

Uma Versão Binária da Meta-heurística Water Flow Optimizer Aplicada à Seleção de Características

Fagner J. Matos Macêdo * Ajalmar R. da Rocha Neto **

* *Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Fortaleza, CE, (e-mail: fagnerjmatosifce@gmail.com).*

** *Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Fortaleza, CE, (e-mail: ajalmar@gmail.com).*

Abstract:

In this work, a binary version of the Water Flow Optimizer (WFO) algorithm, called Binary Water Flow Optimizer (BWFO), is introduced addressing the feature selection problem. WFO is an evolutionary algorithm inspired by the way water flows in nature. In this new approach, the BWFO uses the laminar flow and turbulent flow operators in a binary version, using the accuracy of the Optimum-Path Forest (OPF) classifier as an objective function. The proposed approach is evaluated through a comparative analysis made with classical methods of dimensionality reduction, more specifically with the Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), as well as the metaheuristics Binary Water Wave Optimization (BWWO) and Binary Bat Algorithm (BBA). The computational results demonstrate that the approach is a valid and effective alternative for feature selection problems.

Resumo:

Neste trabalho, é introduzida uma versão binária para o algoritmo *Water Flow Optimizer* (WFO), denominada *Binary Water Flow Optimizer* (BWFO), abordando o problema de seleção de características. O WFO, é um algoritmo evolutivo inspirado na forma do fluxo de água na natureza. Nessa nova abordagem, o BWFO utiliza os operadores de fluxo laminar e fluxo turbulento em uma versão binária, tendo como função-objetivo a acurácia do classificador *Optimum-Path Forest* (OPF). A abordagem proposta é avaliada através de uma análise comparativa feita com métodos clássicos de redução de dimensionalidade, mais especificamente com o *Principal Component Analysis* (PCA) e *Linear Discriminant Analysis* (LDA), bem como as meta-heurísticas *Binary Water Wave Optimization* (BWWO) e *Binary Bat Algorithm* (BBA). Os resultados computacionais demonstram que a abordagem é uma alternativa válida e eficaz para problemas de seleção de características.

Keywords: Feature Selection; Water Flow Optimizer; Evolutionary Computation; Dimensionality Reduction; Artificial Intelligence.

Palavras-chaves: Seleção de Características; Water Flow Optimizer; Computação Evolucionária; Redução de Dimensionalidade; Inteligência Artificial.

1. INTRODUÇÃO

As técnicas de aprendizado de máquina têm sido precursora de vários avanços nas áreas de conhecimento da ciência da computação e suas aplicações, dentre elas: reconhecimento de voz (Chan et al., 2016) e imagens (Foster et al., 2014); robótica (Giusti et al., 2016) e classificação de doenças (Codella et al., 2015). No entanto, mesmo com os avanços, as abordagens mais precisas podem ter seu desempenho ou eficiência degradada devido à alta dimensionalidade dos conjuntos de dados (Papa et al., 2017). Assim, a redução de dimensionalidade surge para atenuar problemas relacionados a ruídos, como também o da alta dimensionalidade dos conjuntos de dados que podem afetar o preditor.

Segundo Ray and Analytics (2015), a redução de dimensionalidade aborda o processo de conversão de um conjunto de dados de alta dimensão para um conjunto de menor dimensão, de forma que as informações sejam mantidas de forma clara e resumidas. A redução de dimensionalidade pode ser feita de duas maneiras diferentes: via seleção de características (*feature selection*), em que selecionamos um subconjunto dos atributos de maior relevância, e via extração de características (*feature extraction*), em que derivamos informações do conjunto de atributos para construir um novo subespaço de atributos.

Dentre os métodos de extração de características, métodos clássicos como o *Principal Component Analysis* (PCA) e o *Linear Discriminant Analysis* (LDA) são bem conhecidos e bastante utilizados para a tarefa de redução de dimensionalidade. Dentre os métodos de seleção de características,

existem diversos trabalhos que abordam o tema, modelando o problema como uma tarefa de otimização através de meta-heurísticas.

Os autores Macêdo et al. (2019), propuseram uma versão binária para a meta-heurística *Water Wave Optimization* (WWO), denominada *Binary Water Wave Optimization* (BWWO). Os autores remodelaram os operadores de propagação, operador de quebra da onda e refração para uma versão binária, obtendo bons resultados para a tarefa de seleção de características. O BWWO utilizou a acurácia retornada pelo *Optimum-Path Forest* (OPF) como função de aptidão para avaliar cada solução durante a fase de pesquisa do algoritmo. Já os autores Nakamura et al. (2012), apresentaram uma versão binária do *Bat Algorithm* (BA), denominada *Binary Bat Algorithm* (BBA). Neste trabalho, os autores modelaram o BBA e validaram na tarefa seleção de características como uma técnica de otimização. A busca é dirigida pelo BBA, e as soluções candidatas são avaliadas junto ao classificador OPF. No mesmo contexto, Rosa et al. (2017) propuseram o *Geometric Semantic Genetic Programming* (GSGP). Assim, demonstra-se que em seus experimentos o GSGP consegue desempenhar satisfatoriamente, obtendo bons resultados. Rodrigues et al. (2013) propuseram o *Binary Cuckoo Search* (BCS), baseado na meta-heurística *Cuckoo Search* (CS), o qual se inspira no comportamento dos pássaros cuco. No BCS, a função a ser maximizada é dada pela acurácia do classificador OPF. Diante disso, perceber-se que o uso de meta-heurísticas tem se destacado como uma forma eficiente na atividade de seleção de características.

Recentemente, uma nova meta-heurística conhecida como *Water Flow Optimizer* (WFO) foi proposta, tendo como inspiração dois tipos de fluxos de água existentes na natureza, são eles: fluxo laminar e o fluxo turbulento. O fluxo laminar tem por característica a baixa velocidade e fluidez contínua, de forma que o fluido passa a se movimentar de forma retilínea. Por sua vez, o fluxo turbulento tem por característica uma fluidez mais rápida, em que o fluido passa a ter movimentos ondulados, fluindo de forma irregular (Luo, 2021). O WFO é um algoritmo recente que tem resultados significativos, e abre espaço para novas versões, inclusive a binária. Além disso, o baixo número de operadores e o baixo número de parâmetros se demonstra vantajoso para a meta-heurística.

