

# Uma Função Objetivo para o Problema de Otimização do Projeto de Quantizadores Vetoriais Robustos

Wesley Vieira de Santana  
Universidade de Pernambuco  
Recife, Brasil  
wvs1@poli.br

Waslon Terllizzie Araújo Lopes  
Universidade Federal da Paraíba  
João Pessoa, Brasil  
waslon@ieee.org

Francisco Madeiro  
Universidade de Pernambuco  
Recife, Brasil  
madeiro@poli.br

**Resumo**—A Quantização Vetorial Robusta (QVR) é um problema de otimização de natureza combinatorial, o qual envolve a atribuição de índices aos vetores-código do dicionário. No processo de otimização da QVR, a função objetivo utilizada é o índice de desordem do dicionário, o qual deve ser minimizado. Neste trabalho, foi proposta e avaliada a utilização de uma função de densidade de probabilidade dos vetores-código para ponderar no cálculo do índice de desordem. Para a tarefa de otimização dos dicionários foram utilizados os algoritmos Busca por Cardume (FSS – *Fish School Search*), Otimização por Enxame de Partícula (PSO – *Particle Swarm Optimization*), *Simulated Annealing* (SA) e o Algoritmo Genético (GA – *Genetic Algorithm*). Os resultados de simulação da transmissão de imagens por Canal Binário Simétrico (BSC – *Binary Symmetric Channel*) mostram que a consideração do índice de desordem proposto, como função objetivo, leva a um melhor desempenho comparado a quando não utilizado, para todos os algoritmos de otimização usados neste trabalho.

**Palavras-chaves**—Transmissão de imagens, quantização vetorial robusta, inteligência de enxames, otimização combinatorial, atribuição de índices.

## I. INTRODUÇÃO

A Quantização Vetorial (QV) é uma técnica de compressão com perdas utilizada em muitos sistemas de comunicação digital [1]–[4]. A QV fundamenta-se na Teoria da Distorção Versus Taxa, formulada por Shanon, na qual é possível encontrar um melhor desempenho de compressão codificando blocos de amostras, ou seja, vetores, em vez de escalares [5].

A QV tem seu desempenho comprometido em canais ruidosos, em virtude dos erros na decodificação. O combate à degradação de desempenho dos sistemas baseados em QV, em canais ruidosos, é um desafio que tem sido alvo de pesquisas [6], [7].

A Quantização Vetorial Robusta (QVR) é uma técnica utilizada para minimizar os impactos provocados por canais ruidosos. Nesta técnica, o projeto do dicionário é realizado considerando um canal sem ruído e, uma vez projetado, é realizada uma Atribuição de Índice (IA – *Index Assignment*) aos vetores-códigos do dicionário por meio de um algoritmo de

otimização. Uma atribuição adequada dos índices aos vetores que compõem o dicionário pode fazer com que os vetores decodificados estejam próximos, em média, dos vetores originais. Este processo de organização torna o dicionário mais robusto aos erros provocados pelo canal [8].

No processo de otimização da QVR, a figura de mérito utilizada é o índice de desordem do dicionário. O alvo da otimização é minimizar o índice de desordem para obter dicionários mais robustos aos erros introduzidos pelo canal. Em [9], Zeger e Gersho apresentaram propriedades assintóticas que assumiam a ocorrência de erros em apenas um único *bit* nos índices transmitidos. Desta forma, a função objetivo a ser minimizada levava em consideração apenas a contribuição dos vetores permuta com distância Hamming igual a 1. Em [10], Lopes propôs uma simplificação adicional admitindo que os vetores-códigos são equiprováveis. Com estas simplificações, a complexidade computacional do processo de otimização torna-se menos dispendiosa.

Zeger e Gersho [9] apresentaram o *Binary Switching Algorithm* (BSA) para realizar a tarefa de organização dos índices do dicionário com o objetivo de torná-lo robusto. Farvardin [8] propôs em 1990 a utilização do *Simulated Annealing* (SA) para a organização dos índices do dicionário. O SA tornou-se um algoritmo amplamente utilizado no problema de atribuição de índice [11], sendo utilizado frequentemente como referência de comparação para avaliação de desempenho de outros algoritmos [7], [12].

Algoritmos de inteligência computacional (IC), particularmente da área de computação bioinspirada, podem ser utilizados para resolver o problema de IA aplicado a QVR, como o algoritmo de Busca por Cardume (FSS – *Fish School Search*) [6].

Este trabalho tem como foco a QVR para transmissão de imagens. É apresentada uma função de densidade de probabilidade não uniforme dos vetores-códigos no cálculo do índice de desordem do dicionário, além de assumir a ocorrência de até dois erros de bit nos índices transmitidos. A tarefa de minimizar o índice de desordem do dicionário foi realizada pelos algoritmos Busca por Cardume (FSS – *Fish School Search*), Otimização por Enxame de Partículas (PSO – *Particle*

Swarm Optimization), Simulated Annealing (SA) e o Algoritmo Genético (GA – *Genetic Algorithm*). A avaliação da qualidade das imagens reconstruídas foi realizada em termos de Similaridade por Características (FSIM – *Feature Similarity*) e Similaridade Estrutural (SSIM – *Structural Similarity*).

Este trabalho está organizado da seguinte forma: a Seção II apresenta a problemática e fundamentos da QVR, a Seção III discute a utilização de uma função de densidade de probabilidade não uniforme para ponderar o índice de desordem e apresenta a função objetivo proposta neste trabalho, a Seção IV apresenta os resultados e a Seção V contém as conclusões.

## II. QUANTIZAÇÃO VETORIAL ROBUSTA

A Figura 1 apresenta um diagrama de blocos para um quantizador vetorial em um canal ruidoso, em que  $i$  e  $j$  representam os índices transmitido e recebido, respectivamente. Tanto o codificador quanto o decodificador possuem o mesmo dicionário projetado contendo os vetores-códigos.

