

Atribuição de Índices para QV Robusta usando o Algoritmo *Variable Neighborhood Search*

J. GALVÃO*, E.A. LIMA e F. MADEIRO

Recebido em 2 dezembro, 2015 / Aceito em 13 abril, 2017

RESUMO. Este trabalho aborda o problema da Quantização Vetorial Robusta no cenário de transmissão de imagens por canal binário simétrico. É introduzida uma aplicação do algoritmo *Variable Neighborhood Search* (VNS) para atribuição de índices para vetores-código do dicionário. Resultados de simulações revelam que o algoritmo VNS, quando comparado ao *Simulated Annealing* e ao Algoritmo Genético, produz dicionários com a mesma robustez aos erros de canal, gastando, para tanto, um menor tempo de execução.

Palavras-chave: Quantização Vetorial Robusta, algoritmos de otimização, processamento digital de imagens.

1 INTRODUÇÃO

O aumento do acesso à Internet, principalmente em dispositivos móveis, como *smartphones* e *tablets*, proporcionou intensa troca de dados entre usuários. Imagens digitais são cada vez mais frequentemente enviadas por essa rede. Cenários como esse evidenciam a importância das técnicas de Processamento de Digital de Imagens (PDI). Um dos objetivos da área é propor mecanismos para transmissão eficiente de imagens digitais.

O armazenamento também é de grande importância quando se fala de imagens digitais. Neste caso, o alvo das técnicas de compressão de imagens, uma subárea de PDI, é diminuir a quantidade de *bits* necessários para representá-las. Uma técnica de compressão muito utilizada é a Quantização Vetorial [6].

Um tópico de interesse envolvendo PDI é a robustez na transmissão de imagens digitais. Durante a transmissão das imagens digitais por canais ruidosos, *bits* podem ser invertidos, ocasionando perda da qualidade da imagem reconstruída. Uma técnica para minimizar o impacto dos ruídos do canal em transmissão baseada em quantização vetorial é a organização do dicionário através da atribuição de índices, ou seja, a Quantização Vetorial Robusta [5, 13]. Para um dicionário de tamanho N , há $N!$ possíveis organizações do dicionário a serem avaliadas. Considerando um

*Autor correspondente: Jair Galvão – E-mail: jair.galvao@upe.br
Escola Politécnica de Pernambuco, Universidade de Pernambuco, 50720-001 Recife, PE, Brasil.
E-mails: eal@poli.br; madeiro@poli.br

dicionário de tamanho 512, por exemplo, há $512! = 3,47 \times 10^{1166}$ organizações a considerar. Alternativas à força bruta são os algoritmos de otimização, como, por exemplo, *Simulated Annealing* [5, 11] e Algoritmo Genético [25, 28].

Este artigo apresenta a utilização do algoritmo *Variable Neighborhood Search* [16] para a organização de dicionários a partir da atribuição de índices aos vetores-código, com o objetivo de reduzir o impacto dos erros de canal na transmissão de imagens digitais.

O restante deste artigo está organizado desta forma: a Seção 2 contempla a Quantização Vetorial e a Quantização Vetorial Robusta (RVQ – *Robust Vector Quantization*); na Seção 3 é apresentado o algoritmo *Variable Neighborhood Search* (VNS) e a aplicação ao problema da RVQ; a metodologia utilizada neste trabalho é apresentada na Seção 4; na Seção 5 são apresentados os resultados das simulações e na Seção 6 são apresentadas as conclusões e trabalhos futuros.

2 QUANTIZAÇÃO VETORIAL

A Quantização Vetorial (QV) é uma técnica que, no cenário de compressão de sinais, tem por objetivo diminuir a quantidade de *bits* para representá-los [6].

A QV tem sido usada amplamente como técnica de compressão de voz [23], vídeo [7] e imagens digitais [4, 26]. A quantização vetorial de imagens digitais é uma técnica de compressão com perda e tem se destacado por obter elevadas taxas de compressão e pela sua simples estrutura de codificação e decodificação.

A QV pode ser definida como um mapeamento Q de um vetor x pertencente ao espaço euclidiano K -dimensional, \mathbb{R}^K , em um vetor pertencente a um subconjunto finito chamado dicionário W , ou seja,

$$Q : \mathbb{R}^K \rightarrow W. \quad (2.1)$$

O dicionário $W = \{w_i; i = 1, 2, \dots, N\}$ é o conjunto de vetores-código K -dimensionais, também chamados de vetores de reconstrução, e N é o tamanho do dicionário (número de níveis de quantização), ou seja, a quantidade de vetores-código [6]. Cada vetor-código w_i pode ser representado por um índice i , associado ao qual existe uma palavra-binária $b_i \in \{0, 1\}^b$, em que $b = \log_2 N$ é o número de *bits*. A taxa de codificação do quantizador vetorial, que mede o número de *bits* por componente do vetor, é definida por $R = \frac{\log_2 N}{K}$. Em codificação de imagens, a taxa de codificação (R) é expressa em *bits* por pixel (*bpp*) [6].

Dentre os problemas relevantes no âmbito de Quantização Vetorial, podemos destacar o projeto de dicionários, e.g. [2, 24], e a concepção de técnicas para reduzir a sensibilidade da QV aos erros de canal, e.g. [3, 12]

2.1 Quantização Vetorial Robusta

Os erros do canal, quando não tratados adequadamente, prejudicam o desempenho do quantizador vetorial, ocasionando alterações na reconstrução da imagem e, conseqüentemente, levando

a uma baixa qualidade da imagem reconstruída. Uma forma de diminuir a degradação da imagem reconstruída a partir dos índices transmitidos é o uso de *bits* de redundância (codificação de canal). Uma outra forma é a atribuição adequada de índices aos vetores-códigos, ou seja, a Quantização Vetorial Robusta [5, 13, 15].

A Quantização Vetorial Robusta tem como objetivo minimizar a degradação das imagens reconstruídas transmitidas por canais ruidosos. A degradação pode ser minimizada realizando a atribuição de índices de tal maneira que vetores-códigos com grande/pequena distância euclidiana estejam associados a palavras-binárias com grande/pequena distância de Hamming. Uma maneira de medir o nível de robustez do dicionário aos erros de canal é o índice de desordem [13], definido como

$$I_d = \sum_{i=1}^N \sum_{j \in H^1(i)} \|w_i - w_j\|^2, \quad (2.2)$$

em que $j \in H^1(i)$ é o conjunto de todos os índices j (palavras-código b_j) cuja distância de Hamming ao índice i (palavras-código b_i) é igual a 1 *bit* e $\|w_i - w_j\|^2$ denota a distância euclidiana quadrática entre os vetores-código w_i e w_j . Quanto menor o I_d maior a robustez inerente do dicionário aos erros de canal.

