

‘*Wolf-Pack Approximation*’ utilizando funções de núcleo

Luís Filipe de Araújo Pessoa

Westfälische Wilhelms-Universität Münster
Leonardo-Campus 3, D-48149 Münster, Germany
filipe.pessoa@wi.uni-muenster.de

Fernando Buarque de Lima Neto, Luís Carlos de Souza Menezes

University of Pernambuco (UPE)
Rua Benfica, 455, 50750-410 – Recife, Brasil
{fbln, lcsm}@ecomp.poli.br

Abstract— Through metaphors of natural phenomena, one can construct computational models that are able to simulate or solve complex problems. Often, those are more intuitive to use and less computationally expensive to run. Within complex problems, function approximation features as a very important class of problems, since functions are good ways to represent real-world problems or to describe reality from a sparse set of entries (e.g. points). Even though the vast range of techniques in existence for dealing with approximation of functions such as linear programming, e.g. the SIMPLEX algorithm, in the worst case, they are of exponential complexity $O(2^n)$. On the other hand, the very good approach of some metaheuristic such as artificial neural networks, completely lack in explanatory power towards the underlying relationships with the data. In this context Wolf-Pack Approximation (WPA), although another metaheuristic, features as a new computational alternative to the approximation problem of a given set of entry points. This paper put forward an enlarged and restructured version of WPA, which had some important limitations in its inception, namely, (i) the lack of communication between the wolves and (ii) the use of simple function (i.e. circles) for modeling the contour of the input points of the problems. Thus, in this work we propose mechanisms to increase the effectiveness of the initial version of WPA. To validate the current proposition, WPA2 was used in complex tasks of digital image representation, where several analyses and comparison were performed.

Keywords—Multiagent Systems, Approximation of Functions, Bio-inspired Computation, Representation of Digital Images, Wolf-Pack Approximation.

I. INTRODUÇÃO

Diversas são as técnicas e algoritmos inspirados no comportamento de animais na natureza para a resolução de problemas que surgiram nas últimas décadas. Essa abordagem mostra-se promissora e já apresentou resultados pelo seu poder de abstração e eficácia, porém ela é desafiadora, já que requer conhecimento em diferentes áreas de conhecimento. Através de metáforas com os fenômenos da Natureza, pode-se construir um modelo computacional que se proponha a simular o ambiente e resolver problemas complexos, geralmente de forma mais intuitiva [1], ou com menos custo computacional. Assim, várias técnicas de Inteligência Artificial (i.e. técnicas simbólicas) e de Inteligência Computacional (i.e. técnicas subsimbólicas) são utilizadas para esses fins. Devido a usual eficácia, esses métodos são aplicados em diversos domínios de difícil resolução, tais como problemas de otimização [2],

classificação, previsão [3], simulação, controle, monitoramento, entre outros.

O motivo para a existência de técnicas e abordagens diferentes dessa área é a sua aplicabilidade (e efetividade) em problemas específicos. Cada uma delas possui vantagens para resolver problemas com características em comum. No que diz respeito a problemas complexos, a aproximação de função é uma classe de problemas bastante importante, uma vez que funções podem representar problemas do mundo real ou descrever alguma realidade a partir de um conjunto de pontos. Além disso, a aproximação de funções é bastante utilizada para a resolução de problemas de classificação e reconhecimento de padrões [4], predição, compressão de dados [5], aprendizagem por reforço, processamento e reconstrução de imagens [6]. A Programação Linear utilizando o algoritmo SIMPLEX possui, no pior caso, complexidade exponencial $O(2n)$ [7] (i.e. escala exponencialmente com o tamanho da entrada do problema de aproximação), o que a torna proibitiva em problemas de grande cardinalidades. Enquanto que Redes Neurais aproximam bem o conjunto de pontos de entrada, mas nenhum entendimento sobre a relação subjacente pode ser adquirida. Ou seja, não é possível analisar ou visualizar de que forma foi obtida a aproximação para o conjunto de pontos de entrada.

Nesse contexto, uma nova metaheurística de Computação Natural foi concebida para ser uma alternativa computacional para o problema de aproximação local de um conjunto de pontos de entrada. A concepção inicial do *Wolf-Pack Approximation* (WPA) [8] baseou-se no comportamento de caça de uma alcateia na natureza, cujo objetivo é encontrar uma possível função que represente e modele o contorno de uma dada nuvem de pontos. Entretanto, a abordagem inicial do WPA possui algumas limitações, tais como (i) a falta de comunicação direta entre os lobos, (ii) a ausência de um mecanismo de ajuste automático do número de lobos no ambiente, (iii) o uso de circunferências para a modelagem do contorno da nuvem de pontos e (iv) apenas funciona em problemas unidimensionais.

Logo, o presente trabalho se propõe a promover melhorias substanciais no algoritmo do WPA inicial, endereçando as limitações apontadas e, principalmente, no que se refere à atribuição de uma função de núcleo para cada um dos lobos e à implementação da comunicação entre eles. Dessa forma, espera-se que as limitações citadas sejam dirimidas e que se aumente a eficácia da solução em problemas complexos.

Para validação da proposta, o WPA foi utilizado para a representação do mesmo conjunto de pontos de entrada utilizados na versão inicial e os resultados obtidos foram comparados com a versão inicial. Além disso, o WPA foi utilizado para a representação de imagens digitais. Esta abordagem mostra-se promissora, tendo em vista que pode ser aprimorada se utilizada em conjunto com outras técnicas de processamento de imagens, inteligência computacional ou métodos estatísticos para prover um novo mecanismo de compressão ou para o processamento de imagens, dentre outras aplicações (do método proposto).

II. PROCESSAMENTO DE IMAGENS DIGITAIS

Uma imagem digital pode ser definida como uma função bidimensional $f(x, y)$, a qual mapeia as entradas finitas e discretas x e y em seus valores finitos de intensidade. Assim, uma imagem digital I é representada computacionalmente através de uma matriz, onde os índices das linhas e colunas indicam o posicionamento do *pixel* (i.e. elementos da imagem), enquanto o valor de cada elemento da matriz indica a sua intensidade. Essa representação está indicada na Equação (1). Para o caso de representação da imagem em tons de cinza, os valores de $f(x, y)$ variam entre 0 e 255.

$$I = f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f(M,0) & f(M,1) & f(M,2) & \dots & f(M,N) \end{bmatrix} \quad (1)$$

No que diz respeito ao armazenamento de imagens, o JPEG (*Joint Photographic Experts Group*) [9] se destaca como um dos padrões de compressão de imagens mais utilizado. Apesar de ser possível obter uma compressão sem perdas, o JPEG é essencialmente uma técnica de compressão com perdas, que explora as limitações visuais humanas. Em termos gerais, o processo de compactação consiste na divisão da imagem em blocos de 8x8 e, para cada bloco, é realizada a transformada discreta do cosseno (DCT). O resultado desse processo é a geração de uma matriz de coeficientes da DCT. Essa matriz possui uma concentração de valores de maior energia na região superior esquerda, a qual representa os componentes de baixa frequência da imagem. Em seguida, aproveitando-se do fato de a matriz de coeficientes da DCT apresentar muitos valores próximos de zero, é realizada uma quantização e normalização, simultaneamente, de forma que o número de coeficientes com valor zero aumente e sejam descartadas informações pouco significativas, de acordo com a prioridade definida pelo ordenador *zig-zag*, conforme a Figura 1.