Neste contexto, o presente trabalho propõe uma versão binária da meta-heurística WFO, chamada *Binary Water Flow Optimizer* (BWFO), aplicando-a ao problema de seleção de características, de forma que o novo algoritmo binário seja uma alternativa eficaz e efetiva na redução de dimensionalidade para conjuntos de dados de alta dimensão, sendo assim, capaz de melhorar a previsão e o processamento sobre quaisquer dados.

O restante deste trabalho está organizado da seguinte forma: na Seção 2, é apresentada a meta-heurística WFO, visando uma melhor compreensão de sua estrutura e de seus operadores. Na Seção 3, o método proposto BWFO é apresentado, trazendo modificações dos novos operadores binário de fluxo laminar e fluxo turbulento. Posteriormente, na Seção 4 os resultados obtidos são apresentados e discutidos. Finalmente, é feita as considerações finais deste artigo na Seção 5.

2. WATER FLOW OPTIMIZER

2.1 Fluxo de água na hidráulica

O cientista britânico Osborne Reynolds, em 1883, conduziu experimentos verificando o comportamento do fluxo de um filete de líquido colorido ao injetá-lo em um tubo de vidro transparente com uma válvula na extremidade. Reynolds observou que quando a válvula foi aberta lentamente, a velocidade do fluxo de água no tubo foi pequena e o líquido colorido fluíu em linha reta por todo o comprimento do tubo. No entanto, a medida em que a válvula foi aberta, houve um aumento gradual da velocidade e o fluxo do líquido tomou formas irregulares, tornando-se primeiramente ondulado e depois surgindo vórtices (Abbott and Minns, 2017). Este fenômeno regular, ocorrido pelo escoamento de menor velocidade é definido como fluxo laminar. Por sua vez, o fenômeno irregular é definido como fluxo turbulento. O fluxo de água pode ser identificado pelo número de Reynolds, que é definido pela razão entre as forças inerciais e as forças viscosas. Em geral, o fluxo de água é laminar se o número de Reynolds for menor que um limiar e turbulento, caso contrário (Abbott and Minns, 2017).

Em suma, o fluxo laminar ocorre quando a velocidade do fluxo de água é pequeno, de forma que as partículas de água distribuídas em diferentes camadas se movem regularmente em linhas retas paralelas, como exemplificado na Figura 1a. Já o fluxo turbulento, ocorre quando a velocidade de água é grande, fazendo com que as partículas se choquem de maneira estocástica e se movam sobre as demais camadas, como pode ser visto na Figura 1b.

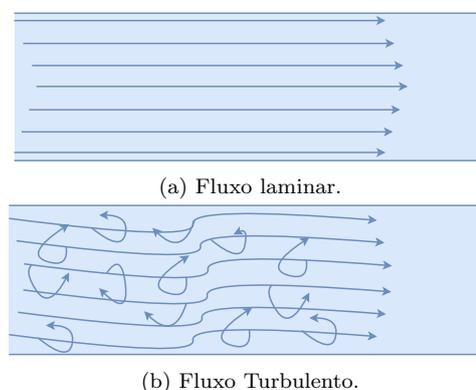


Figura 1. Representação das formas do fluxo da água.

2.2 Modelagem matemática do Water Flow Optimizer

O algoritmo *Water Flow Optimizer* (WFO) (Luo, 2021) é um método evolutivo baseado no fluxo da água para solucionar problemas de otimização global. No WFO, assim como nos demais algoritmos evolutivos, a solução ótima pode ser encontrada por um processo iterativo, buscando a cada iteração a evolução de suas soluções, as quais, são avaliadas por uma função-objetivo. Nesse algoritmo, cada solução é uma partícula de água, e a sua posição é análoga ao valor da solução, bem como a energia potencial da partícula é semelhante a seu valor de aptidão. Assim, a função-objetivo é utilizada para avaliar a

energia potencial de cada partícula no espaço de soluções. O fluxo laminar e o fluxo turbulento presente na mecânica dos fluidos, servem de inspiração para os operadores do WFO, sendo estes projetados para realizarem pesquisas estocásticas regulares e irregulares na otimização (Luo, 2021). Durante o processo de solução do problema, as operações de fluxo laminar e fluxo turbulento são aplicadas para evoluir a população.

Operador laminar No fluxo laminar, todas as partículas se movem em linhas retas paralelas. Devido à viscosidade da água, a velocidade das partículas podem ser diferentes, por exemplo, as partículas em camadas próximas de paredes ou obstáculos perdem velocidade quando comparada as partículas que estão mais afastadas. O movimento regular denotado pelo fluxo laminar foi modelado pela seguinte equação:

$$y_i(t) = x_i(t) + s * \vec{d} \quad \forall i \in \{1, 2, \dots, n\}, \quad (1)$$

$$\vec{d} = x_b(t) - x_k(t), \quad (b \neq k, \quad f(x_b(t)) \leq f(x_k(t))). \quad (2)$$

Assim, t é o número da iteração, i é o índice da partícula na população de soluções e n é o tamanho da população de partículas de água empregadas, $x_i(t)$ é a posição da i -ésima partícula na t -ésima iteração, $y_i(t)$ denota sua possível posição de movimento, o número aleatório $s \in U([0, 1])$ representa o coeficiente de deslocamento de uma partícula de água, e o vetor \vec{d} indica uma direção de movimento comum para todas as partículas.

A direção comum é determinada por duas partículas distintas selecionadas, em que $x_b(t)$ é a melhor solução encontrada na t -ésima iteração, e $x_k(t)$ é uma partícula selecionada aleatoriamente na t -ésima iteração, tal que, considerando um problema de minimização, a aptidão da melhor solução $x_b(t)$ deve ser menor que a aptidão da partícula escolhida aleatoriamente $x_k(t)$, sendo b e k , o índice dessas partículas e obrigatoriamente diferentes. A melhor solução encontrada até o momento é selecionada como partícula guia na implementação do operador laminar.