2.2 Algoritmo Linde-Buzo-Gray

A Quantização Vetorial depende fortemente do projeto de dicionários. O algoritmo LBG [14], também conhecido como GLA (*Generalized Lloyd Algorithm*), aloca os vetores-treino em N classes Cl_i (células de Voronoi) e, utilizando um critério de distorção E , caso o limiar de distorção (ϵ) não seja alcançado, o LBG iterativamente atualiza os vetores-código a partir da regra do centroide em cada $Cl_i(p)$, em que p é o número da iteração e $Cl_i(p)$ é a i -ésima classe ou célula de Voronoi na iteração p .

O pseudo-algoritmo do LBG é descrito a seguir:

- (1) Inicialização: Defina W_0 como um dicionário inicial;
- (2) Faça $p = 0$ e uma distorção inicial $E_{-1} = \infty$;
- (3) Seja W_p o dicionário na p -ésima iteração. Aloque cada vetor-treino t_j , $j = 1, 2, \dots, J$, na respectiva classe do vetor-código (w_i) mais próximo;
- (4) Calcule a distorção E_p , definida por

$$E_p = \sum_{i=1}^N \sum_{t_j \in Cl_i} d(t_j, w_i) \quad (2.3)$$

- (5) Teste o critério de parada: $(E_{p-1} - E_p)/E_p \leq \epsilon$, se for satisfeito retorne W_p como o dicionário projetado, caso contrário, continue;

- (6) Faça $p = p + 1$. Atualize os vetores-código com os centróides, $\text{cent}(\text{Cl}_i(p))$, das respectivas células de Voronoi

$$\begin{aligned} w_i(p) &= \text{cent}(\text{Cl}_i(p)) \\ &= \frac{1}{M_i} \sum_{t_j \in \text{Cl}_i} t_j, 1 \leq i \leq N, \end{aligned} \quad (2.4)$$

em que M_i é o número de vetores t_j da classe Cl_i , e retorne ao Passo 3.

3 VARIABLE NEIGHBORHOOD SEARCH

Algoritmos de busca local são algoritmos de otimização que realizam pesquisas dentro da vizinhança $V(s)$, sendo s a solução local, até encontrar um ótimo local, s' , para uma função objetivo [17]. A cada iteração o algoritmo tenta encontrar uma melhor solução dentro da mesma vizinhança.

Problemas de estagnação são comuns em algoritmos de busca local, os quais podem ficar presos em um mínimo local. Uma estratégia para melhorar o desempenho e escapar de um mínimo local é aumentar o tamanho da vizinhança de busca [1].

O *Variable Neighborhood Search* (VNS) [17] combina a estratégia dos algoritmos de busca local com a mudança dinâmica de vizinhanças, com isso, podendo escapar dos mínimos locais [8, 9]. O pseudo-algoritmo do VNS é descrito a seguir:

- (1) Inicialização: selecione uma estrutura de vizinhança V_m , para $m = 1, \dots, m_{\max}$ que será usada como o espaço de busca. Encontre uma solução inicial s .
- (2) Faça $m = 1$;
- (3) Repita os seguintes passos enquanto $m \leq m_{\max}$:
 - (a) Gere uma solução aleatória s' dentro da m -ésima vizinhança de s ($s' \in V_m(s)$);
 - (b) Aplique métodos de busca local com s' sendo a solução inicial. Seja s'' o ótimo local;
 - (c) Se o ótimo local s'' for melhor que a solução s , então, $s = s''$ e continue na vizinhança $V_{m=1}$. Caso contrário, $m = m + 1$.
- (4) Retorne s .

3.1 VNS na RVQ

Na atribuição de índices aos vetores-código no processo da RVQ, o algoritmo VNS realiza as permutações entre os vetores-código dentro de uma vizinhança, com isso, o VNS encontra novos índices dos vetores-código com o objetivo de minimizar o índice de desordem, apresentado na Equação 2.2.

O tamanho da vizinhança de busca do algoritmo VNS corresponde ao tamanho do dicionário (N). O pseudocódigo do VNS aplicado à quantização vetorial robusta é descrito a seguir:

- (1) Inicialização: selecione uma estrutura de vizinhança V_m , para $m = 1, \dots, N$. Seja L o número de buscas locais.
- (2) Atribua índices (s) aos vetores-códigos e calcule o índice de desordem (I_d) e faça $m = 1$;
- (3) Repita os seguintes passos enquanto $m \leq N$:
 - (a) Escolha m vetores-códigos e realize permutações dentro da m -ésima vizinhança. Encontre uma sequência de novos índices (s'), ou seja, $s' \in V_m(s)$;
 - (b) Faça L buscas locais, ou seja, permuta m vetores-código ao redor de s' e encontre uma nova sequência de índices s'' e calcule o $I_d(L)$;
 - (c) Se a solução s'' for melhor que a solução s , então, $s = s''$ e continue na vizinhança $V_{m=1}$. Caso contrário, $m = m + 1$;
- (4) Retorne a sequência de índices (s) do dicionário.

4 METODOLOGIA

O algoritmo *Variable Neighborhood Search* (VNS) foi aplicado à atribuição de índices aos vetores-código dos dicionários. Os resultados obtidos do VNS foram comparados com os algoritmos *Simulated Annealing* [5, 11] e Algoritmo Genético [25, 28].

Nas simulações, foram utilizados dicionários de tamanho (N) igual a 32, 64, 128, 256 e 512 com dimensão (K) igual a 16 (blocos de 4×4 pixels). Para o conjunto treino dos dicionários das imagens em escala de cinza, foram utilizadas as imagens *Airplane*, *Elaine*, *Goldhill & Peppers* (Fig. 1) codificadas em 8 bpp (256 níveis de cinza) no formato PGM. Para o conjunto treino dos dicionários das imagens coloridas, foram utilizadas as imagens *Lenna*, *Mandrill*, *Sailboat & Tiffany* (Fig. 2) codificadas em 24 bpp (256 níveis de vermelho (*Red*), verde (*Green*) e azul (*Blue*)) no formato PPM. Todas as imagens utilizadas têm dimensão 512×512 pixels. O algoritmo para projeto dos dicionários foi o LBG [14] com limiar de distorção $\epsilon = 10^{-3}$.

A robustez dos dicionários organizados foi avaliada por meio da transmissão de imagens por um canal binário simétrico (BSC – *Binary Symmetric Channel*).

Os algoritmos foram implementados na linguagem C# (C sharp) com a interface de desenvolvimento Visual Studio 2010 da Microsoft® e a configuração dos computadores utilizados foi a seguinte: processador AMD® FX(tm)-6100 de 3,30 GHz com seis núcleos de processamento, 4 GB de memória RAM DDR3, placa mãe Asus® M5A78L-M, HD com 500 GB e sistema operacional Windows 7 Professional®.

