

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

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 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 [5].

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 [3, 10]. Para um dicionário de tamanho N , há $N!$ possíveis organizações do dicionário a serem avaliadas. Considerando um 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 [3, 9] e Algoritmo Genético [8, 15].

Este artigo apresenta a utilização do algoritmo *Variable Neighborhood Search* [13] 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: Seção 2 contempla a Quantização Vetorial e a Quantização Vetorial Robusta (RVQ – *Robust Vector Quantization*); na Seção 3 o algoritmo *Variable Neighborhood Search* (VNS) é descrito; a Seção 4 aborda o algoritmo VNS aplicado a RVQ; a metodologia utilizada neste trabalho é apresentada na Seção 5; na Seção 6 são apresentados os resultados das simulações e na Seção 7 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 [5].

A QV tem sido usada amplamente como técnica de compressão de voz [16], vídeo [4] e imagens digitais [2, 17]. 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 $W = \{w_i : i = 0, 1, \dots, N - 1\}$ 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 [5]. 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*) [5].

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, consequentemente, 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 [3], [10], [12].

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 [10, 12], 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.

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 [13]. 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) [13, 14] combina a estratégia dos algoritmos de busca local com a mudança dinâmica de vizinhanças, com isso, podendo escapar dos mínimos locais [6, 7]. 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 usado como o espaço de busca. Encontre uma solução inicial s .
- (2) Faça $m = 1$;
- (3) Repetir 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 .

4. 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) Repetir 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, permute 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.

5. Metodologia

O algoritmo *Variable Neighborhood Search* (VNS) foi aplicado à atribuição de índices aos vetores-código do dicionários. Os resultados obtidos do VNS foram comparados com os algoritmos Simulated Annealing [3, 9] e Algoritmo Genético [8, 15].

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 foram utilizadas as imagens Airplane, Elaine, Goldhill e Peppers (Fig. 1). Todas as imagens utilizadas têm dimensão 512×512 *pixels* e são codificadas em 8 *bpp* (256 níveis de cinza) no formato PGM. O algoritmo para projeto dos dicionários foi o LBG [11] com limiar de distorção $\epsilon = 10^{-3}$.



(a) Airplane.

(b) Elaine.

(c) Goldhill.

(d) Peppers.

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

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[®].

5.1. 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) utilizados por Lopes [12]. A função de avaliação dos dicionários organizados pelo SA foi o índice de desordem definido na Equação 2.2. 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 a Tabela 1.

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 Tabela 1.

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

N	t_0	t_f	α	<i>Avaliações</i>	ΔId
32	10.000	1	0,95	180.000	53,57%
64	10.000	1	0,95	180000	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%

5.2. 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. 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.

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 $\Delta Id = 69,54\%$.

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

N	P	Tc	Tm	Ps	<i>Avaliações</i>	ΔI_d
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%

5.3. 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. 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	<i>Avaliações</i>	ΔI_d
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%

5.4. 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 e Peppers (Fig. 4) foram utilizadas como conjunto de teste, transmitidas pelo canal BSC 50 vezes para cada probabilidade ϵ .

Os melhores dicionários organizados pelos algoritmos *Simulated Annealing*, Algoritmo Genético e *Variable Neighborhood Search* foram selecionados para reconstruir as imagens após a transmissão pelo canal. Para medir a qualidade das imagens foi utilizado o valor da relação sinal-ruído de pico (*PSNR – Peak Signal-to-Noise Ratio*) entre a imagem original e a imagem recebida após passar pelo canal BSC.

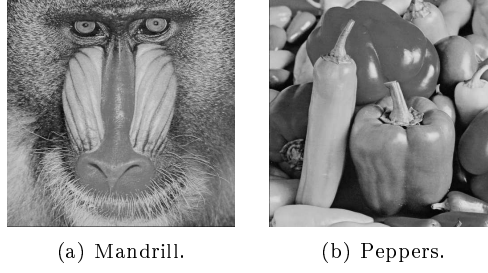


Figura 2: Imagens utilizadas como conjunto de teste, transmitidas no *BSC*.

6. Resultados

Inicialmente, é importante observar que o algoritmo VNS obteve maior 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 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 ao 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 ao 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 ao 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

Tabela 7: *PSNR* (dB) médio em relação ao 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 ao 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

O tempo médio gasto pelo VNS foi menor que o do SA e do GA na organização dos índices dos dicionários. A Fig. 3 apresenta o tempo médio dos algoritmos, onde se observa uma economia de tempo médio de aproximadamente 18% e 32%, respectivamente, em relação aos algoritmos SA e AG, para $N = 256$.

