

# UM ALGORITMO MEMÉTICO PARA A OTIMIZAÇÃO DE QUANTIZADORES VETORIAIS

**Carlos R. B. Azevedo, Renan A. Azevedo, Esdras L. Bispo Júnior**

Departamento de Estatística e Informática  
Universidade Católica de Pernambuco – Recife, PE, Brasil  
{azevedo,renanazevedo,bispojr}@dei.unicap.br

**Tiago A. E. Ferreira**

Departamento de Estatística e Informática  
Universidade Federal Rural de Pernambuco – Recife, PE, Brasil  
tiago@deinfo.ufrpe.br

**Waslon Terlizzie Araújo Lopes**

Departamento de Engenharia Elétrica  
ÁREA 1 – Faculdade de Ciência e Tecnologia – Salvador, BA, Brasil  
waslon.lopes@area1fte.edu.br

**Francisco Madeiro**

Departamento de Estatística e Informática  
Universidade Católica de Pernambuco – Recife, PE, Brasil  
madeiro@dei.unicap.br

**Resumo** – O presente artigo apresenta um Algoritmo Memético (AM) para a otimização de quantizadores vetoriais. O AM proposto atualiza os dicionários produzidos a cada geração de um Algoritmo Genético (AG) modificado mediante a aplicação de iterações de uma versão acelerada do algoritmo Linde-Buzo-Gray (LBG). A partir de simulações envolvendo compressão de imagens, observa-se que o AM proposto leva a melhores dicionários em uma quantidade menor de gerações, quando comparado a um AG integrado ao algoritmo LBG convencional, produzindo, portanto, melhores valores de relação sinal-ruído de pico para as imagens reconstruídas. Finalmente, a complexidade da busca pelo Vizinheiro Mais Próximo (VMP) no método proposto é reduzida mediante a utilização de uma busca por distância parcial (PDS).

**Palavras-Chave** – Quantização vetorial, computação evolutiva, algoritmos meméticos, codificação de imagens.

**Abstract** – This paper presents a Memetic Algorithm (MA) for optimizing vector quantizers. The proposed MA updates the codebooks produced at each generation of a modified Genetic Algorithm (GA) by applying iterations of an accelerated version of the Linde-Buzo-Gray (LBG) algorithm. From simulation results involving image compression, it is observed that the proposed MA leads to better codebooks, in a smaller number of generations, when compared to the GA guided by the standard LBG, leading to better peak signal-to-noise results for the reconstructed images. Finally, the complexity of the search for the nearest neighbour in the proposed MA is reduced by the usage of a Partial Distance Search (PDS).

**Keywords** – Vector quantization, evolutionary computing, memetic algorithms, image coding.

## 1. INTRODUÇÃO

Dentre as vantagens decorrentes da representação digital de sinais, destacam-se o controle eficiente de erros de canal, a aplicação de técnicas criptográficas e a possibilidade de integração de sinais e dados.

A codificação eficiente de sinais (voz, imagem, áudio, vídeo) é essencial para diversas aplicações, tais como, comunicação multimídia, navegação *web*, armazenamento de imagens médicas e documentos de negócios, arquivamento de impressões digitais, transmissão de imagens de sensoriamento remoto obtidas de satélites e comunicações móveis. Nessas aplicações, o propósito fundamental da compressão de sinais é reduzir o número de bits para representá-los adequadamente, de forma a minimizar os requisitos de memória, para armazenamento, e de largura de banda, para transmissão.

A Quantização Vetorial (QV) [1, 2], que pode ser vista como uma extensão da quantização escalar em um espaço multidimensional, encontra-se fundamentada na Teoria da Distorção Versus Taxa, formulada por Shannon, segundo a qual um melhor desempenho é obtido codificando blocos de amostras (isto é, vetores) em vez de amostras individuais (isto é, escalares).

A QV é definida como um mapeamento  $Q$  de um vetor de entrada  $x \in \mathbb{R}^k$ , em um subconjunto finito  $W \subset \mathbb{R}^k$  contendo  $N$  vetores de reprodução distintos. Assim,  $Q: \mathbb{R}^k \rightarrow W$ . O dicionário  $W = \{w_1, \dots, w_N\}$  é o conjunto de vetores-código (vetores de reconstrução),  $k$  é a dimensão dos vetores-código e  $N$  é o tamanho do dicionário. O mapeamento  $Q$  particiona o

espaço  $\mathbb{R}^k$  em  $N$  regiões disjuntas (também chamadas de células de Voronói)  $S_i$ ,  $i = 1, \dots, N$ , em que  $S_i = \{\mathbf{x} : Q(\mathbf{x}) = \mathbf{w}_i\}$ , ou, alternativamente,  $S_i = \{\mathbf{x} : d(\mathbf{x}, \mathbf{w}_i) \leq d(\mathbf{x}, \mathbf{w}_j), \forall j \neq i\}$ , em que  $d(\cdot, \cdot)$  é uma medida de distorção. O vetor-código  $\mathbf{w}_i$  é o vetor representante de todos os vetores de entrada pertencentes à célula  $S_i$  correspondente.

Em diversos sistemas de codificação de voz [3–5] e imagem [6–8], a QV tem sido utilizada com sucesso, permitindo elevadas taxas de compressão. Em [3], a QV é aplicada a vetores com dez parâmetros LSF (*line spectral frequencies*), que são divididos em duas partes, as quais são quantizadas separadamente – com uma quantização realizada com 24 bits/quadro, obtém-se uma distorção espectral média de 1 dB e menos de 2% dos quadros de voz com uma distorção superior a 2 dB. Em [4] é apresentado um algoritmo para projeto de quantizadores vetoriais preditivos fazendo uso de um conjunto de dicionários de descrição múltipla – a técnica é avaliada no cenário de canais com perdas de pacotes. O desempenho do algoritmo proposto é avaliado em codificação de parâmetros LSF, considerando quadros de 20 ms de voz (160 amostras), representados por vetores de dez parâmetros LSF. Chatterjee e Sreenivas apresentam em [5] um método de SVQ (*split vector quantization*) baseado em função densidade de probabilidade condicional. O método é desenvolvido assumindo que a distribuição da fonte LSF é uma gaussiana multivariada. Além disso, os subvetores são quantizados sequencialmente de modo a explorar a correlação entre eles. Com isso, são obtidos resultados superiores aos apresentados pela técnica SVQ convencional. Sasazaki et al. apresentam em [6] um método de quantização vetorial com tamanho de bloco variado usando dimensões fractais locais – o desempenho do método é superior ao apresentado pela quantização vetorial usando FGLA (*Fuzzy generalized Lloyd algorithm*) no tocante à taxa de compressão e qualidade da imagem codificada. Em [7] os autores aplicam a quantização vetorial no domínio da transformada *wavelet* discreta com o propósito de comprimir imagens de tumores de pele. É realizado um estudo envolvendo taxa de compressão, tempo de compressão e tempo de descompressão, e são realizadas comparações de desempenho com JPEG. Em [8] é considerado um problema relevante da quantização vetorial: sua sensibilidade aos erros de canal, que podem levar a bloqueamentos incômodos nas imagens reconstruídas. Os autores combinam a diversidade em modulação com a técnica de atribuição de índices (quantização vetorial robusta), obtida por meio do algoritmo *simulated annealing*, para melhorar a qualidade das imagens transmitidas por um canal com desvanecimento Rayleigh – a diversidade em modulação é usada para reduzir a taxa de erro de bits e a atribuição de índices para reduzir o impacto visual dos erros de canal.