4.1 Configuração do *Simulated Annealing*

O algoritmo *Simulated Annealing* (SA) foi configurado com os parâmetros de temperatura inicial (T_0), temperatura final (T_f) e constante de resfriamento (α) escolhidos empiricamente. A função de avaliação dos dicionários organizados pelo SA foi o índice de desordem definido



(a) Airplane.



(b) Elaine.



(c) Goldhill.

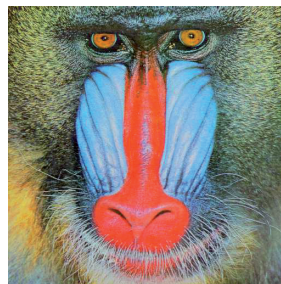


(d) Peppers.

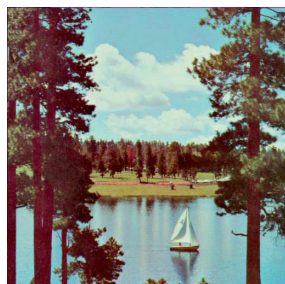
Figura 1: Imagens monocromáticas utilizadas para projetar os dicionários aplicados a RVQ.



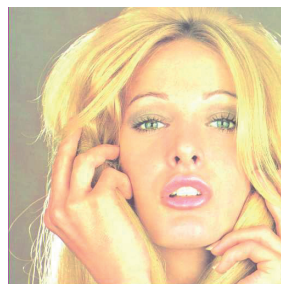
(a) Lenna



(b) Mandrill



(c) Sailboat



(d) Tiffany

Figura 2: Imagens coloridas utilizadas para projetar os dicionários aplicados a RVQ.

na Equação 2.2. Foram utilizadas temperaturas iniciais maiores (T_0) para os dicionários de tamanhos $N = 256$ e 512 . Após realizadas 50 execuções para cada tamanho do dicionário com respectivos parâmetros, foi calculada a média dos valores dos índices de desordem obtidos e calculado o percentual de redução de índice de desordem (ΔId), ou seja, a diferença percentual entre o índice de desordem inicial e o final, conforme mostra a Tabela 1.

Tabela 1: Parâmetros e percentual de redução do índice de desordem do SA.

N	t_0	t_f	α	Avaliações	ΔId
32	10.000	1	0,95	180.000	53,57%
64	10.000	1	0,95	180.000	56,46%
128	10.000	1	0,95	177.653	61,63%
256	50.000	1	0,97	129.484	69,50%
512	50.000	10	0,97	146.174	71,42%

O algoritmo SA reduziu os valores dos índices de desordem dos dicionários. O SA obteve melhor desempenho na organização do dicionário de tamanho $N = 512$, com $\Delta Id = 71,42\%$, conforme se observa na Tabela 1.

4.2 Configuração do Algoritmo Genético

O Algoritmo Genético (AG) foi configurado com valores binários e treinado com vários conjuntos de parâmetros. Para o AG, os valores dos parâmetros com tamanho da população (P), taxa de cruzamento (Tc), taxa de mutação (Tm) e quantidade de pontos de cruzamento (Ps), que levaram ao melhor desempenho (menor I_d) para os tamanhos de dicionário considerados, são apresentados na Tabela 2. Os tamanhos das populações (P) em relação aos tamanhos dos dicionários (N) foram escolhidos a partir do desempenho de diversas configurações de conjuntos de parâmetros. O índice de desordem foi utilizado como função *fitness*, para avaliar cada indivíduo da população. O algoritmo foi executado 50 vezes para cada tamanho do dicionário.

Tabela 2: Parâmetros e percentual de redução do índice de desordem do GA.

N	P	Tc	Tm	Ps	Avaliações	ΔId
32	10	90%	5%	4	70.000	58,71%
64	8	80%	0,5%	4	70.000	58,98%
128	8	80%	0,4%	4	70.000	62,60%
256	8	90%	0,5%	6	120.000	69,24%
512	8	90%	0,2%	4	150.000	69,54%

O Algoritmo Genético reduziu os valores de índice de desordem de todos os tamanhos de dicionários executados. O melhor desempenho em termos de redução percentual do índice de desordem pode ser observado para $N = 512$ com $\Delta Id = 69,54\%$.

4.3 Configuração do *Variable Neighborhood Search*

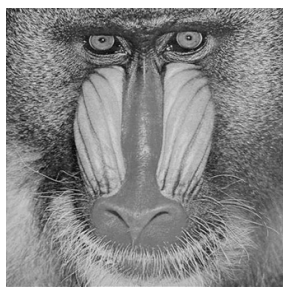
O algoritmo *Variable Neighborhood Search* (VNS) foi configurado conforme mostra a Tabela 3, sendo $Vmax$ o tamanho máximo de vizinhanças, $Vpts$ a quantidade de buscas na vizinhança e Pl a quantidade de permutações locais. Os tamanhos máximos de vizinhanças ($Vmax$) em relação aos tamanhos dos dicionários (N) foram escolhidos de forma empírica e os mesmos levaram a resultados satisfatórios. A função *fitness* utilizada para avaliar os pontos visitados foi o índice de desordem. O VNS foi executado 50 vezes e foram calculados os valores médios dos índices de desordem obtidos. Para o VNS, a maior eficiência de organização pode ser vista no dicionário com tamanho 512, com $\Delta Id = 71, 59\%$.

Tabela 3: Parâmetros e percentual de redução do índice de desordem do VNS.

N	$Vmax$	$Vpts$	Pl	Avaliações	ΔId
32	30	50	18	22.301	58,84%
64	10	80	20	26.191	59,14%
128	15	80	10	48.801	62,74%
256	10	80	20	196.878	69,79%
512	10	70	18	329.883	71,59%

4.4 Configuração do Canal Binário Simétrico

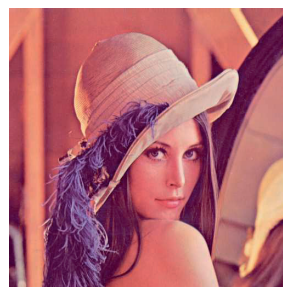
As probabilidades de erro de bit (ϵ) consideradas para o canal binário simétrico (BSC) foram 10^{-4} , 5×10^{-4} , 10^{-3} , 5×10^{-3} , 10^{-2} , 5×10^{-2} , 10^{-1} . As imagens Mandrill, Peppers & Lenna (Fig. 3) foram utilizadas como conjunto de teste, transmitidas pelo canal BSC 50 vezes para cada probabilidade ϵ .



(a) Mandrill



(b) Peppers



(c) Lenna

Figura 3: Imagens utilizadas como conjunto de teste, transmitidas pelo BSC.

