padrões de sinais de voz com um número reduzido de parâmetros (Zeng and Liu, 2006; Mahmood and Farbod, 2008), e a viabilidade do uso de tais ferramentas em Sistemas de Reconhecimento de Sinal de Voz.

3 Sistema de Reconhecimento de Comando de Voz

3.1 Pré-processamento do sinal de Voz

O pré-processamento do sinal de voz é realizado em três etapas: retirada do silêncio inicial e final, alinhamento temporal e janelamento. A retirada do silêncio foi realizada através da análise da energia média deslizante, taxa de cruzamento aplicada a derivada do sinal e análise da porcentagem de sinal vocalizado, como o foco principal do artigo é a otimização este programa não será abordado de forma mais detalhada. O alinhamento temporal foi feito através da técnica de DTW (Dinamic Time Warping), técnica que procura o melhor caminho para o alinhamento de duas seqüências com tempos diferentes. O algoritmo DTW funciona da seguinte forma, têm-se duas séries temporais A e B de tamanhos n e m respectivamente, onde $A = [a_1, \dots, a_n]$ e $B = [b_1, \dots, b_m]$ gera-se uma matriz $n \times m$ onde todo elemento (i, j) da matriz contém a distância $d(a_i, b_j)$ entre os pontos a_i e b_j . Cada elemento (i, j) corresponde ao alinhamento entre os pontos a_i e b_j . Isso é ilustrado na figura 1. Uma trajetória W é um conjunto contíguo de elementos da matriz que define um mapeamento entre A e B. Um elemento de k em W é definido como $w_k = (i, j)_k$. A trajetória W está sujeita a algumas restrições:

a) Limites: $w_1 = (1, 1)$ e $w_k = (m, n)$, isso requer que a trajetória comece e termine em pontos diagonalmente opostos da matriz.

b) Continuidade: Sendo $w_k = (a, b)$ então $w_{(k-1)} = (a, b)$ onde $a - a \leq 1$ e $b - b \leq 1$, isso restringe os possíveis passos da trajetória para células adjacentes.

c) Monotonicidade: Sendo $w_k = (a, b)$ então $w_{(k-1)} = (a, b)$ onde $a - a \leq 0$ e $b - b \leq 0$, isso força os pontos de W a serem monotonicamente espaçados no tempo.

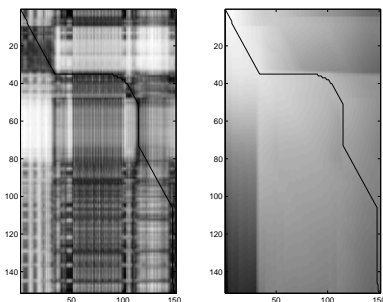


Figura 1: A matriz W corresponde a matriz de custo.

Existem várias trajetórias que satisfazem as condições acima (limites, continuidade e monotonicidade), entretanto, só interessa a trajetória que minimize o custo da função:

$$DTW(A, B) = \min \left\{ \sum_{k=1}^K W_k \right\} \quad (1)$$

Esta trajetória pode ser encontrada utilizando programação dinâmica para buscar os próximos elementos do conjunto W . Assim, a distância cumulativa definida pela programação dinâmica $\gamma(i, j)$ é o mínimo das distâncias cumulativas das células adjacentes, que podem ser expressas como (Jeske, 2011) :

$$\gamma(i, j) = d(a_i, b_j) + \min \{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \} \quad (2)$$

A última etapa é o processo de janelamento, que tem como objetivo selecionar uma pequena parcela deste sinal, a qual será analisada, denominada segmento. A análise de Fourier de curto-prazo efetuada sobre esses segmentos, é chamada análise de sinal segmento por segmento (Picone, 1991). A duração do segmento T_f é definida como a extensão de tempo sobre o qual um conjunto de parâmetros é considerado válido. O período do segmento é utilizado para

determinar a extensão de tempo entre os cálculos de sucessivos parâmetros. Para processamento de voz, tipicamente, o período de segmento está entre 10ms e 30ms. Valores nesta faixa representam um compromisso entre a razão de mudança do espectro e a complexidade do sistema. Devido ao fato de nas extremidades das janelas o sinal analisado sofrer um amortecimento excessivo em suas amostras adotou-se o processo denominado sobreposição para evitar este amortecimento e controlar quão rapidamente os parâmetros do sinal podem mudar de segmento para segmento. Assim, a cada novo segmento apenas uma fração do sinal irá mudar.