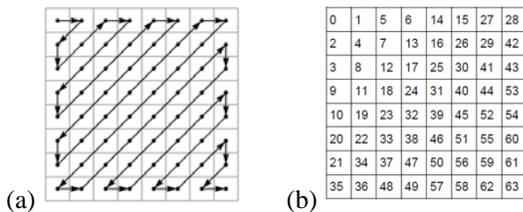


Fig. 1. Ordenação da frequência espacial de acordo com o padrão *zig-zag*.

III. APROXIMAÇÃO DE FUNÇÃO

Basicamente, a aproximação consiste em substituir uma função desconhecida $f(x)$ por uma função $g(x)$, a qual,

geralmente, é calculada como a interpolação de funções previamente definidas. Tal procedimento é necessário, por exemplo, (i) quando se deseja saber o valor de uma determinada variável fora do intervalo da observação dos dados (i.e. extrapolação); (ii) quando não é possível ou é difícil realizar a integração ou diferenciação sobre uma determinada função; (iii) quando se deseja obter uma representação contínua para um dado conjunto de observações discretas; entre outras. No que diz respeito às técnicas, várias são as abordagens de cálculo numérico e métodos estatísticos utilizados para a resolução desse problema, tais como: interpolação, *splines*, séries de potências, regressão por funções de núcleo, entre outros. Há ainda abordagens baseadas em Inteligência Computacional, como *Multilayer Perceptron* (MLP) [10], *Radial Basis Function Network* (RBFN) [11], *Neuro-fuzzy Systems* [12].

Com relação ao uso de aproximação de função em processamento de imagens, ela têm sido utilizada para a reconstrução de imagens e redução de ruído [6], assim como para a compressão de dados [5]. Dentre essas, destaca-se o trabalho de Takeda [6], o qual utiliza o *steering kernel regression* e o *iterative steering kernel regression* para a redução de ruídos, reconstrução de imagens, fusão, redimensionamento, entre outros.

A. Kernel Regression

O *Kernel Regression* é um estimador de regressão por funções de núcleo não paramétrico, baseado no *Kernel Density Estimation* (KDE). O KDE é dado pelas equações (2) e (3), onde $K(u)$ é uma função contínua que satisfaz as equações (4) e (5). Dessa forma, em cada ponto do espaço amostral, é posicionada uma função de núcleo K_h com massa de probabilidade $\frac{1}{n}$. No que diz respeito às funções de núcleo $K(u)$, a mais comumente utilizada é a função Gaussiana [13], conforme a Equação (6).

$$P_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - X_i) \quad (2)$$

$$K_h(u) = \frac{1}{h} K\left(\frac{u}{h}\right) \quad (3)$$

$$\int K(u) du = 1 \quad (4)$$

$$K(u) \geq 0 \quad (5)$$

$$K(u) = \frac{1}{2\pi} \exp\left(-\frac{u^2}{2}\right) \quad (6)$$

A principal vantagem dessa abordagem é que ela resulta em uma função de densidade com curvas mais suaves. Nesse sentido, o parâmetro de suavização h é muito importante para a determinação da curva de densidade, tendo em vista que valores pequenos resultarão em uma estimativa não suave, enquanto valores grandes determinarão uma curva muito suave. Por isso, diversas abordagens e trabalhos são desenvolvidos com a finalidade de contornar esse problema. Por exemplo, Kerm [14] propõe o uso de larguras h variando de forma adaptativa, ao invés de definir valores fixos.

No que diz respeito ao estimador de regressão por funções de núcleo, o seu objetivo é determinar os valores dos pesos $W_{N,i}$, dada a restrição (7), de forma que a função aproximativa $F(x)$ possa ser calculada conforme (8), onde y é o valor de saída esperado do modelo de entrada, de tamanho N . Dessa forma, *Nadaraya-Watson* [15][16] propuseram a função de ponderação normalizada conforme a Equação (9), resultando no *Kernel Regression* de *Nadaraya-Watson* (KRNW).

$$\sum_{i=1}^N W_{N,i}(x) = 1, \quad \forall x \quad (7)$$

$$F(x) = \sum_{i=1}^N W_{N,i}(x)y_i \quad (8)$$

$$W_{N,i}(x) = \frac{K_h(x - X_i)}{\sum_{j=1}^N K_h(x - X_j)}, \quad i = 1, 2, \dots, N \quad (9)$$

IV. WOLF-PACK APPROXIMATION INICIAL

A abordagem inicial do WPA [8] é uma metaheurística baseada em agentes que visa a modelagem do contorno de uma nuvem de pontos a partir da interação entre os agentes no ambiente, os quais são baseados no comportamento de caça dos lobos na natureza. O ambiente operacional do WPA inicial é composto por árvores (*i.e.* cada árvore representa um ponto da nuvem de pontos de entrada), por ovelhas (*i.e.* representam a presa dos lobos) e por lobos (os quais “caçam” as ovelhas através do ajuste de seus raios de atuação e seu posicionamento). Durante a caça, os lobos perseguem as presas através do ajuste de seu posicionamento, assim como ajustam seus raios de atuação, o qual determina as ovelhas que estão sendo caçadas pelos lobos atualmente. Por sua vez, as ovelhas reagem ao ataque dos lobos fugindo em direção à floresta toda vez que se encontrarem dentro do raio de atuação do lobo. Ao final, os lobos estarão estrategicamente posicionados de forma que estejam modelando o contorno da nuvem de pontos de entrada, e esta possa ser representada pelos posicionamentos e raios de atuação (*i.e.* circunferências) de cada lobo do ambiente. A Figura 2 mostra o modelo inicial do WPA.

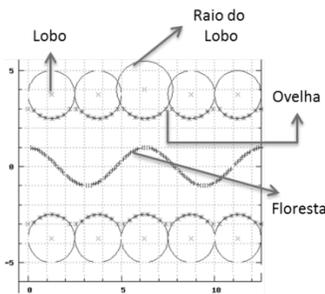


Fig. 2. Modelo computacional do WPA em sua proposição inicial.

Ao analisar os resultados obtidos com a abordagem inicial do WPA é possível identificar algumas limitações e pontos de melhoria, tais como:

- a falta de comunicação direta entre os lobos, pois é um fator limitante no que diz respeito à possibilidade de um melhor ajuste para diminuição do erro global;

- a ausência de um mecanismo de ajuste automático do número de lobos no ambiente, pois pode ser que a quantidade de lobos inicial para a aproximação do contorno dos pontos de entrada não seja suficiente;
- o uso de circunferências para a modelagem do contorno da nuvem de pontos, pois existem diversas situações em que circunferências não conseguem modelar corretamente um conjunto de pontos, principalmente aquelas que apresentam certa linearidade, como na Figura 3;
- apenas funciona em problemas unidimensionais;
- o posicionamento e o raio de atuação se limitam ao ajuste de incremento, o qual restringe as possibilidades de ação e adaptação do lobo.

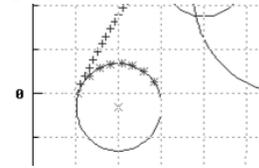


Fig. 3. Situação na qual circunferências não modelam corretamente.

Dessa forma, este trabalho propõe melhorias e novas implementações que diminuam ou resolvam as limitações e pontos de melhoria citados anteriormente, além de aplicar a nova versão do WPA para a representação de imagens digitais.

V. WPA COM FUNÇÕES DE NÚCLEO (WPA2)

A Figura 4 mostra as modificações realizadas no WPA inicial, as quais visam o desenvolvimento dos mecanismos de melhoria do erro local dos agentes e do erro global da coletividade dos agentes, com a incorporação do módulo de comunicação entre os lobos no ambiente para a solicitação de ajuda. Da mesma forma que o WPA inicial, o objetivo da abordagem proposta é determinar o posicionamento e raios de atuação dos lobos, de forma que as ovelhas formem o contorno da floresta (*i.e.* que o somatório das funções de núcleo dos lobos modele o contorno da nuvem de pontos de entrada) e, assim, encontre a aproximação para o conjunto de pontos de entrada.