O valor de \vec{d} na Equação (1), permanece inalterado para todas as partículas da iteração, já o valor de s varia de partícula para partícula. O operador laminar no algoritmo WFO adota um padrão de pesquisa unidirecional paralela regular, em que \vec{d} , visa a busca unidirecional paralela, e a aleatoriedade de s garante que as diferentes partículas de água tenham deslocamentos distintos.

Operador turbulento No fluxo turbulento, há um movimento irregular das partículas de água, que faz com que o fluxo retilíneo seja interrompido. Devido ao rápido fluxo de água, as partículas se chocam de maneira aleatória e se movem sobre as camadas adjacentes, visto que a amplitude da oscilação é grande o suficiente. A força de cisalhamento, faz com que as partículas de água girem, formando redemoinhos (Luo, 2021). No operador de fluxo turbulento, a dimensão do problema é considerada como uma camada no fluxo de água. Assim, o operador simula o fenômeno turbulento por realizar uma transformação entre as dimensões, a fim de produzir um movimento irregular nas partículas.

A movimentação da partícula é produzida por oscilar uma dimensão aleatoriamente conforme a Equação (3):

$$y_i(t) = (\dots, x_{i,j_1-1}(t), u, x_{i,j_1+1}(t), \dots), \quad (3)$$

em que $j_1 \in \{1, 2, \dots, m\}$ é uma dimensão selecionada aleatoriamente de m dimensões, e o valor de mutação u é determinado pela seguinte equação:

$$u = \begin{cases} \psi(x_{i,j_1}(t), x_{k,j_1}(t)), & \text{se } r < p_e \\ \varphi(x_{i,j_1}(t), x_{k,j_2}(t)), & \text{caso contrário.} \end{cases} \quad (4)$$

Deste modo, $k \in \{1, 2, \dots, n\}$ é o índice de uma partícula distinta selecionada aleatoriamente, tal que $k \neq i$, j_2 é uma dimensão distinta selecionada aleatoriamente, em que $j_2 \neq j_1$, r é um número aleatório no intervalo $[0, 1]$ e $p_e \in (0, 1)$ é um parâmetro de controle denominado probabilidade de turbulência.

O operador de fluxo turbulento utiliza uma transformação de turbulência ψ com probabilidade p_e e uma operação de movimentação geral sobre camada φ com probabilidade $1 - p_e$. A transformação de turbulência ψ é construída com base em uma espiral arquimediana, pela semelhança com um redemoinho, conforme a Equação (5), descrita por:

$$\psi(x_{t,j_1}(t), x_{k,j_1}(t)) = x_{i,j_1}(t) + \rho * \theta * \cos(\theta), \quad (5)$$

em que θ é um número aleatório dentro do intervalo $[-\pi, \pi]$ e ρ é determinado dinamicamente por uma regra auto-adaptativa conforme a Equação (6)

$$\rho = |x_{i,j_1}(t) - x_{k,j_1}(t)|. \quad (6)$$

Então, ρ pode ser visto como a força de cisalhamento da k -ésima partícula para a i -ésima partícula. Certamente, também é praticável usar a função seno básica (Equação (5)) devido a $\sin(\theta) = \cos(\theta - [\pi/2])$.

A função φ , por sua vez, aplicada com probabilidade $1 - p_e$, visa simular o comportamento geral das partículas sobre as camadas. A seguinte função de transformação é empregada:

$$\varphi(x_{i,j_1}(t), x_{k,j_2}(t)) = (ub_{j_1} - lb_{j_1}) * \frac{x_{k,j_2}(t) - lb_{j_2}}{ub_{j_2} - lb_{j_2}} + lb_{j_1}, \quad (7)$$

em que lb e ub , respectivamente, representam os limites inferiores e superiores. A transformação linear pode herdar a escala de uma partícula selecionada aleatoriamente para os limites em uma dimensão selecionada.

2.3 Algoritmo

A evolução das partículas de água no WFO, é alcançado através dos operadores de fluxo laminar e fluxo turbulento. Os fluxos ocorrem através de um limiar, no qual, como explicado anteriormente, se dá pelo número de Reynolds, em que, se o fluxo da água for menor que o número de Reynolds, este passa a ser laminar e caso contrário será

turbulento. Esses operadores são simulados de forma estocástica. O operador laminar é executado com probabilidade p_l , já o operador turbulento, por sua vez, é executado com probabilidade de $1 - p_l$. O parâmetro p_l , no algoritmo se comporta como um limiar, visto que com base no seu valor que se determina a operação a ser executada (Luo, 2021).

Em cada iteração o algoritmo avalia suas partículas após a execução dos operadores de fluxo laminar e turbulento, com isso, a partícula avaliada que realiza uma movimentação que aumente sua energia potencial, ou seja, melhore sua *fitness*, é selecionada para a próxima iteração do algoritmo, caso contrário, segue o movimento anterior da partícula, antes da aplicação dos operadores na iteração atual. A Equação (8) apresenta a regra para atualização das partículas para a próxima iteração.

$$x_i(t+1) = \begin{cases} y_i(t), & \text{se } f(y_i(t)) < f(x_i(t)) \\ x_i(t), & \text{caso contrário.} \end{cases} \quad (8)$$

2.4 Inicialização e Operadores

O algoritmo WFO necessita que antes de sua inicialização, os valores para os parâmetros p_l e p_e sejam definidos. Além das configurações dos parâmetros, é necessário que as posições e a energia potencial de cada partícula sejam inicializadas. Luo (2021), em seus experimentos contra os demais algoritmos evolutivos, fixou em seus testes valores como: $p_l = 0.3$, $p_e = 0.7$ e o tamanho da população $n = 50$. A posição da partícula por sua vez é inicializada aleatoriamente dentro do espaço de busca, e após isso, cada partícula é calculada diretamente utilizando a função-objetivo.