3.2 Geração da matriz temporal bidimensional-TCD

Depois de devidamente parametrizado em coeficientes mel-cepstrais, o sinal será codificado pela TCD realizada em uma seqüência de T vetores de observação dos coeficientes mel-cepstrais no eixo do tempo. A codificação pelo TCD é dada pela equação:

$$C_k(n, T) = \frac{1}{N} \sum_{t=1}^T mfcc_k(t) \cos \frac{(2t-1)n\pi}{2T} \quad (3)$$

onde $k, 1 \leq k \leq K$, refere-se a k -ésima linha (nº de coeficientes mel-cepstrais), componente do t -ésimo segmento da matriz e $n, 1 \leq n \leq N$, refere-se a n -ésima coluna (ordem da TCD), $mfcc_k(t)$ representa os coeficientes mel-cepstrais. Dessa forma, obtém-se a matriz de duas dimensões, onde o interesse está nos coeficientes de baixa ordem de k e n que codificam as variações de longo prazo do envelope espectral do sinal de voz (Fissore and Rivera, 1997). Este procedimento é realizado para cada palavra falada. Assim, tem-se uma matriz bidimensional $C_k(n, T)$ para cada sinal de entrada. Os elementos da matriz são obtidos da seguinte forma:

1. Para um dado modelo de uma palavra falada P (dígito), dez exemplos deste modelo são pronunciados. Cada exemplo é devidamente segmentado em T segmentos distribuídos ao longo do eixo do tempo. Deste modo têm-se: $P_0^0, P_1^0, \dots, P_9^0, P_0^1, P_1^1, \dots, P_9^1, P_0^2, P_1^2, \dots, P_9^2, \dots, P_m^j$, where $j=0,1,2,\dots,9$ and $m=0,1,2,\dots,9$.
2. Cada segmento de um dado exemplo do modelo P gera um total de K coeficientes mel-cepstrais e então são obtidas características significantes dentro de cada segmento ao longo do tempo. A TCD de ordem N é então calculada para cada coeficiente mel-cepstral de mesma ordem dentro do segmento, isto é, c_1 do segmento t_1 , c_1 do segmento t_2, \dots, c_1 do segmento t_T , c_2 do segmento t_1 , c_2 do segmento t_2, \dots, c_2 do segmento t_T , e assim por diante, gerando os elementos $\{c_{11}, c_{12}, c_{13}, \dots, c_{1N}\}$, $\{c_{21}, c_{22}, c_{23}, \dots, c_{2N}\}$, $\{c_{K1}, c_{K2}, c_{K3}, \dots, c_{KN}\}$, e a matriz na equação (3). Desta forma, uma matriz temporal bidimensional TCD é gerada para cada exemplo do modelo P , representada por C_{kn}^j . Finalmente, matrizes de média CM_{kn}^j (4) e variância CV_{kn}^j (5) são geradas. Os parâmetros de CM_{kn}^j e CV_{kn}^j são utilizados para gerar matrizes com parâmetros gaussianos C_{kn}^j que serão utilizados como funções de pertinências para a implementação da inferência fuzzy para o reconhecimento. Estes parâmetros serão otimizados pelo Enxame de Partículas.

$$CM_{kn}^j = \frac{1}{M} \sum_{m=0}^{M-1} C_{kn}^{j,m} \quad (4)$$

$$CV_{kn}^j = \frac{1}{M-1} \sum_{m=0}^{M-1} \left[C_{kn}^{j,m} - \left(\frac{1}{M} \sum_{m=0}^{M-1} C_{kn}^{j,m} \right) \right]^2 \quad (5)$$