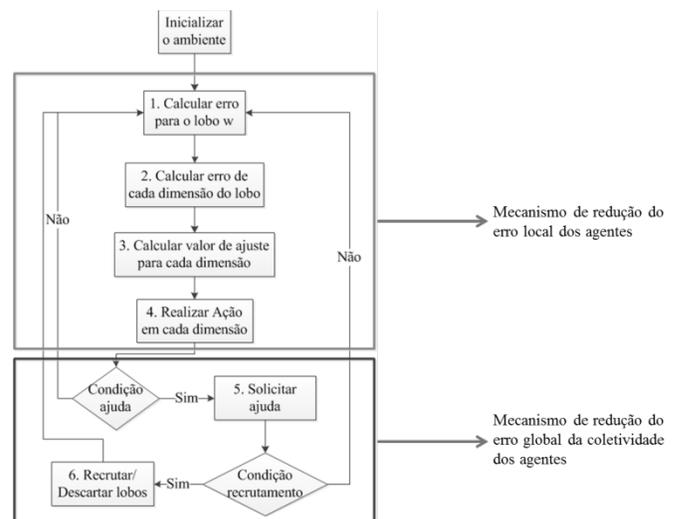


Fig. 4. Fluxograma do *Wolf-Pack Approximation* proposto (WPA2).

Após a inicialização dos lobos, árvores e ovelhas no ambiente, o erro do lobo w é calculado para cada lobo do ambiente. Em seguida, para cada uma das dimensões do problema, calcula-se (i) o erro do lobo em cada dimensão específica, (ii) o valor de ajuste a ser realizado em cada uma das dimensões analisadas, e, (iii) a ação a ser realizada, baseada nos erros calculados e nas regras de ação dos lobos. Dessa forma, a quantidade de ações realizadas pelo lobo é igual a quantidade de dimensões analisadas e as ações podem ser: ajuste de raio, ajuste de posicionamento e ajuste da amplitude do lobo. Após realizar as ações em todas as dimensões, o lobo verifica se deverá solicitar ajuda aos seus vizinhos, em caso dele não conseguir diminuir o seu erro durante épocas consecutivas, objetivando a diminuição do erro global. Em seguida, o lobo verifica se a condição de recrutamento ou descarte foi satisfeita, a qual examina, por exemplo, se algum outro lobo forneceu ajuda ao lobo atual. Caso a condição de recrutamento ou descarte seja satisfeita, o lobo deverá ou (i) recrutar um novo lobo para ajuda-lo a diminuir o erro das piores ovelhas dentro de seu domínio, ou (ii) descartar lobos que estejam colaborando de forma significativa para a redução do erro global. Ao final, será selecionado o próximo lobo para realizar suas deliberações no ambiente.

Além das modificações no algoritmo, foram promovidas modificações no ambiente e no conjunto de ações dos lobos. No ambiente proposto, cada ovelha está associada a um ponto distinto pertencente ao conjunto de pontos de entrada (*i.e.* as árvores), eliminando a necessidade do parâmetro de configuração do número de ovelhas, presente na versão inicial. Outra modificação implementada na proposta atual é a utilização de um único conjunto de lobos no ambiente. Ou seja, não há mais um conjunto de lobos acima e outro abaixo da floresta. Dessa forma, o WPA2 poderá representar a nuvem de pontos de entrada com um menor número de lobos. Por sua vez, no algoritmo proposto os lobos contêm uma função de núcleo e podem (i) ajustar seu posicionamento, (ii) ajustar seu raio, (iii) ajustar amplitude, (iv) solicitar ajuda e (v) solicitar recrutamento. Observa-se ainda que o objetivo de todas as ações realizadas pelo lobo é o ajuste de suas funções de núcleo. A Figura 5 mostra o modelo computacional do WPA2.

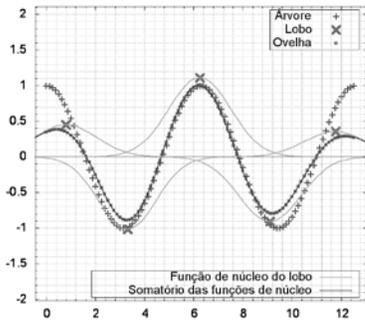


Fig. 5. Modelo computacional da abordagem proposta.

De forma similar ao KRNW, os lobos contêm uma função de núcleo e, através do somatório de suas funções de núcleo, computam a aproximação da nuvem de pontos. Por outro lado, o WPA2 difere do KRNW pela maneira como as funções de núcleo são ajustadas e na forma como o erro é calculado. Por objetividade, neste trabalho foi utilizada a função Gaussiana,

conforme a Equação (10), onde h_w é a amplitude da Gaussiana; o vetor \mathbf{P}_w corresponde ao posicionamento do lobo w nas d dimensões do problema; e, o vetor \mathbf{R}_w corresponde ao raio de atuação do lobo w nas respectivas dimensões do problema. Assim, o conjunto de informações necessárias para que um lobo seja representado corresponde a $\{\mathbf{P}_w, \mathbf{R}_w$ e $h_w\}$.

$$f_w(\mathbf{P}) = h_w e^{-\frac{(\mathbf{P}-\mathbf{P}_w)^2}{2\mathbf{R}_w^2}} \quad (10)$$

A. Inicialização do Ambiente

A inicialização do WPA consiste em:

1. definir o posicionamento das árvores (*i.e.* a nuvem de pontos de entrada, a qual se deseja aproximar);
2. dividir o espaço de entrada em regiões iguais e associar um lobo responsável por determinar uma aproximação para cada uma delas (o posicionamento inicial do lobo será igual ao centro da região que foi atribuída a ele; o raio inicial do lobo para cada uma das dimensões é igual a distância do centro do lobo para as extremidades da sua região de atuação; e, a amplitude inicial do lobo é igual ao menor de todos os valores esperados para essa região); e,
3. inicializar as ovelhas na posição de cada uma das árvores e atribuir seus valores calculados ϕ'_s , resultante do somatório das funções de núcleo de cada lobo, conforme a Equação (11), onde f_w é a função de núcleo de cada lobo w e \mathbf{P}_s é a posição da ovelha S .

$$\phi'_s = \sum_w f_w(\mathbf{P}_s) \quad (11)$$

B. Cálculo do erro do lobo

Como no WPA inicial, o erro do lobo é igual ao somatório dos erros das ovelhas dentro de seu raio de atuação. Com o objetivo de avaliar diferentes aspectos de sucesso do lobo, duas medidas de erro são utilizadas: Err_w (erro médio) e MAE_w (erro médio absoluto), dados pelas equações (12) e (13). Onde, ϕ_T é o valor esperado para a árvore T que está nas mesmas coordenadas de S ; G_w é a região de atuação do lobo w ; e, k é o número de ovelhas que pertencem a G_w . A distinção entre as duas medidas de erro para uma determinada região é que enquanto o MAE_w mede o quão distante as ovelhas estão das árvores, o Err_w indica a prevalência do erro: acima ou abaixo da floresta (*i.e.* maior ou menor do que os valores esperados para a região, respectivamente). No que diz respeito a região G_w , o seu propósito é definir o conjunto de ovelhas ao qual o lobo concentrará seus esforços para diminuir os respectivos erros. Dessa forma, G_w pode ser a região compreendida pelo raio de atuação \mathbf{R}_w do lobo (*i.e.* varia de acordo com o raio do lobo) ou também pode ser definida por uma região fixa do espaço de entrada, a qual o lobo é responsável por aproximá-la.