A QV também tem sido utilizada em outras aplicações, tais como, identificação vocal [9,10], esteganografia e marca d'água digital [11,12]. Em [9] é proposta uma técnica (GVQ, *group vector quantization*) para treinar dicionários voltados para identificação de locutor. Na técnica, os dicionários são projetados para grupo de vetores em vez de vetores individuais. É realizada uma avaliação de desempenho envolvendo os algoritmos LBG (Linde-Buzo-Gray) e LVQ (*learning vector quantization*). Os autores mostram que os dicionários treinados com GVQ podem levar a uma taxa de identificação de locutores maior que a obtida com dicionários treinados com LBG e LVQ. Em [10] os autores apresentam um sistema de identificação vocal baseado em quantização vetorial (QV) paramétrica e modelos de Markov escondidos (HMMs, *hidden Markov models*). A identificação/rejeição é realizada em duas etapas. A primeira, referente à QV, utiliza dicionários de padrões acústicos de referência. A segunda, concernente a HMMs, apresenta-se como um refinamento da primeira. Foram utilizados dois HMMs de referência para cada locutor, um para coeficientes cepstrais e outro para coeficientes delta cepstrais. Com o esquema de identificação utilizado (pré-classificação em sexo, seguida de QV e possível refinamento via HMM), foram obtidas elevadas taxa média de identificação (97,8%) e confiabilidade (99,2%) e baixas taxas médias de falsa aceitação (0,8%) e falsa rejeição (1,5%), considerando uma base de dados de 40 locutores. Em [11] é apresentado um esquema para esconder informações em imagem, o qual é capaz de esconder múltiplas imagens secretas em uma imagem de cobertura de mesmo tamanho. As imagens secretas são comprimidas usando quantização vetorial e submetidas a uma criptografia, e escondidas usando os bits menos significativos (LSB, *least significant bits*) dos pixels da imagem de cobertura (imagem *host*). As qualidades das imagens secretas e das estegoimagens são boas para aplicações práticas. Chiang e Tsai apresentam em [12] um método de esteganografia em imagens digitais. O método consiste em agrupar os vetores-código de um dicionário em subclasses, de acordo com a similaridade. Resultados de simulação mostram que os autores conseguem aumentar a capacidade de esconder bits sem sacrificar a qualidade da imagem.

O algoritmo Linde-Buzo-Gray (LBG) [13], também conhecido como Algoritmo de Lloyd Generalizado, constitui-se na técnica mais utilizada para o projeto de dicionários. Outras abordagens têm sido utilizadas em projeto de dicionários, como por exemplo: algoritmo de Kohonen [14] e outros algoritmos não-supervisionados [15–23]; relaxação estocástica [24]; algoritmos *fuzzy* [25, 26].

Algoritmos de otimização podem ser usados para melhorar o desempenho de quantizadores vetoriais. Dentre esses algoritmos, podem ser citados os Algoritmos Genéticos (AG) [27]. Neste contexto, um AG recebe como entrada dicionários gerados por algum algoritmo de projeto de dicionários e tenta obter um melhor dicionário mediante aplicações sucessivas de operadores, tais como, recombinação (ou cruzamento) e mutação, em um processo evolutivo.

No presente artigo, um AG híbrido modificado é utilizado para a otimização de dicionários projetados com o algoritmo LBG. O AG modificado é composto de um módulo de otimização local. Especificamente, as soluções encontradas ao final de cada geração do AG são atualizadas de acordo com o método de Lee et al. [28], o qual consiste em uma versão acelerada do LBG. A estratégia de integração de módulos de busca local a algoritmos evolutivos, como aqui se propõe, constitui-se em uma característica inerente a Algoritmos Meméticos [29].

O artigo encontra-se organizado como segue: a Seção 2 descreve o algoritmo LBG; a Seção 3 descreve as modificações realizadas na versão acelerada do LBG; a Seção 4 descreve o método de busca por distância parcial (PDS) empregado no algoritmo proposto para a redução da complexidade da busca pelo vizinho mais próximo na atualização dos dicionários; a Seção 5 introduz os conceitos e as questões que envolvem o projeto de um Algoritmo Memético; a Seção 6

fornece a descrição do algoritmo proposto para a otimização de quantizadores vetoriais; finalmente, as Seções 7 e 8 apresentam os resultados e as conclusões obtidas da aplicação do algoritmo proposto à quantização vetorial de imagens monocromáticas.

## 2. ALGORITMO LINDE-BUZO-GRAY (LBG)

Seja a iteração do algoritmo LBG denotada por  $n$ . Dados  $k$ ,  $N$  e um limiar de distorção  $\varepsilon > 0$ , o algoritmo LBG [13] consiste nos seguintes passos:

*Passo 1* (inicialização). Dados um dicionário inicial  $W_0$  e um conjunto de treino  $X = \{\mathbf{x}_m; m = 1, \dots, M\}$ , faça  $n = 0$  e  $D_{-1} = \infty$ .

*Passo 2* (particionamento). Seja  $W_n$  o dicionário na  $n$ -ésima iteração e  $\mathbf{w}_i^n$  o  $i$ -ésimo vetor-código do dicionário. Atribua cada vetor de treino (vetor de entrada) a uma classe correspondente (célula de Voronoi), de acordo com a regra do vizinho mais próximo; determine a distorção

$$D_n = \sum_{i=1}^N \sum_{\mathbf{x}_m \in S_i} d(\mathbf{x}_m, \mathbf{w}_i^n). \quad (1)$$

*Passo 3* (teste de convergência). Defina  $W_n$  como o dicionário final e pare se  $(D_{n-1} - D_n)/D_{n-1} \leq \varepsilon$ ; senão, continue.

*Passo 4* (atualização do dicionário). Calcule os novos vetores-código como

$$\mathbf{w}_i^{n+1} = \mathcal{C}(\mathcal{V}(\mathbf{w}_i^n)), \quad (2)$$

no qual  $\mathcal{C}(\mathcal{V}(\mathbf{w}_i^n))$  é o centróide da partição  $\mathcal{V}(\mathbf{w}_i^n)$ ; faça  $W_{n+1} \leftarrow W_n$ ; faça  $n \leftarrow n+1$  e volte para o Passo 2.

No algoritmo LBG, a função distorção decresce monotonicamente, uma vez que o dicionário é iterativamente atualizado visando satisfazer as condições de centróide e de vizinho mais próximo. A distorção introduzida ao se representarem os vetores do conjunto de treinamento pelos correspondentes vetores-código (centróides) é monitorada a cada iteração. A regra de parada (teste de convergência) do algoritmo baseia-se nessa distorção monitorada – o treinamento do dicionário é encerrado quando  $(D_{n-1} - D_n)/D_{n-1} \leq \varepsilon$ . A velocidade de convergência do algoritmo e o desempenho do dicionário final dependem do dicionário inicial [1, 13].

## 3. ALGORITMO LBG ACELERADO

O algoritmo proposto por Lee *et al.* [28] corresponde a uma modificação introduzida no algoritmo LBG, a qual, em geral, resulta, para uma mesma inicialização, em um melhor dicionário. A modificação diz respeito ao passo de atualização do dicionário. Os primeiros vetores-código são dados ou calculados como os centróides das classes na partição inicial. Em todas as etapas subsequentes, cada novo vetor-código não é mais determinado como o centróide de sua classe, mas sim por um dos pontos na linha que conecta o vetor-código atual com seu ponto refletido, passando pelo novo centróide da classe, conforme ilustra a Figura 1. O algoritmo LBG determina o ponto 2, correspondente ao novo centróide da classe, como o novo vetor-código. A proposta de Lee *et al.*, ao utilizar uma abordagem do tipo *look ahead*, de escolher como novo vetor-código um ponto entre os pontos 2 e 4, visa acelerar a convergência do algoritmo LBG.