3.3 Sistema de Inferência Fuzzy

Uma base de regras fuzzy Ru consiste de um conjunto de regras fuzzy $SE - ENTÃO$. Inferências em um sistema fuzzy baseado em regras é um processo no qual se obtém um conjunto de saída fuzzy dado um conjunto de entrada fuzzy. Em um sistema de inferência fuzzy, o princípio da lógica fuzzy relaciona as regras $SE - ENTÃO$ para uma base de regras Ru em um mapeamento de um conjunto fuzzy A' em U para um conjunto fuzzy B' em V , e as relações das regras $SE - ENTÃO$ são interpretadas como um produto dentro do espaço $U \times V$. Se a base de regra consiste de uma simples regra, então pode-se utilizar o *Modus Ponens* generalizado para especificar um mapeamento do conjunto A' em U para o conjunto B' em V com a seguinte forma (Wang, 1994):

$$SE \ x \text{ é } A \ \text{ENTÃO} \ y \text{ é } B \quad (6)$$

$$SE \ x \text{ é } A' \ \text{ENTÃO} \ y \text{ é } B' \quad (7)$$

Dados um conjunto fuzzy de entrada A' , um conjunto fuzzy de saída B' pode-se obter uma matriz de composição relacional max-t (Mas et al., 2007), esta relação é dada por:

$$B' = A' \circ Ru \quad (8)$$

onde \circ é algum operador relacional. A base de regras de sistemas práticos, usualmente, consiste de mais de uma regra. Há dois modos para inferir um conjunto de regras: Inferência baseada em Composição e Inferência baseada em regras individuais (Wang, 1994; Gang., 2010). Neste artigo utilizou-se a Inferência baseada em composição. De modo geral, uma base de regras fuzzy é dada por:

$$Ru^l : SE \ x_1 \text{ é } A_1^l \ \dots \ x_n \text{ é } A_n^l \ \text{ENTÃO} \ y \text{ é } B^l \quad (9)$$

onde A_i^l e B^l são conjuntos fuzzy em $U_i \subset \mathbb{R}$ e $V \subset \mathbb{R}$, e $x = (x_1, x_2, \dots, x_n)^T \in U$ e $y \in V$ são variáveis de entrada e saída do sistema fuzzy, respectivamente e M é o número de regras em uma base de regras fuzzy; isto é, $l = 1, 2, \dots, M$.

3.3.1 Base de Regras do Sistema de Reconhecimento de Voz

Os coeficientes das matrizes C_{kn}^j com $j = 0, 1, 2, \dots, 9$, $k = 1, 2$ e $n = 1, 2$, geradas durante o processo de treinamento, são utilizados para gerar a base de regras com $M = 40$ regras individuais definidas por:

$$Ru^j : SE \ C_{kn}^j \ \text{ENTÃO} \ y^j \quad (10)$$

O processo de treinamento é baseado na relação fuzzy Ru^j usando-se a implicação Mamdani. A base de regras Ru^j pode ser considerada uma relação $R(X \times Y) \rightarrow [0, 1]$, determinada por:

$$\mu_{Ru}(x, y) = I(\mu_A(x), \mu_B(y)) \quad (11)$$

onde o operador I representa qualquer norma-t (Babuska, 1998; Seki et al., 2010). Dado um conjunto fuzzy de entrada A' , o conjunto fuzzy de saída B' pode ser obtido pela composição relacional *max-min* (Wang, 1994). Para uma norma-t (mínimo) e uma composição relacional max-min tem-se:

$$\mu_{(Ru)}(x, y) = I(\mu_A(x), \mu_B(y)) = \min(\mu_A(x), \mu_B(y)) \quad (12)$$

$$\mu_{(B')} = \max_x \min_{x,y}(\mu_{A'}(x), \mu_{(Ru)}(x, y)) \quad (13)$$

3.4 Sistema de Inferência Fuzzy para a Decisão do Reconhecimento de Voz

A fase de decisão é realizada por um sistema de inferência fuzzy Mamdani baseado no conjunto de regras obtido das matrizes de média e da variância de cada dígito falado, uma vez que a Implicação Mandani possui características de formulação do conhecimento de maneira sistemática, através da interpretação de regras em linguagem natural, fato que

se assemelha com o raciocínio humano nos processos de classificação (Wang, 1994; Babuska, 1998). Neste artigo optou-se por uma matriz com número reduzido de parâmetros (2×2) e que ainda permitisse desempenho de reconhecimento satisfatório quando comparado com o método HMM convencional. Os elementos das matrizes C_{kn}^j são utilizadas por um sistema de inferência fuzzy para gerar quatro funções gaussianas que serão fuzzificadas para a entrada do sistema de fuzzy e correspondem a cada elemento $C_{kn}^j |_{k=1,2;n=1,2}$ da matriz. O conjunto de regras da relação fuzzy é dado por:

$$SE \ C_{kn}^j |_{k=1,2;n=1,2} \ \text{ENTÃO} \ y^j \quad (14)$$

$$SE \ C_{kn}^j |_{k=1,2;n=1,2} \ \text{ENTÃO} \ y'^j \quad (15)$$

Do conjunto de regras e da relação fuzzy entre antecedente e consequente é obtida uma matriz de dados para uma dada implicação. Depois do processo de treinamento, a superfície relacional é gerada através do método de implicação apresentado na seção 2.3. O sinal de voz é codificado para ser reconhecido e seus parâmetros são avaliados em relação às funções de pertinências de cada padrão na superfície e o grau de pertinência obtido. A decisão final para o padrão é obtida de acordo com a composição *max-min* entre os parâmetros de entrada e os dados contidos na superfície relacional. O processo de desfuzzificação para o processo de reconhecimento é baseado no método *mean of maxima (mom)* dado por:

$$y' = mom(\mu_{y'^j}) = mean\{y | \mu_{y'^j} = \max_{y \in Y}(\mu_{y'^j})\} \quad (16)$$

onde

$$\mu_{y'^j} = \mu_{C_{kn}^j} \circ \mu(Ru) \quad (17)$$

Diagrama em blocos do processo de reconhecimento é mostrado na Fig.2.

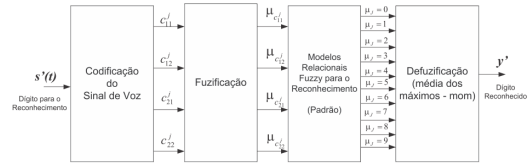


Figura 2: Diagrama em blocos do processo de reconhecimento.

4 Otimização da Superfície Relacional com Enxame de Partículas

O algoritmo Enxame de Partículas (Particle Swarm Optimization - PSO) é um meta-algoritmo de otimização local, pois ele tenta otimizar um conjunto de valores, mas encontra frequentemente máximos locais em vez de um máximo global (Coppin, 2010). Motivado pelo comportamento social através da competição e da cooperação entre indivíduos, assim como na natureza, pode trazer diversos benefícios, encontrando soluções boas de forma eficiente, mantendo certa simplicidade no processo de otimização (T. Silveira and da Silva, n.d.). A PSO emprega uma população de partículas para realizar buscas dentro de um espaço multidimensional restrito. Neste espaço de busca, em um determinado intervalo de tempo, cada partícula ocupa uma posição específica e movimenta-se com uma velocidade, a qual é modificada baseada na experiência adquirida pela própria partícula e na experiência das outras partículas do grupo (enxame). A experiência de cada partícula inclui informações relevantes da sua trajetória no espaço de busca, de forma a armazenar a melhor posição já ocupada pela própria partícula. Os elementos básicos da técnica PSO são (R. Rabêlo and Coury, n.d.): Posição da partícula j ($X_j(t)$), População $Pop(t)$, Velocidade da

partícula $j(V_j(t))$, Função de avaliação $f(X(t))$, Melhor posição prévia da partícula j até o instante de tempo $t(X^*(t))$, Parâmetro cognitivo c_1 , Parâmetro social c_2 , Número de interações e Peso de inércia $w(t)$.

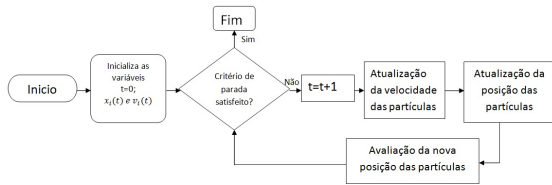


Figura 3: Algoritmo PSO

O movimento da partícula é influenciado por dois fatores: a melhor solução encontrada até aquele momento pela partícula ($pbest$) e a melhor solução encontrada considerando todas as partículas ($gbest$). Estes fatores, designados componentes cognitivos e sociais, influenciam na locomoção das partículas pelo espaço de busca criando forças de atração. Como resultado, as partículas interagem com todos os seus vizinhos e armazenam em sua memória a localização das soluções ótimas. Após cada interação $pbest$ e $gbest$ são atualizados se uma solução melhor for encontrada pela partícula ou pela população, respectivamente. Este processo se repete até que o resultado desejado seja obtido ou um determinado número de interações seja alcançado (G. Chiachia and Marana, n.d.). O algoritmo para implementar o PSO segue os passos descritos na Figura 3 (R. Rabêlo and Coury, n.d.). Os valores escolhidos para os parâmetros c_1 , c_2 e $cons$ foram os utilizados usualmente na literatura, conforme mostra a tabela 1 (L. Valéria, 2011).