$$Err_w = \left\{ \frac{1}{k} \sum_{i=0}^k (\phi'_{S_i} - \phi_{T_i}), \forall S_i \in G_w \text{ and } \forall T_i \in G_w \right\} \quad (12)$$

$$MAE_w = \left\{ \frac{1}{k} \sum_{i=0}^k |\phi'_{S_i} - \phi_{T_i}|, \forall S_i \in G_w \text{ and } \forall T_i \in G_w \right\} \quad (13)$$

Tendo em vista que o lobo realiza uma ação para cada dimensão do espaço, é necessário analisar os erros dos lobos em cada dimensão: $Err_{w,d}$ e $MAE_{w,d}$. Para isso, basta substituir a região G_w por $G_{w,d}$, a qual corresponde à região de atuação do lobo na dimensão d . Tendo em vista que $G_{w,d}$ pode ser diferente para cada dimensão, os erros para cada dimensão terão valores e características diferentes, pois o conjunto de ovelhas consideradas também serão diferentes.

C. Cálculo do fator de ajuste

Com a finalidade de adaptar os ajustes do lobo, foi desenvolvido um mecanismo para que o lobo determine o fator de ajuste $\eta_{w,d}(t)$ tomando por base o valor de $MAE_{w,d}$ nos instantes $t-1$ e $t-2$. Assim, se o erro absoluto diminuir entre os instantes de tempo anteriores (*i.e.* se o lobo obteve êxito ao realizar sua ação), o fator de ajuste é incrementado, caso contrário seu valor é decrementado. A partir de experimentos preliminares realizados, foi possível determinar que quando $\eta_{w,d}(t) \in [0.8, 1]$ e a taxa de incremento foi 0.1, resultou em num decaimento mais acelerado do erro médio e em erros menos ao final da execução. A Equação (14) mostra como esse fator é utilizado pelos operadores dos lobos para determinar o valor a ser corrigido (ΔV_d), onde $MAE_{w,d}(t)$ é o erro absoluto do lobo w na dimensão d no instante t .

$$\Delta V_d(t) = \eta_{w,d}(t) \cdot MAE_{w,d}(t) \quad (14)$$

D. Repertório de ações e regras dos lobos

O conjunto de ações do lobo é composto por: (i) ajuste da amplitude do lobo (*i.e.* da amplitude da Gaussiana h_w), (ii) ajuste do raio de atuação (R_w) e (iii) ajuste do posicionamento (P_w) dos lobos. A principal característica incorporada na heurística proposta é a possibilidade de combinar ajustes em diferentes atributos do lobo. Por exemplo, uma ação do lobo pode consistir no aumento do seu raio de atuação em uma dimensão e na diminuição da amplitude em outra dimensão do problema. De forma que, em cada uma das dimensões, a ação realizada seja aquela que diminua o erro do lobo, bem como o erro global do ambiente. Para definir qual ação será realizada em determinada dimensão, o lobo w deve considerar (i) o desvio-padrão σ_{MAE} do erro médio absoluto MAE_w e (ii) o $Err_{w,d}$ dividido em três regiões distintas: esquerda ($LErr_{w,d}$), central ($CErr_{w,d}$) e direita ($RErr_{w,d}$). As regiões são divididas de acordo com o valor do raio na dimensão d , $R_{w,d}$, conforme a Figura 6. O erro de cada região é calculado considerando apenas as ovelhas contidas em cada região, onde a região central está compreendida no intervalo $[P_w(d) - R_w(d), P_w(d) + R_w(d)]$ e as regiões esquerda e direita são aquelas que estão à esquerda e à direita da região central.

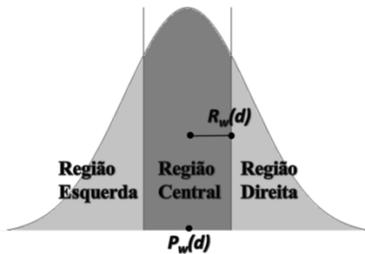


Fig. 6. Região de atuação do lobo dividido nas três regiões de análise.

O conjunto de regras para a tomada de decisão do lobo para cada uma das possíveis situações foi definido de forma heurística e é mostrado na Tabela I. Essas heurísticas foram definidas a partir da observação e estudo das possíveis situações que poderiam ocorrer quando analisados os sinais dos erros em cada uma das regiões de erro do lobo. Para cada uma das situações estudadas, foi definida uma ação que colaborasse mais para a diminuição do erro.

TABELA I. CONJUNTO DE REGRAS HEURÍSTICAS PARA O LOBO.

$LErr_{w,d}$	$CErr_{w,d}$	$RErr_{w,d}$	σ_{MAE}	Ação
-	+	-		$R_w(d) - \Delta V_d$
+	-	+		$R_w(d) + \Delta V_d$
-	-	+		$P_w(d) - \Delta V_d$
-	+	+		
+	+	-		$P_w(d) + \Delta V_d$
+	-	-		
Sinal do erro de todas as regiões iguais			$> \sigma_{min}^a$	$h_w + \Delta V_d$
			$< \sigma_{min}$	$Err_{w,d} \geq 0, R_w(d) - \Delta V_d$ $Err_{w,d} < 0, R_w(d) + \Delta V_d$

^{a.} parâmetro do sistema que indica qual o desvio-padrão mínimo desejado.

Da mesma forma que a abordagem inicial do WPA, o objetivo esperado de cada lobo é a diminuição de seus erros absolutos MAE_w e, como consequência, a diminuição do erro médio absoluto do ambiente. Assim, caso a ação realizada pelo lobo não contribua com seu objetivo, será escolhida outra ação do repertório que reduza o erro, caso exista.

E. Atribuição de estados para os lobos

Tendo em vista que os lobos tomam suas ações baseado no seu erro médio, é possível afirmar que um lobo pode influenciar de maneira positiva ou negativa (*i.e.* diminuindo ou aumentando o erro, respectivamente) os seus vizinhos, os quais agirão de forma a melhorar seus erros médios e, da mesma forma, poderão influenciar seus respectivos lobos vizinhos. Logo, é possível concluir que mesmo sem um mecanismo de comunicação explícito, os lobos podem influenciar as ações de seus vizinhos, porém de uma forma limitada.

Assim, para promover uma maior cooperação entre os lobos e melhorar os resultados do WPA (*i.e.* a melhoria do erro global da coletividade dos agentes), foi desenvolvido um mecanismo de comunicação entre lobos. A finalidade dessa comunicação é possibilitar que um lobo com dificuldades possa solicitar ajuda aos seus vizinhos. Dessa forma, em uma reação em cadeia, os lobos passam a cooperar intencionalmente para a diminuição do erro global da coletividade dos agentes.

Para viabilizar a comunicação, foram definidos diferentes estados para os lobos baseado no seu sucesso em diminuir o seu erro médio. Os possíveis estados para um lobo w são:

- **TRABALHANDO:** se após realizar uma ação, (i) o seu erro MAE_w é maior do que o valor do erro mínimo aceitável (min_erro_lobo) e (ii) quando o percentual de melhoria do erro ($\%M$) for maior do que o percentual mínimo de melhoria ($min_ \%M$), onde

min_erro_lobo e $min_ \%M$ são parâmetros do algoritmo;

- INCOMODADO: se após realizar uma ação, (i) o seu erro MAE_w é maior do que min_erro_lobo e (ii) quando $\%M$ for menor ou igual a $min_ \%M$; e,
- AJUSTADO: se após realizar uma ação, o seu erro MAE_w é menor ou igual a min_erro_lobo .