Em resumo, o operador de fluxo laminar faz com que as partículas de água movam-se de forma serena em direção a solução ótima, o operador de fluxo turbulento intensifica a exploração dentro do espaço de busca, fazendo com que as partículas façam por sua vez movimentos relativamente desordenados. O parâmetro de probabilidade de turbulência p_e , é relativamente propenso a impactar o desempenho do algoritmo, visto que, para valores muito baixos, a chance de operação de movimentação sobre camadas aumenta, conseqüentemente, pode ocorrer um aumento do risco de convergência prematura. Vale salientar que no operador de fluxo laminar o valor de s , como pode ser observado no Algoritmo 1, na linha 9, é distinto para cada partícula, mas idêntico para cada camada da mesma partícula. Já o valor de \vec{d} , linha 7, pode ser diferente em diferentes iterações, mas idêntico para cada partícula de água durante a mesma iteração. É bom observar que, na computação de \vec{d} , a melhor partícula na iteração funciona como um guia, para acelerar a taxa de convergência.

3. PROPOSTA: BINARY WATER FLOW OPTIMIZER

Neste trabalho, o algoritmo WFO é adaptado para ser capaz de resolver problemas binários, visto que em sua versão original o algoritmo é desenvolvido para solucionar apenas problemas de otimização contínuo. Mais especificamente, mostraremos a capacidade da nossa proposta de resolver

Algoritmo 1 Algoritmo WFO

```

1: para cada partícula  $i \in \{1, 2, \dots, n\}$  fazer
2:   para cada dimensão  $j \in \{1, 2, \dots, m\}$  fazer
3:     Gere aleatoriamente uma posição  $x_{i,j} \leftarrow lb_j + rand() * (ub_j - lb_j)$ 
4:     Calcule a energia potencial,  $f_i \leftarrow f(x_i)$ 
5: enquanto o critério de parada não é satisfeito fazer
6:   se  $rand() < p_l$  então
7:     Determine uma direção laminar,  $d \leftarrow x_b - x_k$ 
8:     para cada partícula  $i \in \{1, 2, \dots, n\}$  fazer
9:        $s \leftarrow rand()$ 
10:       $y_i \leftarrow x_i + s * \vec{d}$ 
11:   senão
12:     para cada partícula  $i \in \{1, 2, \dots, n\}$  fazer
13:       Inicialize uma posição temporária,  $y_i \leftarrow x_i$ .
14:       Selecione uma posição aleatória,  $k \in \{1, 2, \dots, n\}$  e  $k \neq i$ .
15:       Selecione aleatoriamente um dimensão,  $j_1 \in \{1, 2, \dots, m\}$ .
16:       se  $rand() < p_e$  então
17:          $\rho \leftarrow |x_{i,j_1} - x_{k,j_1}|$ 
18:          $\theta \leftarrow 2 * \pi * rand() - \pi$ 
19:          $y_{i,j_1} \leftarrow x_{i,j_1} + \rho * \theta * \cos(\theta)$ 
20:       senão
21:         Selecione aleatoriamente um dimensão,  $j_2 \in \{1, 2, \dots, m\}$  e  $j_2 \neq j_1$ .
22:          $y_{i,j_1} \leftarrow (ub_{j_1} - lb_{j_1}) * \frac{x_{i,j_2} - lb_{j_2}}{ub_{j_2} - lb_{j_2}} + lb_{j_1}$ 
23:     para cada partícula  $i \in 1, 2, \dots, n$  fazer
24:       se  $f(y_i) < f_i$  então
25:          $f_i \leftarrow f(y_i)$ 
26:          $x_i \leftarrow y_i$ 
27:       se  $f_i < f_b$  então
28:          $f_b \leftarrow f_i$ 
29:          $x_b \leftarrow x_i$ 
30:          $b \leftarrow i$ 

```

problemas binários por aplicá-la na tarefa de seleção de características aplicadas em problemas de classificação. Para isso, será necessário remodelar os operadores do WFO, sendo eles: o operador de fluxo laminar e o operador de fluxo turbulento. Além disso, será preciso estabelecer uma função de aptidão para guiar o algoritmo em busca da melhor solução.

3.1 Representação do problema

A solução para o problema será uma partícula de água, que por sua vez é formada por um conjunto de dimensões (camadas), em que cada dimensão é associada a uma característica. Mais especificamente, a partícula é representada por um vetor binário $x = [d_1 \ d_2 \ \dots \ d_j \ \dots \ d_{m-1} \ d_m]$, em que $d_j \in \{0, 1\}$ é a j -ésima dimensão da onda e m é o número de atributos da base de dados em questão. O valor 1 determinada se a característica permanecerá para a construção do novo conjunto de dados e o valor 0 definirá se a característica será excluída.

3.2 Operador de Fluxo Laminar Binário

A aplicação da nova abordagem para o operador de fluxo laminar é empregada em cada partícula de água da iteração atual. Portanto, cada partícula pode sofrer ou não alteração em sua camada aleatoriamente, a qual é decidida por um número aleatório $s \in [1, 0]$ gerada para cada camada da partícula na iteração, conforme a equação

$$y_{i,j}(t) = \begin{cases} x_{b,j}(t), & \text{se } s_j < \text{rand}() \\ x_{i,j}(t), & \text{caso contrário.} \end{cases} \quad (9)$$

Assim, cada camada j escolhida para alteração, é substituída pela camada da melhor partícula vigente na iteração. O operador de fluxo laminar binário, faz com que partículas de água consigam obter camadas da partícula de melhor energia potencial na geração atual, realizando uma nova movimentação em direção da melhor partícula.

A Figura 2, exemplifica a execução do operador laminar binário. As flechas direcionais verdes indicam que a primeira condição da Equação (9) foi satisfeita, e assim a j -ésima camada da partícula x_i recebe o valor da respectiva camada da melhor partícula atual x_b . Já as flechas azuis indicam que não houveram alterações nos valores das camadas inpede da condição ter sido satisfeita, ou não, visto os valores das respectivas camada de x_b , ter os mesmos valores da partícula x_i . As flechas vermelhas indicam que a primeira condição não foi satisfeita, assim a camada de x_i continua com o mesmo valor atual.