No método de Lee *et al.* o novo vetor-código é determinado da seguinte forma: *novo vetor-código* = *vetor-código atual* + *escala*  $\times$  (*novo centróide* – *vetor-código atual*), ou seja,

$$\mathbf{w}_i^{n+1} = \mathbf{w}_i^n + s(\mathcal{C}(\mathcal{V}(\mathbf{w}_i^n)) - \mathbf{w}_i^n) \quad (3)$$

em que  $s$  é a escala,  $\mathbf{w}_i^n$  denota o vetor-código  $\mathbf{w}_i$  ao final da  $n$ -ésima iteração e  $\mathcal{C}(\mathcal{V}(\mathbf{w}_i^n))$  é o centróide da partição  $\mathcal{V}(\mathbf{w}_i^n)$ . Esse método demanda praticamente o mesmo esforço computacional do algoritmo LBG para cada iteração.

Considerando o uso da distância quadrática para a partição, Lee *et al.* mostraram experimentalmente que o algoritmo por eles proposto converge mais lentamente em comparação ao algoritmo LBG convencional para  $s < 1$ . Quando  $1 < s < 2$ , converge mais rápido e resulta em um melhor desempenho em termos de erro médio quadrático. Quando  $s > 2$  o algoritmo ou não converge ou converge muito lentamente, com um desempenho ruim. Nas avaliações de Lee *et al.*, quando o valor da escala é em torno de 1,8 esse método alcança o melhor desempenho.

Para  $s = 1$  o algoritmo de Lee *et al.* corresponde ao algoritmo LBG convencional.

## 4. BUSCA POR DISTÂNCIA PARCIAL (PDS)

No *Passo 2* do algoritmo LBG (bem como no algoritmo LBG modificado, proposto por Lee *et al.* [28]), a determinação do vizinho mais próximo (VMP) consiste em selecionar, para um dado vetor de treino (vetor de entrada)  $\mathbf{x}$ , o

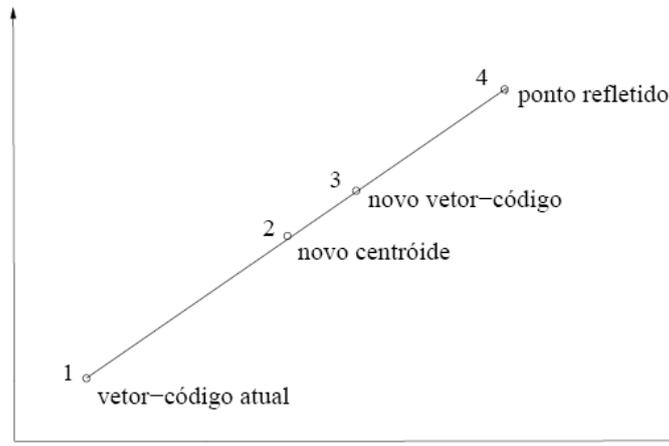


Figura 1: Atualização de um vetor-código pelo método de Lee *et al.* para  $k = 2$ .

vetor-código que se apresenta como o mais semelhante (isto é, como o mais próximo), de acordo com um critério de distância. Determina-se o VMP de  $\mathbf{x}$  calculando sua distância a cada um dos  $N$  vetores-código do dicionário e depois comparando essas distâncias de modo a encontrar o vetor-código mais semelhante, ou seja, o vizinho mais próximo. No processo convencional de busca total (*full search*) tem-se, portanto,  $N$  cálculos de distância e  $N - 1$  comparações. A distância euclidiana quadrática, isto é,

$$d(\mathbf{x}, \mathbf{w}_i) = \sum_{j=1}^k (x_j - w_{ij})^2, \quad (4)$$

em que  $w_{ij}$  é a  $j$ -ésima componente do vetor-código  $\mathbf{w}_i$  e  $x_j$  é a  $j$ -ésima componente do vetor de entrada  $\mathbf{x}$ , requer, para cada cálculo de distância,  $k$  multiplicações,  $k$  subtrações e  $k - 1$  adições. Portanto, para determinar o VMP de cada vetor de entrada,  $kN$  multiplicações,  $kN$  subtrações,  $(k - 1)N$  adições e  $N - 1$  comparações devem ser computadas.

O algoritmo de busca por distância parcial, denotado por PDS (*partial distance search*), proposto em [30], constitui uma das técnicas tradicionais para redução da complexidade computacional envolvida no processo de busca do VMP. O algoritmo PDS decide que o vetor-código em questão não representa o vizinho mais próximo se, para algum  $j < k$ , a distância acumulada (ou seja, a distância parcial) para as primeiras  $j$  amostras do vetor de entrada é maior que a menor distância até então encontrada no processo de busca. Nesse caso, interrompe-se o cálculo de distância para esse vetor-código e inicia-se esse cálculo para o próximo vetor-código do dicionário. Nessa abordagem, o número de multiplicações por amostra é reduzido significativamente. Observa-se, também, redução do número de subtrações/adições por amostra. Apesar de o algoritmo PDS introduzir um aumento do número de comparações, esse método reduz a complexidade global da busca do vizinho mais próximo. A estrutura básica do algoritmo PDS é apresentada a seguir:

```

dmin = ∞
FOR i = 1 TO N
    d = 0
    FOR j = 1 TO k
        d = d + (xj - wij)2
        IF d > dmin THEN
            Break
    ENDFOR
    dmin = d
    VMP = wi
ENDFOR
    
```

No algoritmo PDS,  $d$  denota a distância (distorção),  $d_{min}$  representa a distância mínima (ou seja, a menor distância até então encontrada no processo de procura),  $x_j$  denota a  $j$ -ésima componente do vetor de entrada  $k$ -dimensional  $\mathbf{x}$  e  $w_{ij}$  representa a  $j$ -ésima componente do  $i$ -ésimo vetor-código  $k$ -dimensional  $\mathbf{w}_i$ ,  $i = 1, \dots, N$ .

## 5. ALGORITMOS MEMÉTICOS

Diversos métodos de hibridização têm sido propostos para a integração de AGs a heurísticas de buscas locais em problemas de otimização. Uma das técnicas mais utilizadas de hibridização consiste na aplicação de busca local a cada nova solução gerada por um AG. Nessa abordagem, as soluções são “movidas” para ótimos locais antes de serem introduzidas na próxima geração de soluções. Gen e Cheng [31] ressaltam que a complementariedade dos AGs e dos métodos de buscas locais

é condicionante para a superioridade da técnica híbrida em comparação com quaisquer dos métodos de busca aplicados isoladamente. A combinação de busca local e busca evolutiva denotou-se por Algoritmo Memético (AM) em [32], baseado no conceito de *meme*<sup>1</sup> introduzido por Dawkins em [33].

Radcliffe e Surry [34] observam que, se um otimizador local é integrado a um AG e aplicado a cada solução antes de ser inserida na população, inclusive na população inicial, então tal algoritmo pode ser considerado como um caso especial de busca genética no subespaço de ótimos locais, i.e., um AM. Ainda, notam que, se a recombinação e a mutação produzem geralmente “proto-soluções” fora do subespaço de ótimos locais, o otimizador local, por sua vez, é capaz de “restaurar” essas soluções de tal forma que a solução a ser introduzida na população pertença novamente àquele subespaço. Esse cenário pode ser melhor compreendido ao se observar a Figura 2: a recombinação de duas soluções, **X** e **Y**, produz uma proto-solução, **Z'**, fora do subespaço de ótimos locais; o otimizador local, ao utilizar uma técnica de *hill climbing*, realiza o mapeamento  $Z' \rightarrow Z$ , i.e., move a proto-solução **Z'** para o subespaço de ótimos locais. Entretanto, em aplicações práticas, o custo da aplicação do otimizador local à cada solução é bastante elevado, de forma que aspectos tais como a “intensidade de aprendizado” [35] — quantidade de tempo computacional alocado em uma etapa de aprendizado individual— devem ser considerados.

