

USO DE ALGORITMOS GENÉTICOS NA OTIMIZAÇÃO DE UM SISTEMA DE IDENTIFICAÇÃO DE COMANDOS DE VOZ

Juan C. Pérez Ibarra, jcperezibarra@sc.usp.br¹
Henry Borrero Guerrero, h_borrelo@ieee.org¹
Andrés Eduardo Baquero Velásquez, andresbaquero@sc.usp.br¹
Adriano A. G. Siqueira, siqueira@sc.usp.br¹

¹Escola de Engenharia de São Carlos - Universidade de São Paulo,
Av. Trabalhador São-carlense, 400, Pq Arnold Schimdt, São Carlos, S. P., Brasil

Resumo. Este artigo descreve e compara os resultados do uso de um Algoritmo Genético como técnica de otimização num sistema para a identificação de comandos de voz. O sistema busca comparar duas sequências de vetores que contém a informação espectral de cada comando (codificação LPC). Para fazer que o sistema seja indiferente às possíveis variações que possa apresentar cada comando de voz, as sequencias são normalizadas e alinhadas no tempo. Este alinhamento e normalização envolve o uso de funções de distorção que determinam a distância total acumulada entre os dois comandos. Escolher as “melhores” funções equivale a determinar a distância acumulada mínima entre os dois comandos de voz é implica selecionar entre um grande número de funções de distorção, virando o analise num problema de minimização. Para resolver este problema foi usado um algoritmo genético. O artigo descreve os passos para a implementação de este algoritmo, incluindo o analise dos diferentes parâmetros que são usados no mesmo.

Palavras Chave: Comparação de padrões, Algoritmos Genéticos, Otimização Moderna.

1. INTRODUÇÃO

Um sistema de reconhecimento de voz é propriamente um sistema para o reconhecimento de padrões na voz. Como tal, pode utilizar diferentes abordagens e métodos para identificar tais padrões. No entanto, o sistema de reconhecimento geralmente consiste dos três processos seguintes: primeiro, um processo de extração de características, que utiliza um modelo estadístico para representar o sinal de voz com a mínima quantidade de bits que permitam manter a informação das características relevantes para a identificação; segundo, um processo de classificação de padrões, no qual são comparados os padrões de voz adquiridos com um conjunto de padrões de referência baseando-se em uma medida estabelecida da similitude (distância) entre diferentes sequencias de voz e finalmente, um processo de tomada de decisões para definir o padrão correspondente ao som adquirido. (Rabiner & Juang, 1993)

Em (Perez Ibarra & Borrero, 2010) foi desenvolvido um sistema de identificação de comandos de voz que contém os três processos mencionados antes. Neste sistema, como resultado do processo de extração cada comando de voz é codificado em uma sequência de vetores de predição linear (LPC) que contém a informação espectral e temporal do comando. O processo de classificação de padrões procura comparar a sequência espectral de um padrão “objetivo” com as sequências espectrais de um conjunto de padrões “referência” obtidas previamente em uma fase de “treinamento” do sistema. A distância total acumulada entre os dois comandos é definida como a somatória das distâncias locais ao longo das sequências. Mas, dada a existência de possíveis diferenças de comprimento entre as sequências, assim como entre a distribuição temporal dos sons nas mesmas, tem que se fazer um alinhamento e normalização temporal entre a sequência “objetivo” e cada sequência de referência.

O alinhamento temporal e normalização envolve o uso de pares de funções não lineares de distorção que determinam os índices ao longo dos quais será feita a somatória. Escolher o “melhor” par de funções implica selecionar entre um grande número de funções de distorção virando o analise num problema de minimização. Este documento apresenta um algoritmo genético para melhorar a capacidade de busca global do processo de alinhamento temporal e normalização. São apresentadas a descrição, implementação e o analise dos resultados obtidos com este método. Um trabalho similar é apresentado em (Kwong, et al., 1996)

Este documento está organizado da forma seguinte: a Seção I se apresentam os conceitos gerais relacionados com o processo de classificação de padrões. Na seção II se mostram de forma geral os aspectos relacionados com a implementação do algoritmo de otimização. Na seção III se mostra o analise dos resultados obtidos com cada algoritmo, incluindo o analise de cada parâmetro do mesmo. Por último, são apresentadas as conclusões.

2. CLASSIFICAÇÃO DE PADRÕES DE VOZ

Inicialmente, é definido o padrão que vai ser avaliado como uma concatenação de frames espectrais \mathcal{X} , onde:

$$\mathcal{X} = \{x_1, x_2, x_3, \dots, x_{T_x}\},$$

sendo cada x_i corresponde ao vetor espectral do sinal de voz no instante de tempo i , e T_x é o comprimento do padrão. Também, é definido um conjunto de padrões de referência $\{R^1, R^2, R^3, \dots, R^V\}$ onde cada padrão R^j é também uma sequência de frames espectrais,

$$R^j = \{r_1^j, r_2^j, r_3^j, \dots, r_f^j\},$$

O objetivo do sistema é determinar uma medida da distância entre T e cada R^j para logo identificar o padrão de referência com a menor medida, e por último, associar este ao sinal de voz obtida.