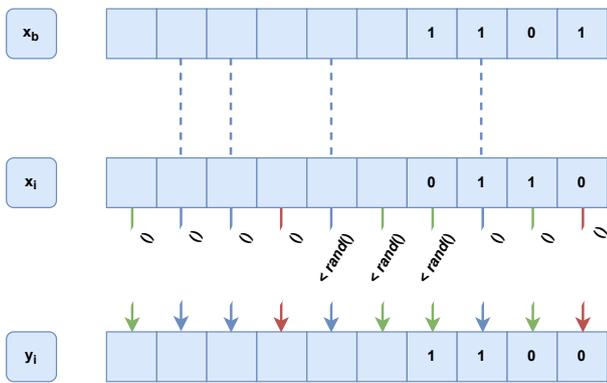


Figura 2. Processo do operador de fluxo laminar binário.

3.3 Operador de Fluxo Turbulento Binário

O novo operador de fluxo turbulento, por sua vez utiliza o parâmetro de probabilidade de turbulência p_e para escolher quais camadas irão sofrer alterações. Quando o operador de turbulência é aplicado, é selecionado uma partícula de água x_k aleatoriamente na população, assim, a partícula a ser modificada recebe a camada da partícula x_k se ambas forem diferentes, caso contrário a camada da partícula atual recebe aleatoriamente um valor $v \in \{0, 1\}$, assumindo assim de forma aleatória o valor 0 ou 1, conforme descrito pela equação

$$y_{i,j}(t) = \begin{cases} x_{k,j}(t), & \text{se } x_{k,j}(t) \neq x_{i,j}(t) \\ v \in \{0, 1\}, & \text{caso contrário.} \end{cases} \quad (10)$$

O operador de fluxo turbulento binário, visa aderir características das demais partículas ou sofrer uma espécie de mutação quando a camada selecionada da partícula x_k , tem o mesmo valor da partícula x_i , trazendo assim, mais diversidade para a pesquisa dentro do espaço de busca.

3.4 Função de Aptidão

A função de aptidão (*fitness*) visa avaliar a evolução das soluções encontradas durante a execução das gerações de um determinado algoritmo evolutivo, neste caso o BWFO. Visto que o problema a ser resolvido é o de seleção de características aplicada ao problema de classificação, a função de aptidão implementada neste trabalho utiliza o valor de acurácia retornado por um classificador após a redução de dimensionalidade de um conjunto de dados. Diante disso, a tarefa é tratada como um problema de maximização. O classificador escolhido é o OPF, visto que o mesmo é uma técnica rápida de reconhecimento de padrões (Papa et al., 2011).

3.5 Framework BWFO

O Algoritmo 2, apresenta a ideia geral para o algoritmo BWFO, aqui utilizado para o problema de seleção de características. O BWFO segue a estrutura do WFO, sendo realizado somente alteração do funcionamento de seus operadores na nova versão proposta. O parâmetro de probabilidade laminar p_l , continua como um limiar de decisão para a aplicação do fluxo laminar ou turbulento. Já no fluxo turbulento, o parâmetro de probabilidade de turbulência p_e é utilizado como um decisor para realizar a modificação de cada camada das partículas. A principal característica do operador laminar, é que o mesmo visa que as partícula venham a aderir características da melhor solução na iteração atual, de forma que a partícula atual movimente-se em direção a melhor. O operador turbulento faz com que a diversidade de soluções aumente, fazendo com que partículas obtenham características das demais soluções, ou por sofrer uma espécie de mutação em seu valor.

Assim como o algoritmo original, o BWFO também é um algoritmo simples de ser implementado sendo um algoritmo de poucos parâmetros, como os parâmetros de probabilidade laminar e o de probabilidade de turbulência.

4. SIMULAÇÕES COMPUTACIONAIS

Os experimentos foram conduzidos utilizando como métodos de comparação ao BWFO as meta-heurísticas *Binary Water Wave Optimization* (BWFO) (Macêdo et al., 2019) e *Binary Bat Algorithm* (BBA) (Nakamura et al., 2012). Além disso, foram utilizados também, os métodos clássicos de redução de dimensionalidade, a saber: PCA e o LDA.

Inicialmente, os conjuntos de dados foram normalizados, embaralhados e particionados em três subconjuntos: conjunto de treinamento (Z_1), validação (Z_2) e teste (Z_3); em que, 30% dos dados foram destinados ao conjunto de treinamento, 30% para o conjunto de validação e os outros 40% restantes para o conjunto de teste. Durante a execução do BWFO, novos subconjuntos Z'_1 , Z'_2 e Z'_3 são gerados de acordo com os valores binários presentes em cada solução gerada aleatoriamente, em que o valor 1 e 0 representam, respectivamente, a presença e a ausência de um atributo nos novos subconjuntos. Então, o algoritmo OPF é treinado com Z'_1 e classifica Z'_2 para avaliar a aptidão de cada solução candidata. Esse processo é repetido durante um determinado número de iterações ou quando

Algoritmo 2 Algoritmo BWFO

Entrada: Conjunto de treinamento Z_1 e validação Z_2 λ -rotulados e um número T de iterações.

