

# Otimização Bio-inspirada Utilizando Enxame de Partículas com Aplicação em Reconhecimento de Voz

Amanda Abelardo Viana Beserra  
Instituto Federal de Educação Ciência  
e Tecnologia do Maranhão  
Av. Getúlio Vargas, nº 04 - Monte Castelo  
São Luís, Maranhão, Brasil 65030-005  
Email:amandaabelardo@hotmail.com

Ginalber L.O. Serra  
Instituto Federal de Educação Ciência  
e Tecnologia do Maranhão  
Av. Getúlio Vargas, nº 04 - Monte Castelo  
São Luís, Maranhão, Brasil 65030-005  
Email:ginalber@ifma.edu.br

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  
Email:washington.wlss@ifma.edu.br

**Abstract**—Este artigo propõe a otimização não linear de um Classificador de Padrões para Reconhecimento de Voz utilizando a técnica de Enxame de Partículas. O Classificador utiliza um sistema de inferência fuzzy para realizar a decisão do reconhecimento. O pré-processamento do sinal de voz inicia-se com a retirada do silêncio com o algoritmo de taxa de cruzamento por zero e energia média deslizante e alinha-se temporalmente os sinais de voz utilizando-se a ‘Dynamic Time Warping’. Em seguida calcula-se os coeficientes mel-cepstrais. Após o pré-processamento utiliza-se a Transformada Cosseno Discreta (TCD) para gerar uma matriz temporal bidimensional. Essa matriz será utilizada como entrada do sistema de implicação fuzzy para gerar os padrões das palavras a serem reconhecidas. Finalmente são mostrados os resultados experimentais obtidos no processo de otimização para o reconhecimento de voz aplicado ao idioma português brasileiro.

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

## I. INTRODUÇÃO

Nas últimas décadas vários algoritmos bio-inspirados baseados em populações e meta-heurísticas vêm sendo usados para resolver problemas de busca e otimização em vários domínios de problemas, para os quais soluções robustas são difíceis ou impossíveis de se obter usando abordagens tradicionais. Esses algoritmos são inspirados em mecanismos biológicos da evolução. Um algoritmo baseado neste princípio, o qual será abordado neste artigo, é o Enxame de Partículas, ou do original “*Particle Swarm Optimization (PSO)*”. O PSO é um método de busca heurística inspirado pelo comportamento colaborativo e individual de populações biológicas [4]. Este método foi introduzido por James Kennedy e Russel Elberhart em 1995 [8] para tratar problemas no domínio contínuo. Uma teoria sócio-cognitiva está por trás da PSO. Cada indivíduo de uma população possui sua própria experiência, que o capacita estimar a qualidade dessa experiência. 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. Portanto, a probabilidade de um determinado indivíduo decidir será uma função de seu conhecimento adquirido e do conhecimento de alguns de vizinhos [1]. Partindo-se deste princípio é definido um espaço de busca, em que as partículas tem sua velocidade e posição próprias, que a cada iteração é atualizada levando-se em conta uma série de variáveis individuais e coletivas. Uma das vantagens desse método é a pequena quantidade de

memória utilizada por cada partícula e a alta velocidade de convergência em comparação às outras técnicas de otimização não linear bio-inspirada.

## A. Formulação do Problema

No sistema de reconhecimento de voz proposto neste artigo tenta-se, através de técnicas não lineares de otimização, diminuir a confusão intrínseca dos modelos gerados para representar a voz. Encontrar uma métrica de separabilidade entre os padrões de forma a otimizar o reconhecimento é o objetivo da otimização PSO usada neste trabalho.

## II. METODOLOGIA

Nesta proposta, um sinal de voz é codificado em um 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 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) [16], [22] 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 aproximar-se 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 [11], [14], [25]. 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 [10], [24], e a viabilidade do uso de tais ferramentas em Sistemas de Reconhecimento de Sinal de Voz.

### III. SISTEMA DE RECONHECIMENTO DE COMANDO DE VOZ

#### A. 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 (Dynamic Time Warping), técnica que procura o melhor caminho para o alinhamento de duas sequências 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 cada 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.