2.1 Medição da distância entre dois padrões

Se $\mathcal{X} = [x_1, x_2, x_3 \dots x_{T_x}]$ é o padrão de um comando de voz para identificar e $R^1 = \mathcal{Y} = [y_1, y_2, y_3 \dots y_{T_y}]$ é o padrão correspondente a um fonema de referência, sendo x_i e y_i os índices correspondentes aos i -ésimos vetores de parâmetros dos padrões objetivo y referência, respectivamente, e T_x e T_y são os comprimentos dos mesmos. A distância entre \mathcal{X} e \mathcal{Y} é definida pela somatória de distâncias locais apresentada na Eq. (1).

$$d(\mathcal{X}, \mathcal{Y}) = \sum_{i_x=1}^{\max(T_x, T_y)} d(x_{i_x}, y_{i_y}), \quad (1)$$

sendo $i_x = \{1, 2, \dots, T_x\}$ e $i_y = \{1, 2, \dots, T_y\}$ os índices que determinam os vetores das sequências \mathcal{X} e \mathcal{Y} para os quais serão calculadas as distâncias locais. Para o nosso sistema a medida local de distância entre dois vetores espectrais x_i e y_i , é definida mediante a Eq. (2).

$$d(x_{i_x}, y_{i_y}) = d(x_i, y_i) = \sum_{n=1}^p (x_i[n] - y_i[n])^2. \quad (2)$$

2.2 Alinhamento temporal e normalização

Normalmente as sequências de voz têm diferenças em seu comprimento e em sua distribuição temporal dos sons. Então, o valor de $d(\mathcal{X}, \mathcal{Y})$ deve ser uma medida subjetivamente significativa, matematicamente coerente e indiferente às flutuações da velocidade da fala. Para alcançar isto, é preciso normalizar tais flutuações e alinhar as sequências \mathcal{X} e \mathcal{Y} no domínio do tempo de modo que a ordem dos sons em cada expressão seja mantido. Para o nosso sistema, o cálculo da distância entre duas sequências é dado pela Eq. (3) e utiliza duas funções de “distorção” $\varphi_x(k)$ e $\varphi_y(k)$ que relacionam os índices i dos dois padrões com um índice comum $k = 1, 2, \dots, T$, permitindo assim a normalização e o alinhamento necessários para determinar a distorção acumulada ao final da expressão a avaliar.

$$d_\varphi(\mathcal{X}, \mathcal{Y}) = \frac{1}{T} \sum_{k=1}^T d(\varphi_x(k), \varphi_y(k)), \quad \text{com } i_x = \varphi_x(k) \quad i_y = \varphi_y(k), \quad (3)$$

sendo $d(\varphi_x(k), \varphi_y(k))$ a medida da distância local entre os vetores indexados pelas funções de distorção.

Uma abordagem útil para entender o problema é apresentar as funções como se fossem “caminhos” ao longo de um plano (i_x, i_y) formado pelas distâncias locais entre as duas expressões para comparar. Assim, escolher o “melhor caminho” equivale a escolher o melhor par de funções de distorção. É comum estabelecer que o melhor caminho é aquele que tem o menor valor de distância acumulada obedecendo um conjunto de restrições dadas. Evidentemente o número de funções de distorção possíveis aumenta exponencialmente com o comprimento dos sinais de voz, então, avaliar um grande número de pares de funções para escolher a função com o menor valor de distância pode precisar de uma grande quantidade de cálculos.

Aqui, este é abordado como um problema de minimização sujeita a restrições que é representado na Eq. (4).

$$d(\mathcal{X}, \mathcal{Y}) \triangleq \min_{\varphi} d_\varphi(\mathcal{X}, \mathcal{Y}). \quad (4)$$

2.3 Restrições de normalização

A normalização temporal das diferentes expressões de um comando de voz precisa de estabelecer um conjunto de restrições para as funções de distorção descritas na seção anterior. Comumente são consideradas necessárias para um adequado alinhamento e normalização temporal as restrições de ponto inicial e final, as condições de monotonia, as restrições de continuidade local e as restrições de caminho global, as quais são descritas nas seguintes subseções.

2.3.1 Restrições de ponto inicial e final

Os limites fixos no tempo das expressões são os pontos inicial e final das funções de distorção, e isto é estabelecido nas restrições mediante a Eq. (5).

$$\varphi_x(1) = 1, \varphi_y(1) = 1, \varphi_x(T) = T_x, \varphi_y(T) = T_y. \quad (5)$$

2.3.2 Condições de monotonia

Para conservar a ordem temporal das sequências espectrais, a Eq. (6) impõe uma condição de monotonia para as funções de distorção.

$$\varphi_x(k+1) \geq \varphi_x(k), \varphi_y(k+1) \geq \varphi_y(k). \quad (6)$$

Esta restrição elimina a possibilidade de aparecer funções de distorção com pendentes negativas, o que significa que a função “caminharia” volta atrás no alinhamento temporal.