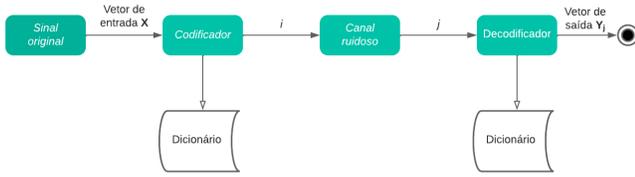


Figura 1: Diagrama de um sistema de transmissão baseado em QV em um canal com ruído.

A QVR visa diminuir, por meio do processo de atribuição de índices, sem a necessidade de adição de *bits* na codificação, a sensibilidade aos erros provocados por um canal ruidoso. Os erros do canal podem modificar o índice enviado pelo codificador, levando o decodificador a escolher de forma errada um vetor-código diferente do codificado. Os efeitos provocados pelo ruído do canal podem resultar em uma distorção relevante do sinal reconstruído no receptor. A degradação do sinal é atenuada com a atribuição de índices com grande/pequena distância de Hamming a vetores-código com grande/pequena distância euclidiana. Esta organização dos índices do dicionário visa fazer com que os vetores decodificados incorretamente estejam próximos, em média, dos vetores originais. Desta forma, a distorção média introduzida pelos erros do canal pode ser minimizada [10], [13].

Para um dicionário com tamanho  $N = 2^b$ , em que  $b$  é um inteiro positivo, cada um dos  $N$  índices que representam os vetores-códigos do dicionário pode ser apresentado na forma binária por  $\{0, 1\}^b$ , em que  $b$  é o comprimento da palavra-código.

A função densidade de probabilidade  $p(\cdot)$  dos vetores-código, determinada pelas estatísticas da fonte, é dada por

$$p(\mathbf{y}_k) = Pr[\mathbf{x} \in R_k], \quad (1)$$

sendo  $Pr[\mathbf{x} \in R_k]$  a probabilidade de o vetor de entrada  $\mathbf{x}$  pertencer à partição  $R_k$ . O valor de  $p(\mathbf{y}_k)$  é a probabilidade de

o vetor-código  $\mathbf{y}_k$  ser selecionado pelo codificador para representar o vetor de entrada. Esta distribuição de probabilidade é determinada pelas estatísticas dos vetores de entrada e pelas partições do dicionário, pois assume-se que os vetores da fonte são independentes do ruído do canal. A probabilidade  $p(\mathbf{y}_k)$  pode ser calculada a *priori* por meio da frequência relativa com a qual um determinado vetor-código é escolhido como a melhor aproximação para o vetor de entrada a ser codificado [10].

Os efeitos provocados pelo ruído do canal podem resultar em uma distorção relevante do sinal reconstruído no receptor. Uma atribuição adequada dos índices aos vetores que compõem o dicionário pode fazer com que os vetores decodificados estejam próximos, em média, dos vetores originais. De forma a otimizar o desempenho do sistema de codificação para um dado dicionário, o valor da distorção provocada pelos erros nos índices deve ser minimizado considerando as possíveis permutações dos vetores-código do dicionário. Para fins de minimização da distorção provocada pelos erros do canal nos índices transmitidos, é suficiente minimizar

$$D_\pi = \min_{\pi \in S_N} E[d(\mathbf{y}_i, \mathbf{y}_j)]. \quad (2)$$

O valor de  $E[d(\mathbf{y}_i, \mathbf{y}_j)]$  representa a distorção esperada dado o vetor-código transmitido em relação ao vetor-código recebido, sobre todas as permutações possíveis em  $S_N$ .

O procedimento para obtenção da permutação ótima envolve uma grande complexidade computacional. Para tanto, é preciso explorar o espaço  $S_N$  com os  $N!$  arranjos possíveis do dicionário. Como exemplo, para um dicionário de tamanho 512 tem-se  $512! = 3,47 \times 10^{1166}$  arranjos possíveis.

Considerando um Canal Binário Simétrico (BSC – *Binary Symmetric Channel*) com probabilidade de erro  $p$ , a probabilidade de uma palavra-código de comprimento  $b$  ser transmitida e recebida no decodificador por uma outra palavra-código particular de distância Hamming  $m$ , é dada utilizando a distribuição de Bernoulli

$$q_m = \binom{b}{m} p^m (1-p)^{b-m}. \quad (3)$$

Considerando cada índice binário  $q \in \{0, 1\}^b$  representando um vetor-código,  $m$  sendo um inteiro,  $1 \leq m \leq b$ , pode-se obter um conjunto  $V^m(q)$  de índices que fazem vizinhança com  $q$  que possuem distância de Hamming  $m$  em relação a  $q$ . Para uma palavra binária  $q$  de comprimento  $b$ , o número de índices com distância de Hamming  $m$  é dado por

$$V^m(q) = \binom{b}{m} = \frac{b!}{(b-m)!m!}. \quad (4)$$

$V^m(q)$  pode ser interpretado como o conjunto de índices para os quais  $q$  pode ser transformado como resultado de erros em  $m$  *bits*.

Supondo  $b = 3$ , os conjuntos para o índice “1” em decimal são:  $V^1(1) = \{0, 3, 5\}$ ,  $V^2(1) = \{2, 4, 7\}$  e  $V^3(1) = \{6\}$ .