Saída: Partícula de melhor acurácia x_b .

```

1: Inicialize aleatoriamente uma população de  $n$  soluções
2: Calcule a energia potencial de cada partícula (acurácia)
3: enquanto o critério de parada não é satisfeito fazer
4:   se  $rand() < p_l$  então
5:     Selecione o índice da melhor partícula vigente  $b$ 
6:   para cada partícula  $i \in \{1, 2, \dots, n\}$  fazer
7:     para cada dimensão  $j \in \{1, 2, \dots, m\}$  fazer
8:        $s \leftarrow rand()$ 
9:       se  $s < rand()$  então
10:         $y_{i,j} \leftarrow x_{b,j}$ 
11:   senão
12:     para cada partícula  $i \in \{1, 2, \dots, n\}$  fazer
13:       para cada dimensão  $j \in \{1, 2, \dots, m\}$  fazer
14:         Selecione aleatoriamente o índice de uma partícula  $k$ 
15:         se  $rand() < p_e$  então
16:           se  $y_{i,j} \neq x_{k,j}$  então
17:              $y_{i,j} \leftarrow x_{k,j}$ 
18:         senão
19:            $v \leftarrow randint(0,1)$ 
20:            $y_{i,j} \leftarrow v$ 
21:   para cada partícula  $i \in 1, 2, \dots, n$  fazer
22:     se  $f(y_i) > f_i$  então
23:        $f_i \leftarrow f(y_i)$ 
24:        $x_i \leftarrow y_i$ 
25:     se  $f_i > f_b$  então
26:        $f_b \leftarrow f_i$ 
27:        $x_b \leftarrow x_i$ 
28:        $b \leftarrow i$ 
    
```

o melhor valor de aptidão de uma onda tenha convergido para um valor específico. Neste trabalho, o número de gerações foi definido como 20, e o valor de aptidão de uma onda é visto como convergido, caso o valor de melhor solução até o momento não progrida durante cinco gerações. Após os critérios de parada serem satisfeitos, o BWFO retorna como resposta a melhor solução encontrada, ou seja, um vetor binário mapeando os atributos da base de dados que permitiram que o classificador obtivesse uma melhor acurácia em Z'_2 . Após isso, o classificador treinado em Z'_1 é utilizado para computar um novo valor de acurácia em Z'_3 .

Neste trabalho, foram utilizadas onze bases de dados obtidas do repositório *UCI Machine Learning*¹, a saber: *Australian Credit Approval* (ACA), *Breast Cancer Wisconsin* (BCW), *Dermatology* (DM), *Vertebral Column* (VC), *Liver Disorders* (BLD), *Pima Indians Diabetes* (PID), *Divorce Predictors* (DIV), *Parkinson's Disease* (PDS), *Turkish Music Emotion* (TKM), *Urban Land Cover* (UBL) e *Arrhythmia* (ARR)². A Tabela 1 apresenta as principais características desses conjuntos.

4.1 Experimentos Realizados

A fim de realizar uma comparação justa entre os métodos, os experimentos foram conduzidos utilizando os mesmos padrões de dados para cada abordagem e o mesmo classificador, o OPF. Portanto, cada método recebe como parâmetro exatamente os mesmos subconjuntos Z_1 , Z_2 e

¹ <https://archive.ics.uci.edu/ml/>

² O conjunto de dados Arrhythmia foi modificado para considerar apenas duas classes no problema, considerando a normalidade e a existência de arritmia nos batimentos cardíacos.

Tabela 1. Descrição dos conjuntos de dados utilizados no trabalho.

Dataset	# Amostras	# Características	# Classes
ACA	690	14	2
BCW	699	10	2
DM	366	33	6
VC	310	6	3
BLD	345	6	2
PID	768	8	2
DIV	170	54	2
PDS	756	753	2
TKM	400	50	4
UBL	540	147	9
ARR	452	279	16

Z_3 utilizados no BWFO. Além disso, para o PCA e o LDA, a quantidade de componentes e discriminantes são escolhidos com base na quantidade de características selecionadas pelo BWFO, na rodada. Visto que o PCA e o LDA comparados ao BWFO, BWWO e BBA não necessitam de um conjunto de validação, apenas os subconjuntos Z_1 e Z_3 foram utilizados por eles. No caso do PCA e LDA, Z_1 e Z_3 são transformados, respectivamente, em Z'_1 e Z'_3 cada qual possuindo um número de componentes igual ao número de características retornadas pelo BWFO. Todo esse processo é repetido 20 vezes para computar os valores de acurácia, desvio padrão, e o número médio de atributos retornados por cada método de redução de dimensionalidade. A Tabela 2, resume as configurações dos parâmetros de cada abordagem utilizada para os experimentos. Os parâmetros utilizados para o BWWO e BBA foram escolhidos com base nos trabalhos de seus autores.

Tabela 2. Resumo dos parâmetros de cada método.

Métodos	Parâmetros
BWFO	$p_l = 0.3$ e $p_e = 0.7$
BWWO	$kMax = \min(12, dim/2)$, $hMax = 3$ e $\alpha = 0.01$
BBA	$\alpha = 0.9$ e $\gamma = 0.9$
PCA	N° componentes = N° atributos retornado pelo BWFO
LDA	N° discriminantes = N° atributos retornado pelo BWFO

4.2 Análise dos resultados

As Tabelas 3 e 4 apresentam os resultados obtidos pelos métodos empregados neste artigo levando em consideração os tamanhos de população 100 e 120, respectivamente. Esses resultados são referentes a acurácia, desvio padrão, e a média do número de características selecionados ($\#NC$), medidos durante 20 execuções independentes para as base de dados PID, VC, ACA, BLD, BCW e DM presentes na Tabela 1. Note que nas tabelas podem ser visualizados somente os valores do número de características do BWFO, BWWO e BBA. Isso ocorre porque o número de componentes e discriminantes utilizados no PCA e LDA, respectivamente, são iguais ao número de características selecionados pelo BWFO, já para o OPF, os experimentos são realizados sem redução de dimensionalidade para os conjuntos de dados.

Pode-se observar na Tabela 3, que para tamanhos de população igual a 100, o BWFO alcançou bons resultados no quesito acurácia. Nos testes realizados, em vários momentos o método proposto superou as técnicas clássicas de redução de dimensionalidade, como o PCA e o

Tabela 3. Resultados do OPF, PCA, LDA, BWFO e BBA versus BWFO com tamanho de população 100.