2.3.3 Restrições de continuidade local

Para diminuir ao mínimo a possibilidade de omitir segmentos significativos do sinal de voz, é incorporado um conjunto de restrições de continuidade local para as funções de distorção. Estas restrições podem tomar diferentes formas, e neste documento determinam os passos possíveis que podem se seguir ao longo de cada caminho. Então, para completar a analogia, pode se definir um caminho P como uma sequência de passos que equivalem a pares de incrementos nos índices i_x e i_y das sequências X e Y :

$$P \rightarrow (p_1, q_1)(p_1, q_1)(p_1, q_1) \dots (p_T, q_T).$$

Assim, as funções de distorção definidas na Eq. (3) são expressadas de novo na Eq. (7).

$$\varphi_x(k) = \sum_{i=1}^k p_i, \quad \varphi_y(k) = \sum_{i=1}^k q_i, \quad (7)$$

Os passos podem ser, como exemplo das restrições de cont. local, $P1 \rightarrow (1,1)(0,1)$, $P2 \rightarrow (1,0)$, $P3 \rightarrow (0,1)(1,0)$,

Então $C = \{(1,1)(0,1)(1,0)(0,1)(1,0)\}$ seria o caminho percorrido pela sequência de passos $P1, P2, P3$; e, dado que a restrição de ponto inicial exige que a posição inicial seja $(1,1)$, o par de funções de distorção ficam $\varphi_x = [1,2,3,4]$ e $\varphi_y = [1,3,3,4]$, determinando que seja nas posições $[(1,1)(2,3)(3,3)(4,4)]$ onde se realize o cálculo de distância.

Dada a dificuldade de fazer a modelagem de todas as possíveis variações temporais de uma sequência de voz, as restrições de continuidade local são geralmente determinadas heurísticamente. Em (Rabiner & Juang, 1993) é apresentado um conjunto de restrições locais usadas, e conforme com sua terminologia, aqui será utilizada a restrição TIPO III.

$$\text{Restrição TIPO III} \quad P1 \rightarrow (2, 1) \quad P2 \rightarrow (1, 1) \quad P3 \rightarrow (1, 2)$$

2.3.4 Restrições de caminho global

Devido às restrições de continuidade local, algumas posições no plano (i_x, i_y) não podem-se alcançar nos traçados dos caminhos e por tanto são excluídas das possíveis soluções ótimas. Assim, é gerada uma nova restrição que determina o espaço de pontos no plano que podem se alcançar para cada caminho que une o ponto inicial $(1, 1)$ com o ponto final (T_x, T_y) . A Eq. (8) apresenta a inequação que determina o espaço de pontos que podem ser alcançados saindo do ponto $(1,1)$ e a Eq. (9) apresenta a inequação que determina o espaço de pontos que permitem chegar ao ponto final (T_x, T_y) .

$$1 + \frac{[\varphi_x(k) - 1]}{Q_{max}} \leq \varphi_y(k) \leq 1 + Q_{max}[\varphi_x(k) - 1]. \quad (8)$$

$$T_y + Q_{max}[\varphi_x(k) - T_x] \leq \varphi_y(k) \leq T_y + \frac{[\varphi_x(k) - T_x]}{Q_{max}}. \quad (9)$$

sendo que o parâmetro Q_{max} indica a máxima expansão possível que pode dar cada passo conforme à restrição de continuidade local selecionada.

3. ALGORITMO GENÉTICO

O problema apresentado neste documento consiste no problema de minimização descrito nas Eq. (2-4) e submetido às restrições presentes nas Eq. (5-9), como técnica de otimização para a sua resolução foi usado um algoritmo genético. O algoritmo implementado começa com uma população inicial de caminhos gerados aleatoriamente, onde cada caminho (indivíduo) está representado como uma sequência de um alfabeto finito (neste caso, o alfabeto consta de [1 2 3] segundo o passo dado desde cada ponto no caminho). Depois cada caminho é submetido a uma função de avaliação (neste caso, a distorção acumulada ao longo do caminho) (Rao, 2009). Como terceiro passo é feita a geração de sucessores onde a probabilidade de ser elegido é inversamente proporcional ao valor resultante na função de avaliação. Aqui são usados um conjunto de operadores genéticos encargados de gerar uma nova população “melhor” da que a anterior. Isto é feito até conseguir uma medida apropriada para algum caminho, o até realizar um determinado número de iterações (gerações) do algoritmo (Russell & Norvig, 2010).