Desta forma, a distorção para um vetor-código em particular  $\mathbf{y}_k$ , em relação à permutação  $\pi$ , é dada por [8]

$$C_{\pi(\mathbf{y}_k)} = p(\mathbf{y}_k) \sum_{m=1}^b q_m d(\mathbf{y}_k, \mathbf{y}_{\pi^{-1}(V^m)}). \quad (5)$$

A função custo, que mede a contribuição total da distorção provocada pelos erros de canal, a qual deve ser minimizada, é dada por [8]

$$D_{\pi} = \sum_{k=0}^{N-1} C_{\pi(\mathbf{y}_k)}. \quad (6)$$

Assumindo a ocorrência de erros em apenas um *bit*, a minimização de (6) não depende do valor de  $p$ , e portanto, demanda uma menor complexidade computacional. Para minimizar  $D_{\pi}$ , neste caso, é suficiente encontrar a permutação  $\pi$  que minimize [9]

$$d_{\pi} = \sum_{k=0}^{N-1} C_{\pi}^{(1)}(\mathbf{y}_k). \quad (7)$$

Em outras palavras, a Equação 7 leva em consideração apenas a contribuição dos vetores permuta com distância Hamming igual a 1.

Uma simplificação adicional pode ser obtida supondo que os vetores  $\mathbf{y}_k$  são equiprováveis. Logo, a minimização deve ser feita em termos de [10]

$$I_{des}(\pi) = \sum_{k=0}^{N-1} \sum_{w \in V^1} d(\mathbf{y}_k, \mathbf{y}_{\pi^{-1}(w)}). \quad (8)$$

Esta equação é chamada índice de desordem do dicionário.

### III. ÍNDICE DE DESORDEM DO DICIONÁRIO PROPOSTO

Zeger e Gersho em [9] apresentaram as propriedades assintóticas, resultando na Equação (7) como suficiente para minimizar a distorção. Essa equação admite a ocorrência de apenas um erro de *bit* no índice transmitido. Desta forma, no cálculo do índice de desordem, apenas os vetores permuta com distância Hamming igual a 1 são considerados.

Uma outra simplificação é apresentada na Equação 8 [10], a qual considera os vetores-código equiprováveis, isto é, assume-se que a função densidade de probabilidade dos vetores-códigos é uniforme. Desde então, trabalhos vêm utilizando essas equações simplificadas [7], [13].

Em 2012, Barreto *et al.* [14] realizaram um estudo sobre a influência da distância Hamming no índice de desordem, em que foram consideradas distâncias Hamming iguais a 1 e 2 e vetores-código equiprováveis para o cálculo do índice de desordem.

No presente trabalho, propõe-se considerar a função densidade de probabilidade dos vetores-código não uniforme, além de admitir até dois erros de *bits* na palavra-código transmitida. Desta forma, tem-se que  $q_m = 0$  para  $m > 2$ . Logo, aplicando essas condições à Equação (5), obtém-se

$$C_{\pi(\mathbf{y}_k)}^{(2)} = p(\mathbf{y}_k) \sum_{m=1}^b q_m d(\mathbf{y}_k, \mathbf{y}_{\pi^{-1}(V^m)}), \quad (9)$$

em que  $C_{\pi(\mathbf{y}_k)}^{(2)}$  é o valor da contribuição da distorção do vetor-código  $\mathbf{y}_k$  para distância Hamming igual 2. O valor da probabilidade  $q_m$  é encontrado por meio da Equação 3.

Ao levar em consideração a função densidade de probabilidade dos vetores-código, determinada pelas estatísticas da fonte, a probabilidade de cada vetor  $p(\mathbf{y}_k)$  terá influências diferentes no valor do índice de desordem do dicionário. Na prática, significa que, no processo de otimização da QVR, os vetores-código com maior probabilidade de acesso terão um maior peso na contribuição do valor do índice de desordem. Portanto, o índice de desordem do dicionário proposto, o qual deve-se minimizar, é dado por

$$I_{des-proposto}(\pi) = \sum_{k=0}^{N-1} C_{\pi(\mathbf{y}_k)}^{(2)}. \quad (10)$$

Para determinar a função densidade de probabilidade dos vetores-código  $p(\mathbf{y}_k)$ , propõe-se utilizar o mesmo conjunto de treino aplicado ao projeto do dicionário, para estimar as estatísticas da fonte.

A Equação (10) implica um aumento na complexidade computacional no processo de atribuição de índices em relação à Equação (8), contudo, a atribuição de índices do dicionário é feita *off-line*, isto é, não ocorre durante a transmissão do sistema baseado em QV. Portanto, o índice de desordem proposto não provoca aumento de complexidade no processo de transmissão.

## IV. RESULTADOS

Na QVR, em particular, os algoritmos são usados para reduzir o índice de desordem dos dicionários. Os resultados das simulações envolvem a transmissão de imagens por um Canal Binário Simétrico (BSC — *Binary Symmetric Channel*).

Todas as simulações foram realizadas em um computador com CPU Intel Core i5 7200U, 8 GB de memória RAM e SSD de 240 GB rodando o Windows 10 64 bits. A linguagem utilizada para implementar os algoritmos foi Python (versão 3.7). Foram projetados dicionários com o algoritmo LBG (*Lind-Buzo-Gray*) [15].

Os dicionários utilizados têm tamanhos  $N = 128, 256$  e  $512$  com dimensão  $K = 16$ , isto é, blocos de imagens com  $4 \times 4$  *pixels*. As imagens, no formato PGM (*Portable Gray Map*), utilizadas para o conjunto de treino e estimativa da função densidade de probabilidade dos vetores-código foram: Barbara, Elaine, Goldhill, Lena e Peppers, com dimensões  $256 \times 256$  *pixels* codificadas a 8 bits por *pixel*.

As funções objetivo a minimizar são: a Equação 8 ( $I_{des}$ ), descrita na Seção II, e a Equação 10 ( $I_{des-proposto}$ ), referente ao índice de desordem proposto. Para fins de cálculo da probabilidade  $q_m$ , necessária para a Equação 10, assume-se uma probabilidade de erro de canal  $p = 0, 1$ , admitindo um cenário com condições severas de transmissão. Os algoritmos FSS, GA, PSO e SA são usados para a tarefa de otimização.