5 RESULTADOS

Os melhores dicionários organizados pelos algoritmos *Simulated Annealing*, Algoritmo Genético e *Variable Neighborhood Search* foram selecionados para reconstruir as imagens após a trans-

missão pelo canal. Para medir a qualidade das imagens, foram utilizados os valores da relação sinal-ruído de pico (PSNR – *Peak Signal-to-Noise Ratio*) e da *Structural Similarity* (SSIM) [27] entre a imagem original e a imagem recebida após passar pelo canal BSC. Os parâmetros utilizados no SSIM foram os mesmos propostos em [18].

Inicialmente, é importante observar que o algoritmo VNS obteve valores superiores na redução dos índices de desordem (ΔId) quando comparados aos algoritmos SA e GA para todos os tamanhos de dicionários.

Os valores médios de PSNR obtidos para as diversas probabilidades de erro de *bit* (ϵ) na transmissão da imagem Peppers monocromática pelo canal BSC com dicionários originais e organizados pelos algoritmos em consideração são apresentados nas Tabelas 4 a 8, onde os melhores resultados estão em negrito. Foi observado que os algoritmos obtiveram valores de PSNR próximos em todas as probabilidades ϵ .

Tabela 4: PSNR (dB) médio em relação à probabilidade de erro (ϵ) do canal com $N = 32$.

Método	$\epsilon = 0,0$	$\epsilon = 10^{-4}$	$\epsilon = 5 \times 10^{-4}$	$\epsilon = 10^{-3}$	$\epsilon = 5 \times 10^{-3}$	$\epsilon = 10^{-2}$	$\epsilon = 5 \times 10^{-2}$	$\epsilon = 10^{-1}$
Original	27,59	27,46	27,00	26,56	23,87	21,88	16,32	13,89
SA	27,59	27,54	27,34	27,12	25,59	24,21	19,29	16,66
AG	27,59	27,54	27,34	27,11	25,58	24,19	19,28	16,65
VNS	27,59	27,54	27,34	27,12	25,63	24,27	19,33	16,70

Tabela 5: PSNR (dB) médio em relação à probabilidade de erro (ϵ) do canal com $N = 64$.

Método	$\epsilon = 0,0$	$\epsilon = 10^{-4}$	$\epsilon = 5 \times 10^{-4}$	$\epsilon = 10^{-3}$	$\epsilon = 5 \times 10^{-3}$	$\epsilon = 10^{-2}$	$\epsilon = 5 \times 10^{-2}$	$\epsilon = 10^{-1}$
Original	28,76	28,60	28,00	27,30	24,04	21,88	16,04	13,64
SA	28,76	28,70	28,45	28,16	26,35	24,78	19,53	16,82
AG	28,76	28,71	28,46	28,17	26,31	24,79	19,50	16,80
VNS	28,76	28,70	28,45	28,17	26,37	24,83	19,57	16,86

Tabela 6: PSNR (dB) médio em relação à probabilidade de erro (ϵ) do canal com $N = 128$.

Método	$\epsilon = 0,0$	$\epsilon = 10^{-4}$	$\epsilon = 5 \times 10^{-4}$	$\epsilon = 10^{-3}$	$\epsilon = 5 \times 10^{-3}$	$\epsilon = 10^{-2}$	$\epsilon = 5 \times 10^{-2}$	$\epsilon = 10^{-1}$
Original	29,92	29,69	28,67	27,88	24,30	21,68	15,63	13,43
SA	29,92	29,84	29,59	29,30	27,23	25,44	19,89	17,27
AG	29,92	28,85	29,45	29,22	27,19	25,31	19,75	17,07
VNS	29,92	28,85	29,49	29,24	27,28	25,34	19,85	17,14

O tempo médio gasto pelo VNS foi menor que o do SA e do GA na organização dos índices dos dicionários. A Figura 4 apresenta o tempo médio dos algoritmos, onde se observa uma economia de tempo médio de aproximadamente 18% e 32% obtida com o VNS, respectivamente, em relação aos algoritmos SA e AG, para $N = 256$.

Tabela 7: PSNR (dB) médio em relação à probabilidade de erro (ϵ) do canal com $N = 256$.

Método	$\epsilon = 0,0$	$\epsilon = 10^{-4}$	$\epsilon = 5 \times 10^{-4}$	$\epsilon = 10^{-3}$	$\epsilon = 5 \times 10^{-3}$	$\epsilon = 10^{-2}$	$\epsilon = 5 \times 10^{-2}$	$\epsilon = 10^{-1}$
Original	30,82	30,44	29,22	28,05	23,52	21,01	14,95	12,68
SA	30,82	30,72	30,32	29,91	27,37	25,45	19,63	16,83
AG	30,82	30,71	30,28	29,80	27,15	25,20	19,37	16,62
VNS	30,82	30,72	30,31	29,86	27,38	25,46	19,63	16,83

Tabela 8: PSNR (dB) médio em relação à probabilidade de erro (ϵ) do canal com $N = 512$.

Método	$\epsilon = 0,0$	$\epsilon = 10^{-4}$	$\epsilon = 5 \times 10^{-4}$	$\epsilon = 10^{-3}$	$\epsilon = 5 \times 10^{-3}$	$\epsilon = 10^{-2}$	$\epsilon = 5 \times 10^{-2}$	$\epsilon = 10^{-1}$
Original	31,55	31,09	29,69	28,31	23,53	20,92	14,91	12,76
SA	31,55	31,44	31,00	30,54	27,90	25,94	20,10	17,31
AG	31,55	31,45	30,98	30,51	27,76	25,82	19,93	17,17
VNS	31,55	31,44	31,00	30,53	27,84	25,93	20,05	17,28

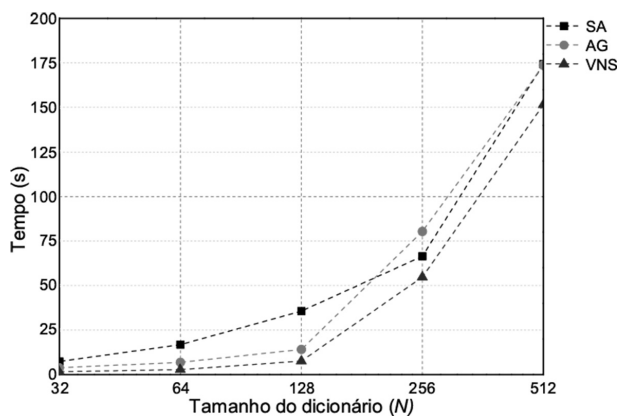


Figura 4: Tempo de execução médio gasto pelos algoritmos.

O desempenho dos algoritmos para a transmissão da imagem Peppers, à luz dos valores de PSNR e SSIM das imagens reconstruídas, pode ser observado nas Figuras 5 e 6 respectivamente. Os valores de SSIM obtidos pelos algoritmos para a imagem Lena após ser transmitida pelo canal BSC pode ser observado na Figura 7. Observa-se que a organização dos dicionários (por SA, AG ou VNS) contribuiu para aumentar a robustez aos erros do canal, levando a imagens reconstruídas com valores de PSNR e SSIM maiores que os obtidos com os dicionários originais.