A continuação é descrito detalhadamente o algoritmo implementado:

1. Gerar um plano (i_x, i_y) mediante a Eq. (10) com as distâncias entre os vetores segundo a Eq. (2)

$$plano(i, j) = d(x_i, y_j) = \sum_{n=1}^p (x_i[n] - y_j[n])^2 \quad \text{Para } i = 1, 2, \dots, T_x \text{ e } j = 1, 2, \dots, T_y \quad (10)$$

2. Normalizar o valor dos planos.
3. Gerar a população inicial, para cada valor de $i = 1, 2, \dots, ps$ é gerado um novo indivíduo mediante o seguinte algoritmo:

3.1. Determinar o ponto inicial conforme à restrição expressada na Eq. (5)

3.2. Determinar as condições para ponto final conforme à restrição expressada na Eq. (5)

$$(i_x(k) = T_x) \vee (i_y(k) = T_y) \vee (k = T_x) \vee (k = T_x) \quad (11)$$

3.3. Gerar um vector de tamanho T com valores inteiros aleatórios entre 1, 2, 3 que correspondem aos passos P1, P2 e P3 indicados na restrição de continuidade local:

$$p(k) = rand\{1,2,3\} \text{ para } k = 1, 2, \dots, T \text{ onde } T = \min(T_x, T_y) \quad (12)$$

3.4. Com a informação do vector p , construir a função de distorção (caminho) correspondente aos índices i_x e i_y . Então dado que cada valor de $p(k)$ corresponde a um passo indicado na restrição de continuidade, o caminho é construído da forma seguinte:

$$\begin{aligned} \text{Para } p(k) = 1 & \quad \begin{cases} i_x(k+1) = i_x(k) + 2 \\ i_y(k+1) = i_y(k) + 1 \end{cases} \\ \text{Para } p(k) = 2 & \quad \begin{cases} i_x(k+1) = i_x(k) + 1 \\ i_y(k+1) = i_y(k) + 1 \end{cases} \\ \text{Para } p(k) = 3 & \quad \begin{cases} i_x(k+1) = i_x(k) + 1 \\ i_y(k+1) = i_y(k) + 2 \end{cases} \end{aligned} \quad (13)$$

3.5. Verificar que a geração do passo obedeça às restrições de caminho global expressadas nas Eq. (8-9). Já que $Q_{max} = 1/2$ para restrições Tipo III, as Eq. (8-9) podem ser rescrevidas nas Eq. (14-15).

$$2 \cdot \varphi_x(k) - 1 < \varphi_y(k) < (\varphi_x(k) + 1)/2 \quad (14)$$

$$T_y + (\varphi_x(k) - T_x)/2 < \varphi_y(k) < T_y + 2 \cdot (\varphi_x(k) - T_x) \quad (15)$$

3.6. Cada indivíduo (caminho) representa uma função de distorção que determina mediante os índices i_x e i_y onde vai se fazer o cálculo da distância total.

4. Avaliar a população atual

4.1. Avaliar de novo que o caminho se ajuste às restrições globais para cada valor de $k = 1, 2, \dots, T$. Assim, como um controle adicional, se algum passo dado em $p(k)$ não satisfaz as restrições é assignado para este caminho um $fi = \infty$, de modo que não seja escolhido para a seguinte geração.

4.2. Gerar um vetor D no qual são guardados os valores de distância local para $k = 1, 2, \dots, T$.

$$D(k) = \text{plano}(i_x(k), i_y(k)) \propto d(\mathcal{X}(i_x), \mathcal{Y}(i_y)) \quad (16)$$

4.3. Fazer a avaliação total do caminho conforme com a Eq. (3). A Eq. (17) calcula a distância total acumulada entre as sequências \mathcal{X} e \mathcal{Y} :

$$fi = \frac{1}{T} \sum_{k=1}^T D(k) \quad (17)$$

5. Aplicar o algoritmo *roulette wheel selection* que gera uma população nova de caminhos do modo seguinte:

5.1. Assignar pesos à roleta, a Eq. (18) calcula o peso de cada caminho.

$$f(i) = \frac{\text{mean}(fi)}{fi(i)} \quad (18)$$

5.2. Calcular para cada caminho seu valor de fitness acumulado, com o objetivo de lhe assignar um lugar na roleta.

$$fps(m) = fps(m-1) + fp(m), \quad \text{para } m = 1:ps \quad (19)$$

5.3. Para cada valor de $i = 1, 2, \dots, ps$ se gera um novo caminho, mediante os operadores recombinação e mutação:

5.3.1. O operador de recombinação é apresentado na Eq. (20), para fazer isto, são escolhidos aleatoriamente dois indivíduos $P1$ e $P2$ e um ponto de recombinação Pc .

$$np = [P1(1), P1(2), \dots, P1(pc), P2(pc+1), P2(pc+2), \dots, P2(k)] \quad (20)$$

5.3.2. O operador de mutação é apresentado na Eq. (21), para fazer isto, se define uma probabilidade de ocorrência Pm . No caso de acontecer, aleatoriamente é determinada uma posição pm , tal que o passo pm do caminho np seja modificado.

$$\begin{aligned} \text{Para } np(pm) = 1 &\rightarrow np(pm) = np(pm) + 1 \\ \text{Para } np(pm) = 2 &\rightarrow np(pm) = np(pm) \pm 1 \\ \text{Para } np(pm) = 3 &\rightarrow np(pm) = np(pm) - 1 \end{aligned} \quad (21)$$

5.3.3. Dado que o algoritmo *roulette wheel selection* gera uns novos caminhos onde foram modificados os passos, tem que se verificar que os caminhos cumpram as restrições estabelecidas.