A Tabela I apresenta os parâmetros tamanho de cardume,  $step_{inst-inicial}$ ,  $step_{inst-final}$  e condição de parada configurados

no algoritmo FSS. Foi utilizado o algoritmo FSS modificado proposto em [6]. Para o algoritmo SA, os parâmetros foram configurados conforme Tabela II, baseado em [10], temperatura inicial ( $T_0$ ), temperatura final ( $T_f$ ) e constante de resfriamento  $\alpha$ . Na Tabela III são definidos os parâmetros do GA, população, taxa de cruzamento ( $T_c$ ), taxa de mutação ( $T_m$ ), elitismo e o critério de parada. Estes parâmetros foram baseados em [7]. Foi utilizada a técnica *Path Relinking* aplicada ao PSO e cujos parâmetros estão descritos na Tabela IV, sendo  $c_1$  o fator de individualidade e  $c_2$  o fator de sociabilidade, baseado em [16], [17].

Tabela I: Parâmetros do FSS.

| Parâmetros            | $N = 128$                                     | $N = 256$ | $N = 512$ |
|-----------------------|---|-----------|-----------|
| Tamanho do cardume    | 30  | 30        | 30        |
| $step_{inst-inicial}$ | 0,8   | 0,8       | 0,8       |
| $step_{inst-final}$   | 0,4   | 0,4       | 0,4       |
| Critério de parada    | 100 iterações sem melhorias ou 2000 iterações |           |           |

Tabela II: Parâmetros do SA.

| Parâmetros | $N = 128$ | $N = 256$ | $N = 512$ |
|------------|-----------|-----------|-----------|
| $T_0$      | 10.000    | 50.000    | 50.000    |
| $T_f$      | 1         | 1         | 10        |
| $\alpha$   | 0,95      | 0,97      | 0,97      |

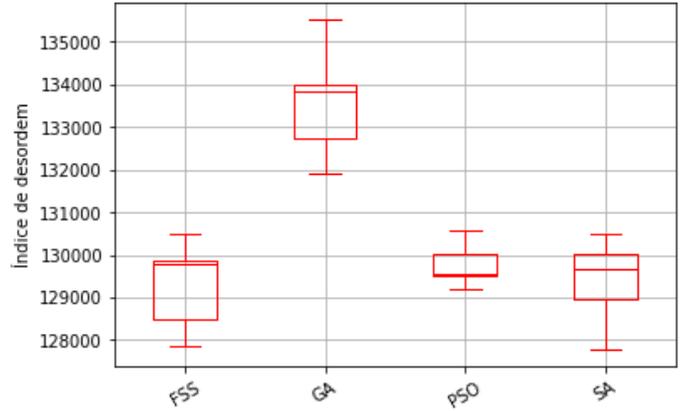
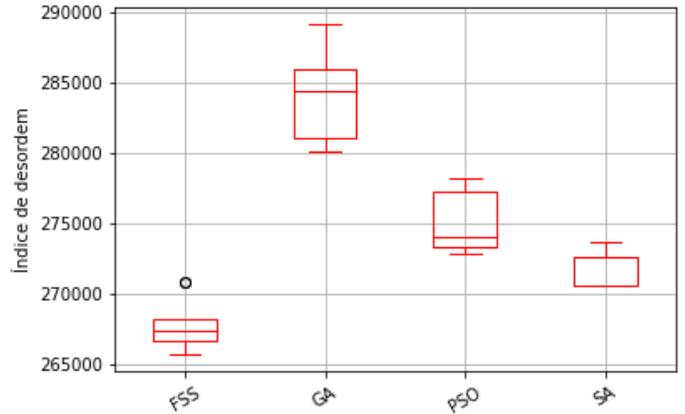
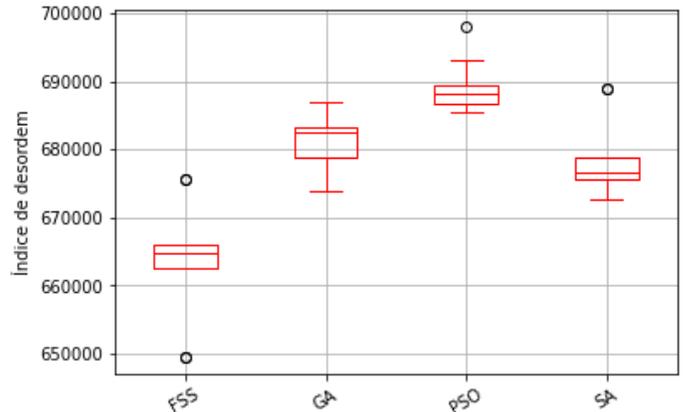
Tabela III: Parâmetros do GA.

| Parâmetros         | $N = 128$                                   | $N = 256$ | $N = 512$ |
|--------------------|---|-----------|-----------|
| População          | 30  | 30        | 30        |
| $T_c$              | 0,8   | 0,9       | 0,9       |
| $T_m$              | 0,004                                       | 0,005     | 0,002     |
| Elitismo           | 0,3   | 0,3       | 0,3       |
| Critério de parada | 100 gerações sem melhorias ou 2000 gerações |           |           |

Tabela IV: Parâmetros do PSO.

| Parâmetros         | $N = 128$                                     | $N = 256$ | $N = 512$ |
|--------------------|---|-----------|-----------|
| Tamanho do enxame  | 30  | 30        | 30        |
| $c_1$              | 0,7   | 0,5       | 0,5       |
| $c_2$              | 0,8   | 0,5       | 0,5       |
| Critério de parada | 100 iterações sem melhorias ou 2000 iterações |           |           |

As Figuras 2 a 7 apresentam o *boxplot* dos valores do índice de desordem como também para o índice de desordem proposto, obtidos pelos algoritmos no processo de atribuição dos índices dos dicionários. Os valores foram obtidos por meio do processamento de 10 dicionários independentes. Destaca-se a superioridade de desempenho do algoritmo FSS em relação ao GA, PSO e SA para os dicionários de tamanhos  $N = 256$  e  $512$ , tanto utilizando o índice de desordem como o índice de desordem proposto. O algoritmo FSS apresentou melhores resultados na maior parte dos casos para o problema da atribuição de índices. Os dicionários de tamanho  $N = 256$  e  $512$  tiveram uma maior redução do índice de desordem com o FSS, sendo a redução mais expressiva para os dicionários de tamanho  $N = 512$ .


