BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **23**, No 4
Sofia • 2023 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2023-0038

A Competitive Parkinson-Based Binary Volleyball Premier League Metaheuristic Algorithm for Feature Selection

Edjola Naka

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev Str. bl.2, 1113 Sofia, Bulgaria
E-mail: edjola.naka@iict.bas.bg

Abstract: *A novel proposed Binary Volleyball Premier League algorithm* (*BVPL*) *has shown some promising results in a Parkinson's Disease* (*PD*) *dataset related to fitness and accuracy* [1]. *This paper evaluates and provides an overview of the efficiency of BVPL in feature selection compared to various metaheuristic optimization algorithms and PD datasets. Moreover, an improved variant of BVPL is proposed that integrates the opposite-based solution to enlarge search domains and increase the possibility of getting rid of the local optima. The performance of BVPL is validated using the accuracy of the k-Nearest Neighbor Algorithm. The superiority of BVPL over the competing algorithms for each dataset is measured using statistical tests. The conclusive results indicate that the BVPL exhibits significant competitiveness compared to most metaheuristic algorithms, thereby establishing its potential for accurate prediction of PD. Overall, BVPL shows high potential to be employed in feature selection.*

Keywords: *Binary Volleyball Premier League, Feature Selection* (*FS*), *k-Nearest Neighbor* (*kNN*), *Parkinson's Disease* (*PD*), *MetaHeuristic optimization Algorithms* (*MHA*).

1. Introduction

Currently, there is a significant amount of research being conducted on the issue of feature selection. Various techniques, including filter, wrapper, and embedding methods, have been utilized and suggested over time to address the issue of feature selection. The selection of features in filter methods is independent of the choice of a machine learning classifier, whereas wrapper methods rely on the performance of the classifier algorithm when evaluating different subsets of features. Lastly, embedded methods incorporate feature selection as an integral part of the classifier algorithm. Details of them have been discussed in previous review papers [2-4]. Nevertheless, it is worth noting that there exists a distinct category known as metaheuristics, which has gained significant popularity in the field of Feature Selection (FS) over the past few decades [5-8]. This preference can be explained by the good things about metaheuristics, like their ability to work without gradients, their adaptability, their simplicity, and the fact that they don't depend on the specific problem [9]. In recent decades, there has been a boom of novel Metaheuristic Optimization Algorithms (MHAs) as well as enhancements to existing ones, along with an increasing number of hybrid methods. Yet, a significant portion of contemporary algorithms developed by the younger generation exhibits a lack of originality and bear a resemblance to pre-existing algorithms, such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Differential Algorithm (DE), Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC) [10]. Accuracy, stability, scalability, and computing cost are the main difficulties that researchers encounter when using metaheuristics [5, 8].

The process of FS, which aims to identify and retain only the most useful features while discarding noisy, non-informative, irrelevant, and redundant information, can improve machine-learning models. Metaheuristics typically produce continuous values, making them unsuitable to be applied directly for Feature Selection (FS). In their study, C r a w f o r d et al. [11] have looked at a lot of different ways to turn continuous MHAs into discrete or binary ones. One often employed methodology is known as the two-step binarization technique, which involves the utilization of Transfer Functions (TF) to transform continuous values within the range of 0 to 1, and then methods as standard, or complement converts them in binary values, 0 or 1. In FS 1 means that the feature is selected; otherwise, it is not selected. These methodologies facilitate the utilization of continuous metaheuristics without necessitating any modifications to the operators. In addition to the S-shaped and V-shaped TFs [12], there are other types such as the X-shaped [13], U-shaped [14], and linear [15] TFs. The utilization of quadratic functions has also been suggested [16]. Nevertheless, it cannot be assured that just one metaheuristic will be able to identify the optimal subset of features from various domains. Given the aforementioned concerns, it is plausible that the utilization of novel metaheuristics in feature selection could yield improved outcomes. Several binary metaheuristics have been developed recently, employing various approaches to TFs. These include moth-flame optimization [17], quadratic Harris hawk optimization [16], seagull optimizer [18], gradient-based optimizer [19], ant lion [20], Artificial Algae Algorithm [21], atom search optimization [22], bamboo forest growth optimization [23], Giza pyramids construction [24], golden eagle optimizer [25], gaining-sharing knowledge-based optimization [26], and Manta ray foraging optimization [27], horse herd optimization [28], among others. The Volleyball Premier League (VPL) Algorithm is an MHA that draws inspiration from the competitive nature of volleyball teams participating in a league over the course of a season [29]. The first proposal of the binary VPL Algorithm has been documented in [1]. In order to assess its robustness, the algorithm will now be tested on a large number of datasets and in conjunction with other MHAs. An Opposition-Based BVPL (OBL_BVPL) Algorithm is proposed also to improve the exploration of the optimum. The primary aim of this research is to strengthen the utilization of metaheuristics in the domain of feature selection, specifically focusing on the identification of the most significant features. The main focus of this investigation is on datasets related to Parkinson's Disease (PD), aiming to develop a model that can improve the accuracy of predicting people affected by PD.

The article is organized into five sections. Following the introductory part Section 2, the next section provides a concise overview of the BVPL, opposition-based learning, and k-Nearest Neighbor Algorithm. Section 3 outlines the experimental conditions covering the methodology, the datasets, the binary metaheuristics employed for comparative analysis, and the assessment measures. Section 4 summarizes the results of the metrics for all the MHAs and the eight TFs. The final results are completed along with convergence curves and statistical tests. Additionally, the distinction between BVPL and OBL_BVPL in performance is emphasized. The Conclusion Section 5 presents the most significant findings of these experiments.

2. Materials and methods

2.1. Binary Volleyball Premier League (BVPL) Algorithm

The foundational algorithm of this work is the Volleyball Premier League Algorithm, which Moghdani and Salimifard [29] first created. VPL is considered a human-based algorithm that is inspired by the volleyball league. The composition of players consists of active players, who are those who participate in a game or competition from the initial stages, and passive players, who are substitutes who have the potential to enhance the team's overall performance and are selected by the coach. In VPL, a league represents a population, a team represents a solution, an iteration is a season, a week means the schedule, and the winning team at the end of each season represents the best solution. The VPL Algorithm encompasses 11 distinct steps. The initial phase involves the initialization process, which starts with the utilization of two matrices named formation and substitution with random values. Their dimensions are the team's number of players and the dataset's number of features. The second phase consists of the setting of the match schedule, which determines the schedule and order of the competitions among the participating teams. In this competitive setting, two teams compete with each other and afterwards, a winner is determined. Following this, both the winning and losing teams proceed to form new formations and implement four strategies accordingly. The implementation of three strategies, namely knowledge sharing, repositioning, and substitution, might be seen as advantageous for a team experiencing a decline in performance, as these strategies have the potential to enhance the overall quality of the team. Conversely, a winning team may choose to follow the leading role approach to maintain their success. The subsequent stage involves the implementation of a learning phase, during which the remaining teams are restructured based on the top three performing teams. The three concluding stages employed to enhance the efficacy of the proposed solutions are season transfers, promotions, and relegations. The original study written by the researchers explains in depth the mathematical details associated with each phase.

