

# A Binary Water Wave Optimization Algorithm Applied to Feature Selection

Fagner J. de Matos Macêdo<sup>1</sup>, Gabriel Santos Barbosa<sup>1</sup>, Ajalmar R. da Rocha Neto<sup>1</sup>

<sup>1</sup>Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)  
Maracanaú – CE – Brazil

{fagnerjmatosifce, gabrielsantos.ifce, ajalmar}@gmail.com

**Abstract.** *In this work, the problem of feature selection is assessed by introducing a new binary version for the Water Wave Optimization (WWO) algorithm, called Binary Water Wave Optimization (BWWO). The WWO algorithm, in its original version, is used to solve only continuous optimization problems. The method proposed here combines the optimization characteristics of WWO with the speed of the optimum-path forest (OPF) classifier in order to provide a framework able to solve the feature selection task, which is a discrete problem, in an efficient way. In order to evaluate the performance obtained by BWWO, a comparative analysis against traditional methods of dimensionality reduction such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) is made. On the basis of our experiments, we can say that BWWO is a valid alternative for feature selection tasks.*

**Resumo.** *Neste trabalho, o problema de seleção de características é abordado através da introdução de uma nova versão binária para o algoritmo Water Wave Optimization (WWO), chamada Binary Water Wave Optimization (BWWO). O WWO, em sua versão original, é utilizado apenas para resolver problemas de otimização contínuos. O método aqui proposto combina as características de otimização presentes no WWO juntamente com a velocidade de treinamento do algoritmo Optimum-Path Forest (OPF) a fim de providenciar um framework capaz de resolver problemas de seleção de características, que são problemas discretos, de forma eficaz. Para avaliar o desempenho do BWWO, uma análise comparativa é feita com métodos clássicos de redução de dimensionalidade, mais especificamente com Principal Component Analysis (PCA) e Linear Discriminant Analysis (LDA). Com base nos experimentos, pode-se afirmar que BWWO é uma alternativa válida para problemas de seleção de características.*

## 1. Introdução

A redução de dimensionalidade se refere ao processo de conversão de um conjunto de dados de alta dimensão para um de menor dimensão garantindo que as informações mantidas sejam concisas. Estas técnicas são tipicamente usadas em problemas de aprendizagem de máquina a fim de obter os principais atributos para tarefas de classificação ou regressão [Van Der Maaten et al. 2009]. A redução de dimensionalidade pode ser feita de duas maneiras diferentes: via extração de características (*feature extraction*), onde a redução de dimensionalidade é realizada através da combinação das características originais, e via seleção de características (*feature selection*), que busca efetuar a redução de

dimensionalidade por remover características redundantes e irrelevantes do conjunto de dados [Verónica et al. 2015].

Dentre os métodos de extração de características, destacam-se o PCA e o LDA. O PCA é uma técnica de transformação linear não supervisionada para a redução de dimensionalidade. O PCA permite identificar padrões nos dados com base na correlação entre os atributos. Esse método funciona por encontrar as direções de maior variância em dados de alta dimensão e projetá-los em um novo subespaço com dimensões iguais ou menores que a original [Raschka 2015]. O conceito geral por trás do LDA é muito semelhante ao PCA; enquanto o PCA tenta encontrar os eixos ortogonais do componente de máxima variância em um conjunto de dados, o objetivo do LDA é encontrar o subespaço de recursos que otimiza a separabilidade das classes [Mika et al. 1999].

Com respeito aos estudos envolvendo seleção de características, alguns trabalhos abordaram essa técnica como um problema de otimização através do uso de metaheurísticas. Por exemplo, em [Rodrigues et al. 2014] os autores apresentaram um algoritmo de seleção de características baseado no *Bat Algorithm* (BA) juntamente com a acurácia do classificador *Optimum-Path Forest* (OPF) como função de aptidão. Nesse trabalho, o problema de seleção de características foi modelado como uma técnica de otimização binária. A busca pela solução ótima é guiada pelo BA enquanto o conjunto de dados reduzido é validado usando a acurácia do classificador OPF. Em [Prudhvi et al. 2018], os autores combinaram o método *Hybrid Particle Swarm Optimization* com o *Gravitational Search Algorithm* a fim de gerar um algoritmo de otimização binária chamado *Binary Hybrid Particle Swarm Optimization-Gravitational Search Algorithm* (BPSOGSA). Esse algoritmo ajuda a escolher os melhores atributos com base na aptidão obtida pela rede neural *Radial Basis Network* (RBF). A função de aptidão por sua vez foi projetada de modo a garantir um número mínimo de recursos com valores de sensibilidade e especificidade balanceados. Já em [Rodrigues et al. 2013], um novo método para seleção de características baseado no algoritmo *Cuckoo Search* (CS) foi proposto. Tal método foi chamado *Binary Cuckoo Search* (BCS), o qual é baseado no comportamento dos pássaros cuco. Os experimentos foram realizados no contexto de detecção de roubo em sistemas de distribuição de energia em dois conjuntos de dados obtidos de uma empresa brasileira de energia elétrica.