Base	Modelo	Acurácia	#NC
PID	BWFO	64,28 ± 2,45	3,95
	BWFO	64,26 ± 2,47	3,81
	BBA	64,85 ± 2,44	4,05
	OPF	64,69 ± 2,86	-
	PCA	62,43 ± 2,74	-
VC	LDA	62,69 ± 3,25	-
	BWFO	78,67 ± 3,33	3,55
	BWFO	78,67 ± 3,55	3,45
	BBA	78,39 ± 3,50	3,50
	OPF	78,04 ± 2,84	-
ACA	PCA	75,19 ± 4,01	-
	LDA	76,30 ± 5,33	-
	BWFO	82,72 ± 3,70	5,71
	BWFO	83,14 ± 2,72	5,14
	BBA	82,54 ± 3,86	5,57
BLD	OPF	77,79 ± 2,07	-
	PCA	75,77 ± 3,16	-
	LDA	78,18 ± 2,71	-
	BWFO	54,92 ± 4,26	3,29
	BWFO	54,81 ± 4,23	3,19
BCW	BBA	54,92 ± 4,26	3,29
	OPF	58,13 ± 5,31	-
	PCA	53,62 ± 4,46	-
	LDA	57,64 ± 4,89	-
	BWFO	93,70 ± 1,47	5,43
DM	BWFO	94,07 ± 1,24	5,38
	BBA	93,71 ± 1,28	5,81
	OPF	94,27 ± 0,99	-
	PCA	94,15 ± 1,62	-
	LDA	93,70 ± 2,05	-
	BWFO	96,07 ± 1,31	20,14
	BWFO	95,89 ± 1,41	19,26
	BBA	95,97 ± 1,56	19,47
	OPF	95,75 ± 0,99	-
	PCA	95,61 ± 1,11	-
	LDA	91,69 ± 6,61	-

LDA, obtendo melhores resultados em relação ao PCA em quase todos os experimentos. Por sua vez, o PCA esteve à frente do BWFO somente na base dados BCW. Já quando comparado ao LDA, o BWFO também superou o LDA na maior parte dos experimentos, sendo superado somente na base de dados BLD e igualando seus valores na base BCW.

Já contra as demais meta-heurísticas, o BWFO esteve à frente do BWFO em três oportunidades, sendo elas nas bases PID, BLD e DM, e empatando em termos de acurácia na base de dados VC. Contra o BBA, o BWFO o superou em três oportunidades, sendo elas, nas bases VC, ACA e DM, tendo os dois empatados na base BLD. Pode-se notar que o método proposto desempenha bem nos confrontos com os demais métodos, apesar de reduzir um pouco menos o número de características.

Com base na Tabela 4, por sua vez com tamanho de população 120, o BWFO mantém bons resultados sobre os métodos clássicos de redução de dimensionalidade. O BWFO superou o PCA nas bases PID, VC, ACA, BLD e DM. Contra o LDA, o BWFO também obteve melhores resultados em cinco oportunidades, nas bases PID, VC, ACA, BCW e DM. É notório a eficácia dos métodos baseados em meta-heurísticas sobre os métodos clássicos, pode-se perceber com base nas tabelas de resultados a superioridade desse tipo de abordagem, porém o custo computacional é maior por conta da fase de busca desses algoritmos.

Tabela 4. Resultados do OPF, PCA, LDA, BWFO e BBA versus BWFO com tamanho de população 120.

Base	Modelo	Acurácia	#NC
PID	BWFO	63,58 ± 3,50	4,05
	BWFO	63,57 ± 2,97	3,85
	BBA	63,43 ± 3,10	3,85
	OPF	64,07 ± 2,88	-
	PCA	62,59 ± 3,13	-
VC	LDA	62,37 ± 4,09	-
	BWFO	78,19 ± 3,00	3,45
	BWFO	78,19 ± 3,00	3,45
	BBA	78,01 ± 3,03	3,50
	OPF	77,52 ± 2,20	-
ACA	PCA	74,36 ± 4,16	-
	LDA	74,31 ± 5,41	-
	BWFO	81,40 ± 2,92	6,30
	BWFO	82,33 ± 3,17	5,05
	BBA	81,10 ± 3,60	5,90
BLD	OPF	78,19 ± 2,53	-
	PCA	74,20 ± 3,49	-
	LDA	78,28 ± 3,55	-
	BWFO	57,24 ± 3,87	3,45
	BWFO	57,24 ± 3,87	3,45
BCW	BBA	57,37 ± 3,92	3,55
	OPF	58,11 ± 4,18	-
	PCA	52,21 ± 3,24	-
	LDA	57,51 ± 4,38	-
	BWFO	94,18 ± 1,46	5,90
DM	BWFO	94,38 ± 1,64	5,80
	BBA	94,85 ± 1,43	5,50
	OPF	94,46 ± 1,58	-
	PCA	94,98 ± 1,33	-
	LDA	94,17 ± 1,67	-
	BWFO	95,82 ± 1,52	20,40
	BWFO	95,76 ± 1,56	19,20
	BBA	95,48 ± 2,24	19,40
	OPF	94,76 ± 1,25	-
	PCA	94,94 ± 0,93	-
	LDA	90,75 ± 6,84	-

A Tabela 5 apresenta os resultados dos métodos para os conjuntos de dados DIV, PDS, TKM, UBL e ARR, também presentes na Tabela 1. Os testes levam em consideração o tamanho de população igual a 100 para as meta-heurísticas. O BWFO tem um melhor desempenho nos conjuntos de dados com dimensões igual ou superior a 50, em que das cinco bases de dados, o BWFO foi superior a todos os outros métodos nas bases PDS, TKM, UBL e ARR, levando em consideração sua acurácia.

Tabela 5. Resultados dos métodos para as bases DIV, PDS, TKM, UBL e ARR.