The binary version of BVPL is constructed by selecting an appropriate S- or V-shaped transfer function for each dataset. Additionally, the application of a suggested cost function determines the evaluation of team quality. This function assesses the solution's quality by considering the accuracy of the k-NN machine learning method. The following is a general pseudocode illustration.

Algorithm 1. BVPL Algorithm

Input: $t = 0$, parameters, cost function

Output: mean, and standard deviation of fitness, average of selected features, average accuracy

```
Initialization
For nruns = 1 to nruns
t =1;
 While t < max_iteration
    Generate a league schedule
    For i = 1: (N – 1)
  Best team =Select best team according to the cost function
  // ---TF is applied each time that the fitness of the team is calculated ---//
  // --- Cost function is applied each time that the fitness of the team is calculated -//
      For (each match in schedule table of week i)
         Apply Competition procedure between team A, and B
         Determine loser and winner teams
         Apply different strategies for loser and winner teams
         Update Best team
         Apply learning phase
      End For
  i=i+1
    End For
  Apply promotion and relegation process
  Apply season transfer process
  t = t + 1
  End While
End For
```

2.2. Opposition-based learning

T i z h o o s h [30] was the first to propose the technique known as Opposition-Based Learning (OBL) in the field of intelligence computation. The main principle of OBL is to evaluate simultaneously the fitness values of the current solution and its corresponding opposite solution, then retain the dominant individual to continue with the next iteration, thus effectively strengthening population diversity. Therefore, OBL has been widely implemented to enhance the optimization performance of many basic metaheuristics. In our case, OBL equation is implemented in the final phase of BVPL, where after the best solution is provided so far, it will also be the opposite solution. The solution with the best fitness will be selected for the next iteration. OBL is integrated with the aim of searching for a better solution than that provided by the BVPL. The mathematical equation for OBL is presented as follows, where $x_{\mathrm{op\_sol}}$ is the opposite solution, and $x_{\mathrm{act\_sol}}$ is the actual solution found better so far:

(1) $$x_{\mathrm{op\_sol}} = \text{lower boundary} + \text{upper boundary} - x_{\mathrm{act\_sol}}.$$

2.3. K-Nearest Neighbor (k-NN)

A popular and widespread supervised learning technique for feature selection is k-Nearest Neighbor [8, 7, 31]. It is a non-parametric, supervised learning algorithm that employs distance to classify or predict how a particular data point will be grouped. The parameter k in the k-NN Algorithm specifies how many neighbors will be examined to determine an instance classification. In our situation, k-NN is employed as an evaluator of the team's fitness in order to determine whether a person has PD or not.

3. Experiment design

3.1. Methodology

In this section, a description of the applied experiments is given.

- Firstly, certain assessments are employed to determine the most suitable transfer function that aligns with BVPL in each dataset.
- Secondly, BVPL is compared with 20 other popular metaheuristics listed in Table 2. The two-step binarization method has not been applied to GA, DE, and ACO as they provide themselves binary outcomes.
- At last, an innovative approach is introduced that integrates OBL into BVPL, yielding highly favorable outcomes in comparison to BVPL itself.

To avoid overfitting, k-fold cross-validation with k-fold = 5 was used, which divides datasets into k-folds. The classifier used the k – 1 folds for training data and the 1-fold for test data. As a fitness function for our evaluation, we have used k-nearest neighbor classifier accuracy with an Euclidean distance metric and k-neighbor = 5 to measure the quality of the solutions. Every execution has been implemented on a PC with an Intel(R) Core(TM) i5-8365U CPU @ 1.60 GHz and 1.90 GHz and 16 GB of RAM. All the codes have been executed in the RStudio environment.

3.2. The PD datasets

Our experiments are entirely focused on Parkinson's disease, and we base our analysis on nine publicly available PD datasets [32, 33], and one PPMI dataset [34] (Dataset D8 used in preparation of this article has been obtained on August 01, 2022 from the Parkinson's Progression Markers Initiative (PPMI) database (**www.ppmi-info.org/access-data-specimens/download-data**), RRID: SCR 006431. For up-to-date information on the study, visit **www.ppmi-info.org**). The names of the datasets and feature names correspond with the same names as in the origin sites. The following dataset is associated with three categories of Parkinson's disease: speech (vocal), writing tests, and gait processes. All the features of the datasets are normalized between 0 and 1 using a min-max normalization. Columns that identify patients, that have null values, or for which there is no data for healthy subjects have been removed from the analysis. A summary of the dimensions and number of classes is shown in Table 1. The number in brackets for the third column shows the final features used for the computations in each dataset. The selected datasets can be categorized into low, medium, or high dimensions, spanning the scalability analysis of BVPL.

Table 1. Summarized information about the PD datasets

Dataset name	ID	Dimension	Class
Parkinson	D1	195×23 (23)	2
HandPD spiral	D2_S	368×16 (13)	2
HandPD meander	D2_M	368×16 (13)	2
NewHandPD spiral	D3_S	264×16 (13)	2
NewHandPD meander	D3_M	264×16 (13)	2
Early biomarkers of PD based on naturally connected speech	D4	130×65 (27)	3
Parkinson's disease classification speech-based	D5	756×754 (754)	2
Replicated acoustic features of Parkinson	D6	240×48 (46)	2
Parkinson dataset with multiple types of sound recordings	D7	1040×29 (27)	2
Gait data arm swing	D8	148×58 (55)	2

3.3. The binary metaheuristics optimization algorithms parameters

Each binary MHA has its own parameters, which are important in the evaluation of their performance. The parameters are presented in Table 2.