Recentemente, uma nova metaheurística conhecida como *Water Wave Optimization* (WWO) foi proposta tendo como inspiração modelos de ondas de águas rasas [Zheng 2015]. O algoritmo WWO é composto de três operadores capazes de balancear as capacidades de busca local e global; são eles: propagação, refração e quebra de onda. No entanto, o algoritmo WWO em sua versão original é capaz de resolver apenas problemas contínuos. Para contornar essa limitação, alguns autores modificaram o WWO para que também pudesse resolver problemas discretos [Wu et al. 2015], [Zheng 2015], [Manickam et al. 2016], [Zhao et al. 2018]. Embora esses métodos tenham sido empregados para resolver problemas discretos, ainda existe espaço para que novas versões do algoritmo WWO sejam propostas, em especial para abordar aplicações que podem ser codificadas de forma binária.

Nesse contexto, este trabalho propõe uma versão binária do algoritmo WWO, chamada *Binary Water Wave Optimization* (BWWO), aplicada em seleção de características. Para tal finalidade, será preciso que cada onda (solução do problema) seja representada

por um conjunto de coordenadas binárias que denotem se uma característica pertencerá ou não ao conjunto final de características. A função objetivo a ser maximizada será fornecida pela acurácia de um classificador. O classificador utilizado para auxiliar esse processo é o OPF [Papa et al. 2009a], visto que o mesmo tem demonstrado ser tão eficaz quanto máquina de vetores suporte (SVM, do inglês: *Support Vector Machine*) e possui uma rápida fase de treinamento [Papa et al. 2012]. Além disso, o método será aplicado em bases de dados do mundo real e comparado com OPF (sem redução de atributos), PCA e LDA.

O restante deste trabalho está organizado da seguinte forma: na Seção 2, os fundamentos do algoritmo WWO são apresentados. Na Seção 3, o método proposto neste trabalho é explanado. Então, na Seção 4 os resultados obtidos são apresentados e discutidos. Finalmente, o artigo é concluído na Seção 5.

## 2. Water Wave Optimization

O algoritmo *Water Wave Optimization* (WWO) [Zheng 2015] é um método evolutivo inspirado na teoria de ondas de águas rasas utilizado para resolver problemas de otimização contínuos. Nesse algoritmo, o espaço de solução é análogo à área do fundo do mar e a aptidão de um ponto no espaço é medida em relação à profundidade do mar. Em uma população, cada onda possui dois atributos: altura de onda  $h \in \mathbb{Z}^+$  e comprimento de onda  $\lambda \in \mathbb{R}^+$ . De acordo com a teoria de ondas de águas rasas, quanto menor for a distância entre o fundo do mar e a onda, menor será o valor  $\lambda$  e maiores serão os valores de aptidão e  $h$ . Na inicialização, para cada onda,  $h$  assume o valor da constante  $h_{max}$  e  $\lambda$  é definido para o valor 0.5.

Durante o processo de solução do problema, as operações de propagação, refração e quebra são aplicadas para evoluir a população.

### 2.1. Operador de Propagação

O processo de propagação pode ser considerado como um processo de movimento (mudanças) de águas profundas para águas rasas. O operador de propagação cria uma nova onda  $x'$  pelo deslocamento de cada dimensão  $d$  da onda original  $x$  como

$$x'(d) = x(d) + rand(-1, 1) \cdot \lambda L(d), \quad (1)$$

em que  $rand(-1, 1)$  é um número aleatório uniformemente distribuído no intervalo  $[-1, 1]$ , e  $L(d)$  é o tamanho da  $d$ -ésima dimensão do espaço de busca. Se a nova posição estiver fora do intervalo factível, ela será redefinida para uma posição aleatória no intervalo. Após cada iteração, o comprimento de cada onda é atualizado conforme

$$\lambda = \lambda \cdot a^{-(f(x) - f_{min} + \epsilon) / (f_{max} - f_{min} + \epsilon)}, \quad (2)$$

em que  $f_{max}$  e  $f_{min}$  são, respectivamente, os valores de aptidão máxima e mínima na população atual,  $a$  é o coeficiente de redução de comprimento de onda, e  $\epsilon$  é um pequeno valor positivo que evita divisões por zero.

## 2.2. Operador de Refração

No operador de refração, se uma onda após várias iterações não foi melhorada, e sua altura após sucessivos decrementos chegou a zero, a operação de refração é ativada para atualizar essa onda. Sua fórmula de cálculo é dada por

$$x'(d) = N\left(\frac{x^*(d) + x(d)}{2}, \frac{|x^*(d) - x(d)|}{2}\right), \quad (3)$$

em que  $N(\mu, \sigma)$  denota a distribuição normal com média  $\mu$  e desvio padrão  $\sigma$  e  $x^*$  é a melhor solução atual. Este mecanismo faz com que  $x$  aprenda com a onda ótima atual  $x^*$ . Após a refração, a altura ( $h$ ) da onda  $x$  é redefinida para  $h_{max}$ , e seu comprimento de onda ( $\lambda$ ) é redefinido por

$$\lambda' = \lambda_i \frac{f(x)}{f(x')}. \quad (4)$$

## 2.3. Operador de Quebra

O operador de quebra é aplicado apenas em ondas que encontram uma melhor solução. A operação específica é escolher  $k$  dimensões aleatoriamente ( $k$  é um número aleatório entre 1 e  $k_{max}$ ) e redefini-las em  $x'$  como na Equação 5, sendo  $\beta$  o coeficiente de quebra.