6. Para determinar se finalizar as iterações, é avaliado se o valor de avaliação convergiu num valor adequado, ou porque a nova geração ultrapassou o valor máximo de g . Daí o caminho escolhido é aquele que tenha a menor valor de função de avaliação.

4. RESULTADOS

O algoritmo foi implementado no programa desenvolvido em (Perez Ibarra & Borrero, 2010) no entorno MATLAB® para o sistema de identificação de comando de voz. Para determinar a eficiência do algoritmo genético desenvolvido, foi realizada uma prova que consistia em comparar uma sequência correspondente a um comando de voz com duas sequências diferentes, a primeira delas correspondente a outra expressão do mesmo comando de voz e a segunda correspondente a um comando diferente com duração similar. Inicialmente foi executado o algoritmo com 250 indivíduos, durante 50 gerações, sem mutações de modo que a evolução da população fora apreciada melhor ao longo das iterações. Embora cada execução deste terá uma evolução diferente devido a que o algoritmo genético é meramente estocástico, é possível identificar tendências que permitem entre outras coisas determinar os valores adequados para os parâmetros do algoritmo. As Figuras 4-9 foram desenvolvidas no entorno MATLAB® e mostram o comportamento do valor de avaliação (fitness) contra o número de gerações, mostrando em magenta os valores de todos os caminhos gerados, as linhas azuis correspondem ao valor médio da população, e a linha vermelha corresponde ao mínimo valor de fitness.

4.1. Convergência da média para o valor mínimo

A Figura 4.a. mostra a evolução dos valores de avaliação para um algoritmo executado entre duas expressões de um mesmo comando de voz. A partir da informação presente na Figura podemos evidenciar que o valor mínimo é alcançado muito cedo pelo algoritmo (geralmente antes da geração 15), e que em diante o valor da media converge para tal valor mínimo, o que indica uma concentração de indivíduos em tal valor de avaliação. Este fato permite que em implementações práticas se estabeleça um número de gerações entre 20 e 25 já que é muito provável que o mínimo para o qual se convergirá já tinha sido gerado.

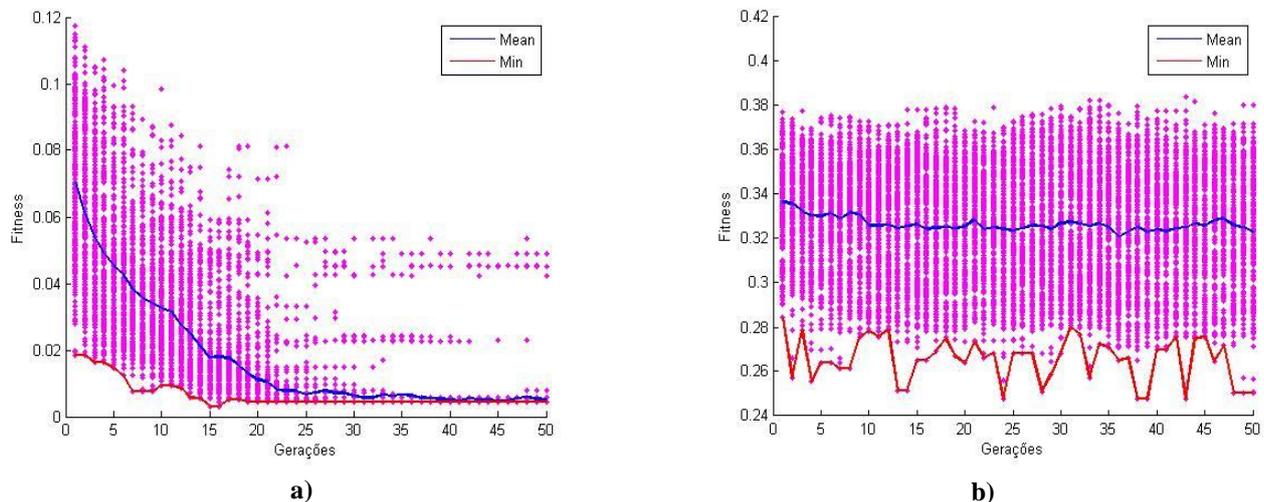


Figura 4. Evolução do valor de avaliação para duas expressões com $P_s = 250$ e $g = 50$
a) Com dois expressões de um mesmo comando b) Com dois comandos diferentes

4.2. Persistência de mínimos locais

No entanto, também se apresenta uma marcada tendência a convergir em mínimos locais, para o caso mostrado na Figura isto se evidencia em que o valor mínimo final no corresponde ao mínimo global, já que na geração 15 apareceu ao menos um indivíduo com menor valor de fitness que não foi conservado pelo algoritmo. Esta tendência pode ser consequência da “persistência” que possam apresentar alguns mínimos locais que “travam” o algoritmo num grupo de indivíduos cada vez mais pequeno. Embora, estes mínimos locais têm valores suficientemente baixos que permitem determinar si as duas sequências são ou não expressões de um mesmo comando. O uso do operador mutação dá ao algoritmo a variabilidade suficiente para sair destes “bloqueios”. (Ver resultado 6)