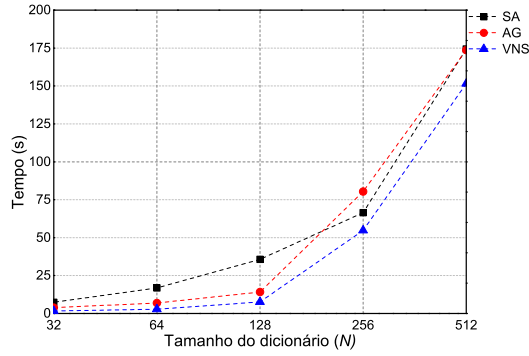


Figura 3: Tempo de execução médio gasto pelos algoritmos.

O desempenho dos algoritmos para a transmissão da imagem Peppers, à luz dos valores de *PSNR* das imagens reconstruídas, pode ser observado nas Fig. 4. Observa-se que a organização dos dicionários (por SA, GA ou VNS) contribuiu para aumentar a robustez aos erros do canal, levando a imagens reconstruídas com *PSNR* maior que as obtidas 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 Fig. 5. Inspeções visuais revelam que as imagens reconstruídas com dicionários organizados pelos algoritmos SA, GA e VNS tiveram praticamente a mesma qualidade.

A Fig. 6 apresenta a imagem Mandrill reconstruída com os dicionários organiza-

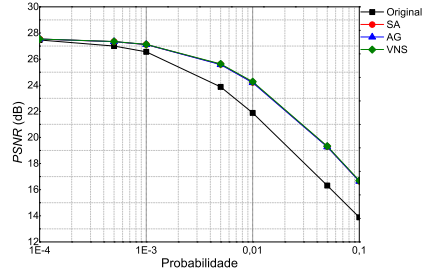
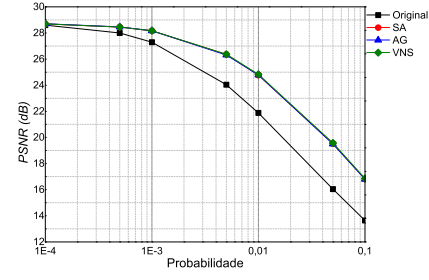
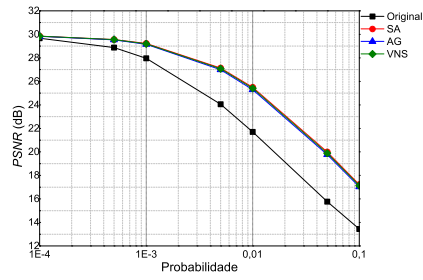
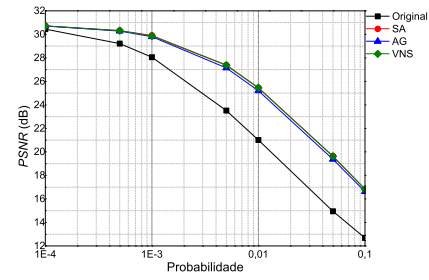
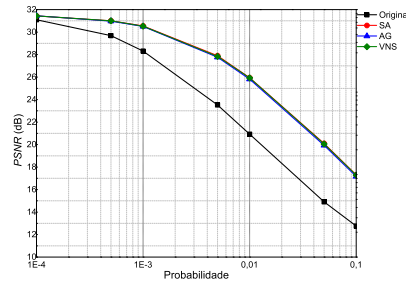
(a) $N = 32$.(b) $N = 64$.(c) $N = 128$.(d) $N = 256$.(e) $N = 512$.

Figura 4: Desempenho dos algoritmos por probabilidade de erro de *bit* (ϵ) do canal *BSC*.

dos 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 robu-

tez aos erros de canal obtida com a atribuição de índices pelo algoritmo VNS.

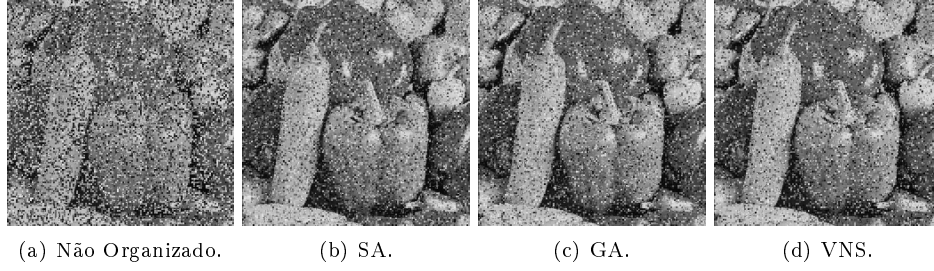


Figura 5: Imagens reconstruídas com os dicionários organizados pelos algoritmos na transmissão da imagem Peppers com $N=256$ e pelo canal BSC.