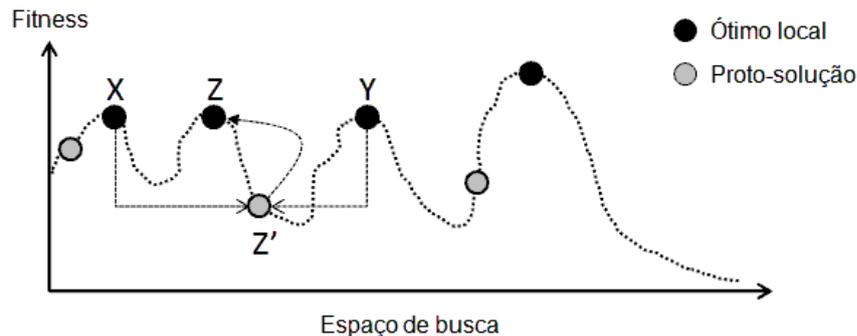


Figura 2: Operação de um Algoritmo Memético em um subespaço de ótimos locais.

## 5.1. ALGORITMOS MEMÉTICOS EM QUANTIZAÇÃO VETORIAL

O LBG pode ser visto como um algoritmo de projeto de dicionários o qual adota uma estratégia de busca *hill climbing* [36]. Portanto, supõe-se que a busca alcançará ótimos locais (partições sub-ótimas). No entanto, abordagens estocásticas, tais como AGs, podem ser utilizadas para procurar a partição ótima, uma vez que podem escapar de mínimos locais. Considerando a população em um AG como sendo constituída de vários dicionários, Fränti [37] apresenta três possibilidades para a integração entre AGs e o LBG como um módulo de busca local: (i) aplicar o LBG à saída do AG (AG + LBG); (ii) aplicar o AG à saída do LBG (LBG + AG); e (iii) aplicar o LBG a todos os dicionários presentes na população. Também é possível adaptar o AG para QV sem o uso de técnicas integradas de *hill climbing*, tal como o LBG. Demonstrou-se, porém, que tais AGs não possuem um bom desempenho em comparação com os AGs que utilizam conjuntamente um algoritmo de agrupamento [37]. Uma vez que o LBG procura satisfazer as condições de otimalidade requeridas pela QV [1], a sua integração com AGs em algoritmos meméticos é considerada benéfica [36, 37].

## 6. O ALGORITMO PROPOSTO

O algoritmo proposto no presente artigo deriva da estratégia AG + LBG, em que um indivíduo é um dicionário composto por  $N$  genes, os quais correspondem a  $N$  vetores-código. Entretanto, algumas diferenças podem ser apontadas, ao compará-lo com os métodos descritos em [37]:

- i. modificou-se a estratégia de recolocação de forma a aceitar todos os filhos que representam melhores soluções em relação àquelas já conhecidas;
- ii. com probabilidade  $1 - p_{acc}$  (em que  $p_{acc}$  é a probabilidade de aceitação), rejeitam-se as soluções produzidas que não obtiverem ganhos sobre o pior indivíduo presente na população. Quando aceito, um filho irá sempre substituir o pior indivíduo;
- iii. substituiu-se o algoritmo LBG convencional pela versão acelerada proposta em [28] para o módulo de otimização local; e (iv) em vez de aplicar o LBG acelerado ao melhor indivíduo a cada geração, aplica-se o algoritmo à nova solução gerada.

A Figura 3 ilustra o processo geral realizado pelo otimizador híbrido evolutivo proposto neste trabalho. As justificativas para as modificações são descritas nos parágrafos seguintes.

As modificações (i) e (ii) estão relacionadas à probabilidade de aceitação ( $p_{acc}$ ). Para  $p_{acc} = 0$ , essa estratégia de recolocação equivale a uma busca genética gulosa [38, 39], enquanto  $p_{acc} = 1$  faz com que a estratégia seja aquela

<sup>1</sup> Um meme é descrito por Dawkins como uma “unidade de evolução” que compete com seus alelos (i.e., memes rivais) por recursos (i.e., tempo e espaço) nas “máquinas de sobrevivência” (i.e., cérebros).

implementada no AG canônico (aceitam-se todos os filhos) [31]. Esse parâmetro adicional fornece um controle mais preciso dos níveis de diversidade do AG, garantindo que boas soluções não sejam rejeitadas. Essa modificação foi amplamente investigada em Leung *et al.* [40], sendo também utilizada para treinar redes neurais do tipo Perceptron multi-camadas (MLP – *multilayer Perceptron*) naquele trabalho.

A modificação (iii) constitui a principal contribuição do presente trabalho, pois leva a melhores dicionários em um menor número de gerações em relação à estratégia AG + LBG convencional, conforme apresentado na Seção 7.

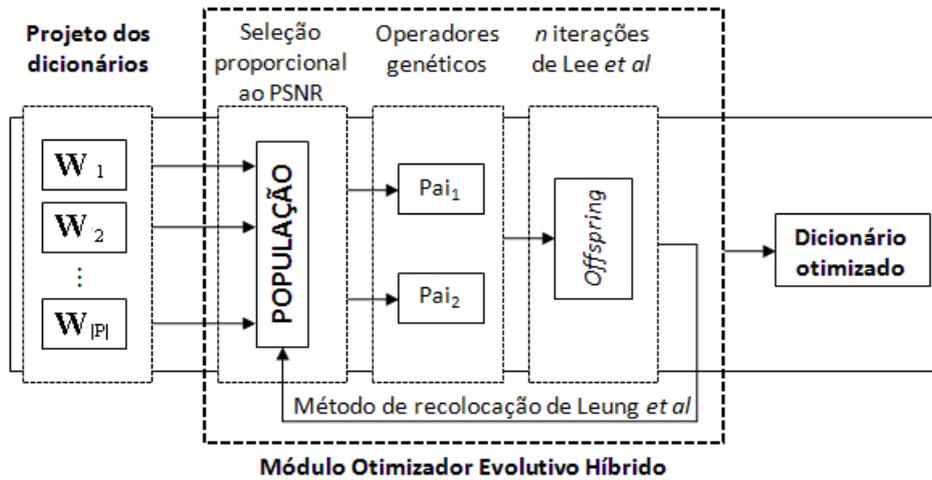


Figura 3: O método proposto para otimização de dicionários da QV.

Finalmente, justifica-se a modificação (iv) mediante os objetivos de manter a diversidade e acelerar a convergência: em aplicações de AGs, o controle de diversidade é comumente realizado por meio do ajuste da taxa de mutação [31], *i.e.*, a frequência de aplicação do operador de mutação sobre o filho gerado. Nesse sentido, a integração do método de Lee *et al.* ao AG pode ser considerada como a concepção de um módulo de otimização local, o qual contribui para a obtenção de um maior nível de diversidade. A versão acelerada do LBG [28], quando comparada ao algoritmo LBG convencional, em geral, produz, em um menor número de iterações, dicionários que levam a uma menor distorção dos sinais quantizados. Portanto, espera-se que as aplicações do método de Lee *et al.* no AG melhorem as soluções geradas, de tal forma que um indivíduo tenha maiores chances de representar uma solução mais apta do que aquelas representadas pelos dicionários que o geraram. Cumpre mencionar que a estratégia descrita na modificação (iv) foi utilizada em [36] para o projeto de dicionários, levando a resultados adequados. Os demais mecanismos do método proposto são explicados a seguir:

**Inicialização** – Denota-se a população inicial utilizada como o conjunto  $P = \{W_1, \dots, W_{psize}\}$  de dicionários iniciais projetados, em que  $|P| = psize$  denota o tamanho da população. Neste trabalho, experimentou-se o método proposto com dois tipos de população: a primeira projetada com o LBG convencional e a segunda projetada pelo método de Lee *et al.* [28]. Nos dois casos, o limiar de distorção utilizado  $\epsilon = 10^{-3}$ .