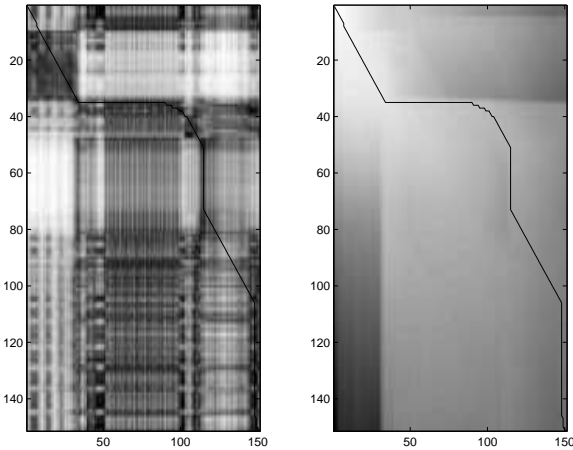


Figure 1. A matriz  $W$  corresponde a matriz de custo.

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-se 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 [7] :

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

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 [9]. 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.

#### B. Geração da matriz temporal bidimensional-TCD

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

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

onde  $k, 1 \leq k \leq K$ , refere-se a  $k$ -ésima linha ( $n^\circ$  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),  $m f c c_k(t)$  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 [17]. 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  $P$  (digito), dez exemplos deste modelo são pronunciados. Cada exemplo é devidamente segmentado em  $T$  partes distribuídas 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  $\mathbf{P}$ , representada por  $C_{kn}^{jm}$ . 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.

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

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

### C. Sistema de Inferência Fuzzy

Uma base de regras fuzzy  $\mathbf{Ru}$  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  $\mathbf{Ru}$  em um mapeamento de um conjunto fuzzy  $A'$  em  $\mathbf{U}$  para um conjunto fuzzy  $B'$  em  $\mathbf{V}$ , e as relações das regras  $SE - ENTÃO$  são interpretadas como um produto dentro do espaço  $\mathbf{U} \times \mathbf{V}$ . 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  $\mathbf{U}$  para o conjunto  $B'$  em  $\mathbf{V}$  com a seguinte forma [13]:

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

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

Dados um conjunto fuzzy de entrada  $A'$ , um conjunto fuzzy de saída  $B'$  pode-se obter uma matriz de composição relacional max-t [15], 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 [3], [13]. Neste artigo utilizou-se a Inferência baseada em composição. De modo geral, uma base de regras fuzzy é dada por:

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

onde  $A_i^l$  e  $B^l$  são conjuntos fuzzy em  $U_i \subset \mathfrak{R}$  e  $V \subset \mathfrak{R}$ , 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,  $l = 1, 2, \dots, M$ .

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 [6], [18]. Dado um conjunto fuzzy de entrada  $A'$ , o conjunto fuzzy de saída  $B'$  pode ser obtido pela composição relacional *max-min* [13]. Para uma norma-t (mínimo) e uma composição relacional max-min tem-se:

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

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

### D. 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 [13], [18]. Neste artigo optou-se por uma matriz com número reduzido de parâmetros ( $2 \times 2$ ) e que ainda permitesse 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:

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

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

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 defuzzificação para o processo de reconhecimento é baseado no método *mean of maxima (mom)* dado por:

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

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.



Figure 2. Diagrama em blocos do processo de reconhecimento.

#### IV. 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 mínimos locais em vez de um mínimo global [2]. 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 [20]. 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 que é modificada com base 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 modo a armazenar a melhor posição já ocupada pela própria partícula. Os elementos básicos da técnica PSO são definidos a seguir [19]:

- 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 iterações;
- Peso de inércia  $w(t)$ .

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

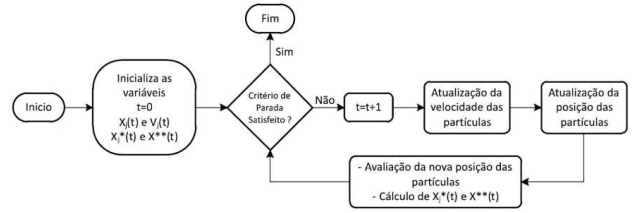


Figure 3. Algoritmo PSO;

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 iterações seja alcançado [5]. O algoritmo para implementar o PSO segue os passos descritos na Figura 3 [19]. Os valores escolhidos para os parâmetros  $c_1$ ,  $c_2$  e  $cons$  foram os utilizados usualmente na literatura, conforme mostra a tabela 1 [12].