A qualidade das imagens Peppers reconstruídas, com o tamanho dicionário $N = 256$ e BSC com $\epsilon = 10^{-1}$, pode ser observada na Figura 8. Inspeções visuais revelam que as imagens reconstruídas com dicionários organizados pelos algoritmos SA, AG e VNS tiveram praticamente a mesma qualidade.

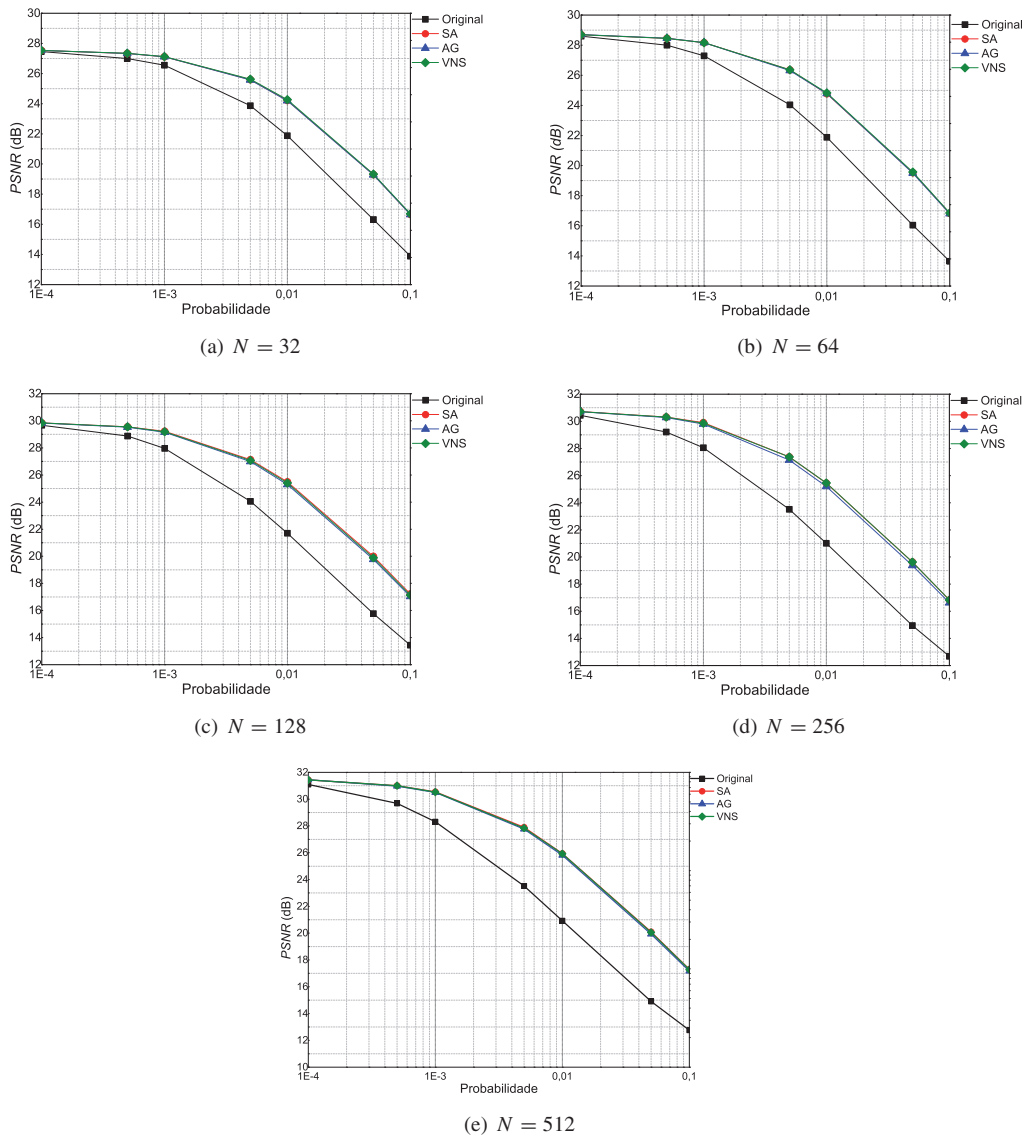


Figura 5: PSNR das imagens Peppers reconstruídas versus probabilidade de erro de *bit* (ϵ) do canal BSC.

As Figuras 9 e 10 apresentam as imagens Mandrill & Lena, respectivamente, reconstruídas com os dicionários organizados pelo VNS para $N = 512$, após transmissão pelo BSC com $\epsilon = 10^{-4}$, 10^{-3} , 10^{-2} e 10^{-1} . Inspeções visuais revelam que as imagens reconstruídas pelos dicionários organizados pelo VNS apresentam melhor qualidade quando comparadas às imagens reconstruídas com uso do dicionário original. Desta forma, observamos a robustez aos erros de canal obtida com a atribuição de índices pelo algoritmo VNS.