4.3. Ausência de convergência para comandos diferentes

A Figura 4.b. mostra em troca a evolução dos valores de avaliação para um algoritmo executado entre duas expressões correspondentes a dois comandos diferentes. A partir do comportamento da média na Figura podemos inferir que não existe convergência para nenhum caminho, o que é um indicador da falta de relação entre as duas sequências. Assim, embora ter valores mínimos, estes ficam muito mais altos que os achados no caso de expressões de um mesmo comando, de modo que o sistema de identificação pode determinar que os comandos são diferentes.

4.4. Efeito da redução no número de gerações

A Figura 5 mostra duas execuções do algoritmo genético para duas expressões de um mesmo comando. Na Figura 5 pode se observar que tal como foi falado no ponto 1 de esta seção, a convergência da média ao valor mínimo permite prever que na geração 25 já tem se alcançado o valor mínimo para o qual se convergirá. A Figura 6 mostra duas execuções neste caso para dois comandos diferentes, e nela pode se conferir que perto da geração 25 não se apresenta convergência nenhuma.

4.5. Efeito da redução no número de indivíduos

As análises anteriores foram feitas com uma população suficientemente grande para que o algoritmo tenha uma grande variabilidade que permitirá observar as tendências presentes. No entanto, nos usos práticos avaliar 250 indivíduos cada uma das 25 gerações por cada sequência a comparar, implica um custo computacional demasiado alto. Então, se estudo o efeito de diminuir o número de indivíduos sobre as tendências analisadas, a Figura 7 mostra as execuções do algoritmo para os casos antes estudados, estabelecendo agora o número de indivíduos em 25 e 50. Se observo que as tendências de convergência se conservavam ainda a grande diminuição de Ps.

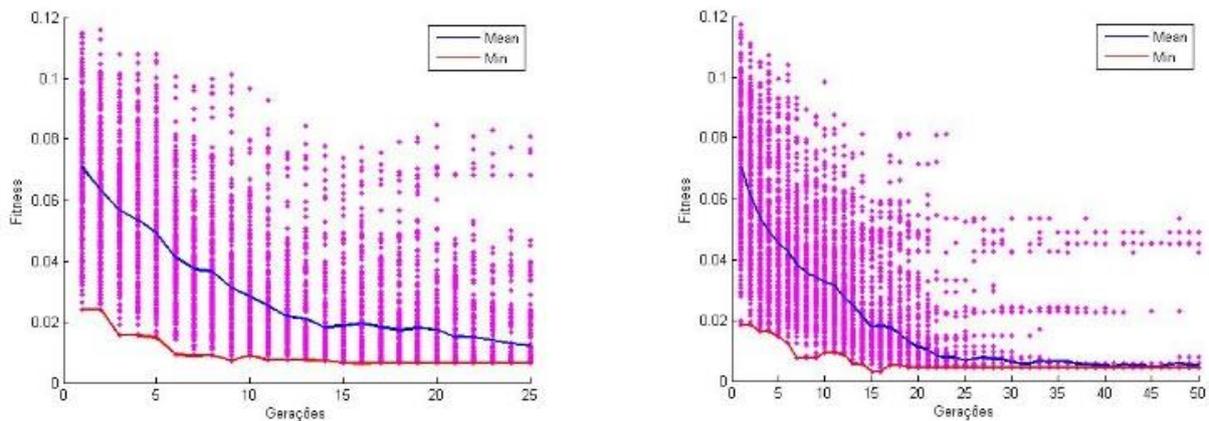


Figura 5. Evolução do valor de avaliação para duas expressões de um mesmo comando com $P_s = 250$ para a) $g = 25$ b) $g = 50$