Table 2. MHAs parameters

Algorithm	Reference	Parameters
General	–	nRuns = 20; maxiter = 100, population = 6; alpha_cost = 0.99, k=5-fold
BVPL	–	fall_rate=0.15, transport_rate = 0.5, β=2, b is from β to 0
ACO	[35]	τ =1, η = 1, α = 1, β = 0.1, ρ = 0.2
ABC	[36]	Acceleration coefficient a =1
ALO	[20]	-
ASO	[37]	V_{max} = 6, ε = 0.001, Depth weight α = 50, multiplier weight β = 0.2
BA	[38]	Loudness A = 0.25, pulse rate r = 0.1, Q_{min}=0, Q_{max}=2
DE	[39]	Crossover probability CR = 0.9
DF	[40]	D_{max} = 6
FA	[41]	Light absorption coefficient γ = 1, Attraction coefficient β_0 = 2, Mutation coefficient α = 0.2, Mutation coefficient damping ratio alpha_damp = 0.98
GWO	[42]	a linearly decreases from 2 to 0, C1, C2, and C3 are random numbers
HHO	[16]	β = 1.5
MFO	[43]	a linearly decreases from –1 to –2
PSO	[44]	Cognitive factor C1 =2, Social factor C2 = 2, W_{max} = 0.9, W_{min} = 0.4, V_{max} = 6
SSA	[45]	C2, C3 = rand
TGA	[46]	Number of trees in first group N_1 = 3, Number of trees in second group N_2 = 5, Number of trees in fourth group N_4 = 3, Tree reduction rate τ = 0.8, Parameter controls nearest tree λ = 0.5
WOA	[47]	a decreases linearly from 2 to 0, a_2 linearly decreases from –1 to –2, r_1, r_2, p are random numbers in interval (0, 1), b =1
EOA	[48]	Thres = 0.5, V = 1, a_1 =2, a_2=1, GP = 0.5
GA	[49]	Crossover Rate CR = 0.8, Mutation Rate MR = 0.3
SCA	[50]	r_1, decreases linearly from α to 0, α = 2, r_2, r_3, r_4 are random numbers
TLBO	[51]	-
GOA	[52]	c_{max}=1, c_{min}=0.00004

In order to see the detailed performance of BVPL, we have compared it with 20 other binary MHAs. The selected algorithms are the binary variants of: ACO, ABC, Ant Lion Optimization (ALO), Atom Search Optimization (ASO), Bat Algorithm

(BA), DE, Dragon Fly (DF), Firefly Algorithm (FA), Grey Wolf Optimization (GWO), Harris Hawk Optimization (HHO), Moth Flame Optimization (MFO), PSO, Salp Swarm (SSA), Tree Growth Algorithm (TGA), Whale Optimization Algorithm (WOA), Equilibrium Optimizer Algorithm (EOA), GA, Sine-Cosine Algorithm (SCA), Teaching Learning-Based Optimization (TLBO), and Grasshopper Optimization Algorithm (GOA) algorithms. The parameters are chosen as in the reference works or cited papers.

3.4. The performance metrics

To validate the performance of the BVPL vs. the eight TFs and the other metaheuristics, some well-known metrics have been used. In general, the FS problem has two conflicting objectives, such as choosing the smallest number of features while maintaining the highest classification accuracy. The fitness function is used to find a balance between the number of selected features and classification accuracy. The formula is as below:

(2) $$\text{Fitness} = 0.99 * \text{CE} + 0.01 * n_\text{f} / n_\text{t} \,,$$

where n_f is the number of selected features, n_t is the number of total features, and CE is the classification error = 1 – accuracy (generated by the selected features of the test dataset of the KNN Algorithm). Due to the stochastic nature of metaheuristics, for each method, 20 independent runs were performed (each run includes 100 iterations), then the average value of the performance metrics was recorded over the 20 runs. The other metrics are formulated below:

- **Classification accuracy.** It is a metric that defines how accurate a classification model is for a given set of features. It is calculated as:

(3) $$\text{Average accuracy} = \frac{1}{\text{nruns}} \sum_{j=1}^{\text{nruns}} \frac{1}{N} \sum_{i=1}^{N} \text{match}(C_i, L_i),$$

where N is the number of test points, C_i denotes the output label for data point i, match denotes the comparator that returns 0 when two labels are not identical, and 1 when they are same, and L_i denotes the reference label for i.

- **Average number of selected features.** It represents the average of the number of selected features (n_f) over nruns times and is defined as follows:

(4) $$\text{Average no features} = \frac{1}{\text{nruns}} \sum_{i=1}^{\text{nruns}} n_f(i) \,.$$

- **Average fitness.** It represents the mean of each best fitness where fi is the fitness in each run,

(5) $$\text{Average fitness} = \frac{1}{\text{nruns}} \sum_{i=1}^{\text{nruns}} f_i.$$

The last metric used, standard deviation describes the variation of the optimal fitness in all runs.

4. Experiment results

4.1. Results from the S-shaped and V-shaped Transfer Function (TF)

This subsection presents the optimal outcomes attained by BVPL utilizing eight transfer functions for each dataset. Tables 9-12 in Appendix A show the results for the mean and standard deviation of fitness, the average accuracy, and the average number of selected features. The objective was to identify the most prominent TF for the BVPL based on the average fitness, as presented in Table 9. When the average fitness is equal, the subsequent criterion considered is the maximum average accuracy, which is shown in Table 11. The successful transfer functions for each dataset are as follows: D1 (V3), D2_S (V3), D2_M (S2), D3_S (S3), D3_M (S3), D4 (S2), D5 (S3), D6 (V4), D7 (S2), and D8 (V3). The TF mentioned is utilized for the subsequent experiments in the other MHAs.

4.2. Comparison of BVPL and metaheuristics on the datasets

The results from the metrics for the MHAs are presented in Tables 3-7. The optimal outcomes are shown through the utilization of both italics and bold formatting. In these tables, we will henceforth refer to the metrics as average fitness (f_{avg}), the standard deviation of the fitness (f_{sd}), average accuracy (acc_{avg}), and the average number of features ($feat_{avg}$). In general, it can be observed that BVPL produces a smaller number of features compared to the other methods, particularly in the cases of D1, D2_M, D3_S, D3_M, and D7. It is important to note that BVPL has a predetermined minimum number of features, with the maximum being half of the total number of features in each dataset.

In reference to D1 (Table 3), it can be shown that ACO outperforms BVPL in all metrics, with the exception of the average number of features. BVPL is among the third-best algorithms after ACO and GA. According to the information presented in Table 3 for the D2_S dataset, it is evident that the BVPL Algorithm ranks as the second most effective approach, surpassed only by the ACO Algorithm.