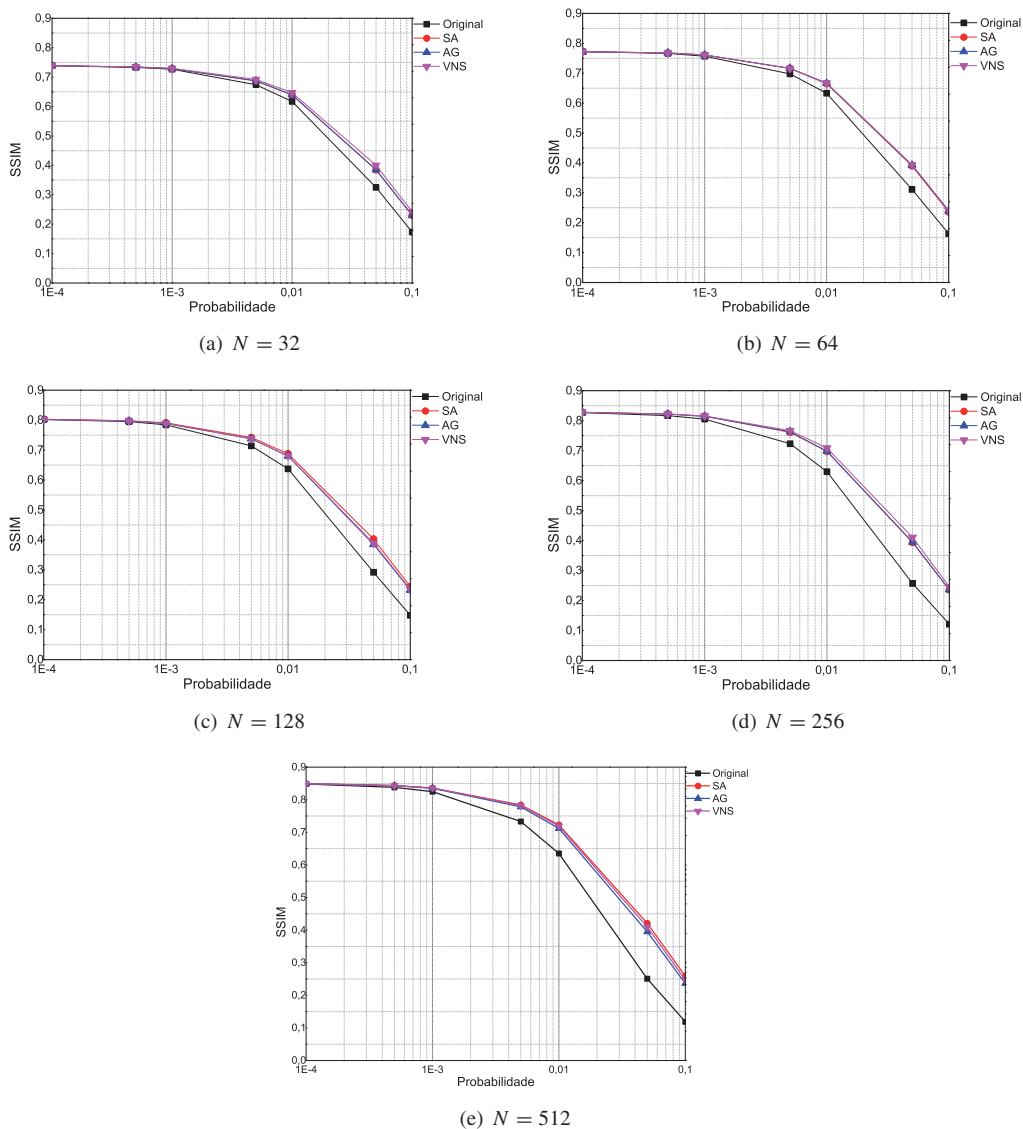


Figura 6: SSIM das imagens Peppers reconstruídas versus probabilidade de erro de bit (ϵ) do canal BSC.

6 CONCLUSÃO

O algoritmo *Variable Neighborhood Search* (VNS) foi apresentado como uma alternativa para atribuição de índices na organização de dicionários utilizados na quantização vetorial robusta. Foi realizada uma avaliação comparativa de desempenho com os algoritmos *Simulated Annealing* (SA) e Genético (AG). Os algoritmos VNS, SA e AG levaram à redução dos índices de desordem em todos os tamanhos de dicionário.

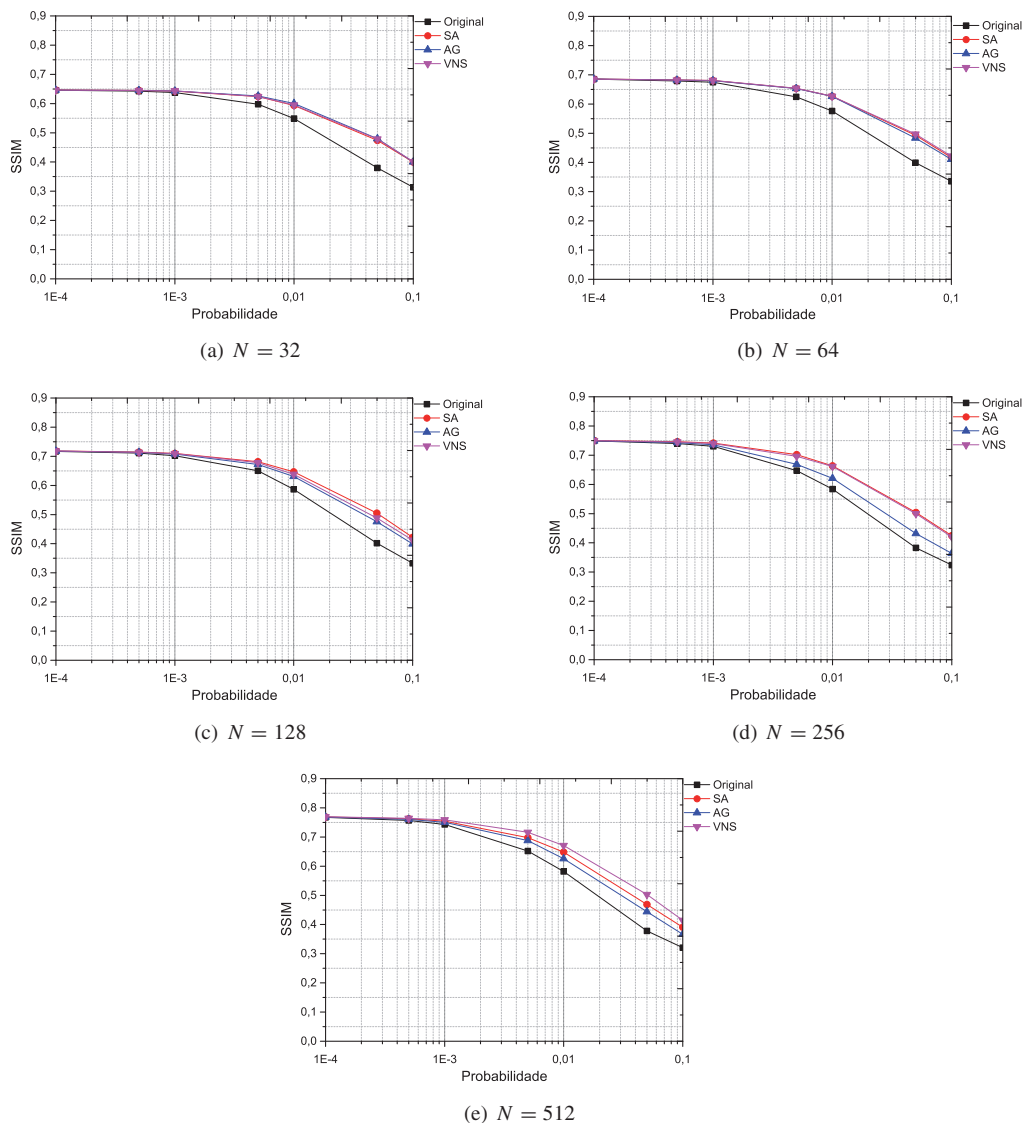
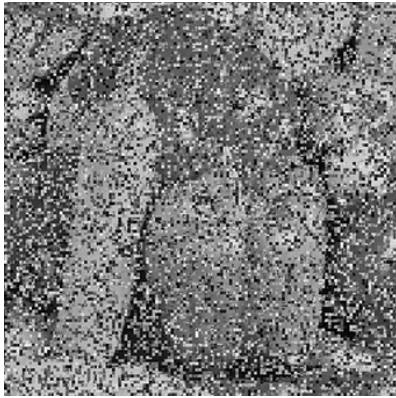


Figura 7: SSIM das imagens Lena reconstruídas versus probabilidade de erro de *bit* (ϵ) do canal BSC.