Table I. PARÂMETROS DO PSO

Tamanho da população	100
Números de iterações	100
$c_1$ parâmetro cognitivo	1.5
$c_2$ parâmetro social	2
$w_{min}$	0.4
$w_{max}$	1.2
Fator de ponderação - 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 [21], 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 iterações:

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

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 [23]. 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 [1]. 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 na trajetória que cada partícula efetua durante o seu vôo. A inclusão da turbulência em cada velocidade relativa a uma dimensão de vôo 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 iterações 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 vôo da partícula é feita independentemente,  $v_{id} = v_{id}.iturb.rand_i$

## V. RESULTADOS

Os padrões que foram utilizados no processo de reconhecimento foram obtidos de dez locutores que pronunciaram, dez vezes cada um, 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. Para o processo de otimização foram implementadas 6 realizações do PSO, cujos resultados e características dos testes são mostrados na Tabela II, o item Var refere-se a quantidade de vezes que a velocidade pode variar.

Table II. :RESULTADOS

$c_1$	$c_2$	cons	Cont. peso	Turb.	Var.	Ret. Silêncio	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	83%

O melhor indivíduo resultante do processo de treinamento pelo PSO, relacionado à base de regras do sistema de inferência *fuzzy* para reconhecimento do comando de voz, é mostrado na figura ???. O desempenho total usando-se o PSO foi de 83 dígitos identificados corretamente no processo de

treinamento. Na figura ?? é mostrado o gráfico do teste feito sem a adição de turbulência, obteve-se o máximo de 79 acertos.

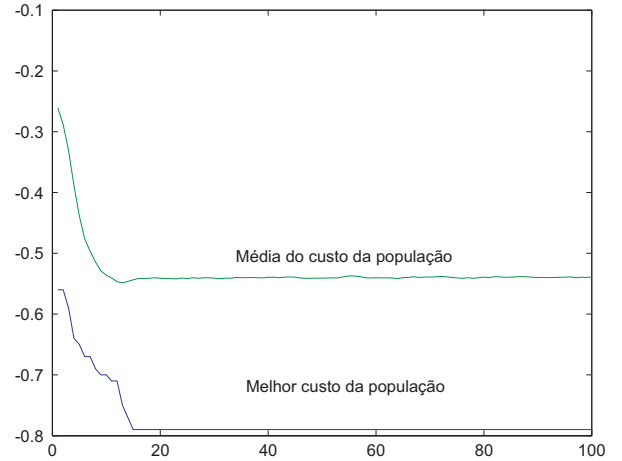


Figure 4. Resultado para o PSO sem turbulência.

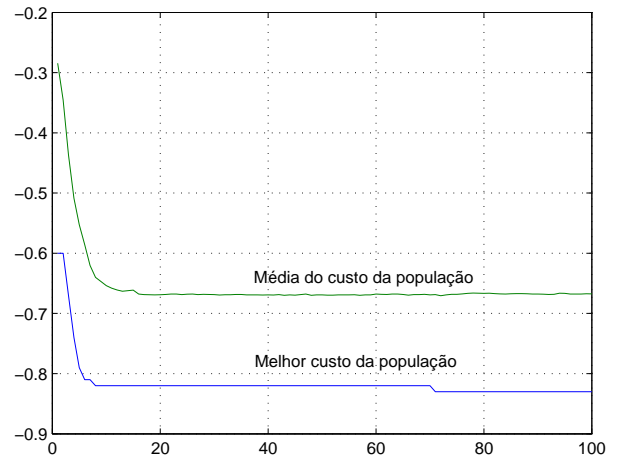


Figure 5. Melhor Resultado para o PSO.

## VI. CONCLUSÃO

Analisando-se os testes realizados com o PSO proposto neste artigo, observou-se que o melhor resultado atingido foi no ultimo teste, em que todos os recursos estudados e aplicados foram utilizados: Retirada do Silêncio, Alinhamento temporal através de DTW, controle do peso inercial e turbulência. Com essas alterações no PSO notou-se também que o algoritmo convergiu de forma mais rápida e eficaz que o PSO tradicional. Como proposta futura tem-se a inclusão de outros parâmetros e mudança na topologia, a fim de levar a PSO a conseguir melhores resultados para o mínimo global.

## AGRADECIMENTOS

Os autores gostariam de agradecer ao Programa Jovens Talentos para a Ciência da Capes, pelo suporte financeiro a esta pesquisa e 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.