$$x'(d) = x(d) + N(0, 1) \cdot \beta L(d). \quad (5)$$

## 2.4. Visão geral do WWO

Segundo [Zheng 2015], testes empíricos mostram que o WWO apresenta um bom desempenho com um pequeno tamanho de população  $n$  (entre 5 e 10), o que torna o algoritmo mais eficiente computacionalmente. Além do tamanho da população, existem quatro parâmetros de controle no WWO: altura máxima de onda  $h_{max}$ , coeficiente de redução de comprimento de onda  $\alpha$ , coeficiente de quebra  $\beta$  e o número máximo de quebra da onda  $k_{max}$ . Em geral, quanto maior o valor de  $h_{max}$ , maior será o tempo de vida da onda; caso contrário, se o valor de  $h_{max}$  for pequeno (*e.g.* 1 ou 2), as ondas serão frequentemente substituídas por novas ondas, aumentando, portanto, a diversidade de soluções. Grandes valores para  $\alpha$  fazem com que o algoritmo explore áreas maiores, enquanto que valores pequenos fazem com que o algoritmo realize uma exploração mais intensiva. Além disso, para  $\beta$  com pequenos valores e com um grande  $k_{max}$ , o algoritmo busca melhores soluções, o que resulta em uma maior velocidade de convergência. Empiricamente é recomendado que  $h_{max}$  tenha valor 5 ou 6,  $\alpha$  possua valores entre 1.001 e 1.01, e  $k_{max}$  seja definido como o menor valor em  $\{12, D/2\}$ , em que  $D$  denota a dimensão do problema.

## 2.5. Algoritmo WWO

O algoritmo WWO é descrito a seguir.

---

**Algoritmo 1** Algoritmo WWO

---

```
1: Inicialize aleatoriamente uma população P de  $n$  soluções (ondas);
2: enquanto o critério de parada não é satisfeito fazer
3:   para cada onda  $x$  de P fazer
4:     Propaga  $x$  para um novo  $x'$  conforme a Equação 1;
5:     se  $f(x') > f(x)$  então
6:       se  $f(x') > f(x^*)$  então
7:         Quebra  $x$  para um novo  $x'$  conforme a Equação 5;
8:         Atualiza  $x^*$  com  $x'$ ;
9:         Atualiza  $x$  com  $x'$ ;
10:    senão
11:       $x.h \leftarrow x.h - 1$ ;
12:      se  $x.h = 0$  então
13:        Refrata  $x$  para um novo  $x'$  conforme a Equação 3 e 4;
14:        Redefine  $x.h$  para  $h_{max}$ ;
15:    Atualiza os comprimentos de onda conforme a Equação 2;
16: return  $x^*$ 
```

---

### 3. Proposta: BWWO

Neste trabalho, é proposta uma nova versão do algoritmo WWO capaz de resolver problemas binários. O método aqui proposto, chamado *Binary Water Wave Optimization* (BWWO), é aplicado na tarefa de seleção de características em classificação de padrões. Para isso, os operadores do WWO serão remodelados no BWWO. Além disso, é preciso estabelecer uma função de aptidão para guiar o BWWO em busca da melhor solução. Tal função de aptidão é baseada na acurácia de um classificador sobre um conjunto de validação. O classificador utilizado neste trabalho é o OPF [Papa et al. 2009b], por possuir uma rápida fase de treinamento. No entanto, outros classificadores podem ser utilizados.

#### 3.1. Representação do problema

A solução para o problema é uma onda codificada em um vetor binário  $x = [d_1 d_2 \dots d_i \dots d_{N-1} d_N]$ , em que  $d_i \in \{0, 1\}$  é a  $d$ -ésima dimensão da onda e  $N$  é o número de atributos de uma determinada base de dados. O valor 1 e 0 representam, respectivamente, a presença e ausência de uma característica na construção do novo conjunto de dados reduzido. Por exemplo, considere uma onda  $x = [1, 1, 0, 1, 0]$  referente a uma base de dados contendo cinco atributos. Nesse caso, apenas a primeira, a segunda e a quarta característica serão selecionadas para compor o novo conjunto de dados. A seguir é mostrado como os operadores foram remodelados.

#### 3.2. Operador de Propagação Binário

No novo operador de propagação o comprimento de onda  $\lambda$  é utilizado como parâmetro para adicionar ou remover características na solução. Por exemplo, ondas com altos valores  $\lambda$  são consideradas ondas de baixa aptidão, e tendem a remover mais características visando melhorar a solução; ondas com baixos valores  $\lambda$  são consideradas boas soluções e tendem a remover menos características do conjunto de dados avaliado.