Tabela 1: Parâmetros do PSO

Tamanho da população	100
Números de interações	100
c_1 parâmetro cognitivo	1.5
c_2 parâmetro social	2
w_{min}	0.4
w_{max}	1.2
cons	0.2

O algoritmo básico da Otimização por Exame de Partículas sugere que as partículas, espalhadas no espaço de busca, aproximem-se do ponto ótimo desse espaço, com o peso de inércia w (fator que influencia a movimentação da partícula no espaço) decaindo monotonicamente. Como esse peso somente sofre decaimento (somente perde energia), caso haja um estacionamento das partículas em um mínimo local, não haverá meio de sair desse ponto, pois a partícula não se acelera (T. Silveira and de Oliveira, n.d.), apresentando problemas para se aproximar da solução ótima. Para resolver este problema foi acrescentado o controle de inércia não monotônico. No método tradicional o peso de inércia w decresce linearmente de 0.4 a 0.9, com o controle de inércia o seu valor oscila segundo a função cosseno entre 0.4 a 1.2 e é dado pela equação abaixo, em que i é a interação atual e I é o número total de interações:

$$w = \left[\cos\left(\frac{\pi i}{2I}\right) m \right] + s \quad (18)$$

onde

$$m = \frac{(w_{max} + w_{min})}{2} \quad (19)$$

e

$$s = m + w_{min} \quad (20)$$

Com o valor do peso de inércia assumindo esse comportamento não monotônico, a otimização ocorreria da seguinte forma: quando o peso de inércia da PSO reduz, tendendo

ao seu valor mínimo, as partículas perdem energia, se estabilizando em torno de um ponto de mínimo. Se este peso de inércia recebe valores maiores que 1, a tendência é a velocidade da partícula no instante anterior possuir maior importância, podendo causar um espalhamento do enxame. Se o peso de inércia atingir novamente valores menores, a tendência será o enxame se aproximar novamente, efetuando uma busca em torno de um mínimo local. Este mínimo, não será necessariamente igual ao primeiro. Portanto, o espalhamento do enxame pode ajudar o processo da PSO a fugir de mínimos locais, pois propicia uma busca global em um momento no qual seria menos provável encontrar outra melhor solução. Com isso, a inércia w pode atuar na diversificação e intensificação da PSO. O número de oscilações da função que controla a inércia ao longo da PSO é o número de tentativas de fugas de mínimos locais, já que a cada vez que w ultrapassar 1, o enxame pode se espalhar (T. Silveira and Salgado, 2009). Outro avanço adicionado ao PSO foi a turbulência atmosférica no cenário computacional, que utiliza um novo espaço de busca para superar o problema da estagnação da exploração das partículas, possibilitando que a partícula continue a mover-se e a manter a diversidade da população até a convergência do algoritmo (Serapião, 2009). Essa nova variável aproxima o ambiente virtual criado em computador com o ambiente real, fonte de inspiração original do algoritmo, interferindo de maneira independente, aleatória e esporádica a trajetória que cada partícula efetua durante o seu voo. A inclusão da turbulência em cada velocidade relativa a uma dimensão de voo da partícula é regida pela equação abaixo, que determina a intensidade de aplicação.

$$iturb = (w + c_1 + c_2)G(x_{best})\frac{1}{T - t} \quad (21)$$

Sendo $G(x_{best})$ o ajuste, ou *fitness* da posição da melhor partícula para o problema, T o total de interações reservado para a execução do algoritmo e t a interação atual. Após a determinação da intensidade da turbulência, $iturb$, a sua aplicação a cada velocidade relativa à dimensão de voo da partícula é feita independentemente, $v_{id} = v_{id}.iturb.rand_i$