Table 3. The results of 6 metrics for D1 (left) and D2_S (right)

MHA	f_{avg}	(f_{sd})	acc_{avg}	$feat_{avg}$	f_{avg}	(f_{sd})	acc_{avg}	$feat_{avg}$
BVPL	0.05338	0.01874	0.94723	**2.5**	0.06385	0.01782	0.93724	2.05
ACO	**0.04030**	0.02004	**0.96379**	9.95	**0.05837**	0.01429	**0.94495**	4.75
ABC	0.11009	0.02708	0.89052	2.95	0.12738	0.04765	0.87294	2.05
ALO	0.05949	0.02450	0.94310	6.75	0.08581	0.04358	0.91606	2.8
ASO	0.06754	0.02601	0.93621	9.65	0.07619	0.017780	0.92569	3.15
BA	0.07099	0.03129	0.93362	11.6	0.09661	0.02855	0.90734	5.85
DE	0.05733	0.02612	0.94741	11.6	0.07337	0.01636	0.93119	6.3
DF	0.08042	0.02547	0.92414	11.7	0.08423	0.02352	0.92064	6.8
FA	0.10744	0.03267	0.89310	3.55	0.12738	0.05275	0.87294	**1.9**
GWO	0.05761	0.02221	0.94741	12.2	0.07440	0.01922	0.93028	6.45
HHO	0.06503	0.02657	0.93707	6	0.08095	0.02206	0.92156	3.95
MFO	0.07499	0.02574	0.92586	3.5	0.09121	0.04623	0.90963	2.1
PSO	0.07726	0.02155	0.92759	12.25	0.08802	0.02536	0.91606	5.9
SSA	0.10635	0.03638	0.89483	3.05	0.17589	0.04722	0.82431	2.05
TGA	0.05684	0.02242	0.94828	12.4	0.06544	0.01638	0.93853	5.5
WOA	0.05721	0.02530	0.94483	5.7	0.06641	0.01579	0.93532	2.85
EOA	0.08080	0.03317	0.91983	9.7	0.11483	0.05546	0.88578	6.05
GA	0.04904	0.01659	0.95517	10.25	0.06713	0.01844	0.93670	5.35
SCA	0.07658	0.01947	0.92414	2.7	0.09888	0.03708	0.90180	2.1
TLBO	0.09030	0.02952	0.91035	3.4	0.12556	0.05023	0.87477	**1.9**
GOA	0.09256	0.02409	0.90862	4.6	0.10967	0.04740	0.89174	3

The ACO Algorithm demonstrates superior performance in terms of average fitness and accuracy. BVPL Algorithm exhibits a high level of rivalry in terms of accuracy when compared to TGA, WOA, and GA. There is a slight distinction between BVPL and FA, as well as TLBO, in terms of the number of features.

In relation to the D2_M dataset, as illustrated in Table 4, it can be established that BVPL gives the greatest average fitness, along with average accuracy and the selected features. The ACO remains highly competitive. According to the data presented in Table 4 (D3_S), BVPL Algorithm exhibits substantially better outcomes in terms of average fitness and accuracy compared to ACO. Their results are better than those of the other MHAs.

Table 4. The results of 6 metrics for D2_M (left), and D3_S (right)

MHA	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$
BVPL	**0.05716**	0.01763	**0.94433**	**2.45**	**0.14256**	0.03281	**0.85924**	**3.85**
ACO	0.05977	0.02083	0.94358	4.85	0.14590	0.03043	0.85823	6.8
ABC	0.10143	0.03725	0.90138	4.65	0.20743	0.02453	0.79557	5.4
ALO	0.07313	0.01890	0.93211	7.15	0.16586	0.03436	0.83987	8.6
ASO	0.06607	0.02793	0.93532	**2.45**	0.17314	0.03374	0.82848	4
BA	0.08402	0.02626	0.92064	6.55	0.17735	0.02515	0.82595	6.05
DE	0.07990	0.02709	0.92523	7.05	0.16929	0.03239	0.83481	6.9
DF	0.06998	0.01959	0.93440	6.05	0.18040	0.03475	0.82279	5.95
FA	0.11725	0.04133	0.88578	5	0.22355	0.04155	0.77911	5.85
GWO	0.07577	0.02597	0.92890	6.45	0.15396	0.03695	0.85063	7.3
HHO	0.07436	0.02302	0.92982	5.85	0.15467	0.03317	0.84873	5.9
MFO	0.11023	0.05208	0.89312	5.3	0.17004	0.03198	0.83354	6.3
PSO	0.08836	0.03218	0.91606	6.3	0.20972	0.04757	0.79367	6.55
SSA	0.15769	0.04766	0.84312	4.35	0.29933	0.06355	0.70063	5.5
TGA	0.06899	0.01860	0.93486	5.4	0.15542	0.03411	0.84937	7.55
WOA	0.06738	0.01862	0.93716	6.2	0.15609	0.02941	0.84747	6.1
EOA	0.08653	0.04672	0.91697	5.4	0.17861	0.04447	0.82468	4.45
GA	0.06915	0.02361	0.93486	5.6	0.15684	0.03629	0.84684	6.25
SCA	0.10139	0.04792	0.90184	4.45	0.17618	0.03727	0.82722	5.8
TLBO	0.08632	0.02717	0.91697	4.95	0.18345	0.04716	0.81962	5.85
GOA	0.10469	0.04931	0.89817	4.65	0.18884	0.04317	0.81392s	5.55

According to the findings presented in Table 5 of D3_M, the SCA method demonstrates superior performance in terms of accuracy and fitness. According to the ranking, ACO is considered the second most favorable alternative, followed by BVPL. The results from the D4 dataset, which are shown in Table 5, show that BVPL does better than the other MHs on all of the criteria that have been looked at. ACO is the second-best one, and the others are far away from this result.

The results from the seventh dataset, D5 (Table 6), provide further confirmation that BVPL outperforms the other MHAs. The competition between ACO, BA, and WOA is evident across various indicators. In the D6 dataset, as presented in Table 6, it can be observed that BVPL presents greater performance in terms of average fitness. However, ACO demonstrates higher average accuracy. ALO reveals strong concurrence with ACO in terms of metrics.