**Seleção para a reprodução** – Adota-se uma estratégia de seleção proporcional ao *fitness*, conhecida como método da roleta (*roulette wheel*): a população é ordenada pelos valores de *fitness* dos indivíduos e dois pais são escolhidos com probabilidade proporcional aos seus respectivos *fitnesses*, em um cenário de simulação estocástica. **Operadores genéticos** – Para a recombinação dos pais, implementa-se um operador generalizado de cruzamento da seguinte forma: primeiro, um número inteiro aleatório, *segsz*, é amostrado do conjunto  $\{1, \dots, N/2\}$ , em que *segsz* denota um tamanho de segmento de genes e  $N$  é o número de genes (vetores-código). Então, para cada bloco composto por *segsz* genes, escolhe-se aleatoriamente o bloco correspondente a um dos pais, compondo os genes do filho. Esse processo segue até que  $N$  genes tenham sido selecionados. Observa-se que esse algoritmo equivale a um cruzamento uniforme quando *segsz* = 1 e a um cruzamento de um ponto quando *segsz* =  $N/2$  com o ponto de *locus* na metade do cromossomo. Para a mutação, seja  $\mu$  uma variável aleatória uniforme no intervalo  $[0,8; 1,2]$ . Então, realiza-se o produto entre o valor escalar  $\mu$  e um gene escolhido aleatoriamente com probabilidade *pmut*.

## 7. RESULTADOS

O desempenho dos métodos de otimização foram analisados usando as seguintes imagens  $256 \times 256$  pixels, monocromáticas, originalmente codificadas a 8 bpp: Boat, Barbara, Clock, Elaine, Goldhill, Lena, Mandrill, Peppers e Tiffany (Figura 4). Em todos os experimentos, para uma dada imagem e um  $N$  fixo, foram utilizados os mesmos dicionários iniciais e limiar de distorção  $\epsilon = 10^{-3}$ . Foram geradas duas populações iniciais compostas por 20 dicionários para cada experimento: a primeira com dicionários projetados pelo algoritmo LBG convencional e a segunda com dicionários projetados pelo algoritmo de Lee *et al.* com um fator de escala fixo em  $s = 1,8$ . Especificamente para o método proposto, escolheu-se  $p_{acc} = 0,1$  e realizaram-se as simulações variando o número de aplicações do módulo de otimização local de Lee *et al.* em  $n = 1, 2$  e 3. Utilizaram-se, ainda, os seguintes valores para os parâmetros comuns em ambos os métodos em estudo: taxa de mutação igual

a 0,2, taxa de recombinação igual a 1 (filhos são sempre produzidos a cada geração) e um número máximo de 500 gerações para os métodos confrontados. O valor da Relação Sinal-Ruído de Pico (PSNR) foi usado para medir a qualidade objetiva das imagens reconstruídas com os dicionários produzidos e utilizado como função de *fitness* nos AGs. A função *Peak Signal-to-Noise Ratio* (PSNR) é computada como

$$\text{PSNR} = 10 \times \log_{10}(255^2/\text{MSE}), \quad (5)$$

em que

$$\text{MSE} = \frac{1}{K|X|} \times \sum_{i=1}^N \sum_{\mathbf{x}_m \in S_i} \|\mathbf{x}_m - \mathbf{w}_i\|^2. \quad (6)$$

Note-se que 255 representa o valor de pico para imagens monocromáticas originalmente codificadas a 8 bpp. Ainda, vê-se que quanto menor o valor do MSE, maior o valor indicado pela medida PSNR, medida em dB.

A abordagem proposta, assim como os demais métodos, foi implementada utilizando a linguagem de programação C++ e compilador *GNU Compiler Collection* (GCC). Todas as simulações foram realizadas em PCs equipados com processadores Athlon AMD64 com 1 GB de memória RAM e sistema operacional Windows XP.



(a) Boat.



(b) Barbara.



(c) Clock.



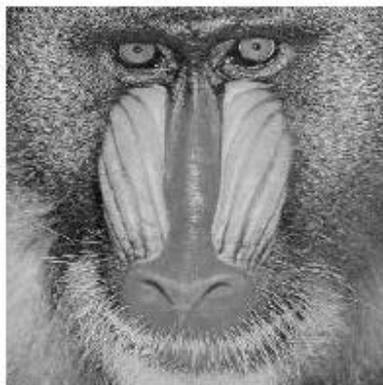
(d) Elaine.



(e) Goldhill.



(f) Lena.



(g) Mandrill.



(h) Peppers.



(i) Tiffany.

Figura 4: Conjunto de imagens de treino.

O quantizador vetorial foi projetado com dimensão  $k = 16$ , o que corresponde ao uso de blocos de  $4 \times 4$  pixels. Por sua vez, utilizaram-se os tamanhos de dicionário  $N = 32, 64, 128, 256$  e  $512$ , correspondendo às taxas de codificação de  $0,3125, 0,375, 0,4375, 0,5$  e  $0,5625$  bpp. A Tabela 1 apresenta os resultados obtidos após 50 experimentos com cada abordagem considerada para a imagem Lena, em que foi utilizado um fator de escala  $s = 1,5$  em uma aplicação de Lee *et al.* no método proposto. Nesse exemplo, a população inicial foi projetada com o LBG convencional.

Tabela 1: Valores médios de PSNR (em dB) obtidos a partir de 50 experimentos com a imagem Lena.

N	LBG	AG + LBG	Proposto
32	26,59	26,69	26,70
64	27,71	27,91	27,91
128	28,80	29,07	29,11
256	29,90	30,55	30,73
512	31,11	32,33	32,62

Observando a Tabela 1 nota-se que o método proposto superou as técnicas convencionais investigadas. Além disso, o aumento do tamanho do dicionário ( $N$ ) leva a ganhos maiores ao se utilizar o AG proposto.

Um estudo experimental também foi conduzido para ajustar o valor do fator de escala ( $s$ ) usado nas simulações. Para essa tarefa, valores do fator de escala do conjunto  $\{1,1; 1,2; \dots, 2,0\}$  foram testados para cada combinação de imagem e tamanho de dicionário ( $N$ ). Os resultados para os melhores valores do fator de escala e os respectivos ganhos obtidos, em termos de PSNR, em relação aos dicionários iniciais (população inicial) projetados com o LBG para imagens codificadas a  $0,5625$  bpp, são apresentados na Tabela 2. Com a utilização do método proposto, alcançaram-se os maiores ganhos médios para as imagens Clock, Lena, Peppers, Tiffany, Boat e Elaine: em relação ao melhor dicionário presente na população inicial, os ganhos obtidos variaram de  $1,10$  a  $1,58$  dB, enquanto, em relação ao valor médio de PSNR obtido nos experimentos com a abordagem AG + LBG, os ganhos obtidos com o método proposto variaram de  $0,20$  a  $0,33$  dB.

Tabela 2: Ganhos médios (em dB) do método proposto sobre os demais para os melhores valores do fator de escala ( $s$ ) utilizados nas imagens quantizadas a  $0,5625$  bpp.