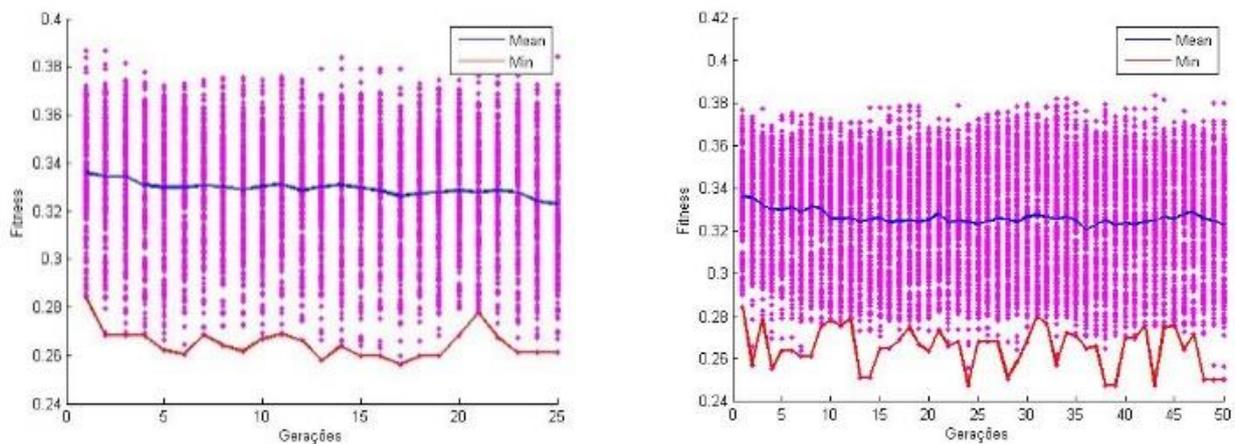


Figura 6. Evolução do valor de avaliação para dois comandos diferentes com $P_s = 250$ e a) $g = 25$ b) $g = 50$

4.6. Efeito do operador Mutação

O operador mutação é utilizado para gerar um indivíduo na vizinhança dos indivíduos já gerados, o que tem importantes consequências sobre o processo de otimização: primeiro se contribui na busca local perto do indivíduo modificado, segundo se aumenta a diversidade da população, e terceiro permite ao sistema “fugir” dos mínimos locais.

Os resultados para o uso deste operador são apresentados na Figura 8, como principal fato se evidencia que um major probabilidade de mutação ocasiona uma diminuição na taxa de convergência da média sobre o mínimo, isto devido a que a mutação compensa o fenómeno observado da persistência dos mínimos locais. Para o caso de duas sequências não correspondentes se observa que a presença do operador não produz variações sobre o comportamento da média.

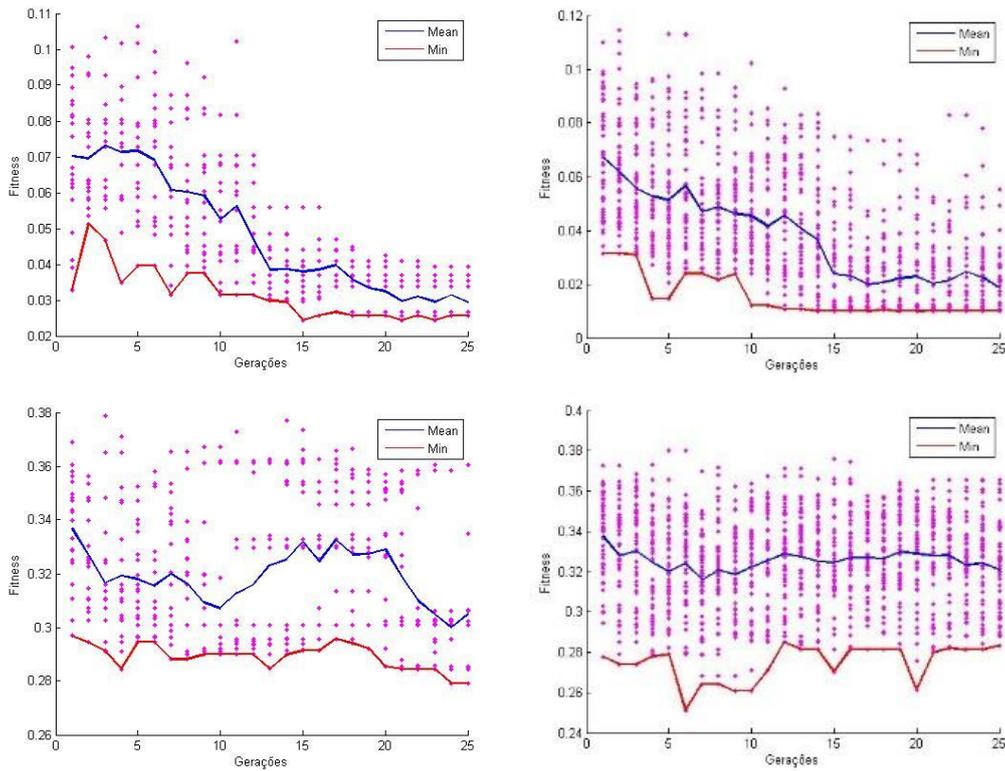


Figura 7. Evolução do valor de avaliação com $g = 25$ para a) duas expressões de um mesmo comando com a) $P_s = 20$ b) $P_s = 50$ e para dois comandos diferentes com c) $P_s = 20$ d). $P_s = 50$