Table 5. The results of 6 metrics for D3_M (left), and D4 (right)

MHA	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$
BVPL	0.13584	0.02114	0.86557	**3.3**	**0.34154**	0.03427	**0.65643**	3.65
ACO	0.12869	0.02737	0.87532	6.45	0.37252	0.04392	0.62821	12
ABC	0.18324	0.03017	0.81962	5.45	0.481	0.05230	0.51795	9.55
ALO	0.16381	0.02369	0.84177	8.85	0.42017	0.03796	0.58333	19.7
ASO	0.16077	0.03347	0.84051	3.45	0.43267	0.04637	0.56410	**2.95**
BA	0.18696	0.04335	0.81646	6.3	0.43515	0.05336	0.56539	12.7
DE	0.16645	0.03761	0.83797	7.25	0.42367	0.05521	0.57821	15.85
DF	0.16327	0.02266	0.84051	6.45	0.43883	0.04639	0.56154	12.35
FA	0.22121	0.05310	0.78101	5.3	0.49485	0.04527	0.50384	9.5
GWO	0.15166	0.03789	0.85317	7.55	0.416	0.04499	0.58590	15.7
HHO	0.14385	0.02511	0.86013	6.45	0.40635	0.04903	0.59359	10.4
MFO	0.15421	0.03190	0.84937	6.1	0.41321	0.03936	0.58590	8.45
PSO	0.19677	0.03463	0.80696	6.8	0.49506	0.05337	0.50513	13.35
SSA	0.25526	0.06306	0.74494	4.7	0.46537	0.05410	0.53205	10.15
TGA	0.14953	0.02683	0.85443	6.5	0.40173	0.03892	0.6	14.9
WOA	0.14573	0.02331	0.85823	6.45	0.39673	0.03614	0.60513	15.1
EOA	0.16766	0.02287	0.83544	5.2	0.40215	0.04229	0.59744	10.9
GA	0.14707	0.02602	0.85633	5.8	0.43825	0.04543	0.56154	10.85
SCA	**0.11151**	*0.04781*	**0.89220**	5.25	0.40267	0.04265	0.59744	11.4
TLBO	0.18228	0.04361	0.82025	5.2	0.43529	0.03697	0.56410	9.75
GOA	0.17113	0.03209	0.83165	5.35	0.43435	0.04199	0.56410	7.3

Table 6. The results of 6 metrics for D5 (left), and D6 (right)

MHA	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$
BVPL	**0.08522**	0.01227	**0.91593**	149.8	**0.10983**	0.02630	0.89040	5.95
ACO	0.09819	0.01492	0.90553	351.7	0.11227	0.01997	**0.89097**	19.7
ABC	0.12731	0.01741	0.87589	331.6	0.16207	0.02344	0.83819	8.2
ALO	0.10986	0.01564	0.89712	605.4	0.11520	0.01930	0.8875	16.9
ASO	0.10473	0.01292	0.89602	**134.6**	0.15249	0.02125	0.84792	8.7
BA	0.09687	0.01310	0.90708	367.7	0.15248	0.02947	0.85069	21
DE	0.10996	0.01631	0.89513	462.3	0.13757	0.02791	0.86667	25.1
DF	0.11513	0.01214	0.88872	373.3	0.17192	0.02570	0.83125	21.9
FA	0.12204	0.01559	0.88120	332.9	0.16833	0.02649	0.83194	8.8
GWO	0.09813	0.01367	0.90774	511.9	0.13015	0.02325	0.87431	25.7
HHO	0.10991	0.01650	0.89381	359.9	0.13889	0.02635	0.86181	9.4
MFO	0.10492	0.01664	0.89845	330	0.12981	0.01735	0.87083	8.7
PSO	0.12457	0.01712	0.87920	374.7	0.16834	0.02626	0.83472	21.2
SSA	0.11879	0.01712	0.88252	336.6	0.15796	0.02849	0.84306	8.2
TGA	0.10641	0.01481	0.89867	459.3	0.13511	0.01844	0.86944	26.4
WOA	0.10327	0.01346	0.90155	436.9	0.13913	0.02421	0.86042	4.25
EOA	0.10829	0.01556	0.89513	295.1	0.12934	0.01938	0.87083	16.45
GA	0.10979	0.01378	0.89381	350.5	0.13516	0.02037	0.86806	20.4
SCA	0.10739	0.01431	0.89624	351.3	0.12829	0.02698	0.87153	**3.4**
TLBO	0.11524	0.01484	0.88805	331.9	0.14520	0.02107	0.85486	6.8
GOA	0.11153	0.01648	0.89071	251.2	0.15037	0.02646	0.85070	11.5

In D7 (Table 7), it can be observed that BVPL indicates weak efficiency, while ACO yields superior outcomes in terms of average fitness and accuracy. The ALO, GWO, HHO, TGA, and WOA exhibit significant competition with the ACO. In the final dataset (Table 7), it can be observed that BVPL implies better performance across all measures, with the exception of the number of features, where SCA displays the most positive outcomes. For the same dataset, ACO and ALO indicate a considerable level of similarity.

Table 7. The results of 6 metrics for D7 (left), and D8 (right)

MHA	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$
BVPL	0.32260	0.01530	0.67564	**3.85**	**0.15493**	0.03445	**0.84429**	4.2
ACO	**0.29192**	0.01316	**0.71026**	13.5	0.16540	0.04240	0.8375	24.6
ABC	0.33930	0.01734	0.66122	10	0.24681	0.05044	0.75227	8.1
ALO	0.30902	0.01684	0.69615	21.35	0.16177	0.05892	0.83977	17.1
ASO	0.31488	0.01947	0.68478	7.3	0.22156	0.05046	0.77727	5.7
BA	0.31183	0.02068	0.69022	13.4	0.21408	0.05738	0.78864	26.1
DE	0.31067	0.01824	0.69311	17.8	0.20351	0.05208	0.8	29.8
DF	0.31903	0.01359	0.68301	13.6	0.25707	0.05711	0.74546	27.4
FA	0.34018	0.01799	0.66010	9.6	0.24349	0.05011	0.75568	8.7
GWO	0.30059	0.01698	0.70272	16.4	0.19273	0.04837	0.81136	32.3
HHO	0.30341	0.01564	0.69792	11.3	0.21054	0.05312	0.78864	6.95
MFO	0.31164	0.01737	0.68958	11.3	0.17367	0.04461	0.82614	8.4
PSO	0.32716	0.02074	0.67468	13.3	0.25237	0.05098	0.75	26.3
SSA	0.33484	0.02285	0.66555	9.9	0.24825	0.05944	0.75114	8.3
TGA	0.30115	0.00596	0.70401	21.1	0.21772	0.01143	0.78523	27.5
WOA	0.29696	0.01591	0.70657	16.8	0.18185	0.04795	0.81705	3.9
EOA	0.31606	0.01780	0.68510	9.7	0.18704	0.04309	0.8125	21.4
GA	0.31019	0.01461	0.69183	13.3	0.20466	0.04393	0.79773	23.8
SCA	0.31166	0.01995	0.68942	12.2	0.18071	0.04426	0.81818	**3.35**
TLBO	0.31445	0.01843	0.68638	10.3	0.22857	0.03077	0.77046	7.1
GOA	0.32323	0.01769	0.67772	10.9	0.22477	0.04204	0.775	10.9

4.3. Convergences curves and statistical difference

The convergence curves can visually illustrate the variations in the performance of all the MHAs across different criteria. Figs 1 and 2 illustrate the convergence curves that correspond to the average fitness observed throughout each iteration. Each graph shown represents a distinct dataset. It can be observed that among the three datasets, namely D2_M, D4, and D5, the BVPL algorithm exhibits a quicker convergence speed compared to the other metaheuristics. Moreover, in the cases of D2_S, D3_M, D3_S, D6, and D8, the level of competitiveness of BVPL is notably high. Additionally, it is observed that BVPL in D8, D6, and D3_S exhibit a faster rate of convergence compared to the other MHAs, after the 75th iteration. SSA reveals a major shift in convergence across all datasets, surpassing the other algorithms in terms of fitness values. The ACO algorithm exhibits a notable convergence speed when searching for the global optimum in datasets D1, D2_S, and D7.