 Figura 2: *Boxplot* dos valores do índice de desordem obtidos pelos algoritmos FSS, GA, PSO e SA para dicionários de tamanho 128.

 Figura 3: *Boxplot* dos valores do índice de desordem obtidos pelos algoritmos FSS, GA, PSO e SA para dicionários de tamanho 256.

 Figura 4: *Boxplot* dos valores do índice de desordem obtidos pelos algoritmos FSS, GA, PSO e SA para dicionários de tamanho 512.

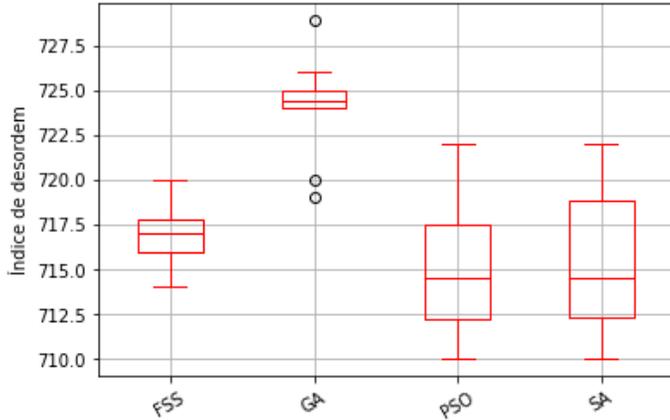


Figura 5: *Boxplot* dos valores do índice de desordem proposto obtidos pelos algoritmos FSS, GA, PSO e SA para dicionários de tamanho 128.

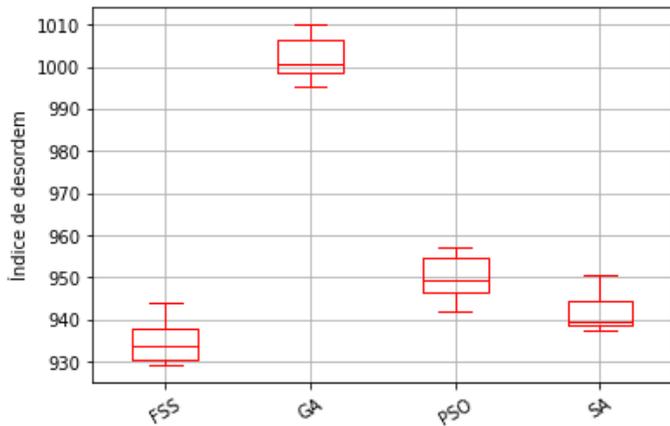


Figura 6: *Boxplot* dos valores do índice de desordem proposto obtidos pelos algoritmos FSS, GA, PSO e SA para dicionários de tamanho 256.

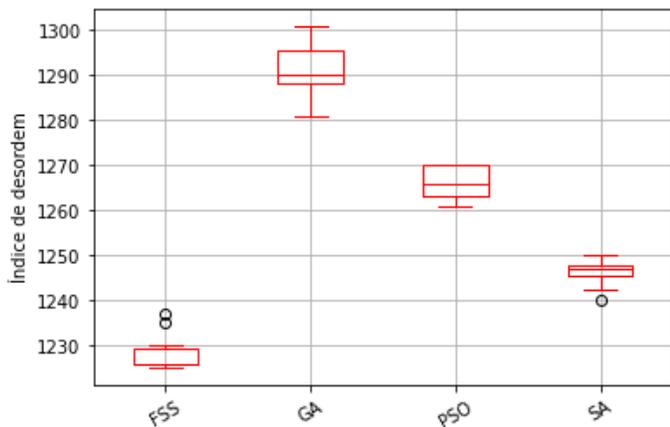


Figura 7: *Boxplot* dos valores do índice de desordem proposto obtidos pelos algoritmos FSS, GA, PSO e SA para dicionários de tamanho 512.

Para avaliar a robustez dos dicionários organizados usando os algoritmos de otimização, as imagens Lena, Mandrill e Boat, apresentadas nas Figuras 8 a 10, foram transmitidas 10 vezes por um canal BSC considerando vários valores de probabilidade de erro de *bit*,  $p$  (0,0005, 0,001, 0,005, 0,01, 0,05 e 0,1). Para cada algoritmo, foi utilizado o dicionário com menor índice de desordem. A avaliação da qualidade das imagens reconstruídas foi realizada em termos de Similaridade por Características (FSIM – *Feature Similarity*) [18] e Similaridade Estrutural (SSIM – *Structural Similarity*) [19].

As Figuras 11 a 16 apresentam as curvas dos valores médios de FSIM e SSIM em função da probabilidade de erro de *bit* de canal para a transmissão das três imagens, sendo cada imagem transmitida 10 vezes. Nas figuras, são apresentados os desempenhos dos dicionários originais, dos dicionários organizados com a função objetivo índice de desordem e os organizados com a função objetivo índice de desordem proposta. Observa-se, para todos os tamanhos, que os dicionários organizados apresentaram superioridade ao dicionário original. Realizando um comparativo entre os dicionários organizados pelo mesmo algoritmo, em todos os tamanhos, os dicionários organizados, utilizando o índice de desordem proposto, destacam-se em relação aos organizados com a função índice de desordem, apresentando melhor desempenho nas duas métricas utilizadas. Nos tamanhos 256 e 512, esta superioridade fica mais acentuada.