A Equação 6 mostra como é realizada a atualização do comprimento de onda de cada solução  $x$  na população. Nessa equação, o coeficiente de redução de comprimento da onda  $\alpha$  assume o valor 0.01, visando melhor distribuir o valor  $\lambda$  para as soluções.

$$\lambda' = 1 - \alpha^{1-f(x)/f(x^*)}. \quad (6)$$

A aplicação da nova abordagem para o operador de propagação é empregada sobre  $k = \{k \in \mathbb{Z}^+ | 1 \leq k \leq k_{max}\}$  dimensões escolhidas aleatoriamente em um intervalo uniforme, em que  $k_{max}$  é definido como  $\min(12, D/2)$ . Assim, cada dimensão selecionada gera um número randômico real  $r \in [0, 1]$  que logo após é comparado com  $\lambda$ . Portanto, se  $r > \lambda$ , a dimensão assume o valor 1, selecionando a característica referente a dimensão, caso contrário a dimensão recebe o valor 0 e não seleciona a característica correspondente.

### 3.3. Operador de Refração Binário

Como no operador de refração original do WWO, se uma onda após várias iterações não obteve melhorias e se sua altura chegou a zero, o operador de refração é ativado. Entretanto,  $k$  dimensões são selecionadas de forma aleatória, conforme a Seção 3.2, e cada dimensão selecionada da onda assume o valor da respectiva dimensão da melhor solução  $x^*$ . Após a refração, a altura  $h$  da onda é redefinida para  $h_{max}$ . O objetivo é que ondas de baixo valor de solução absorvam características da melhor solução atual.

### 3.4. Operador de Quebra Binário

Ondas que foram aprimoradas após a propagação e obtiveram melhor aptidão em relação a onda ótima atual  $x^*$ , passam pelo operador de quebra. Assim como no operador original do WWO,  $k$  dimensões são selecionadas aleatoriamente. No entanto, no novo operador, para cada dimensão de um onda  $x'$  propagada a partir de  $x$  (onda original), o seu valor é invertido, ou seja, para dimensão de valor 1 o seu valor passa a ser 0 e vice-versa. Caso o valor de aptidão de  $x'$  seja maior que o valor de  $x$ , a onda  $x$  é substituída por  $x'$ . O mesmo acontece para  $x^*$  caso  $x'$  a supere. O operador de quebra faz com que o BWWO conduza uma busca local ao redor da onda aprimorada, que por sua vez torna-se a melhor onda na geração atual.

### 3.5. Solução do Problema

Para comprovar a eficiência da abordagem proposta, o método será aplicado no problema de seleção de características. Para isso, o algoritmo deve iniciar com uma população formada por  $n$  ondas geradas aleatoriamente, de forma que a partir de cada geração o método consiga aprimorar as soluções de acordo com os seus valores de aptidão. No final de sua execução, o BWWO deve retornar a onda que selecionar as características mais relevantes de um determinado conjunto de dados sem afetar, ou até mesmo melhorar, a acurácia do classificador em questão.

Em relação ao algoritmo original, o método aqui proposto utiliza um número menor de parâmetros; os parâmetros escolhidos empiricamente para o novo método são: altura máxima da onda  $h_{max}$ , que assume o valor 3; número máximo de direções de quebra  $k_{max}$ , no qual seu valor é dado por  $\min(12, D/2)$ , em que  $D$  é o tamanho da dimensão do problema; e o coeficiente de redução de comprimento da onda  $\alpha$ , que assume o valor

0.01. O valor escolhido para  $h_{max}$  faz com que as ondas sejam frequentemente modificadas por novas ondas no operador de refração, dando assim uma maior diversidade nas soluções. Valores altos para  $k_{max}$  ajudam a melhorar os resultados e aumentar a velocidade de convergência e o valor assumido por  $\alpha$  ajuda a distribuir melhor o comprimento de onda das soluções em cada geração. O Algoritmo 2 mostra a ideia geral do método BWWO.

---

**Algoritmo 2** Algoritmo BWWO

---

```

1: Inicialize aleatoriamente uma população P de  $n$  ondas (soluções);
2: enquanto o critério de parada não é satisfeito fazer
3:   para cada onda  $x$  de  $P$  fazer
4:      $x' \leftarrow x$ ;
5:     Selecione  $k$  dimensões aleatórias de  $x$ ;
6:     para cada dimensão  $k$  fazer
7:       Gerar  $r$  distribuído uniformemente entre  $[0, 1]$ ;
8:        $x'[d] \leftarrow$  se  $r > x.\lambda$  então 1 senão 0;
9:     se  $f(x') > f(x)$  então
10:      Atualiza  $x$  com  $x'$ ;
11:     se  $f(x') > f(x^*)$  então
12:      Atualiza  $x^*$  com  $x'$ ;
13:     Selecione  $k$  dimensões aleatórias;
14:     para cada dimensão  $k$  fazer
15:        $x'[k] \leftarrow$  se  $x'[k] == 1$  então 0 senão 1;
16:     se  $f(x') > f(x)$  então
17:      Atualiza  $x$  com  $x'$ ;
18:     se  $f(x') > f(x^*)$  então
19:      Atualiza  $x^*$  com  $x'$ ;
20:   senão
21:      $x.h \leftarrow x.h - 1$ ;
22:   se  $x.h == 0$  então
23:     Selecione  $k$  dimensões aleatórias de  $x$ ;
24:     para cada dimensão  $k$  fazer
25:        $x[k] \leftarrow x^*[k]$ ;
26:     Redefine  $x.h$  para  $h_{max}$ ;
27:   Atualizar os comprimentos de onda conforme a Equação 6;
28: return  $x^*$ 