F. Solicitação de ajuda

Dado os possíveis estados de um lobo, torna-se possível definir estratégias para determinar quando um lobo irá solicitar ajuda aos seus vizinhos, assim como para determinar se um lobo ajudará o seu vizinho. Assim, o lobo solicitará ajuda aos seus L vizinhos se permanecer “incomodado” durante t_c épocas, onde L e t_c são parâmetros do algoritmo. Essa abordagem procura evitar uma instabilidade no algoritmo, tendo em vista que ao ajudar seu vizinho o lobo poderá aumentar o seu erro. Além disso, é necessário que o lobo “incomodado” tente otimizar (*i.e.* ajustar) sozinho o seu desempenho sem necessitar da ajuda de outros lobos.

Por outro lado, e para evitar prejuízos individuais, os lobos que recebem a solicitação de ajuda deverão decidir se ajudarão o seu vizinho. Por exemplo, lobos que estão “ajustados” (*i.e.* funcionando adequadamente) não ajudarão o seu vizinho, tendo em vista que eles aparentemente encontraram suas configurações ótimas para aproximar a região sob sua responsabilidade. Logo, para que o lobo ajude o seu vizinho, é necessária que a seguinte condição de ajuda seja satisfeita:

- estado atual do lobo é diferente de “ajustados”, e
- sua ação conseguiu diminuir o MAE do lobo que solicitou ajuda, e
- seu erro não aumentou mais do que uma porcentagem definida pelo parâmetro $\%AEL_{MAX}$, e
- não aumentou o erro do ambiente.

Finalmente, os lobos possuem um fator de altruísmo A que indica a probabilidade de um lobo ajudar outro (A é um parâmetro do sistema). Através desse fator, é possível mudar o comportamento dos lobos para uma determinada execução do WPA2, aumentando ou diminuindo a quantidade de lobos que ajudarão o lobo “incomodado”. Esse fator instancia o conceito de sociabilidade do lobo, importante para sistemas cooperativos e baseados em Computação Social.

Dessa forma, o lobo que solicita ajuda deverá informar aos seus vizinhos: (i) o MAE_w ; (ii) o somatório do erro das piores ovelhas $POErr$ dentro da sua região de atuação; e, (iii) o baricentro B do erro das piores ovelhas dentro da sua região de atuação. Por sua vez, o lobo que ajudará o seu vizinho executará a seguinte sequência de passos:

1) *Calcular o ajuste de ajuda ΔV_A para seu raio e amplitude, de acordo com a Equação (15), onde $\Delta V_{\%A}$ indica o percentual de ajuda a ser aplicado e NPO indica o número de piores ovelhas que foram consideradas*

$$\Delta V_A = \frac{POErr}{NPO} \Delta V_{\%A} \quad (15)$$

2) *Calcular o vetor de ajuste de ajuda ΔV_P para o seu posicionamento, de acordo com a Equação (16), de forma que o seu posicionamento seja ajustado na direção do baricentro do erro das piores ovelhas*

$$\Delta V_P = |\mathbf{B} - \mathbf{P}_w| \Delta V_{\%A} \quad (16)$$

3) *Para cada dimensão d do problema, escolher qual ação do repertório diminui mais o erro do lobo que solicitou ajuda.*

4) *Retornar mensagem ao lobo que solicitou ajuda informando que:*

a) *não ajudou – quando o lobo está “ajustado” ou quando ele não quer ajudar, baseado no seu fator de altruísmo;*

b) *tentou ajudar – quando o lobo tentou executar uma ação para ajudar o seu vizinho, mas a condição de ajuda não foi satisfeita para nenhuma das dimensões;*

c) *ajudou – quando o lobo executou uma ação de ajuda em, ao menos, uma das dimensões.*

G. Recrutamento e descarte de lobos

O mecanismo de recrutamento de novos lobos tem um papel fundamental para promover a ajuda necessária ao lobo “incomodado”, quando esse não obteve ajuda de seus vizinhos. Outra motivação importante para o uso do recrutamento é que ele torna o algoritmo menos dependente do número de lobos inicial. Além disso, o recrutamento permite que sejam criadas hierarquias de lobos, assim como acontece na natureza. Logo, é possível dizer que o lobo recrutado é dominado pelo lobo que solicitou o recrutamento (*i.e.* lobo dominante).

Por outro lado, é necessário que haja um controle do número máximo de lobos no ambiente, de forma que a informação necessária para representar os lobos contidos no ambiente seja menor do que para representar a nuvem de pontos de entrada. No WPA2, um lobo é representado por seus vetores de raio R_w e posicionamento P_w e pelo escalar da amplitude h_w da função de núcleo. Como a quantidade de elementos dos vetores R_w e P_w são iguais ao número de dimensões d do conjunto de entrada, logo a quantidade de informação necessária para representar um lobo é igual a $2d + 1$. Por outro lado, cada ponto do conjunto de entrada é representado por suas d dimensões mais o valor esperado para cada um dos pontos. Logo, a quantidade máxima de lobos permitida no ambiente é igual à razão entre a quantidade de informação necessária para representar todo o conjunto de entrada e a quantidade de informação necessária para representar cada um dos lobos, conforme a Equação (17).

$$\#w_{max} = \frac{qtdInformacaoEntrada}{qtdInformacaoLobo} = \frac{m(d + 1)}{2d + 1} \quad (17)$$

Para recrutar um lobo, é necessário que: (i) a região do lobo contenha mais de uma ovelha; (ii) o lobo esteja “incomodado” durante t_R iterações (onde t_R é um parâmetro do WPA2); (iii) o número de lobos no ambiente seja menor do que $\#w_{max}$; e, (iv) nenhum dos vizinhos o tenha ajudado. Uma vez que essa condição seja verdade, as seguintes ações são realizadas:

1) Selecionar as ovelhas que o lobo recrutado será responsável por diminuir seus respectivos erros.

2) Calcular o baricentro \mathbf{B} das ovelhas selecionadas no passo 1.

3) Definir a amplitude h_w igual a 90% do erro da pior ovelha selecionada no passo 1, de forma que o lobo recrutado “empurre” a ovelha para junto da floresta e colabore de forma significativa já no momento de sua inclusão no ambiente.

4) Definir R_w de forma que todas as ovelhas do passo 1 estejam contidas no raio do lobo recrutado.

5) Definir a sua posição igual ao baricentro \mathbf{B} .

6) Incluir o lobo recrutado na lista de lobos dominados pelo lobo que solicitou o recrutamento.

O recrutamento é um operador muito importante para a redução do erro global do sistema, uma vez que permite que regiões com maior dificuldade para a aproximação sejam tratadas com um conjunto maior de lobos. Por outro lado, é possível que existam lobos que não estejam colaborando significativamente para a obtenção de uma melhor aproximação dos pontos de entrada, pois possuem uma amplitude ou raios próximos de zero. Logo, é necessário um mecanismo análogo, só que de descarte de lobos no ambiente do WPA2 para promover a auto-regulação de lobos no ambiente. Os lobos serão descartados nas seguintes situações, caso a sua remoção não resulte em um aumento do erro médio global maior do que $\%AEL_{MAX}$:

- caso não existam ovelhas dentro do raio do lobo, ou
- caso a divisão entre a amplitude atual do lobo e o menor valor esperado para as ovelhas dentro do raio do lobo for menor do que $min_ \%M$, pois indica que a amplitude atual do lobo é muito pequena e não exerce uma melhoria significativa para a diminuição do erro global.