As Figuras 17 e 18 apresentam as imagens Mandrill reconstruídas após transmissão por um canal BSC com probabilidade de erro de *bit* igual a 0,01. As imagens foram submetidas às mesmas condições de ruído de canal. Na Figura 17, para os dicionários organizados (GA, PSO, SA, FSS), foi utilizada a função índice de desordem. Enquanto que na Figura 18, a função índice de desordem proposta foi aplicada aos dicionários organizados. Observa-se que a qualidade visual das imagens reconstruídas com dicionários que utilizaram o índice de desordem proposto, em geral, foi superior às que não utilizaram. Através de inspeções visuais das regiões destacadas das imagens reconstruídas, é possível perceber que os erros de decodificação foram menos incômodos para os dicionários que utilizaram o índice de desordem proposto.

## V. CONCLUSÕES

Neste trabalho, propôs-se a utilização de uma função de densidade de probabilidade dos vetores-código para ponderar no cálculo do índice de desordem, o qual é a figura de mérito utilizada na otimização da QVR. As simulações com imagens apontaram para um melhor desempenho do sistema quando utilizados dicionários otimizados com o índice de desordem proposto, especialmente para os dicionários de tamanhos  $N = 256$  e 512. O ganho de desempenho foi evidenciado para todos os algoritmos utilizados.

Para trabalhos futuros, pretende-se avaliar a complexidade computacional da técnica proposta. Pretende-se também, comparar o desempenho de outras versões do FSS, como, por exemplo, o *Weight Based Fish School Search* (wFSS) [20],



Figura 8: Lena.

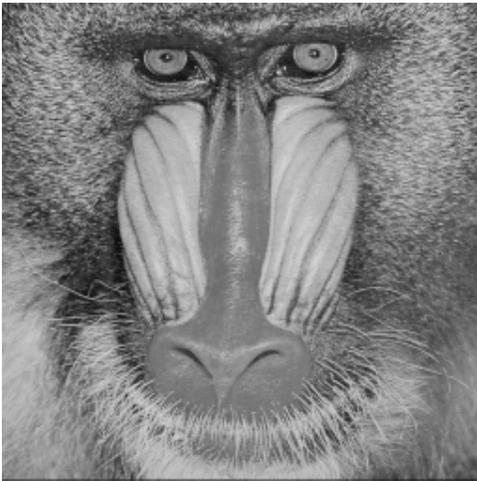


Figura 9: Mandrill.



Figura 10: Boat.

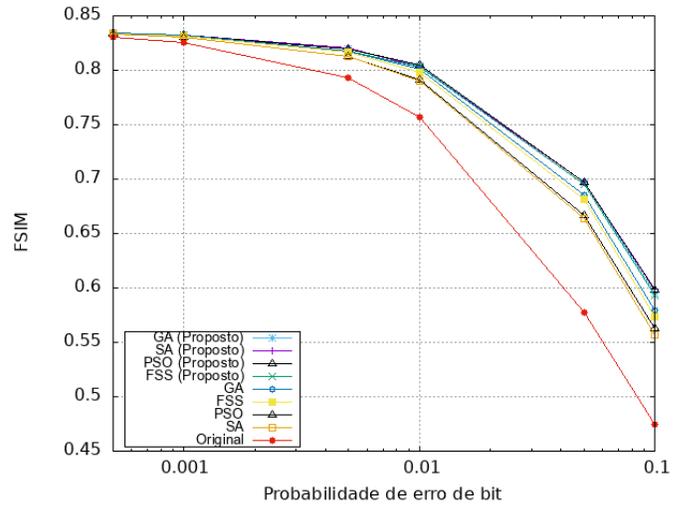


Figura 11: FSIM para os dicionários com tamanho 128 originais, organizados com a função índice de desordem e com o índice de desordem proposto.

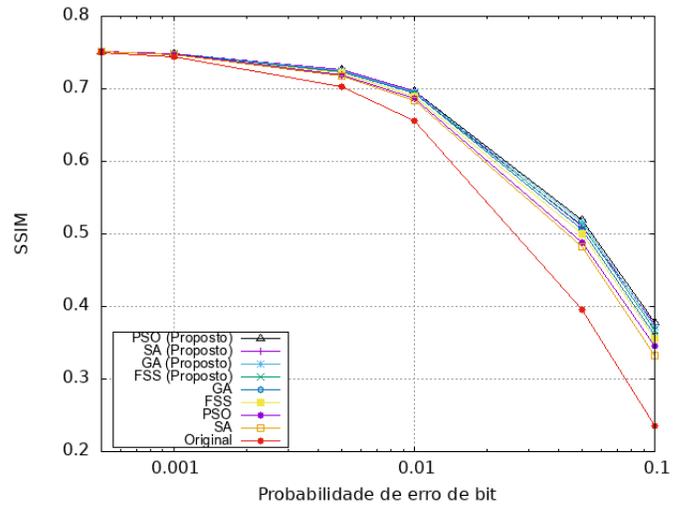


Figura 12: SSIM para os dicionários com tamanho 128 originais, organizados com a função índice de desordem e com o índice de desordem proposto.

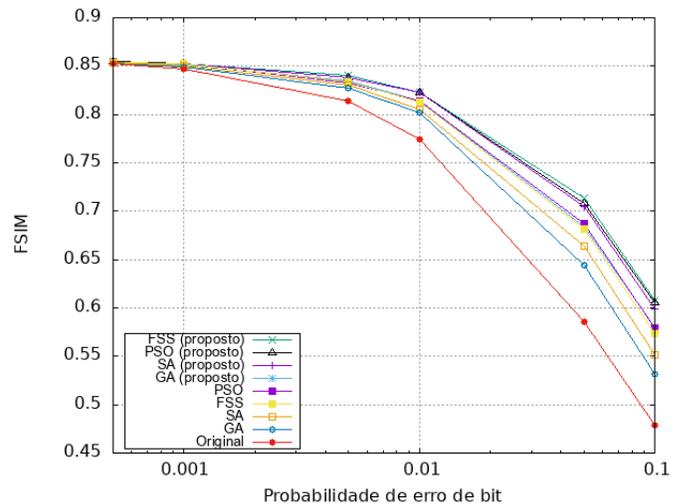


Figura 13: FSIM para os dicionários com tamanho 256 originais, organizados com a função índice de desordem e com o índice de desordem proposto.