```

---

Na linha 1 é iniciada aleatoriamente uma população de  $n$  ondas (soluções) seguindo a codificação mostrada na Seção 3.1. Nas linhas 4-8, é realizada a propagação binária, assim, a partir de uma onda  $x$  é gerada uma nova onda  $x'$ , em que  $k$  dimensões aleatórias de  $x'$  são modificadas a partir do comprimento de onda  $\lambda$ , o qual é utilizado como uma probabilidade para selecionar ou remover características na solução. A partir das linhas 12-19 é realizada a operação de quebra binária, responsável por fazer uma busca intensiva em  $x'$  por inverter o valor de  $k$  dimensões selecionadas aleatoriamente. Nas linhas 23-25 o operador de refração é encarregado de alterar ondas de baixa aptidão, quando a altura  $h$  de tais ondas sofrem sucessivos decrementos até chegar a zero. Assim, o operador de refração seleciona  $k$  dimensões aleatórias para serem substituídas pelas respectivas dimensões da melhor solução atual  $x^*$ , fazendo com que tais ondas absorvam

características da melhor onda atual e melhorem sua aptidão. Na linha 27 o valor  $\lambda$  é atualizado com base na Equação 6. Por fim, quando o critério de parada do BWWO é satisfeito a linha 28 retorna a melhor solução encontrada.

#### 4. Resultados e Discussões

Para avaliar a eficiência do algoritmo BWWO na tarefa de seleção de características, foram utilizadas cinco bases de dados do mundo real, a saber: *Australian Credit Approval* (ACA), *Breast Cancer Wisconsin* (BCW), *Dermatology* (DM), *Sonar (Mine vs Rocks)* (SNR) e *Pima Indians Diabetes* (PID). A Tabela 1 apresenta as principais características dessas bases de dados. Além disso, o algoritmo BWWO proposto é comparado com o OPF tradicional, com o OPF após a aplicação do PCA e do LDA. Os experimentos foram realizados em um computador com um processador Intel Core i3<sup>®</sup> 2.00GHz x 4, com 8GB de memória RAM, sistema operacional Linux Ubuntu 16.04 LTS e tendo o *software* MATLAB<sup>®</sup> (2017a) como ambiente de desenvolvimento dos algoritmos.

**Tabela 1. Descrição dos conjuntos de dados utilizados neste trabalho.**

Conjunto de dados	# Amostras	# Características
ACA	690	14
BCW	699	10
DM	366	33
SNR	208	60
PID	768	8