H. Parâmetros requeridos pelo WPA2

Os parâmetros utilizados no WPA proposto estão descritos na Tabela II. Nela é possível perceber que apenas quatro dentre os parâmetros dependem do problema em questão (*i.e.* a coluna D.P.). Isso porque:

- em um problema mais complexo, o número de épocas NU_EPOCAS necessárias será maior;
- o valor do parâmetro σ_{min} dependerá da tolerância do nível de dispersão do resultado em relação ao erro médio global, para o problema em questão;
- o parâmetro $nu_lobos_inicial$, apesar de formalmente ainda dependente do problema, torna-se menos dada as novas características do WPA2, tendo em vista que o número de lobos no ambiente será ajustado automaticamente pelos operadores de recrutamento e descarte;
- o valor do parâmetro min_erro_lobo indica o nível de sucesso que cada lobo deverá perseguir para atingir seu objetivo;
- demais parâmetros se referem apenas a configuração do *modus operandi* da comunicação e recrutamento.

TABELA II. PARÂMETROS DO WPA2.

Parâmetro	Descrição	D.P.
NU_EPOCAS	Número de épocas a serem executadas.	Sim
σ_{min}	Indica qual o desvio-padrão mínimo desejado para o erro médio.	Sim
$nu_lobos_inicial$	Indica o número de lobos utilizados no início da execução.	Sim
min_erro_lobo	Valor do erro mínimo desejado para o lobo.	Sim
$min_ \%M$	Percentual mínimo desejado de melhoria para o lobo após executar uma ação.	Não
L	Número de vizinhos, os quais o lobo pedirá ajuda.	Não
t_c	Número de épocas que o lobo deve ficar “incomodado” antes de solicitar ajuda aos seus vizinhos.	Não
$\%AEL_{MAX}$	Porcentagem que indica o aumento máximo do erro de um lobo ao ajudar seus vizinhos.	Não
A	Fator de altruísmo dos lobos. Indica a probabilidade de um lobo ajudar seus vizinhos.	Não
NPO	Quantidade das ovelhas com maiores erros dentro da região de atuação do lobo que serão consideradas para a solicitação de ajuda.	Não
$\Delta V_{\%A}$	Percentual de ajuda a ser aplicado na média do erro das piores ovelhas do lobo que solicitou ajuda.	Não
t_R	Número de época que o lobo ficará “incomodado” antes de solicitar o recrutamento.	Não

VI. EXPERIMENTOS

Tendo em vista as modificações promovidas no WPA2, foram realizados experimentos em problemas de uma dimensão com o objetivo de comparar o resultado deste trabalho com o WPA inicial. Assim, foram utilizadas as mesmas configurações do conjunto de pontos de entrada e o mesmo número de épocas da versão inicial, variando o número de lobos inicial para avaliar a capacidade de representação da mesma nuvem de pontos com o uso de menos lobos no ambiente. Com a finalidade de avaliar os novos mecanismos incorporados no WPA2, os experimentos foram divididos nas seguintes categorias: (i) apenas com o mecanismo de melhoria do erro local dos agentes – com 5, 8, 11, 14, 17 e 20 lobos no ambiente – e (ii) com o mecanismo de recrutamento e comunicação ativados – com 5 e 11 lobos iniciais, até um máximo de 15 ou 20 lobos. Os valores dos parâmetros utilizados são descritos na Tabela III.

TABELA III. VALORES DOS PARÂMETROS UTILIZADOS.

Parâmetro	Valor	Parâmetro	Valor
NU_EPOCAS	60	$\%AEL_{MAX}$	10%
σ_{min}	0.001	A	100%
min_erro_lobo	0.008	NPO	8 ovelhas
$min_ \%M$	5%	$\Delta V_{\%A}$	4 épocas
L	Decaimento linear = [5,0]	t_R	Decaimento linear = [0,7,0]
t_c	3 épocas		

Essa configuração dos parâmetros tem o objetivo de incentivar a colaboração e comunicação entre os agentes nos estágios iniciais da execução e desencoraja-la nos estágios finais. Isso, porque a ação de ajudar pode aumentar o erro do

lobo que executou a ação de ajuda, o que é ruim para a convergência do algoritmo em seu estágio final, mas é bom nos estágios iniciais para diminuir o erro da coletividade dos agentes. Assim, nos estágios finais, os lobos estarão buscando diminuir o seu erro local.

Com o objetivo de aplicar o WPA2 em um domínio de problemas mais desafiador, ele foi utilizado para a representação de imagens digitais. Para utilizá-lo nesse domínio de problema, as imagens foram divididas em blocos de 8×8 pixels, transformadas do espaço matricial para o cartesiano e os valores dos pixels normalizados. Assim, o WPA2 foi inicializado e executado em cada bloco em paralelo e, ao final, as imagens aproximadas resultantes foram reconstruídas no espaço matricial a partir das informações dos raios, posicionamentos e amplitude dos lobos. Os valores dos parâmetros utilizados para a representação de imagens são os mesmos daqueles apresentados na Tabela III, com exceção dos seguintes parâmetros: *NU_EPOCAS* (igual a 70), *min_erro_lobo* (igual a 0.03) e *NPO* (igual a 3 ovelhas).

A Figura 7 mostra a transformação do espaço matricial para o espaço cartesiano. É possível observar que o problema é bastante desafiador, tendo em vista que a imagem não obedece a uma lei de formação específica e, em algumas regiões, há uma grande variação dos valores dos pixels. Assim, os experimentos realizados tiveram o objetivo de avaliar o desempenho do algoritmo proposto utilizando uma menor quantidade de funções de núcleo, quando comparado com o *kernel regression* de *Nadaraya-Watson*, e com uma menor quantidade de informação, quando comparado com a representação da imagem no espaço cartesiano. Logo, os experimentos foram realizados em dois tamanhos diferentes da imagem (104×104 e 208×208) e em três diferentes imagens bem conhecidas: *lena*, *pepper* e *cameraman*. Como cada pixel da imagem é representado por x , y e z no espaço cartesiano, o número máximo de lobos para representar cada imagem é 6422 e 25688, respectivamente, conforme a Equação 17.

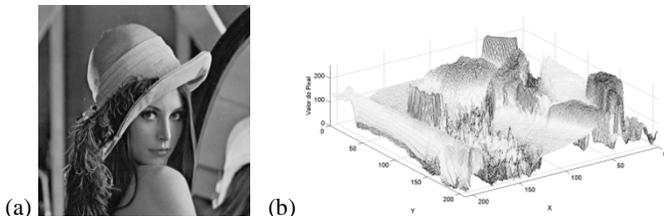


Fig. 7. (a) Imagem original. (b) Imagem original no espaço cartesiano.

No problema de representação de imagens, cada lobo ficou responsável por aproximar uma região de mesmo tamanho do bloco. Assim três configurações diferentes de número de lobos iniciais foram utilizadas por bloco: 8, 16 e 32, onde cada lobo ficou responsável, respectivamente, por 8, 4 e 2 pontos.

As imagens resultantes foram comparadas com aquelas geradas com o uso da DCT e da aplicação de uma máscara de descarte dada pelo filtro de ordenação *zig-zag*, através da métrica PSNR (*Peak Signal-to-Noise Ratio*) [17], onde valores típicos para representação de imagens com perdas estão entre 20dB e 40dB. Além disso, foi utilizada a métrica de qualidade *Structural Similarity* (SSIM) [18], a qual é baseada no sistema visual humano, para calcular o índice de similaridade entra a

imagem original e a calculada. O valor do SSIM varia entre $[-1, 1]$, e será igual a 1 se ambas as imagens forem iguais.