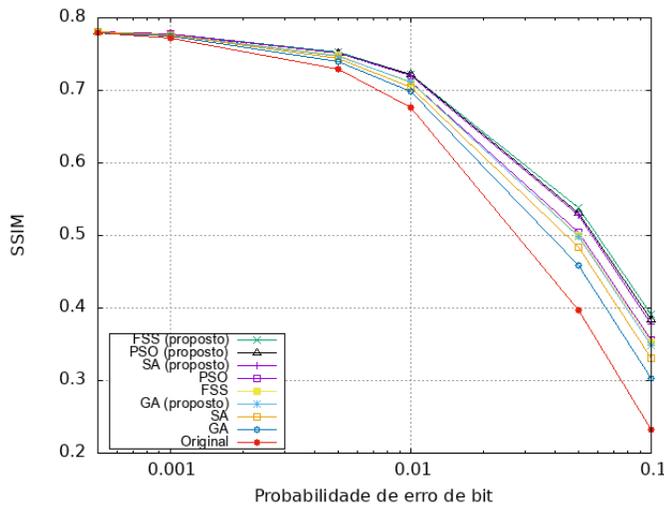


Figura 14: SSIM para os dicionários com tamanho 256 originais, organizados com a função índice de desordem e com o índice de desordem proposto.

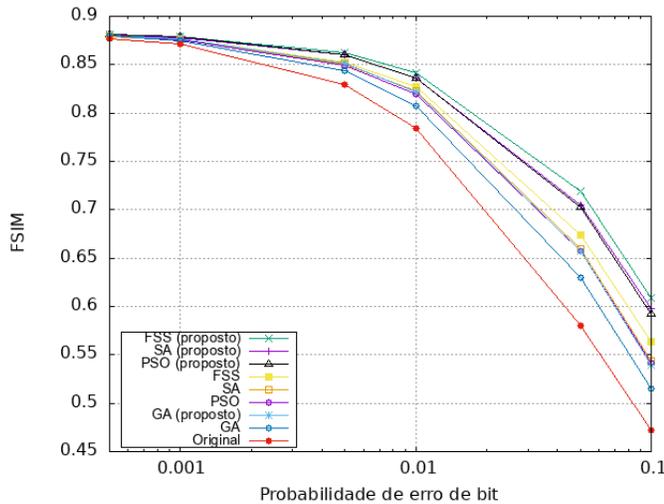


Figura 15: FSIM para os dicionários com tamanho 512 originais, organizados com a função índice de desordem e com o índice de desordem proposto.

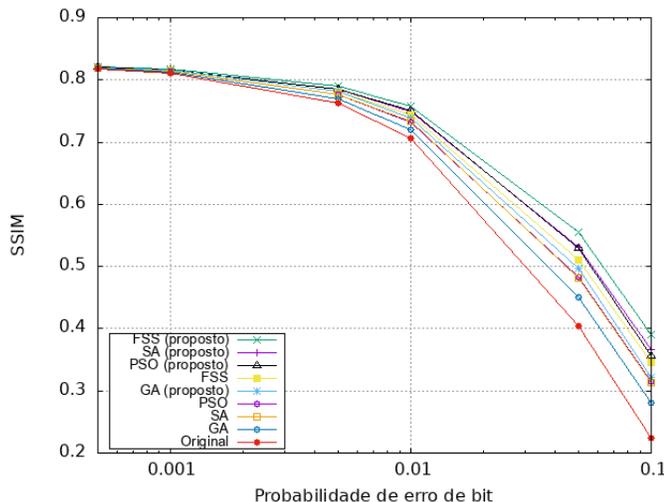


Figura 16: SSIM para os dicionários com tamanho 512 originais, organizados com a função índice de desordem e com o índice de desordem proposto.

além de investigar o desempenho da QVR para outros tipos de sinais, como imagens coloridas e vídeo.