5 Otimização da Superfície Relacional com Algoritmo Genético

O algoritmo genético usado neste artigo é mostrado na Fig.4 (Haupt and Haupt, 2004; Zhou and Khotand, 2007). O algoritmo genético contínuo foi configurado com uma população de tamanho igual a 100, 300 gerações, com probabilidade de mutação igual a 15% e dois cromossomos com quarenta genes em cada um, para otimizar a função custo com 80 variáveis (médias e variâncias de cada padrão) para o processo de reconhecimento do sistema fuzzy proposto. O algoritmo genético foi utilizado para otimizar as variações de média e variância de cada padrão com intuito de se obter o melhor desempenho do sistema de reconhecimento. Por exemplo, para a geração do padrão da palavra "zero" são geradas dez matrizes temporais bidimensionais. Cada elemento da matriz C_{kn}^j estará, para o algoritmo genético, dentro de uma faixa determinada pelas variações máxima e mínima de cada coeficiente. Estas variações são obtidas pelas diferenças inerentes aos modelos, assim, os coeficientes serão especificados da seguinte forma: $c_{11} \in [c_{11}(minimum) \ c_{11}(maximum)]$, $c_{12} \in [c_{12}(minimum) \ c_{12}(maximum)]$, $c_{21} \in [c_{21}(minimum) \ c_{21}(maximum)]$, $c_{22} \in [c_{22}(minimum) \ c_{22}(maximum)]$. Observa-se que para um único padrão são gerados oito parâmetros variantes no tempo que correspondem aos oito parâmetros que serão otimizados pelo AG, (Tang et al., 1998; Weihong et al., 2010).

6 Resultados

Os padrões que foram utilizados no processo de reconhecimento foram obtidos de dez locutores que pronunciaram

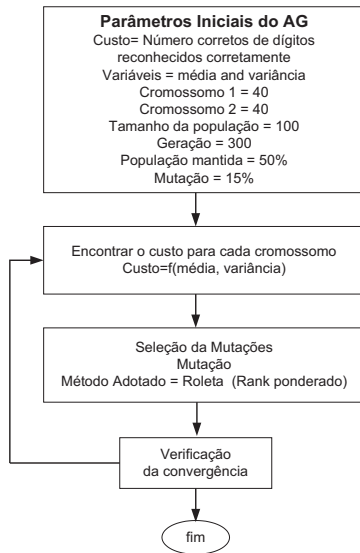


Figura 4: Fluxograma do AG utilizado.

os dígitos: zero até nove. Depois de realizado o pré-processamento do sinal de voz, a codificação e a geração da matriz temporal bidimensional C_{kn}^j , os modelos foram fuzzificados gerando a matriz C_{kn}^j que representa os coeficientes de entrada do sistema fuzzy de reconhecimento a serem otimizados pelo PSO e pelo AG. Para o processo de otimização foram implementadas 16 realizações do AG e 7 realizações do PSO, cujos resultados são mostrados na Fig.5 e na Tab.2.

Tabela 2: Resultados dos 7 testes do PSO

c1	c2	cons	peso	turb	var	sil	DTW	GBEST
1	3	1	x	x	1.5x	x	x	29%
1	2	1	x	x	6x	x	x	58%
1.5	2	0.2	x	x	6x	SIM	SIM	74%
1.5	2	0.2	x	x	6x	SIM	SIM	75%
1.5	2	0.2	SIM	x	6x	SIM	SIM	74%
1.5	2	0.2	SIM	SIM	6x	SIM	SIM	80%
1.5	2	0.2	SIM	SIM	6x	SIM	SIM	82%
1.5	2	0.2	SIM	SIM	6x	SIM	SIM	83%

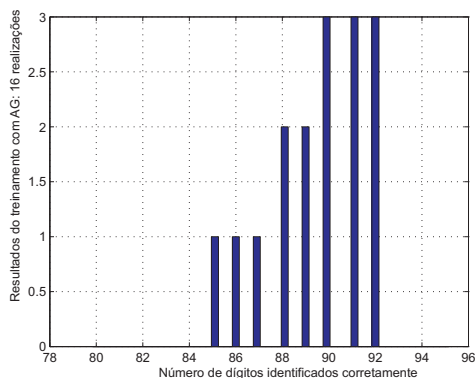


Figura 5: Histograma dos resultados para 16 realizações do processo de treinamento do AG.

O melhor indivíduo obtido processo de treinamento do AG e do PSO respectivamente, relacionado à base de regras do sistema de inferência fuzzy para reconhecimento do comando de voz, é mostrado na Fig.7 e na Fig.6. O desempenho total usando-se o AG foi de 92 dígitos identificados corretamente no processo de treinamento e do PSO 83 dígitos.