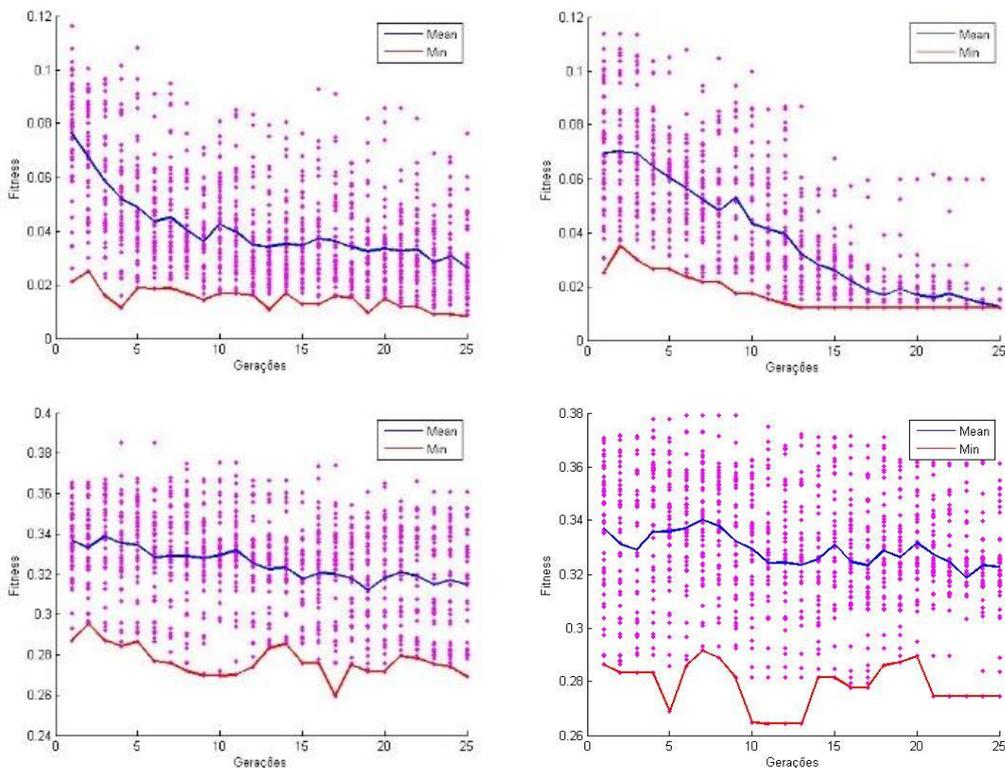


Figura 8. Evolução do valor de avaliação com $g = 25$ e $P_s = 50$ para um mesmo comando com a) $P_m = 0.5$ b) $P_m = 0.2$ e para dois comandos diferentes com c) $P_m = 0.5$ d) $P_s = 0.2$

5. REFERÊNCIAS

- Chou, W., 2003. "Pattern recognition in speech and language processing." S. I. Ed. CRC Press.
- Kwong, S., He, Q. H. & Man, K. F., 1996. "Genetic Time Warping for Isolated Word Recognition." International Journal of Pattern Recognition and Artificial Intelligence, Vol. 10, Núm. 07, pp. 849-865.
- Oppenheim, A., Schafer, R. & Buck, J., 1999. "Discrete-Time Signal Processing." 2da ed. Ed. Prentice Hall., New Jersey.
- Perez Ibarra, J. & Borrero, H., 2010. "Identificación de comandos de voz en MATLAB® utilizando LPC y algoritmos genéticos." Revista CINTEX, Vol. 15, p. 36.
- Rabiner, L. & Juang, B.-H., 1993. "Fundamentals of Speech Recognition." Ed. Prentice-Hall, Inc., New Jersey.
- Rao, S., 2009. "Engineering Optimization: Theory and Practice." 4 ed., S. I., Ed. John Wiley & Sons, Inc..
- Russell, S. & Norvig, P., 2010. "Artificial Intelligence, a Modern Approach." 3 ed., Ed. Pearson Education, Inc., New Jersey.

USE OF GENETIC ALGORITHMS IN OPTIMIZING A SPEECH COMMAND RECOGNITION SYSTEM

Juan C. Pérez Ibarra, jcperezibarra@sc.usp.br¹
Henry Borrero Guerrero, h_borrelo@ieee.org¹
Andrés Eduardo Baquero Velásquez, andresbaquero@sc.usp.br¹
Adriano A. G. Siqueira, siqueira@sc.usp.br¹

¹ Escola de Engenharia de São Carlos - Universidade de São Paulo,
Av. Trabalhador São-carlense, 400, Pq Arnold Schimdt, São Carlos, S. P., Brasil

Abstract. *This article describes and compares the results of use a Genetic Algorithm as optimization technique in a system for the speech commands identification. The system seeks to compare two sequences of vectors that containing the spectral information of each command (linear prediction coding). To make the system indifferent to the possible variations that may present each speech command, the signals are time aligned and normalized. Time alignment and normalization involve the use of pairs of distortion functions that determine the total distortion accumulated between the two commands. Choosing the "best" path equates to determine the function with minimal accumulated distortion, this has to determine the function with the lowest accumulated distortion value is means selecting among a large number of distortion functions turning the analysis in a minimization problem. To solve this problem was used the Genetic Algorithm. The article describes the steps for implementing this algorithm, including the analysis of the various parameters used in this.*

Key words: *Pattern comparison, Genetic Algorithm, Modern Optimization.*