Tables 13 and 14 (Appendix B), present the *p*-values resulting from the use of *t*-tests or Wilcoxon sum-rank tests. These tests have been conducted to examine the difference in average fitness between BVPL and the remaining 20 MHAs. If the *p*-value is less than 0.05, it can be concluded that there is a significant difference in the average fitness of BVPL when compared to the other algorithm. In the event that BVPL demonstrates superior performance, it will be represented by the symbol “+”. Conversely, if BVPL exhibits equivalent performance, it will be signified by the symbol “=”. Lastly, if BVPL demonstrates inferior performance, it will be indicated by the symbol “– “. The performance of BVPL surpasses that of 17 MHAs, exhibiting superior results in over 50% of the datasets. The performance of BVPL does not appear to be superior to ACO, except in the case of four specific datasets. Additionally, when considering the performance of WOA and GA, they are found to be equally superior to BVPL.

4.4. Results of BVPL vs BVPL_OBL

This section provides a summary of the four metrics used to evaluate the FS problem BVPL against BVPL_OBL, as presented in Table 8.

Table 8. Results BVPL against OBL_BVPL

Algorithm	BVPL Algorithm				OBL_BVPL Algorithm			
Dataset	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$
D1	0.05338	0.01874	0.94723	2.5	0.01848	0.00539	0.98274	3.05
D2_S	0.06385	0.01782	0.93724	2.05	0.05601	0.00747	0.94654	3.7
D2_M	0.05716	0.01763	0.94433	2.45	0.05082	0.00864	0.95216	4.15
D3_S	0.14256	0.03281	0.85924	3.85	0.10058	0.01898	0.90241	4.75
D3_M	0.13584	0.02114	0.86557	3.3	0.09235	0.00385	0.91013	4.05
D4	0.34154	0.03427	0.65643	3.65	0.30865	0.01973	0.68971	3.8
D5	0.08522	0.01227	0.91593	149.8	0.08392	0.01209	0.91728	152.3
D6	0.10983	0.02630	0.89040	5.95	0.08794	0.01235	0.91249	5.85
D7	0.32260	0.01530	0.67564	3.85	0.28404	0.01429	0.71523	5.5
D8	0.15493	0.03445	0.84429	4.2	0.11701	0.02745	0.88298	6.3

The suggested methodology exhibits significant enhancements in terms of average fitness and accuracy across all datasets. Incorporating the opposing approach leads to a considerable increase in accuracy and a decrease in fitness. The observed improvement in precision ranges from 0.135% in D5 to 4.456%. In relation to efficacy, this technique has demonstrated major relevance in the prediction of Parkinson’s disease.

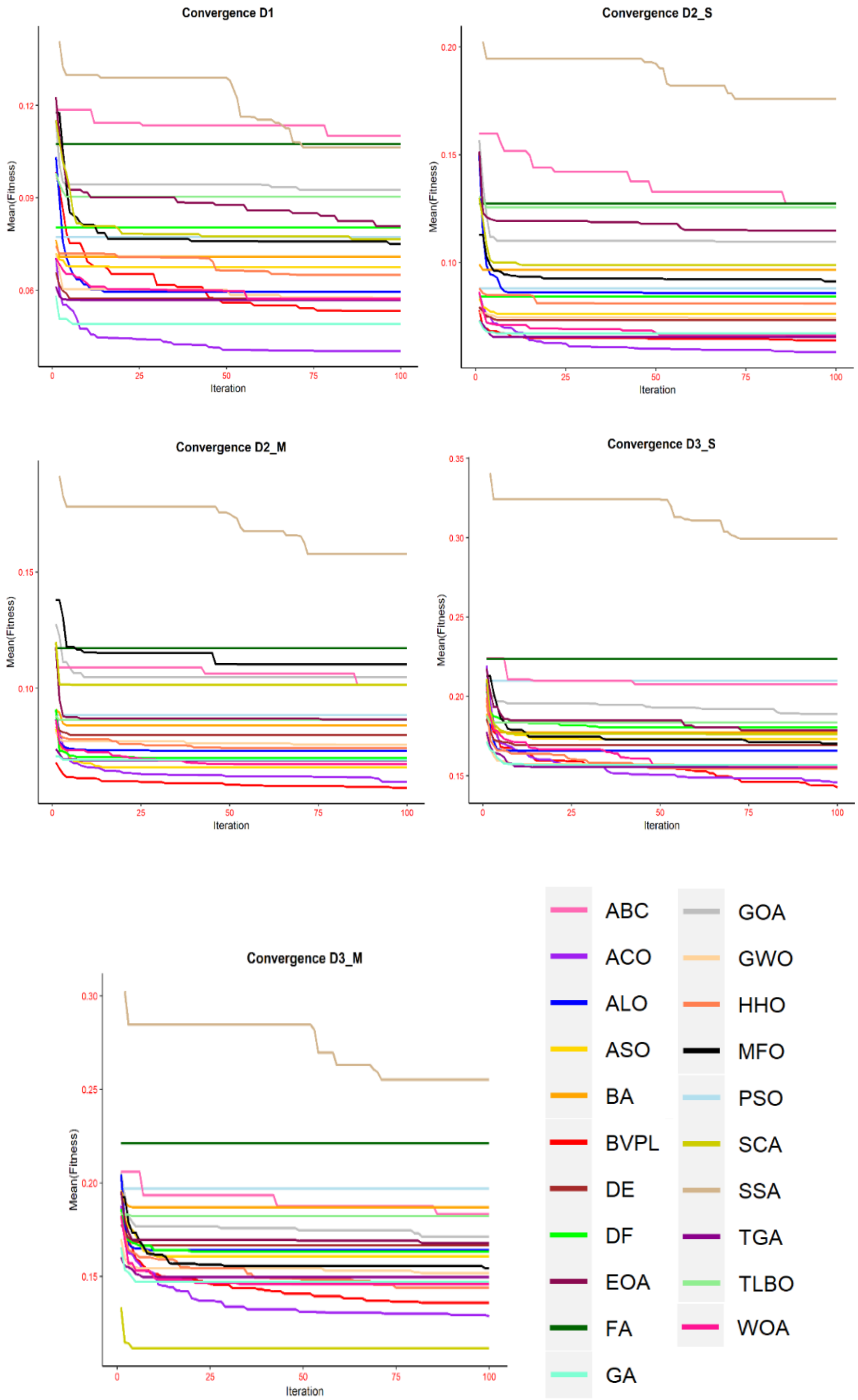

Fig. 1. The convergence curves of B-VPL versus the other MHAs for the first five datasets