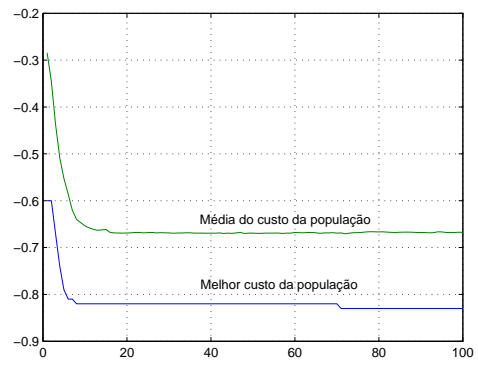


Figura 6: Melhor indivíduo do PSO.

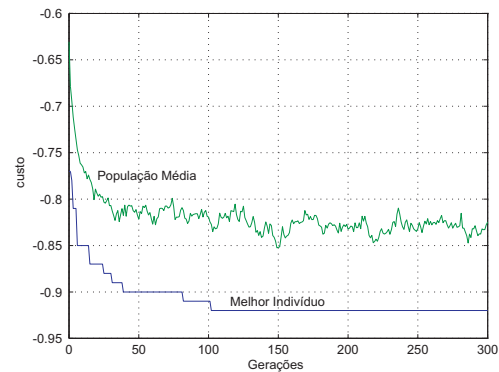


Figura 7: Melhor indivíduo do AG.

7 Conclusão

Analisando-se os testes realizados com o AG e o PSO proposto neste artigo, observou-se que o melhor resultado atingido foi com o AG, que atingiu 92 acertos contra 83 do PSO, entretanto vale ressaltar que o algoritmo PSO proposto tem poucas modificações entre tantas que ainda podem ser feitas, futuramente pretende-se conseguir uma diferença mínima, se possível nula entre as taxas de acertos das duas técnicas e chegar-se também mais próximo do número total de acertos da função.

Agradecimentos

Os autores gostariam de agradecer ao Programa Jovens Talentos para a Ciência da Capes, pelo suporte financeiro a esta pesquisa, ao Grupo de Instrumentação Eletrônica Aplicada a Tecnologia do Instituto Federal de Educação, Ciência e Tecnologia do Maranhão pela infraestrutura utilizada no desenvolvimento desta pesquisa e ao Programa de Pós-Graduação da Universidade Federal do Maranhão - UFMA.

Referências

- Ahmed, T. N. N. and Rao, K. (1974). *Discrete Cosine Transform*, vol.c-24 edn, IEEE Transaction on Computers.
- Babuska, R. (1998). *Fuzzy Modeling for Control*, Kluwer Academic Publishers.
- Coppin, B. (2010). *Inteligência Artificial*, Tradução e Revisão Técnica de Jorge Duarte Pires Valério, LTC, Brasil, Rio de Janeiro.