Imagem	Ganho (dB)	
	LBG	AG + LBG
Barbara ( $s = 1,2$ )	0,62	0,15
Boat ( $s = 1,4$ )	1,12	0,21
Clock ( $s = 1,1$ )	1,58	0,33
Elaine ( $s = 1,3$ )	1,10	0,20
Goldhill ( $s = 1,2$ )	0,50	0,00
Lena ( $s = 1,5$ )	1,50	0,29
Mandrill ( $s = 1,1$ )	0,39	0,05
Peppers ( $s = 1,2$ )	1,49	0,26
Tiffany ( $s = 1,5$ )	1,47	0,27

A Figura 5 mostra a evolução de PSNR realizada pelo método proposto e pela estratégia AG + LBG para a imagem Lena com taxa de codificação  $0,5625$  bpp. Observa-se que, ao fim de 500 gerações, o método proposto alcançou um valor médio de PSNR (após 50 experimentos) para a imagem reconstruída de  $32,62$  dB, enquanto o método AG + LBG alcançou um valor de  $32,33$  dB. A Figura 5 também mostra que, após cerca de 257 gerações, o valor de PSNR obtido pelo método AG + LBG estabiliza-se. O método proposto, por sua vez, requer apenas 161 gerações para alcançar o mesmo resultado ( $32,33$  dB) obtido pelo método convencional. Considerando as gerações nas quais o método AG + LBG estabiliza, vê-se que o método proposto apresenta uma redução média de  $37\%$  em relação ao número de gerações necessárias para alcançar um resultado equivalente ao desempenho obtido pelo método AG + LBG.

Por sua vez, a Figura 6 mostra o efeito da variação do parâmetro fator de escala ( $s$ ) para a imagem Peppers codificada a  $0,5625$  bpp. Os resultados mostram que o método proposto obtém para  $1, 0 < s < 1,6$  melhores valores médios de PSNR em relação ao método AG + LBG. Altos valores de  $s$  sugerem que há uma quantidade excessiva de perturbação no processo evolutivo, afetando negativamente a velocidade de convergência. Ainda assim, para  $s > 1,6$ , o método proposto melhora os dicionários iniciais projetados pelo algoritmo LBG.

Tabela 3: PSNR (em dB) médio nas populações iniciais utilizadas para as imagens Clock e Tiffany a  $0,4375$  e  $0,5$  bpp.

Imagem	N	População	
		Pop I	Pop II
Clock	128	29,47	29,77
	256	30,90	31,32
Tiffany	128	32,40	32,72
	256	33,61	34,15

A Tabela 3 apresenta os valores médios de PSNR das imagens Clock e Tiffany reconstruídas a partir dos dicionários presentes na população inicial do método proposto, projetados com o LBG (*Pop I*) e com o algoritmo de Lee *et al.* (*Pop II*). Percebe-se que os dicionários projetados utilizando o método de Lee *et al.* levam a imagens reconstruídas com valores de PSNR mais altos em comparação com algoritmo LBG.

A Figura 7 ilustra o efeito de inicializar o método proposto com dicionários projetados pelo algoritmo LBG e pelo método de Lee *et al.* para a imagem Clock codificada a 0,5625 bpp utilizando duas aplicações de Lee *et al.* como o módulo de otimização local do AG. Da Figura 7, observa-se que a curva que representa o valor médio de PSNR para a população *Pop II* inicia acima da curva para a população *Pop I*. Ainda, ao longo do processo evolutivo, na média, observa-se que a evolução da população *Pop I* obtém, por volta da geração de número 125, indivíduos de aptidão média equivalente à evolução da população *Pop II*. Entretanto, vê-se que, conforme o número de gerações aumenta, o ganho da população *Pop II* cresce, alcançando o valor de 0,13 dB após 500 gerações.

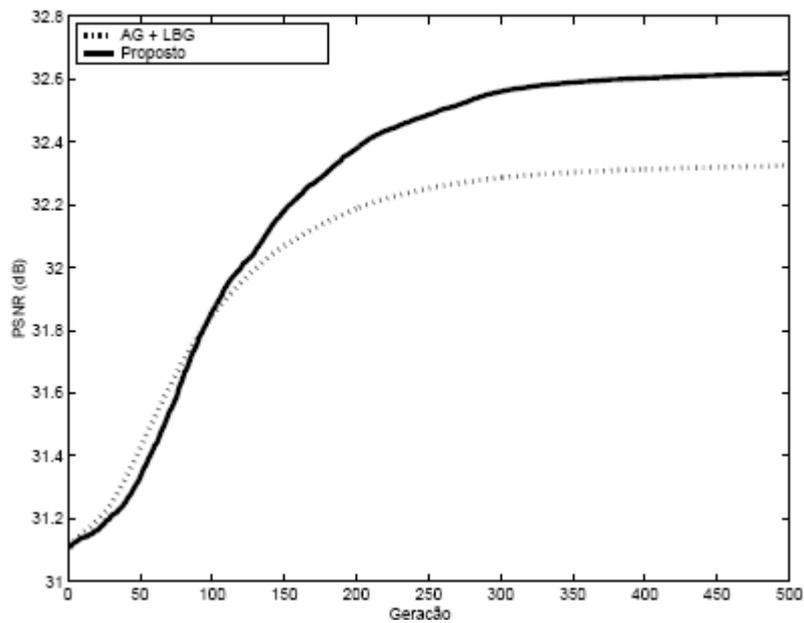


Figura 5: Evolução média do PSNR da imagem Lena com  $N = 512$  calculada a partir de 50 experimentos independentes.

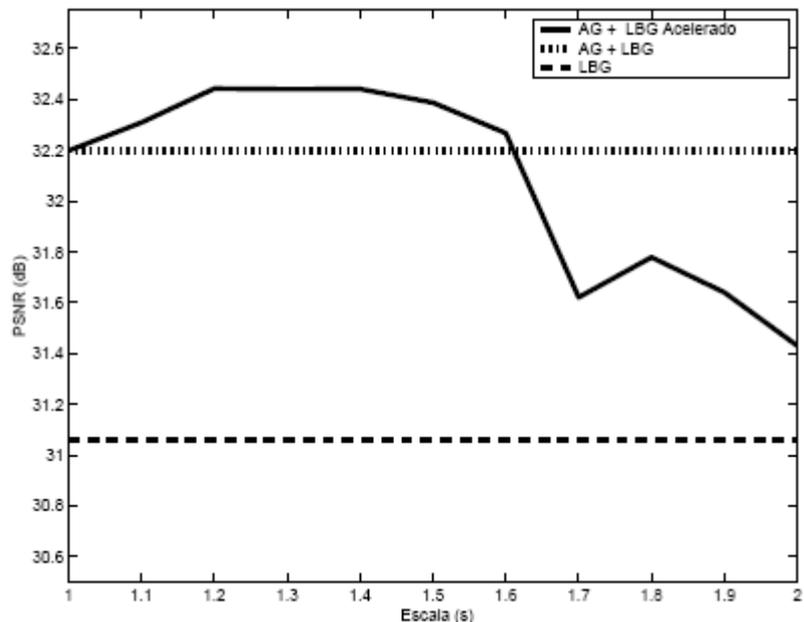


Figura 6: Efeito da variação do fator de escala no PSNR final da imagem Peppers com  $N = 512$ .

Os resultados apresentados na Tabela 4 mostram que alimentar o AG proposto com uma população composta de dicionários projetados com o método de Lee *et al.* pode ou não levar a maiores ganhos médios (em termos de PSNR final) quando se compara com a inicialização a qual utiliza dicionários projetados com o algoritmo LBG convencional. O ganho dependerá do tamanho do dicionário e do número de iterações do módulo de otimização local utilizado no processo evolutivo.

Para tamanhos de dicionários menores que 512, a evolução do método proposto é beneficiada se o algoritmo for inicializado com dicionários projetados por Lee *et al.* Para  $N = 512$ , o método obtém ganhos (em termos de PSNR final) se o número de aplicações do módulo de otimização local for maior que 1.