VII. RESULTADOS

A. Comparação com a versão inicial do WPA

Para todas as configurações de pontos de entrada testadas, o WPA proposto obteve melhores resultados, mesmo utilizando um número bem menor de lobos e sem o mecanismo de comunicação e recrutamento ativado. Por exemplo, mesmo quando foram utilizados 5 lobos no ambiente, melhores representações foram obtidas para o cosseno, tangente hiperbólica e quadrática, com erros médio 0.01070, 0.04262 e 0.04855, respectivamente. Ou seja, mesmo utilizando 12.5% dos lobos, foi possível reduzir o erro em, no mínimo, 64%. As Tabelas IV e V mostram os melhores resultados obtidos com o WPA inicial e com o WPA2, respectivamente. Onde, *I. Lobos* indica o número de lobos inicial e *F. Lobos* indica o número de lobos ao final da execução. Além disso, a Figura 8 mostra a representação final para os melhores resultados obtidos para a função quadrática com o WPA inicial e o WPA2, onde nota-se que mesmo utilizando metade dos lobos, o WPA2 obteve uma aproximação mais precisa e ajustada da nuvem de pontos.

TABELA IV. MELHORES RESULTADOS DA ABORDAGEM INICIAL.

Tipo	Lobos	Erro Médio	Desvio-padrão
<i>cos</i>	40	0.11100	0.07625
<i>linear</i>	40	0.09772	0.05912
<i>quadrática</i>	40	0.11890	0.09962
<i>tanh</i>	40	0.13242	0.08611

TABELA V. MELHORES RESULTADOS DO WPA PROPOSTO (WPA2).

Tipo	I. Lobos	F. Lobos	Erro Médio	MSE	Desvio-padrão
<i>cos</i>	11	-	0.00436	3.22E-5	0.00364
<i>cos^b</i>	5	20	0.00138	9.25E-6	0.05912
<i>linear</i>	17	-	0.07331	0.00761	0.04760
<i>linear^b</i>	5	15	0.03204	0.00399	0.00445
<i>quadrática</i>	17	-	0.03262	0.00177	0.02678
<i>quadrática^b</i>	5	20	0.00256	1.33E-5	0.00261
<i>tanh</i>	14	-	0.01115	2.49E-4	0.01125
<i>tanh^b</i>	5	20	0.00552	6.86E-5	0.00619

b.

Experimentos com comunicação e recrutamento ativados.

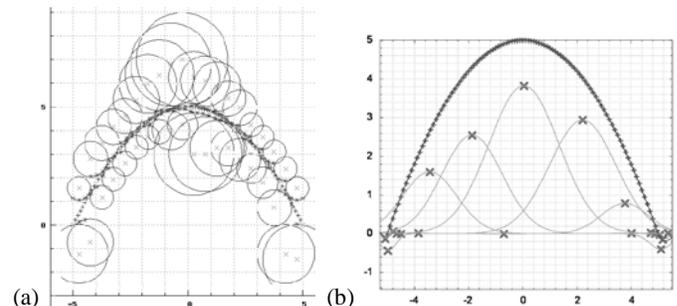


Fig. 8. Melhores resultados para função quadrática: (a) WPA inicial. (b) WPA2.

Ao analisar a Tabela V, é possível perceber que os melhores resultados foram aqueles que utilizaram o mecanismo de comunicação e recrutamento. Com o uso desses operadores, houve vários casos em que mesmo quando um menor número de lobos foi utilizado para representar os pontos de entrada,

foram obtidos melhores resultados do que sem o uso. Por exemplo, quando os operadores de comunicação e recrutamento foram utilizados para a função linear, um menor erro foi obtido (0.03204), mesmo utilizando menos lobos (*i.e.* 15) do que sem o uso desses mecanismos (*i.e.* 17), o que resultou em uma redução de 67% do erro médio e 92% do desvio-padrão.

B. Representação de Imagens Digitais com o WPA2

A Figura 9 mostra as imagens aproximadas com o WPA2, enquanto a Tabela VI sumariza os melhores resultados obtidos para a representação de imagens, onde E.M. significa erro médio; D.P. significa desvio-padrão; %FN significa o percentual de funções de núcleo que foram utilizadas em comparação com o KRNW; e, %Erro significa o percentual de erro. Observa-se que todos os melhores resultados foram obtidos quando os mecanismos de comunicação e recrutamento estavam ativados, o que ressalta a importância desses operadores para a diminuição do erro global.



Fig. 9. Imagens originais e imagens aproximadas com o WPA2.

TABELA VI. MELHORES RESULTADOS DO WPA2 PARA A REPRESENTAÇÃO DE IMAGENS.

	208x208			104x104		
	lena	pepper	cameraman	lena	pepper	cameraman
E.M.	0.623	0.567	0.815	1.209	0.986	1.226
D.P	1.874	1.591	1.989	4.183	2.798	4.794
PSNR	42.224	43.572	41.494	35.384	38.723	34.303
SSIM	0.979	0.981	0.873	0.985	0.988	0.939
Lobos	25283	25125	20946	5834	5544	6285
%FN	58.44	58.07	48.41	53.94	51.26	58.11
%Erro	0.965	0.771	1.065	2.461	1.252	3.564

No que diz respeito à qualidade da representação da imagem, foram obtidos valores de SSIM próximos de 1, o que significa que as imagens representadas com o WPA2 tem um

alto nível de similaridade entre a imagem calculada e a imagem original. O pior índice de similaridade foi obtido para a imagem do cameraman com o tamanho 208x208. Porém, ao final da execução, nota-se que foi utilizado um conjunto menor de lobos do que nas outras imagens de mesmo tamanho. Ao comparar os melhores resultados com o DCT, foi possível constatar que a qualidade da imagem resultante do uso do WPA2 foi equivalente, para a maioria dos casos analisados, à representação dada pela DCT considerando entre 49 e 58 de seus coeficientes mais relevantes (exceto para a imagem pepper no tamanho 208x208, o qual foi equivalente ao DCT considerando 43 de seus coeficientes).

Ao analisar as medidas de erro, nota-se que foram obtidos melhores resultados para a imagem de maior tamanho. Isso se torna evidente quando é observado o desvio-padrão. Para a imagem menor, o desvio-padrão é em torno de duas vezes maior do que a execução da mesma configuração na imagem maior, o que significa uma maior dispersão dos dados em torno da aproximação encontrada. A razão para isso é uma maior ocorrência de regiões de alta frequência na imagem, tornando-a mais difícil de ser representada. Esse mesmo comportamento foi observado ao analisar os resultados do DCT com a métrica PSNR. Para a imagem de menor tamanho, o PSNR é menor do que para a de maior tamanho. A partir do *heatmap* do erro para as imagens *pepper* apresentados na Figura 10, é possível perceber que as regiões de maior ocorrência de erro na imagem menor são aquelas onde há uma frequência mais alta e, normalmente, são pouco perceptíveis aos olhos humanos.

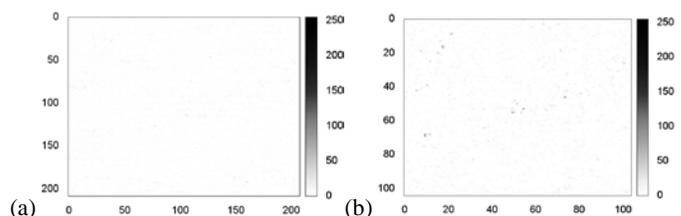


Fig. 10. *Heatmap* do erro para *pepper*: (a) 208x208. (b) 104x104.