### REFERÊNCIAS

- [1] A. A. Hameed, A. Jamil, Z. Orman, e S. A. Shah, “Application of generalized relevance linear vector quantization for diabetes diagnosis,” in *2021 International Conference on Innovative Computing (ICIC)*, 2021, pp. 1–5.
- [2] S. M. Darwish e A. A. Almajtomi, “Metaheuristic-based vector quantization approach: a new paradigm for neural network-based video compression,” *Multimedia Tools and Applications*, vol. 80, n.º. 5, pp. 7367–7396, 2021.
- [3] E. M. da Silva Filho, P. H. E. Santo, e F. Madeiro, “Inserção de marca d’água em imagens usando múltiplos dicionários para quantização vetorial,” in *Congresso Brasileiro de Automática-CBA*, vol. 1, n.º. 1, 2019.
- [4] H. Gunawan, S. Soeheri, D. Adhar, H. Hardianto, L. Wahyuni, e C. B. Harahap, “Learning vector quantization (LVQ) for colorectal cancer identification based on microscopic network image,” in *2022 4th International Conference on Cybernetics and Intelligent System (ICORIS)*, 2022, pp. 1–5.
- [5] R. Gray, “Vector quantization,” *IEEE ASSP Magazine*, vol. 1, n.º. 2, pp. 4–29, 1984.
- [6] W. V. Santana, W. Lopes, e F. Madeiro, “Algoritmo FSS modificado aplicado à quantização vetorial robusta de imagens,” in *Anais do 15 Congresso Brasileiro de Inteligência Computacional*. Joinville, SC: SBIC, 2021, pp. 1–8.
- [7] J. Galvão, E. Lima, e F. Madeiro, “Atribuição de índices para QV robusta usando o algoritmo variable neighborhood search,” *Trends in Computational and Applied Mathematics*, vol. 18, n.º. 2, 2017.
- [8] N. Farvardin, “A study of vector quantization for noisy channels,” *IEEE Transactions on Information Theory*, vol. 36, n.º. 4, pp. 799–809, 1990.
- [9] K. Zeger e A. Gersho, “Pseudo-Gray coding,” *IEEE Transactions on Communications*, vol. 38, n.º. 12, pp. 2147–2158, 1990.
- [10] W. T. A. Lopes, “Diversidade em modulação aplicada à transmissão de imagens em canais com desvanecimento.” Tese (Doutorado em Engenharia Elétrica), Universidade Federal de Campina Grande, 2003.
- [11] T. N. Tuan, N. Q. Trung, e T. N. Khanh, “Improving the simulated annealing algorithm for source codeword index assignment by using the mechanism of tabu search algorithm,” in *2016 International Conference on Advanced Technologies for Communications (ATC)*, 2016, pp. 91–96.
- [12] E. Lima, G. Melo, W. Lopes, e F. Madeiro, “Um novo algoritmo para atribuição de índices: Avaliação em quantização vetorial de imagens,” *TEMA - Tendências em Matemática Aplicada e Computacional*, vol. 10, 08 2009.
- [13] R. A. Azevedo, F. Madeiro, W. T. A. Lopes, e E. A. O. Lima, “A quasi random symbol interleaving technique applied to image transmission by noisy channels,” *IEEE Latin America Transactions*, vol. 14, n.º. 3, pp. 1078–1085, 2016.
- [14] A. R. S. Barreto, E. Lima, e F. Madeiro, “Sobre o índice de desordem como figura de mérito em quantização vetorial robusta,” in *Congresso de Matemática Aplicada e Computacional-Nordeste (CMACNE)*. Natal-RN, Brasil: Nov, 2012.
- [15] Y. Linde, A. Buzo, e R. Gray, “An algorithm for vector quantizer design,” *IEEE Transactions on Communications*, vol. 28, n.º. 1, pp. 84–95, 1980.
- [16] B. D. Agostini, “Uma análise experimental de abordagens topológicas aplicadas ao problema do caixeiro viajante através de otimização por nuvem de partículas,” Dissertação (Mestrado em Matemática), Universidade Federal do Paraná, 2015.
- [17] M. Rosendo, “Um algoritmo de otimização por nuvem de partículas para resolução de problemas combinatórios,” Dissertação (Mestrado em Informática), Universidade Federal do Paraná, 2010.
- [18] L. Zhang, L. Zhang, X. Mou, e D. Zhang, “FSIM: A feature similarity index for image quality assessment,” *IEEE Transactions on Image Processing*, vol. 20, n.º. 8, pp. 2378–2386, 2011.
- [19] Z. Wang, A. Bovik, H. Sheikh, e E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, n.º. 4, pp. 600–612, 2004.
- [20] F. B. de Lima Neto e M. G. P. de Lacerda, “Weight based fish school search,” in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 270–277.

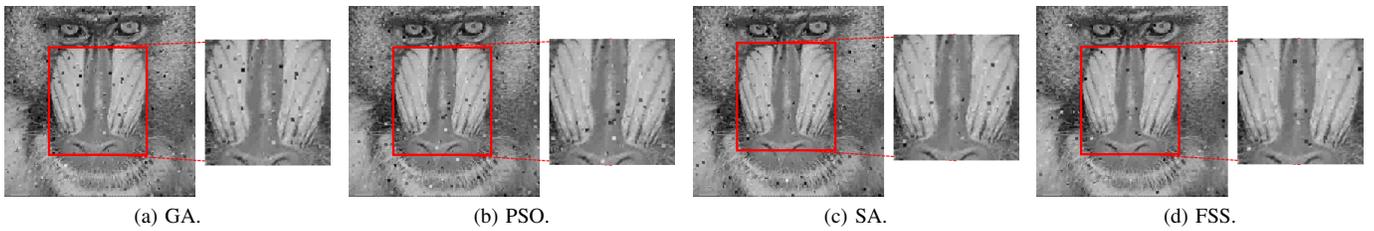


Figura 17: Imagem *Mandrill* reconstruída após transmissão por canal BSC utilizando dicionários organizados com a função índice de desordem.

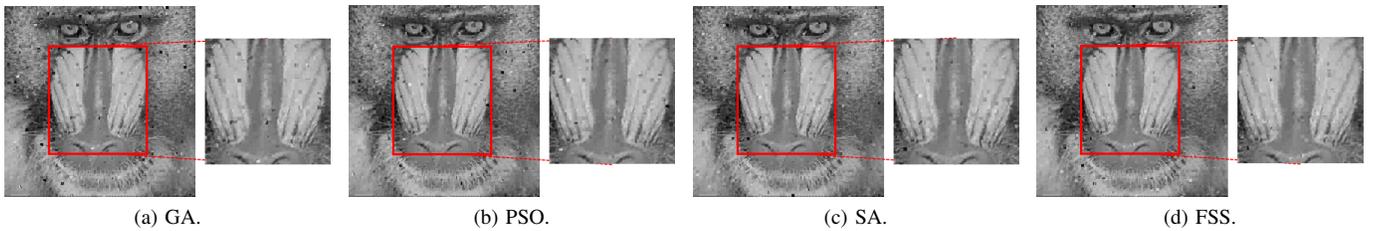


Figura 18: Imagem *Mandrill* reconstruída após transmissão por canal BSC utilizando dicionários organizados com a função índice de desordem proposta.