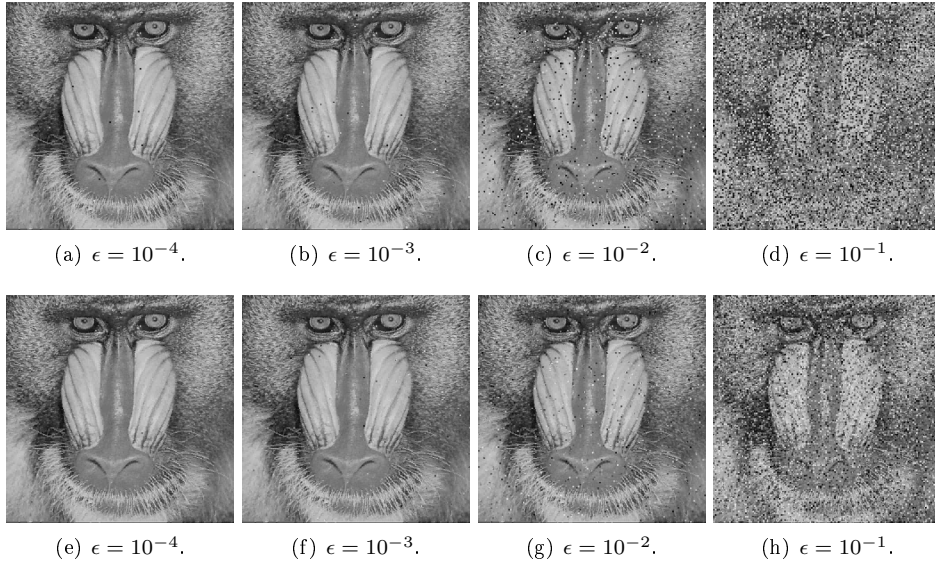


Figura 6: Imagens reconstruídas com os dicionários não organizados (*a*, *b*, *c* e *d*) e organizados com o VNS (*e*, *f*, *g* e *h*) na transmissão da imagem Mandrill com $N=256$ pelo canal BSC.

7. 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.

Em relação aos valores de PSNR 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.

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.

Agradecimentos

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

Index Assignment for Robust VQ Using Variable Neighborhood Search Algorithm

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, and A. P. Punnen, A survey of very large-scale neighborhood search techniques, *Discret. Appl. Math.*, vol. 123, *n*º. 1–3, pp. 75–102, 2002.
- [2] M. Canejo, J. Galvão, F. Madeiro, Inserção de marca d'água digital em imagens digitais utilizando quantização vetorial com particionamento do dicionário por PSO, *XII Congresso Brasileiro de Inteligência Computacional*, Curitiba, PR, 2015.
- [3] N. Farvardin, A study of vector quantization for noisy channels, *IEEE Trans. Inf. Theory*, vol. 36, *n*º. 4, pp. 799–809, Jul. 1990.
- [4] 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.

- [5] A. Gersho and R. Gray, Vector quantization and signal compression. *Springer*, 1992.
- [6] P. Hansen and N. Mladenovic, A tutorial on variable neighborhood search, em "Groupe D'études et de Recherche en Analyse des Décisions", 2003.
- [7] P. Hansen, N. Mladenovic, and D. Perez-Britos, Variable neighborhood decomposition search, *J. Heuristics*, vol. 7, n^o. 4, pp. 335–350, 2001.
- [8] J. H. Holland, "Adaptation in natural and artificial systems", The University of Michigan Press, 1975.
- [9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, *Science*, vol. 220, n^o. 4598, pp. 671–680, 1983.
- [10] E. A. Lima, G. G. M. Melo, W. T. A. Lopes, and F. Madeiro, Um novo algoritmo para atribuição de índices: Avaliação em quantização vetorial de imagem, *TEMA – Tendências em Matemática Aplicada e Computacional*, vol. 10, pp. 167–177, 2009.
- [11] Y. Linde, A. Buzo, and R. Gray, An algorithm for vector quantizer design, *IEEE Trans. Commun.*, vol. 28, n^o. 1, pp. 84–95, Jan. 1980
- [12] W. T. A. Lopes, F. Madeiro, J. F. Galdino, B. G. Aguiar, and 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*, vol. 19, n^o. 1, pp. 1–13, 2004.
- [13] N. Mladenovic and P. Hansen, Variable neighborhood search, *Comput. Oper. Res.*, vol. 24, n^o. 11, pp. 1097–1100, 1997.
- [14] N. Mladenovic, D. Urošević, and S. Hanafi, Variable neighborhood search for the travelling deliveryman problem, *4OR*, vol. 11, n^o. 1, pp. 57–73, 2012.
- [15] J. S. Pan, F. R. McInnes, and M. A. Jack, VQ codevector index assignment using genetic algorithms for noisy channels, em "Fourth International Conference on Spoken Language Processing (ICSLP)", vol. 1, pp. 295–298, 1996.
- [16] 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;
- [17] D. Tsolakis, G. E. Tsekouras, and J. Tsimikas, Fuzzy vector quantization for image compression based on competitive agglomeration and a novel codeword migration strategy, *Eng. Appl. Artif. Intell.*, vol. 25, n^o. 6 (2012), pp. 1212–1225.