Em relação aos valores de PSNR e SSIM das imagens reconstruídas, os algoritmos supracitados conseguiram valores muito próximos e na qualidade visual das imagens reconstruídas tiveram praticamente o mesmo desempenho.

O VNS é o algoritmo com melhor desempenho, com menor tempo médio de execução para fins de RVQ e maior redução do índice de desordem dos dicionários, quando comparado aos algoritmos *Simulated Annealing* e Genético.



(a) Não organizado, SSIM = 0,121, PSNR = 12,68



(b) SA, SSIM = 0,236, PSNR = 16,83



(c) GA, SSIM = 0,236, PSNR = 16,62



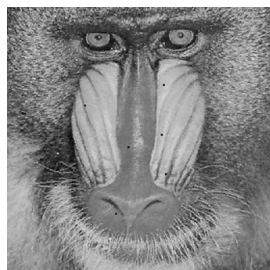
(d) VNS, SSIM = 0,246, PSNR = 16,83

Figura 8: Imagens reconstruídas com os dicionários organizados pelos algoritmos na transmissão da imagem Peppers com $N=256$ e pelo canal BSC com $\epsilon = 10^{-1}$.

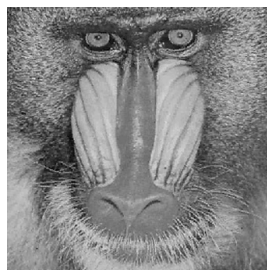
Como trabalhos futuros, outras técnicas de otimização serão utilizadas na atribuição de índices para minimizar o índice de desordem dos dicionários, como, por exemplo, algoritmos de enxames, como *Particle Swarm Optimization* (PSO) e *Firefly Algorithm* (FA). Pretende-se, também, avaliar a robustez dos dicionários organizados em outros tipos de canais de comunicação, tais como os canais com desvanecimento Rayleigh, Rice & Nakagami [19–22].

AGRADECIMENTOS

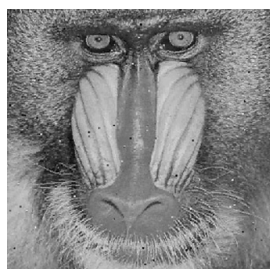
Agradecimentos pelo apoio financeiro da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).



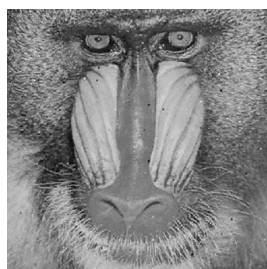
(a) $\epsilon = 10^{-4}$, SSIM = 0,663,
PSNR = 23,19



(b) $\epsilon = 10^{-4}$, SSIM = 0,664,
PSNR = 23,25



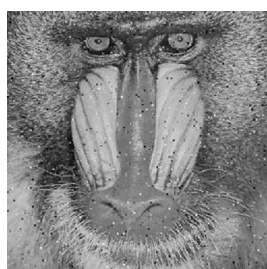
(c) $\epsilon = 10^{-3}$, SSIM = 0,652,
PSNR = 22,86



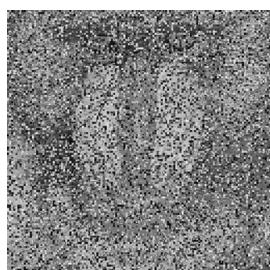
(d) $\epsilon = 10^{-3}$, SSIM = 0,658,
PSNR = 23,11



(e) $\epsilon = 10^{-2}$, SSIM = 0,550,
PSNR = 19,83.



(f) $\epsilon = 10^{-2}$, SSIM = 0,598,
PSNR = 21,82

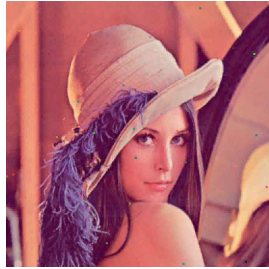


(g) $\epsilon = 10^{-1}$, SSIM = 0,160,
PSNR = 13,60

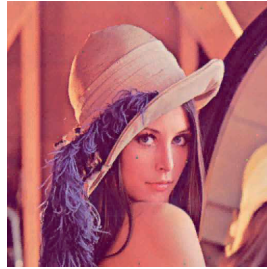


(h) $\epsilon = 10^{-1}$, SSIM = 0,282,
PSNR = 16,49

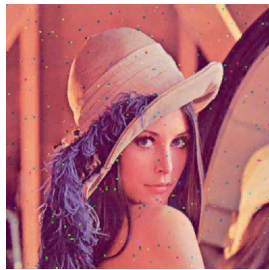
Figura 9: Imagens reconstruídas com os dicionários não organizados (*a, b, c e d*) e organizados com o VNS (*e, f, g e h*) na transmissão da imagem Mandrill com $N = 512$ pelo canal BSC.



(a) $\epsilon = 10^{-4}$, SSIM = 0,767, PSNR = 32,17



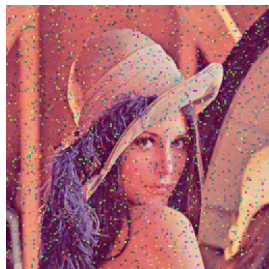
(b) $\epsilon = 10^{-4}$, SSIM = 0,769, PSNR = 32,41



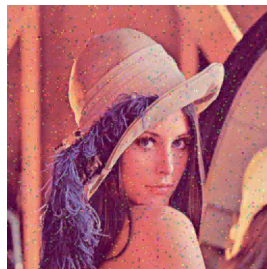
(c) $\epsilon = 10^{-3}$, SSIM = 0,743, PSNR = 30,10



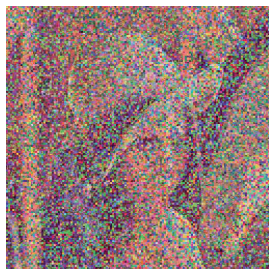
(d) $\epsilon = 10^{-3}$, SSIM = 0,759, PSNR = 31,78



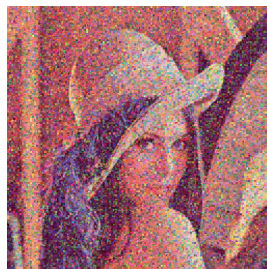
(e) $\epsilon = 10^{-2}$, SSIM = 0,648, PSNR = 23,66



(f) $\epsilon = 10^{-2}$, SSIM = 0,671, PSNR = 28,09



(g) $\epsilon = 10^{-1}$, SSIM = 0,390, PSNR = 15,09



(h) $\epsilon = 10^{-1}$, SSIM = 0,414, PSNR = 19,84

Figura 10: Imagens reconstruídas com os dicionários não organizados (*a*, *b*, *c* e *d*) e organizados com o VNS (*e*, *f*, *g* e *h*) na transmissão da imagem Lena com $N = 512$ pelo canal BSC.