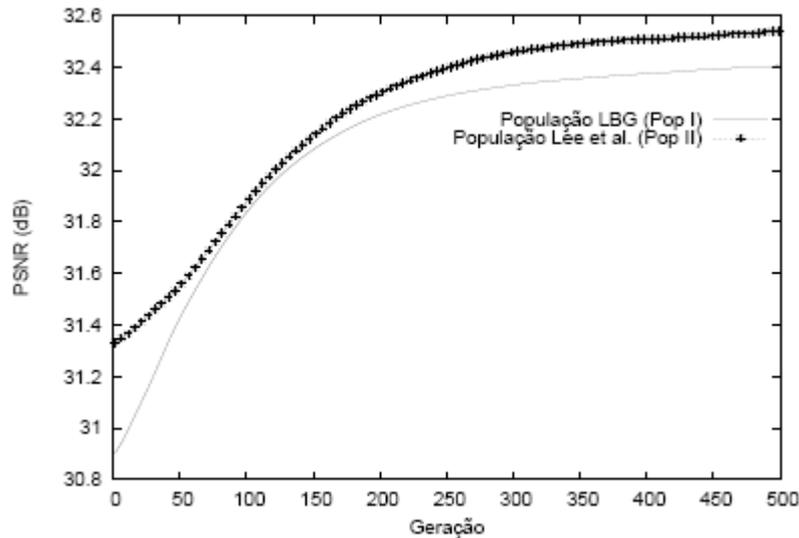


Figura 7: Evolução média obtida pelo método proposto com as duas populações iniciais utilizadas para a imagem Clock codificada a 0,5625 bpp.

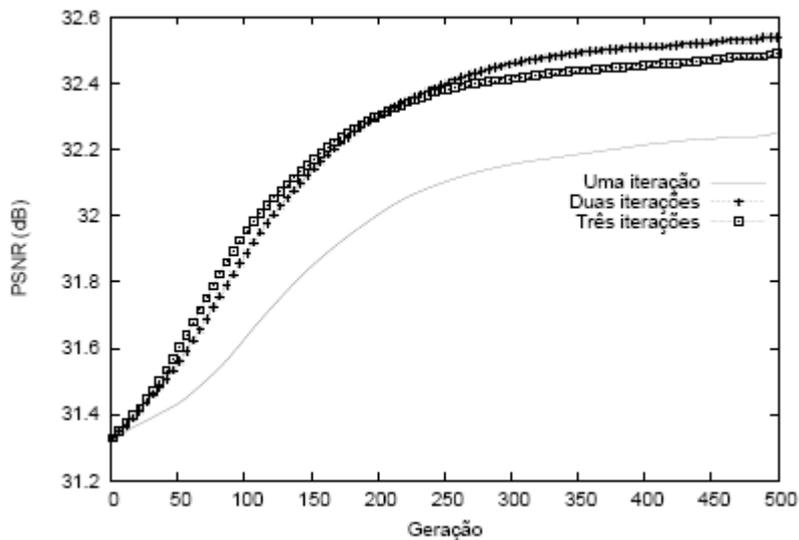


Figura 8: Evolução média obtida pelo método proposto em relação ao número de iterações do algoritmo de Lee *et al.* para a imagem Clock codificada a 0,5625 bpp.

Tabela 4: Valores de PSNR (em dB) obtidos variando o número de iterações de Lee *et al.*, o tamanho dos dicionários e a população inicial para a imagem Clock.

População	Iterações	N	
		256	512
Pop I	1	30,45	32,47
	2	30,65	32,41
	3	30,71	32,45
Pop II	1	30,46	32,25
	2	30,75	32,54
	3	30,77	32,49

O estudo do efeito de elevar o número de aplicações do módulo de otimização local no método proposto pode ser observado na Figura 8 para a imagem Clock codificada a 0,5625 bpp. Baseado nesse estudo, recomenda-se que duas iterações de Lee *et al.* sejam utilizadas no método proposto para a obtenção de melhores dicionários em estágios incipientes do processo

evolutivo. O uso de mais iterações não é recomendado, visto que o método pode ou não obter ganhos em termos de PSNR em relação a duas iterações. Adicionalmente, o custo computacional não justifica os pequenos ganhos obtidos com três aplicações.

Por fim, observou-se que a introdução do algoritmo PDS no Passo 2 do algoritmo LBG e do algoritmo de Lee *et al.* (ou seja, nos módulos de otimização local) – reduziu o tempo médio de simulação em torno de 15%. Por exemplo, para a imagem Tiffany codificada a 0,5 bpp, o módulo otimizador híbrido evolutivo gastou 233 segundos, em média, quando a busca do vizinho mais próximo foi realizada com a busca total, enquanto que gastou 204,2 segundos, em média, com a implementação do algoritmo PDS, considerando todas as 500 gerações em um ambiente computacional Windows XP, processador AMD 64 e 1GByte de memória RAM.

## 8. CONCLUSÃO

O presente artigo apresentou uma abordagem híbrida evolutiva para a otimização de dicionários para quantização vetorial de imagens, mediante o uso de iterações de Lee *et al.* em vez do LBG convencional como um novo módulo de otimização local em um AG modificado. Resultados de simulação mostram que o método proposto otimiza dicionários de entrada gerados com algoritmos de projeto de dicionários. Especificamente, o método produziu, em estágios incipientes do processo evolutivo, melhores dicionários, os quais são utilizados para reconstruir imagens com valores médios de PSNR mais altos em relação à abordagem AG + LBG. Observou-se, ainda, que o uso de duas iterações do método Lee *et al.* introduz o melhor custo-benefício em termos de velocidade de convergência, PSNR final e tempo de simulação. Ainda, recomenda-se utilizar o método de Lee *et al.* para projetar os dicionários iniciais a serem otimizados pelo método proposto. Finalmente, atestou-se o benefício da integração do algoritmo PDS ao método proposto como forma de reduzir a complexidade computacional do otimizador como um todo, provendo um menor tempo médio de otimização dos dicionários de entrada.

A abordagem apresentada possui algumas ressalvas, dentre elas, a necessidade de determinação *a priori* de valores apropriados do fator de escala para o método de aceleração de Lee *et al.* na etapa de otimização local dos dicionários. Como demonstram os resultados, escolhas de valores inapropriados para o fator de escala podem resultar em um pior desempenho para a abordagem proposta em relação à abordagem AG + LBG convencional. Sugere-se, portanto, a utilização de algoritmos auto-adaptativos que possam realizar o ajuste da escala durante o processo evolutivo.

Estudos futuros devem considerar a investigação de um critério de parada baseado na convergência do algoritmo. Os trabalhos correntes dos autores incluem a investigação de um AG baseado em terreno [41] para derivar heurísticas que ajustem o parâmetro do fator de escala ( $s$ ) do algoritmo de Lee *et al.* durante o processo evolutivo.

## REFERÊNCIAS

- [1] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Boston, 1992.
- [2] R. M. Gray. “Vector Quantization”. *IEEE ASSP Magazine*, pp. 4–29, April 1984.
- [3] K. K. Paliwal and B. S. Atal. “Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame”. *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp. 3–14, January 1993.
- [4] P. Yahampath and P. Rondeau. “Multiple-Description Predictive-Vector Quantization With Applications to Low Bit-Rate Speech Coding Over Networks”. *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 749–755, March 2007.
- [5] S. Chatterjee and T. Sreenivas. “Conditional PDF-Based Split Vector Quantization of Wideband LSF Parameters”. *IEEE Signal Processing Letters*, vol. 14, no. 9, pp. 641–644, September 2007.
- [6] K. Sasazaki, S. Saga, J. Maeda and Y. Suzuki. “Vector Quantization of Images with Variable Block Size”. *Applied Soft Computing*, no. 8, pp. 634–645, 2008.
- [7] A. Kjoelen, S. E. Umbaugh and M. Zuke. “Compression of Skin Tumor Images –Wavelet/Vector Quantization Methods for Reducing the Time, Cost and Bandwidth of Storing and Transmitting Data”. *IEEE Engineering in Medicine and Biology*, pp. 73–80, May/June 1998.
- [8] W. T. A. Lopes, F. Madeiro, B. G. Aguiar Neto and M. S. Alencar. “Combining Modulation Diversity and Index Assignment to Improve Image VQ for a Rayleigh Fading Channel”. *Learning and Nonlinear Models – Revista da Sociedade Brasileira de Redes Neurais*, vol. 1, no. 3, pp. 160–171, 2004.
- [9] J. He, L. Liu and G. Palm. “A Discriminative Training Algorithm for VQ-based Speaker Identification”. *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 353–356, May 1999.