Também foi observado durante os experimentos realizados que o erro decai rapidamente com o número de épocas, conforme o gráfico do erro ilustrado na Figura 11. Apesar de extraído do experimento da imagem *lena* 208x208, o mesmo comportamento foi observado para os demais experimentos.

I. CONCLUSÃO

Este trabalho investigou novos mecanismos e alterações a serem incorporadas no modelo inicial do WPA, com o objetivo de (i) garantir uma melhor modelagem do contorno dos pontos de entrada utilizando um conjunto ainda menor de lobos, (ii) resolver as limitações identificadas e (iii) possibilitar a efetiva criação de uma nova metaheurística (WPA2) para a resolução de problemas de aproximação, como a representação de imagens digitais. As principais limitações da abordagem inicial foram tratadas pela nova abordagem, tais como o uso de circunferências para a modelagem dos pontos de entrada e a falta de um mecanismo de comunicação e ajuste do número de lobos no ambiente (a comunicação entre os lobos foi seminal para isso acontecer de forma automática). Assim, foram utilizadas funções de núcleo para permitir uma melhor aproximação dos dados de entrada, com o uso de menos

funções de núcleo que o KRNW (*i.e.* o número de lobo tem que ser menor do que a quantidade de pontos de entrada). Além disso, o mecanismo de recrutamento e descarte de lobos apresentaram avanços significativos no que diz respeito à eliminação do parâmetro que define o número de entidades no ambiente.

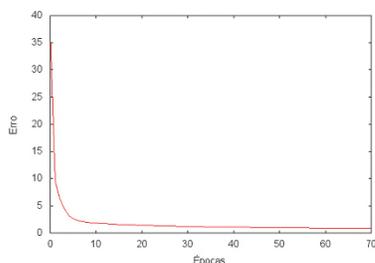


Fig. 11. Gráfico de decaimento do erro médio para a imagem de lena.

A implementação de tais mecanismos viabilizaram a redução significativa do erro e desvio-padrão em relação ao WPA inicial, mesmo utilizando menos da metade do número de lobos. Tendo em vista que todos os melhores resultados obtidos na versão inicial utilizaram 40 lobos no ambiente, na abordagem proposta neste trabalho, menores erros foram alcançados com os seguintes números de lobos (e sem o uso do mecanismo de comunicação e recrutamento): 5 para a função cosseno; 11 para a função linear; 5 para a função quadrática; e, 5 para a função tangente hiperbólica. Ao considerar os melhores resultados obtidos para cada função, nota-se que no pior caso (*i.e.* função linear) o erro diminuiu 67% e o desvio-padrão 92%, enquanto para o melhor caso (*i.e.* função cosseno) o erro diminuiu 99% e o desvio-padrão 96%.

Com o objetivo de demonstrar possíveis desdobramentos e aplicações da metaheurística proposta neste trabalho, o WPA2 também foi utilizado para a representação de imagens digitais, tendo em vista que desdobramentos deste trabalho podem ocorrer no sentido de viabilizar um novo mecanismo de compressão ou de tratamento de imagens digitais. Porém, apesar dos bons resultados evidenciados pelas medidas de qualidade PSNR e SSIM, foi observado que em regiões de maior frequência, o WPA2 apresentou mais dificuldade em aproximar o contorno dos pontos. Esse fato foi evidenciado nos resultados obtidos para as imagens de menor tamanho. O WPA2 conseguiu alcançar plenamente seu objetivo de bem representar a imagem (*i.e.* do conjunto de pontos) com o uso de um conjunto reduzido de funções de núcleo de forma que a representação por lobos não seja maior do que a representação dos pontos de entrada no espaço cartesiano. Espera-se que esta nova abordagem para aproximação de um conjunto de pontos apresentada neste trabalho possa ser estendida e utilizada tanto no domínio de processamento de imagens quanto em problemas de classificação com características complexas.

II. TRABALHOS FUTUROS

Esse trabalho pode ser estendido nos seguintes pontos:

- Desenvolvimento de um mecanismo de inferência de novas ações para a adaptação das ações dos lobos;
- Eliminação de parâmetros do sistema, principalmente os relacionados com o mecanismo de comunicação;

- Desenvolvimento de um mecanismo de penalidade para o recrutamento e para as ações que influenciem negativamente os lobos vizinhos, incorporando um conceito de competição no sistema;
- Utilização de lobos com diferentes funções de núcleo, de modo a aproximar melhor regiões do espaço com diferentes configurações;
- Estudo de viabilidade de utilização do WPA2 para compactação ou processamento de imagens digitais, tais como redução de ruído e reconstrução de imagens.

REFERÊNCIAS

- [1] L. N. de Castro, "Fundamentals of natural computing: an overview," *Physics of Life Reviews*, vol. 4, no. 1, pp. 1–36, Mar. 2007.
- [2] C. J. A. Bastos Filho, F. B. de Lima Neto, A. J. C. C. Lins, A. I. S. Nascimento, and M. P. Lima, "A Novel Search Algorithm based on Fish School Behavior," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2008.
- [3] J. Khan, *et. al.*, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks.," *Nature medicine*, vol. 7, no. 6, pp. 673–9, Jun. 2001.
- [4] H. Bunke and P. S. P. Wang, *Handbook of Character Recognition and Document Image Analysis*. Singapura: World Scientific Pub Co Inc, 1997.
- [5] S. R. Safavian, H. R. Rabiee, and M. Fardaneh, "Projection Pursuit Image Compression with Variable Block Size Segmentation," *IEEE Signal Processing Letters*, vol. 4, no. 5, pp. 117–120, 2007.
- [6] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction.," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 16, no. 2, pp. 349–66, Mar. 2007.
- [7] J. Dobganski, "Programação Linear para Aproximação de Funções Aplicada ao Projeto de Filtros Digitais," Universidade de Campinas - UNICAMP, 1997.
- [8] L. F. A. Pessoa, A. Leonel, L. C. S. Menezes, and F. B. de Lima Neto, "Automatic Neighborhood Control for Intelligent Agents Inspired in Wolf Pack Behavior," *Learning and Nonlinear Models*, vol. 8, no. 4, pp. 216–226, 2010.
- [9] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001.
- [10] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989.
- [11] Y. Wu, H. Wang, B. Zhang, and K.-L. Du, "Using Radial Basis Function Networks for Function Approximation and Classification," *ISRN Applied Mathematics*, vol. 2012, p. 34, 2012.
- [12] D. Nauck and R. Kruse, "Neuro-fuzzy systems for function approximation," *Fuzzy Sets and Systems*, vol. 101, no. 2, pp. 261–271, Jan. 1999.
- [13] A. R. Webb, *Statistical Pattern Recognition*, 2nd ed., vol. 9. Wiley, 2002.
- [14] P. Van Kerm, "Adaptive kernel density estimation," *The Stata Journal*, vol. 3, no. 2, pp. 148–156, 2003.
- [15] E. A. Nadaraya, "On Estimating Regression," *Theory of Probability and its Applications*, vol. 9, no. 1, pp. 141–142, 1964.
- [16] G. S. Watson, "Smooth regression analysis," *Sankhyā: The Indian Journal of Statistics*, vol. 26, no. 4, pp. 359–372, 1964.
- [17] D. Salomon, *Data Compression: The Complete Reference*, 4th ed. Springer Verlag, 2006.
- [18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transaction on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.