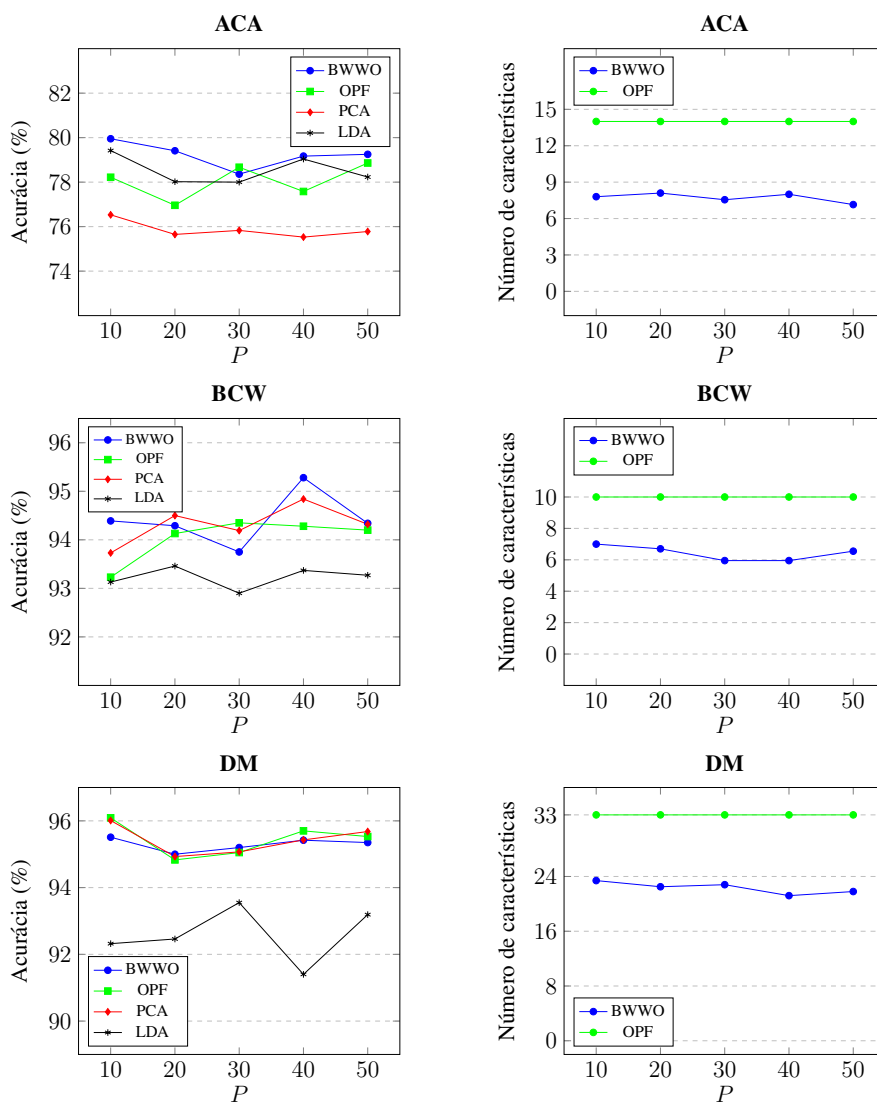
Inicialmente, os conjuntos de dados foram normalizados, embaralhados e particionados em três subconjuntos: conjunto de treinamento ( $Z_1$ ), validação ( $Z_2$ ) e teste ( $Z_3$ ); onde, 30% dos dados foram destinados ao conjunto de treinamento, 30% para o conjunto de validação e os outros 40% restantes para o conjunto de teste. Durante a execução do BWWO, novos subconjuntos  $Z'_1$ ,  $Z'_2$  e  $Z'_3$  são gerados de acordo com os valores binários presentes em cada onda gerada aleatoriamente, em que o valor 1 e 0 representam, respectivamente, a presença e ausência de um atributo nos novos subconjuntos. Então, o algoritmo OPF é treinado em  $Z'_1$  e classificado em  $Z'_2$  para avaliar a aptidão de cada onda. Esse processo é repetido durante um determinado número de iterações (gerações) ou quando 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 tendo convergido caso não mude durante cinco gerações. Após os critérios de parada serem satisfeitos, o BWWO 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  $Z'_3$ .

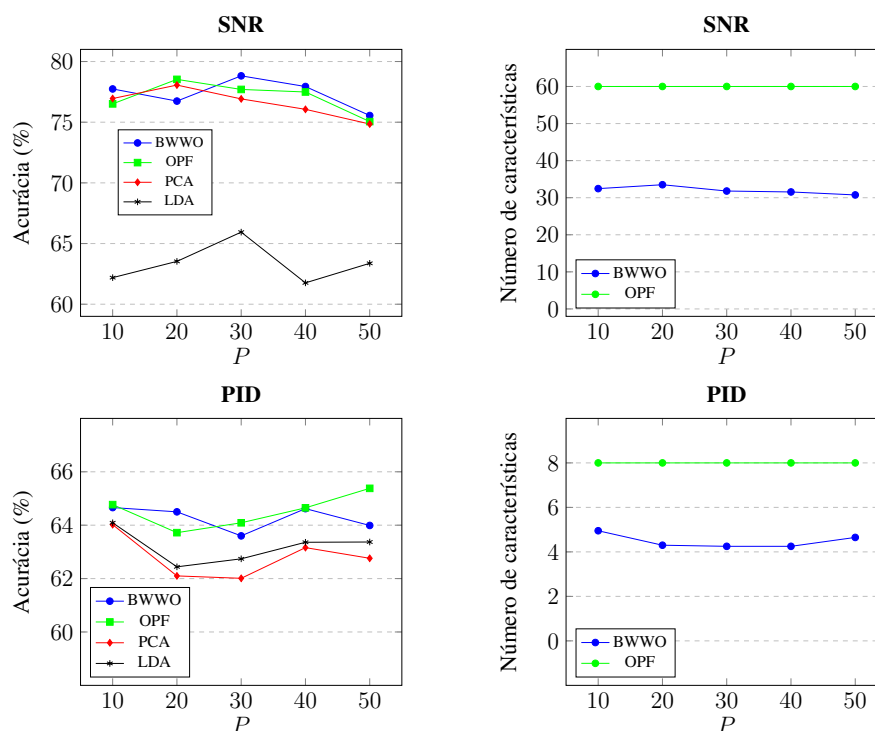
A fim de realizar uma comparação justa entre os métodos estudados neste trabalho, os experimentos foram conduzidos utilizando os mesmos padrões de dados para cada abordagem. Ou seja, cada método recebe como parâmetro exatamente os mesmos subconjuntos  $Z_1$ ,  $Z_2$  e  $Z_3$  utilizados no BWWO. Além disso, o PCA e o LDA possuem como quantidade de componentes e discriminantes o número de características retornado pelo BWWO. Visto que os outros métodos comparados com o BWWO 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 (dimensões) igual ao número de características retornadas pelo BWWO. A seguir, para cada método, o classificador OPF é treinado em  $Z'_1$  e classificado em  $Z'_3$ . Então, o valor de acurácia alcançado é comparado com àquele obtido pelo BWWO. Um procedimento similar acontece para obter o valor de acurácia para o classificador OPF tradicional. No entanto, utilizam-se os conjuntos  $Z_1$  e  $Z_3$  sem redução de dimensionalidade. Todo esse processo é repetido 20 vezes para computar o valor de acurácia média e desvio padrão, bem como o número médio de atributos retornados por cada método de redução de dimensionalidade.