- da Luz, E. (2008). *Estimação de Fonte de Poluição Atmosférica Usando Otimização por Enxame de Partículas*, Instituto Nacional de Pesquisas Espaciais, Tese de Mestrado, Brasil, São José dos Campos.
- da Motta Salles Barreto, A. (2003). *Algoritmo Genético dos Mínimos Quadrados Ortogonal para o Treinamento de Redes RBF*, Tese submetida para a obtenção do título de Mestre em Ciências em Engenharia Civil ao Programa de Pós-Graduação de Engenharia Civil da COPPE/UFRJ, Brasil, Rio de Janeiro.
- Effros, M., Feng, H. and Zeger, K. (2004). *Suboptimality of the Karhunen Loeve Transform for Transform Coding*, vol.50, n.08 edn, IEEE Transactions on Information Theory.
- Fissore, P. L. L. and Rivera, E. (1997). *Using word temporal structure in HMM Speech recognition*, vol.02 edn, ICASSP 97.
- Fu, K. (1968). *Sequential Methods in Pattern Recognition and Machine Learning*, Academic Press, New York.
- G. Chiachia, B. P. and Marana, A. (n.d.). *Fusão de Métodos de Reconhecimento Facial Através da Otimização por Enxame de Partículas*, UNESP - Faculdade de Ciências, Brasil, Bauru - São Paulo.
- Gang., C. (2010). *Discussion of Approximation Properties of Minimum Inference Fuzzy System*, Proceedings of the 29th Chinese Control Conference.
- Haupt, R. L. and Haupt, S. E. (2004). *Practical Genetic Algorithms*, John Wiley & Sons, Inc.
- Hua, Y. and Liu, W. (1998). *Generalized Karhunen Loeve Transform*, vol.05, n.06 edn, IEEE Signal Processing Letters.
- Jeske, J. (2011). *Similiaridade de Series Temporais na Bolsa de Valores*, Universidade Federal do Rio Grande do Sul, Trabalho de Conclusão Apresentado como Requisito Parcial para Obtenção do grau de Bacharel em Ciências da Computação, Brasil, Porto Alegre.
- Kennedy, J. and Eberhart, R. (n.d.). *Particle Swarm Optimization*, IEEE International Conference on Neural Networks, Vol.4, DOI:10.1109/ICNN.1995.488968.1995, Pag. 1942 - 1948, Austrália.
- L. Valéria, L. Iesfer, H. L. (2011). *Identificação de Parâmetros por Enxame de Partículas Através do Compromisso entre Variância e Polarização*, X Simpósio Brasileiro de Automação Inteligente, Brasil, São João Del Rei - MG.
- Mahmood, Y. A. and Farbod, R. (2008). *A DCT Based Nonlinear Predictive Coding for Feature Extraction in Speech Recognition Systems*, 2008-IEEE International Conference on Computational Intelligence for Measurement Systems and Applications.
- Mas, M., Monserrat, M., Torrens, J. and Trillas, E. (2007). *A Survey on Fuzzy Implication Functions*, vol.15, n.7 edn, IEEE Transactions on Fuzzy Systems.
- Picone, J. (1991). *Signal modeling techniques in speech recognition*, vol.79, 2ed. edn, IEEE Transactions on Computer.
- R. Rabêlo, D. Barbosa, I. d. S. M. O. and Coury, D. (n.d.). *Uma Estimação do PSO na Qualidade da Energia Elétrica*, Universidade Salvador, Mestrado em Energia, Brasil, Salvador - Bahia.
- Seki, H., Ishii, H. and Mizumoto, M. (2010). *On the Monotonicity of Fuzzy Inference Methods Related to TS Inference Method*, vol.18, n.3 edn, IEEE Transactions on Fuzzy Systems.
- Serapião, A. (2009). *Fundamentos de Otimização por Inteligência de Enxames: Uma visão geral*, Revista Controle & automação, Vol.20, n.º03, Brasil: Sociedade Brasileira de Automática.
- T. Silveira, H. d. O. and da Silva, L. (n.d.). *Controle de Inércia para Fuga de Mínimos Locais de Funções Não-Lineares na Otimização por Enxame de Partículas*, Universidade Federal de Alfenas, Brasil, Minas Gerais.
- T. Silveira, H. de Oliveira, L. d. S. and Salgado, R. (2009). *Controle de Inércia Não Monotônico na Otimização por Enxame de Partículas*, Interdisciplinary Studies in Computer Science 69-82.
- T. Silveira, V. M. and de Oliveira, H. (n.d.). *Controle de Inércia Não-Monotônico para Fuga de Mínimos Locais na Otimização por Enxame de Partículas*, Universidade Federal de Alfenas, Brasil, Minas Gerais.
- Tang, K.-s., Man, K.-f., Liu, Z.-f. and Kwong, S. (1998). *Minimal Fuzzy Memberships and Rules Using Hierarchical Genetic Algorithms*, vol.45, n.01 edn, IEEE Transactions on Industrial Electronics.
- Wang, L. (1994). *A course in Fuzzy Systems and Control*, Prentice Hall.
- Weihong, Z., Shunqing, X. and Ting, M. (2010). *A Fuzzy Classifier Based on Mamdani Fuzzy Logic System and Genetic Algorithm*, Information Computing and Telecommunications (YC-ICT), 2010 IEEE Youth Conference on.
- Zeng, J. and Liu, Z.-Q. (2006). *Type-2 Fuzzy Hidden Markov Models and Their Application to Speech Recognition*, vol.14, n.03 edn, IEEE Transactions on Fuzzy Systems.
- Zhou, E. and Khotand, A. (2007). *Fuzzy Classifier Design Genetic Algorithms*, Elsevier.
- Zhou, P. C. J. (2009). *Generalized Discrete Cosine Transform*, Pacific-Asia Conference on Circuits, Communications and System.