ABSTRACT. This work addresses the problem of robust vector quantization in the scenario of image transmission over a binary symmetric channel. The application of the algorithm Variable Neighborhood Search (VNS) is introduced for the purpose of index assignment of codevectors. Simulation results reveal that the algorithm VNS, when compared to Simulated Annealing and Genetic algorithm, produces codebooks with the same robustness to channel errors with the spent of minor time.

Keywords: Robust Vector Quantization, optimization algorithms, digital image processing.

REFERÊNCIAS

- [1] R.K. Ahuja, Ö. Ergun, J.B. Orlin & A.P. Punnen. A survey of very large-scale neighborhood search techniques. *Discret. Appl. Math.*, **123**(1–3) (2002), 75–102.
- [2] S. Alkhalaf, O. Alfarraj & A.M. Hemeida. Fuzzy-VQ Image compression based hybrid PSO/GSA optimization algorithm. *International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1-6, (2015).
- [3] R.A. Azevedo, F. Madeiro, W.T.A. Lopes & E.A.O. Lima. A quasi random symbol interleaving technique applied to image transmission by noisy channels. *IEEE Latin America Transactions*, **14** (2016), 1078–1085.
- [4] R. de M. Estevão Filho, J.G.R.C. Gomes & A. Petraglia. Codebook calibration method for vector quantizers implemented at the focal plane of CMOS imagers. *IEEE Transactions on Circuits and Systems for Video Technology*, **26**(4) (2016), 750–761, April.
- [5] N. Farvardin. A study of vector quantization for noisy channels. *IEEE Trans. Inf. Theory*, **36**(4) (1990), 799–809, July.
- [6] A. Gersho & R. Gray. Vector quantization and signal compression. *Springer* (1992).
- [7] B. Huang, F. Henry & C. Guillemot. Mode dependent vector quantization with a rate-distortion optimized codebook for residue coding in video compression. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (2015).
- [8] P. Hansen & N. Mladenovic. A tutorial on variable neighborhood search. *Groupe D'études et de Recherche en Analyse des Décisions*, (2003).
- [9] P. Hansen, N. Mladenovic & D. Perez-Britos. Variable neighborhood decomposition search. *J. Heuristics*, **7**(4) (2001), 335–350.
- [10] C. Karri & U. Jena. Fast vector quantization using a bat algorithm for image compression. *Engineering Science and Technology*, **19**(2) (2016), 769–781.
- [11] S. Kirkpatrick, C.D. Gelatt & M.P. Vecchi. Optimization by simulated annealing. *Science*, **220**(4598) (1983), 671–680.
- [12] H.A.S. Leitão, W.T.A. Lopes & F. Madeiro. PSO algorithm applied to codebook design for channel-optimized vector quantization. *IEEE Latin America Transactions*, **13**(4) (2015), 961–967, April.
- [13] E.A. Lima, G.G.M. Melo, W.T.A. Lopes & F. Madeiro. Um novo algoritmo para atribuição de índices: avaliação em quantização vetorial de imagem. *TEMA – Tendências em Matemática Aplicada e Computacional*, **10** (2009), 167–177.

- [14] Y. Linde, A. Buzo & R. Gray. An algorithm for vector quantizer design. *IEEE Trans. Commun.*, **28**(1) (1980), 84–95, January.
- [15] W.T.A. Lopes, F. Madeiro, J.F. Galdino, B.G. Aguiar & M.S. Alencar. Diversidade em modulação aplicada a canais de comunicações móveis: Efeito dos erros de estimação de canal na transmissão de imagens. *Revista da Sociedade Brasileira de Telecomunicações*, **19**(1) (2004), 1–13.
- [16] N. Mladenovic & P. Hansen. Variable neighborhood search. *Comput. Oper. Res.*, **24**(11) (1997), 1097–1100.
- [17] N. Mladenovic, D. Urošević & S. Hanafi. Variable neighborhood search for the travelling deliveryman problem. *4OR*, **11**(1) (2012), 57–73.
- [18] M.A.S. Nasr, M.F. AlRahmawy & A.S. Tolba. Multi-scale structural similarity index for motion detection. *Journal of King Saud University – Computer and Information Sciences*, April (2016).
- [19] W.J.L. Queiroz, F. Madeiro, W.T.A. Lopes & M.S. Alencar. Error probability of multichannel reception with Theta-QAM scheme under correlated Nakagami-m fading. *Journal of Communication and Information Systems*, **29** (2014), 37–45.
- [20] W.J.L. Queiroz, F. Madeiro, W.T.A. Lopes & M. S. Alencar. Performance analysis of generalized QAM modulation under Eta-mu and Kappa-mu fading. *EURASIP Journal on Advances in Signal Processing*, pp. 1–10, (2013).
- [21] W.J.L. Queiroz, W.T.A. Lopes, F. Madeiro & M. S. Alencar. An alternative method to compute the bit error probability of modulation schemes subject to Nakagami-m fading. *EURASIP Journal on Advances in Signal Processing*, pp. 1–12, (2010).
- [22] W.J.L. Queiroz, M.S. Alencar, W.T.A. Lopes & F. Madeiro. Error probability in multichannel reception with M-QAM and R-QAM schemes under generalized fading. *IEICE Transactions on Communications*, pp. 2677–2687, (2010).
- [23] V. Ramdas, D. Mishra & S.S. Gorthi. Speech coding and enhancement using quantized compressive sensing measurements. *IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, pp. 1–5, (2015).
- [24] V. Severo, H.A.S. Leitão, J.B. Lima, W.T.A. Lopes & F. Madeiro. Algoritmo FA modificado aplicado ao projeto de quantizadores vetoriais, 12º Congresso Brasileiro de Inteligência Artificial (CBIC), (2015).
- [25] N. Siddique & H. Adeli. Computational Intelligence: synergies of fuzzy, neural networks and evolutionary computing. *John Wiley & Sons*, (2013).
- [26] D. Tsolakis, G.E. Tsekouras & J. Tsimikas. Fuzzy vector quantization for image compression based on competitive agglomeration and a novel codeword migration strategy. *Eng. Appl. Artif. Intell.*, **25**(6) (2012), 1212–1225.
- [27] Z. Wang, A.C. Bovik, H.R. Sheikh & E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, **13**(4) (2004), 600–612, April.
- [28] D. Whitley & A.M. Sutton. Genetic algorithms: a survey of models and methods. *Handbook of Natural Computing*, pp. 637–671, (2012).