A Figura 1 apresenta os resultados obtidos pelos métodos empregados neste artigo, variando o tamanho da população  $P$  do algoritmo BWWO em 10, 20, 30, 40 e 50 ondas. Esses resultados são a acurácia média e o número de características médio medidos durante 20 execuções independentes para cada tamanho de população  $P$  e para cada base de dados presente na Tabela 1. Note que no gráfico mostrando os valores do número de características, apenas os valores do BWWO e do OPF foram apresentados. Isso se dá porque o número de componentes e discriminantes utilizados no PCA e LDA, respectivamente, é igual ao número de características selecionado pelo BWWO.





**Figura 1. Comparação do OPF, PCA e LDA versus BWWO com tamanho de população P variando em 10, 20, 30, 40 e 50 ondas.**

É possível notar a partir da Figura 1 que o BWWO obteve bons resultados em termos de acurácia e redução de características. Outro fator interessante é que, assim como no algoritmo WWO original, o método aqui proposto é capaz de obter bons resultados mesmo com um tamanho pequeno de população; isso faz com que o BWWO alcance os resultados desejados de maneira eficiente.

A Tabela 2 apresenta mais detalhes dos resultados obtidos pelo OPF, PCA e LDA quando comparados com o BWWO tendo tamanho de população variando em 10, 20, 30, 40 e 50 ondas, e com um número de gerações constante definido como 20. Nessa tabela, os resultados são denotados como  $x[y]z$ , em que  $x$  representa os valores de acurácia,  $y$  o número de características selecionado e  $z$  o desvio padrão dos valores de acurácia. O número de características não mostrados para o OPF, PCA e LDA dá-se porque não houve redução de dimensionalidade para os experimentos com o OPF, e o PCA e LDA tiveram o mesmo número de características selecionados que o BWWO, indicado pelo símbolo “\*”.

Pode-se observar, na Tabela 2, que o BWWO alcançou bons resultados em relação às demais técnicas de redução de dimensionalidade (PCA e LDA), superando o LDA em todos experimentos. Já o PCA se sobressaiu em 4 dos 25 experimentos realizados, enquanto que em 14 experimentos o BWWO superou as demais técnicas. Além disso, na maioria dos casos o BWWO alcançou os maiores níveis de acurácia entre todos os métodos, com exceção apenas das bases DM e PID. De acordo com os experimentos, o BWWO pode ser visto como uma técnica bastante interessante e efetiva, contribuindo na difusão do uso de técnicas evolutivas no contexto de seleção de características.