Base	Métrica	BWFO	BWFO	BBA	OPF	LDA	PCA
DIV	Acc(%)	96,91	96,98	97,44	97,37	97,37	91,11
	Std	1,4714	1,46	1,50	1,68	1,68	9,56
	#NC	22,00	26,35	25,80	-	-	-
PDS	Acc(%)	76,19	75,13	74,73	74,55	74,07	68,62
	Std	3,0240	3,48	3,49	3,27	3,17	3,76
	#NC	336,60	374,80	284,90	-	-	-
TKM	Acc(%)	75,04	73,53	73,55	74,50	73,26	69,57
	Std	2,2540	2,40	2,44	2,27	2,18	3,71
	#NC	24,45	25,05	25,15	-	-	-
UBL	Acc(%)	83,83	83,67	83,40	82,69	82,54	78,76
	Std	1,1031	1,61	1,82	1,31	1,47	3,94
	#NC	62,60	69,55	51,30	-	-	-
ARR	Acc(%)	63,89	62,08	61,91	60,26	60,22	58,75
	Std	3,1662	4,03	3,64	3,95	4,03	3,84
	#NC	115,30	136,65	95,95	-	-	-

Os resultados discutidos demonstram que o BWFO é uma boa alternativa para a tarefa de seleção de características, visto que nos experimentos o método proposto teve um bom desempenho quando confrontado com o BWWO e o BBA. Por sua vez o BWWO consegue podar um pouco mais atributos em relação ao BWFO e BBA para conjuntos de dados com dimensão inferior a 50, entretanto o BBA supera o BWWO quando esse valor passa a ser maior ou igual a 50. Todavia, o método proposto em grande parte dos testes está próximo em relação ao BWWO e BBA quando se trata de reduzir a quantidade de atributos, superando até mesmo o BBA, em alguns dos testes realizados.

5. CONCLUSÃO

Neste trabalho foi proposto uma nova técnica de seleção de características baseada no algoritmo WFO, por sua vez denominado *Binary Water Flow Optimizer* (BWFO), que tem como ideia principal a remodelagem dos operadores originais do WFO para uma versão binária, a saber: operador de fluxo laminar e o operador fluxo turbulento, visando assim, construir novos operadores para a obtenção de uma versão binária do algoritmo. Além das alterações dos operadores, cada solução é representada por um vetor binário, mapeando cada característica de uma determinada base de dados, no qual os valores 1 e 0 em uma dada dimensão representam respectivamente, a presença e ausência de uma característica no conjunto de dados final. A fim de avaliar a qualidade das soluções, foi utilizada como função de aptidão a acurácia do classificador *Optimum-Path Forest* (OPF).

Os experimentos foram conduzidos por avaliar o BWFO comparado-o com técnicas clássicas de redução de dimensionalidade, como o PCA e LDA, e com os métodos evolutivos empregado também na tarefa de seleção de características: o *Binary Water Wave Optimization* e o *Binary Bat Algorithm*. Os métodos foram aplicados em onze bases de dados obtidas do repositório *UCI Machine Learning*. Todos os algoritmos foram investigados de acordo com a acurácia, desvio padrão e o número de características selecionados. Com base nos experimentos, pode-se afirmar que o método proposto é uma alternativa válida para problemas de seleção de características, visto ter sido capaz de superar na maior parte dos testes as técnicas clássicas de redução de dimensionalidade, PCA e o LDA. Além disso o BWFO também conseguiu superar em boas situações o BWWO e o BBA utilizando a acurácia como critério, e conseguindo em alguns experimentos igualar os resultados em relação a redução de atributos.

REFERÊNCIAS

Abbott, M.B. and Minns, A.W. (2017). *Computational Hydraulics (2nd ed.)*. Taylor and Francis. doi:10.4324/9781315259703.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 4960–4964. IEEE.

Codella, N., Cai, J., Abedini, M., Garnavi, R., Halpern, A., and Smith, J.R. (2015). Deep learning, sparse coding,

and svm for melanoma recognition in dermoscopy images. In *International Workshop on Machine Learning in Medical Imaging*, 118–126. Springer.

Foster, K.R., Koprowski, R., and Skufca, J.D. (2014). Machine learning, medical diagnosis, and biomedical engineering research-commentary. *Biomedical engineering online*, 13(1), 94.

Giusti, A., Guzzi, J., Ciresan, D.C., He, F.L., Rodríguez, J.P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Di Caro, G., et al. (2016). A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2), 661–667.

Luo, K. (2021). Water flow optimizer: A nature-inspired evolutionary algorithm for global optimization. *IEEE Transactions on Cybernetics*, 1–12. doi:10.1109/TCYB.2021.3049607.

Macêdo, F., Barbosa, G., and Neto, A.R. (2019). A binary water wave optimization algorithm applied to feature selection. In *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, 448–459. SBC, Porto Alegre, RS, Brasil. doi:10.5753/eniac.2019.9305. URL <https://sol.sbc.org.br/index.php/eniac/article/view/9305>.

Nakamura, R.Y.M., Pereira, L.A.M., Costa, K.A., Rodrigues, D., Papa, J.P., and Yang, X.S. (2012). Bba: A binary bat algorithm for feature selection. In *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, 291–297. doi:10.1109/SIBGRAPI.2012.47.

Papa, J., Pagnin, A., Schellini, S., Augusto Spadotto, A., Guido, R., Ponti, M., Chiachia, G., and Falcão, A. (2011). Feature selection through gravitational search algorithm. 2052–2055. doi:10.1109/ICASSP.2011.5946916.

Papa, J.P., Rosa, G.H., and Papa, L.P. (2017). A binary-constrained geometric semantic genetic programming for feature selection purposes. *Pattern Recognition Letters*, 100, 59 – 66. doi:<https://doi.org/10.1016/j.patrec.2017.10.002>. URL <http://www.sciencedirect.com/science/article/pii/S0167865517303641>.

Ray, S. and Analytics, B. (2015). Beginners guide to learn dimension reduction techniques. URL <https://www.analyticsvidhya.com/blog/2015/07/dimension-reduction-methods/>. Acesso em: 30 mar. 2019.

Rodrigues, D., Pereira, L.A.M., Almeida, T.N.S., Papa, J.P., Souza, A.N., Ramos, C.C.O., and Yang, X. (2013). Bcs: A binary cuckoo search algorithm for feature selection. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, 465–468. doi:10.1109/ISCAS.2013.6571881.

Rosa, G.H., Papa, J.P., and Papa, L.P. (2017). Feature selection using geometric semantic genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '17*, 253–254. ACM, New York, NY, USA. doi:10.1145/3067695.3076020. URL <http://doi.acm.org/10.1145/3067695.3076020>.