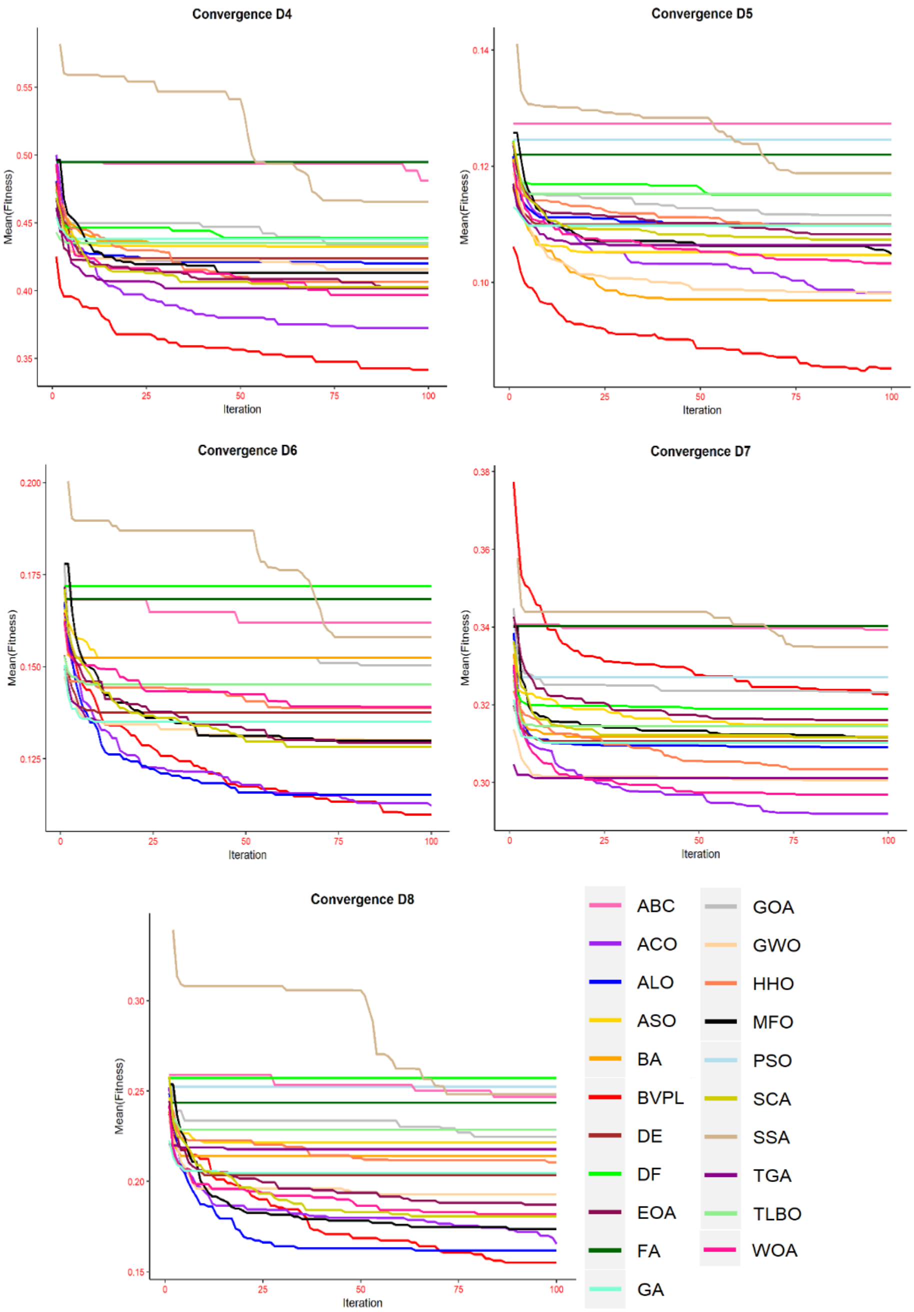

Fig. 2. The convergence curves of B-VPL versus the other MHAs for the last five datasets

5. Conclusions

This paper has employed a metaheuristic, BVPL in feature selection problem which has provided a higher accuracy in predicting PD, in 10 different datasets, compared with a large list of metaheuristics. BVPL outperforms ACO in fitness and accuracy across five datasets, while ACO outperforms in one. SCA outperforms ASO in one dataset, with the lowest values across all datasets. The BVPL algorithm demonstrates an acceptable speed of convergence and effectiveness in searching across a wide range of datasets, consistently ranking among the top three among other MHAs, and superior in three of them. Opposition-based learning can enhance the prediction of PD above 90% in 7 out of 10 datasets.

***Acknowledgments*:** This work has been supported by the Bulgarian National Science Fund by the project "Mathematical models, methods and algorithms for solving hard optimization problems to achieve high security in communications and better economic sustainability" Grant No KP-06-N52/7.