- [10] F. Madeiro, J. M. Fechine, W. T. A. Lopes, B. G. Aguiar Neto and M. S. Alencar. "Identificação Vocal por Frequência Fundamental, QV e HMMS". In *Em-TOM-Ação: A Prosódia em Perspectiva*, edited by M. A. M. Aguiar and F. Madeiro, chapter 4, pp. 91–120. Editora Universitária da UFPE, Recife, PE, first edition, 2007.
- [11] Y. Hu. "High-Capacity Image Hiding Scheme Based on Vector Quantization". *Pattern Recognition*, vol. 39, pp. 1715–1724, 2006.
- [12] Y. K. Chiang and P. Tsai. "Steganography Using Overlapping Codebook Partition". *Signal Processing*, vol. 88, no. 5, pp. 1203–1215, May 2008.
- [13] Y. Linde, A. Buzo and R. M. Gray. "An Algorithm for Vector Quantizer Design". *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, January 1980.
- [14] T. Kohonen. "The Self-Organizing Map". *Proceedings of the IEEE*, Vol. 78, No. 9, pp. 1464–1480, September 1990.
- [15] A. K. Krishnamurthy, S. C. Ahalt, D. E. Melton, and P. Chen. "Neural Networks for Vector Quantization of Speech and Images". *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 8, pp. 1449–1457, October 1990.
- [16] O. T.-C. Chen, B. J. Sheu, and W.-C., Fang. "Image Compression Using Self-Organization Networks". *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No. 5, pp. 480–489, October 1994.
- [17] C. Zhu and L. M. Po. "Partial Distortion Sensitive Competitive Learning Algorithm for Optimal Codebook Design". *Electronics Letters*, vol. 32, no. 19, pp. 1757–1758, September 1996.
- [18] E. Yair, K. Zeger, and A. Gersho. "Competitive Learning and Soft Competition for Vector Quantizer Design". *IEEE Transactions on Signal Processing*, Vol. 40, No. 2, pp. 294–309, February 1992.
- [19] F. Madeiro, W. T. A. Lopes, M. S. Alencar and B. G. Aguiar Neto. "Complexidade Computacional de um Algoritmo Competitivo Aplicado ao Projeto de Quantizadores Vetoriais". In *Anais do VI Congresso Brasileiro de Redes Neurais (CBRN'2003)*, pp. 43–48, São Paulo, SP, Brasil, Junho 2003.
- [20] F. Madeiro, W. T. A. Lopes, M. S. Alencar and B. G. Aguiar Neto. "Aprendizagem Competitiva versus Algoritmo LBG Quanto à Complexidade Computacional". In *Anais do VI Congresso Brasileiro de Redes Neurais (CBRN'2003)*, pp. 37–42, São Paulo, SP, Brasil, Junho 2003.
- [21] F. Madeiro, W. T. A. Lopes, M. S. Alencar and B. G. Aguiar Neto. "Construção de Dicionários Voltados para a Redução da Complexidade Computacional da Etapa de Codificação da Quantização Vetorial". In *Anais do VI Congresso Brasileiro de Redes Neurais (CBRN'2003)*, pp. 439–444, São Paulo, SP, Brasil, Junho 2003.
- [22] P. H. E. S. Lima, R. C. de Albuquerque, D. C. da Cunha and F. Madeiro. "On Frequency Sensitive Competitive Learning for VQ Codebook Design". In *Proceedings of the Brazilian Symposium on Neural Networks (SBRN'2008)*, Salvador, BA, Brazil, 2008.
- [23] F. Madeiro, W. T. A. Lopes, M. S. Alencar and B. G. Aguiar Neto. "Aprendizagem Competitiva com Consciência Aplicada ao Projeto de Dicionários para Quantização Vetorial de Voz". In *Anais do Congresso Brasileiro de Redes Neurais (CBRN'2005)*, Natal, RN, Brasil, 2005.
- [24] K. Zeger, J. Vaisey, and A. Gersho. "Globally Optimal Vector Quantizer Design by Stochastic Relaxation". *IEEE Transactions on Signal Processing*, Vol. 40, No. 2, pp. 310–322, February 1992.
- [25] N. B. Karayiannis, and P.-I. Pai. "Fuzzy Vector Quantization Algorithms and Their Applications in Image Compression". *IEEE Transactions on Image Processing*, Vol. 4, No. 9, pp. 1193–1201, September 1995.
- [26] N. B. Karayiannis, P.-I. Pai and N. Zervos. "Image Compression Based on Fuzzy Algorithms for Learning Vector Quantization and Wavelet Image Decomposition". *IEEE Transactions on Image Processing*, vol. 7, no. 8, pp. 1223–1230, August 1998.
- [27] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, Cambridge, MA, United States, 1992.
- [28] D. Lee, S. Baek and K. Sung. "Modified K-means Algorithm for Vector Quantizer Design". *IEEE Signal Processing Letters*, vol. 4, no. 1, pp. 2–4, January 1997.

- [29] N. Krasnogor and J. Smith. "A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues". *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 474–488, 2005.
- [30] C. Bei and R. M. Gray. "An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization". *IEEE Transactions on Communications*, vol. 33, no. 10, pp. 1132–1133, October 1985.
- [31] M. Gen and R. Cheng. *Genetic Algorithms and Engineering Optimization*. Wiley-Interscience, Hoboken, 1999.
- [32] P. Moscato. "On Evolution, Search, Optimization, GAs and Martial Arts: Toward Memetic Algorithms". Prog. rep., California Inst. Technol., 1989.
- [33] R. Dawkins. *The Selfish Gene*. Oxford Univ. Press, New York, 1976.
- [34] N. J. Radcliffe and P. D. Surry. "Formal Memetic Algorithms". In *Evolutionary Computing: AISB Workshop*, volume 865 of LNCS, pp. 1–16, Heidelberg, 1994. Springer.
- [35] Q. H. Nguyen, Y.-S. Ong and N. Krasnogor. "A Study on the Design Issues of Memetic Algorithm". In *Proceedings of the IEEE Congress on Evolutionary Computation, (CEC '07)*, pp. 2390–2397, 2007.
- [36] K. Krishna and M. Murty. "Genetic K-Means Algorithm". *IEEE Transactions on Systems, Man and Cybernetics*, vol. 29, no. 3, pp. 433–439, June 1999.
- [37] P. Fränti. "Genetic Algorithm with Deterministic Crossover for Vector Quantization". *Pattern Recognition Letters*, vol. 21, no. 1, pp. 61–68, 2000.
- [38] J. Smith. "On Replacement Strategies in Steady State Evolutionary Algorithms". *Evolutionary Computation*, vol. 15, no. 1, pp. 29–59, 2007.
- [39] B. A. Julstrom. "Greedy, Genetic, and Greedy Genetic Algorithms for the Quadratic Knapsack Problem". In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '05)*, pp. 607–614, New York, 2005.
- [40] F. Leung, H. Lam, S. Ling and P. Tam. "Tuning of the Structure and Parameters of a Neural Network Using an Improved Genetic Algorithm". *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 79–88, January 2003.
- [41] S. Gordon, R. Pirie, A. Wachter and S. Sharp. "Terrain-Based Genetic Algorithm (TBGA): Modeling Parameter Space as Terrain". In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '99)*. Morgan Kaufmann, 1999.