## REFERENCES

- [1] A. Serapião, *Fundamentos de Otimização por Inteligência de Enxames: Uma visão geral*, Revista Controle & automação, Vol.20,nº03 ,Brasil: Sociedade Brasileira de Automática, 2009.
- [2] B. Coppin, *Inteligência Artificial*, Tradução e Revisão Técnica de Jorge Duarte Pires Valério, LTC ,Brasil: Rio de Janeiro, 2010.
- [3] CHEN Gang., *Discussion of Approximation Properties of Minimum Inference Fuzzy System*, Proceedings of the 29th Chinese Control Conference, Vol.15, Nº06 ,July, 2010.
- [4] E. da Luz, *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, 2008.
- [5] G. Chiachia, B. Penteado and A. Marana *Fusão de Métodos de Reconhecimento Facial Através da Otimização por Enxame de Partículas*,UNESP - Faculdade de Ciências ,Brasil: Bauru - São Paulo.
- [6] H. Seki, H. Ishii and M. Mizumoto,*On the Monotonicity of Fuzzy Inference Methods Related to TS Inference Method*,IEEE Transactions on Fuzzy Systems, Vol.18, Nº 03 ,June, 2003.
- [7] J. Jeske,*Similaridade de Series Temporaisna 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, 2011.
- [8] J. Kennedy and R. Eberhart, *Particle Swarm Optimization*.IEEE International Conference on Neural Networks.Vol.4. DOI:10.1109/ICNN.1995.488968.1995.Pag.1942 - 1948. Australia.
- [9] J.W. Picone, *Fundamentos de Otimização por Inteligência de Enxames: Uma visão geral*, IEEE Transactions on Computer, Vol.79, 2ed. , April, 1991.
- [10] J. Zeng and Z. Liu, *Type-2 Fuzzy Hidden Markov Models and Their Application to Speech Recognition*,IEEE Transactions on Fuzzy Systems, Vol.14, Nº 03 , June, 2003.
- [11] K. Fu, *Sequential Methods in Pattern Recognition and Machine Learning*, Acadmic Press ,New York: April, 1968.
- [12] L. Valéria, L. Iesfer, H. Lopes, *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, 2011.
- [13] L.X. Wang, *A course in Fuzzy Systems and Control*, Prentice Hall, Vol.1. , 1994.
- [14] M. Effros, H. Feng and K. Zeger, *Suboptimality of the Karhunen Loeve Transform for Transform Coding*, IEEE Transactions on Information Theory, Vol.50, Nº 08 , August, 2004.
- [15] M. Mas and M. Monserrat and J. Torrens and E. Trillas, *A Survey on Fuzzy Implication Functions*, IEEE Transactions on Fuzzy Systems, Vol.15, Nº 06 , December, 2007.
- [16] P. C. Jianqin Zhou, *Generalized Discrete Cosine Transform*, Pacific-Asia Conference on Circuits, Communications and System , May, 2009.
- [17] P. L. L. Fissore and E. Rivera, *Using word temporal structure in HMM Speech recognition*, ICASSP 97 , April, 1997.
- [18] R. Babuska, *Fuzzy Modeling for Control*, Kluwer Academic Publishers, Vol.15, Nº 06 , July, 1998.
- [19] R. Rabêlo, D. Barbosa, I. da Silva, M. Oleskovicz and D. Coury *Uma Estimação do PSO na Qualidade da Energia Elétrica*,Universidade Salvador, Mestrado em Energia ,Brasil, Salvador - Bahia.
- [20] T. Silveira, H. de Oliveira and L. da Silva, *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.
- [21] T. Silveira, V. Miranda and H. de Oliveira, *Controle de Inércia Não-Monotônico para Fuga de Mínimos Locais na Otimização por Enxame de Partículas*, Universidade Federal de Alfenas , Brasil, Minas Gerais.
- [22] T. N. N. Ahmed and K. Rao, *Discrete Cosine Transform*, IEEE Transaction on Computers, Vol.c-24 , January, 1974.
- [23] T. Silveira, H. de Oliveira, L. da Silva and R. Salgado *Controle de Inércia Não Monotônico na Otimização por Enxame de Partículas*,Interdisciplinary Studies in Computer Science 69-82. , July/December, 2009.
- [24] Y. Azar Mahmood and R. Farbod, *A DCT Based Nonlinear Predictive Coding for Feature Extraction in Speech Recognition Systems*, Kluwer Academic Publishers , July, 2008.
- [25] Y. Hua and W. Liu, *Generalized Karhunen Loeve Transform*, IEEE Signal Processing Letters, Vol.5, Nº 06 , June, 1998.