**Tabela 2. Resultados do OPF, PCA e LDA versus BWWO com tamanho de população P variando em 10, 20, 30, 40 e 50 ondas.**

#P	Base	BWWO	OPF	PCA	LDA
10	ACA	<b>79.9%</b> [7.8]2.8	78.2%[-]2.9	76.5%[*]2.6	79.4%[*]3.1
	BCW	<b>94.3%</b> [7]1.6	93.2%[-]2.5	93.7%[*]1.9	93.1%[*]1.5
	DM	95.5%[23.4]1.0	<b>96.0%</b> [-]1.1	96.0%[*]1.4	92.3%[*]2.3
	SNR	<b>77.7%</b> [32.4]4.4	76.5%[-]5.8	76.9%[*]5.8	62.1%[*]6.2
	PID	64.6%[4.9]2.6	<b>64.7%</b> [-]2.8	64.0%[*]3.1	64.0%[*]3.0
20	ACA	<b>79.4%</b> [8.1]2.4	76.9%[-]3.1	75.6%[*]3.0	78.0%[*]3.1
	BCW	94.2%[6.7]1.4	94.1%[-]1.4	<b>94.5%</b> [*]1.6	93.4%[*]1.7
	DM	<b>95.1%</b> [22.5]1.8	94.8%[-]1.1	94.9%[*]1.2	92.4%[*]1.6
	SNR	76.7%[33.5]4.2	<b>78.5%</b> [-]4.5	78.0%[*]4.3	63.5%[*]6.7
	PID	<b>64.5%</b> [4.3]1.7	63.7%[-]3.1	62.1%[*]3.2	62.4%[*]3.0
30	ACA	78.3%[7.5]4.3	<b>78.6%</b> [-]2.3	75.8%[*]3.2	78.0%[*]2.8
	BCW	93.7%[5.9]1.8	<b>94.3%</b> [-]1.7	94.1%[*]1.6	92.9%[*]2.0
	DM	<b>95.2%</b> [22.8]1.3	95.0%[-]1.4	95.0%[*]1.2	93.5%[*]2.1
	SNR	<b>78.8%</b> [31.8]3.9	77.7%[-]3.1	76.9%[*]4.2	65.9%[*]6.4
	PID	63.6%[4.2]2.6	<b>64.0%</b> [-]3.3	62.0%[*]3.6	62.7%[*]3.0
40	ACA	<b>79.1%</b> [8]3.3	77.5%[-]2.8	75.5%[*]2.3	79.0%[*]2.4
	BCW	94.2%[5.9]1.8	94.2%[-]1.6	<b>94.8%</b> [*]1.8	93.3%[*]1.7
	DM	95.4%[21.2]1.2	<b>95.7%</b> [-]1.2	95.4%[*]1.4	91.4%[*]7.0
	SNR	<b>77.9%</b> [31.5]5.8	77.4%[-]5.7	76.0%[*]5.6	61.7%[*]7.0
	PID	<b>64.6%</b> [4.2]2.5	<b>64.6%</b> [-]2.4	63.1%[*]2.5	63.3%[*]3.5
50	ACA	<b>79.2%</b> [7.1]6.7	78.8%[-]2.4	75.7%[*]2.8	78.2%[*]3.1
	BCW	<b>94.3%</b> [6.5]1.7	94.2%[-]1.5	<b>94.3%</b> [*]1.2	93.2%[*]1.5
	DM	95.3%[21.8]1.6	95.5%[-]1.4	<b>95.6%</b> [*]1.5	93.1%[*]2.7
	SNR	<b>75.5%</b> [30.7]5.0	75.0%[-]5.1	74.8%[*]4.3	63.3%[*]6.9
	PID	63.9%[4.6]3.0	<b>65.3%</b> [-]2.2	62.7%[*]1.3	63.3%[*]2.9

## 5. Conclusão

Neste trabalho foi proposta uma nova técnica de seleção de características baseada no algoritmo WWO. A técnica é chamada *Binary Water Wave Optimization* (BWWO) e tem como ideia principal a remodelagem dos operadores originais do WWO para uma versão binária. Nesse sentido, cada solução é representada por um vetor (onda) binário, 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. Para avaliar a qualidade de uma solução, foi utilizada como função de aptidão a acurácia de um classificador, mais especificamente o classificador OPF.

O BWWO foi comparado com técnicas clássicas de redução de dimensionalidade, como PCA e LDA, bem como com o classificador OPF, por aplicá-los em cinco bases de dados do mundo real. Todos os algoritmos foram investigados de acordo com dois critérios: acurácia e número de características selecionadas. Com base nos experimentos, pode-se afirmar que o BWWO é uma alternativa válida para problemas de seleção de características, pois foi capaz de superar os outros métodos em termos de acurácia, mesmo com um número de características equivalente ao PCA e LDA. Para trabalhos futuros, pretende-se avaliar mais profundamente a influência dos parâmetros do BWWO, bem

como compará-lo com outras metaheurísticas utilizadas em problemas binários.

## Referências

- Manickam, S., Balamurugan, R., and Lakshminarasimman, L. (2016). Water wave optimization algorithm for solving combined economic and emission dispatch problem. 12.
- Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, K.-R. (1999). Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, pages 41–48. Ieee.
- Papa, J., Falcão, A., and Suzuki, C. (2009a). Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19:120 – 131.
- Papa, J. P., Falcao, A. X., and Suzuki, C. T. (2009b). Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19(2):120–131.
- Papa, J. P., Falcão, A. X., de Albuquerque, V. H. C., and Tavares, J. M. R. (2012). Efficient supervised optimum-path forest classification for large datasets. *Pattern Recognition*, 45(1):512 – 520.
- Prudhvi, R. B., Sai, K. V., Abhishek, Y., and Venkatanaresbhabu, K. (2018). Feature selection using binary psogsa and radial basis network with a novel fitness function. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6.
- Raschka, S. (2015). *Python Machine Learning*. Packt Publishing.
- Rodrigues, D., Pereira, L. A., Nakamura, R. Y., Costa, K. A., Yang, X.-S., Souza, A. N., and Papa, J. P. (2014). A wrapper approach for feature selection based on bat algorithm and optimum-path forest. *Expert Systems with Applications*, 41(5):2250 – 2258.
- 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)*, pages 465–468.
- Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13.
- Verónica, B.-C., Noelia, S.-M., and Amparo, A.-B. (2015). *Feature Selection for High-Dimensional Data*.
- Wu, X.-B., Liao, J., and Wang, Z.-C. (2015). Water wave optimization for the traveling salesman problem. 9225:137–146.
- Zhao, F., Liu, H., Zhang, Y., Ma, W., and Zhang, C. (2018). A discrete water wave optimization algorithm for no-wait flow shop scheduling problem. *Expert Systems with Applications*, 91:347 – 363.
- Zheng, Y.-J. (2015). Water wave optimization: A new nature-inspired metaheuristic. *Computers & Operations Research*, 55:1 – 11.