References

1. Naka, E., V. Guliashki. B-VPL: A Binary Volleyball Premier League Optimization Algorithm for Feature Selection. – In: Proc. of 29th International Conference on Systems, Signals and Image Processing (IWSSIP'22) – IEEE Xplore, 2022, pp. 1-4.
2. Pudjihartono, N, T. Fadason, A. W. Kempa-Liehr, J. M. O'Sullivan. A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction. – Frontiers in Bioinformatics, Vol. **2**, 2022, pp. 1-17.
3. Venkatesh, B., J. Anuradha. A Review of Feature Selection and Its Methods – Cybernetics and Information Technologies, Vol. **19**, 2019, No 1, pp. 3-26.
4. Remeseiro, B., V. Bolon-Canedo. A Review of Feature Selection Methods in Medical Applications. – Computers in Biology and Medicine, Vol. **112**, 2019, pp. 1-35.
5. Liu, W., J. Wang. A Brief Survey on Nature-Inspired Metaheuristics for Feature Selection in Classification in This Decade. – In: Proc. of 16th IEEE International Conference on Networking, Sensing and Control (ICNSC'19) – IEEE Xplore, 2019, pp. 424-429.
6. Sharma, M., P. Kaur. A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem. – Archives of Computational Methods in Engineering, Vol. **28**, 2021, pp. 1103-1127.
7. Agrawal, P., H. F. Abutarboush, T. Ganesh, A. W. Mohamed. Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019) – IEEE Access, Vol. **9**, 2021, pp. 26766-26791.
8. Dokeroglu, T., A. Deniz, H. E. Kiziloz. A Comprehensive Survey on Recent Metaheuristics for Feature Selection. – Neurocomputing, Vol. **494**, 2022, pp. 269-296.
9. Sörensen, K., F. W. Glover. Metaheuristics. – In: S. I. Gass, M. C. Fu, Eds. Encyclopedia of Operations Research and Management Science. Boston, MA, Springer, 2013, pp. 960-970.
10. Rajwar, K., K. Deep, S. Das. An Exhaustive Review of the Metaheuristic Algorithms for Search and Optimization: Taxonomy, Applications, and Open Challenges. – Artificial Intelligence Review, 2023, pp. 1-71.
11. Crawford, B., R. Soto, G. Astorga, J. García, C. Castro, F. Paredes. Putting Continuous Metaheuristics to Work in Binary Search Spaces. – Complexity, Vol. **2017**, 2017, pp. 1-19.
12. Mirjalili, S., A. Lewis. S-Shaped vs. V-Shaped Transfer Functions for Binary Particle Swarm Optimization. – Swarm and Evolutionary Computation, Vol. **9**, 2013, pp. 1-14.
13. Beheshti, Z. A Novel x-Shaped Binary Particle Swarm Optimization. – Soft Computing, Vol. **25**, 2021, pp. 3013-3042.
14. Mirjalili, S., H. Zhang, S. Mirjalili, S. Chalup, N. Noman. A Novel U-Shaped Transfer Function for Binary Particle Swarm Optimisation. – In: A. Nagar, K. Deep, J. Bansal, K. Das, Eds. Soft Computing for Problem Solving 2019. – Advances in Intelligent Systems and Computing, Springer, Singapore, Vol. **1138**, 2020, pp. 241-259.

15. W a n g, L., X. W a n g, J. F u, L. Z h e n. A Novel Probability Binary Particle Swarm Optimization Algorithm and Its Application. – Journal of Software, Vol. **3**, 2008, No 9, pp. 28-35.
16. T o o, J., A. R. A b d u l l a h, N. M o h d S a a d. A New Quadratic Binary Harris Hawk Optimization for Feature Selection. – Electronics, Vol. **8**, 2019, pp. 1-27.
17. N a d i m i-S h a h r a k i, M. H., M. B a n a i e-D e z f o u l i, H. Z a m a n i, S. T a g h i a n, S. M i r j a l i l i. B-MFO: A Binary Moth-Flame Optimization for Feature Selection from Medical Datasets. – Computers, Vol. **10**, 2021, pp. 1-18.
18. K u m a r, V., D. K u m a r, M. K a u r, D. S i n g h, S. A. I d r i s, H. A l s h a z l y. A Novel Binary Seagull Optimizer and Its Application to Feature Selection Problem. – IEEE Access, Vol. **9**, 2021, pp. 103481-103496.
19. J i a n g, Y., Q. L u o, Y. W e i, L. A b u a l i g a h, Y. Z h o u. An Efficient Binary Gradient-Based Optimizer for Feature Selection. – Mathematical Biosciences and Engineering, Vol. **18**, 2021, No 4, pp. 3813-3854.
20. E m a r y, E., H. M. Z a w b a a, A. E. H a s s a n i e n. Binary Ant Lion Approaches for Feature Selection. – Neurocomputing, Vol. **213**, 2016, pp. 54-65.
21. T u r k o g l u, B., S. A. U y m a z, E. K a y a. Binary Artificial Algae Algorithm for Feature Selection. – Applied Soft Computing, Vol. **120**, 2022, pp. 1-19.
22. T o o, J., A. R. A b d u l l a h. Binary Atom Search Optimisation Approaches for Feature Selection. – Connection Science, Vol. **32**, 2020, No 4, pp. 406-430.
23. P a n, J.-S., L. Y u e, S.-C. C h u, P. H u, B. Y a n, H. Y a n g. Binary Bamboo Forest Growth Optimization Algorithm for Feature Selection. – Entropy, Vol. **25,** 2013, pp. 1-25
24. N s s i b i, M., G. M a n i t a, O. K o r b a a. Binary Giza Pyramids Construction for Feature Selection. – Procedia Computer Science, Vol. **192**, 2021, pp. 676-687.
25. E l u r i, R. K., N. D e v a r a k o n d a. Binary Golden Eagle Optimizer with Time-Varying Flight Length for Feature Selection. – Knowledge-Based Systems, Vol. **247**, 2022, pp. 1-28.
26. A g r a w a l, P., T. G a n e s h, D. O l i v a, A. W. M o h a m e d. S-Shaped and V-Shaped Gaining-Sharing Knowledge-Based Algorithm for Feature Selection. – Applied Intelligence, Vol. **52**, 2022, pp. 81-112.
27. G h o s h, K. K., R. G u h a, S. K. B e r a, N. K u m a r, R. S a r k a r. S-Shaped vs. V-Shaped Transfer Functions for Binary Manta Ray Foraging Optimization in Feature Selection Problem. – Neural Computing & Application, Vol. **33**, 2021, pp. 11027-11041.
28. A w a d a l l a h, M. A., A. I. H a m m o u r i, M. A. A l-B e t a, M. S. B r a i k, M. A. E l a z i z. Binary Horse Herd Optimization Algorithm with Crossover Operators for Feature Selection. – Computers in Biology and Medicine, Vol. **141**, 2022, pp. 105152.
29. M o g h d a n i, R., K. S a l i m i f a r d. Volleyball Premier League Algorithm. – Applied Soft Computing, Vol. **64**, 2018, pp. 161-185.
30. T i z h o o s h, H. R. Opposition-Based Learning: A New Scheme for Machine Intelligence. – In: Proc. of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, Vienna, Austria, 2005, pp. 695-701.
31. N a k a, E. K. Review of Metaheuristic Algorithms in Feature Selection Based on Parkinson Disease. – In: Proc. of 24th International Conference on Control Systems and Computer Science (CSCS'23) – IEEE Xplore, 2023, pp. 221-228.
32. UCI Machine Learning Repository (Accessed in August 2022). **https://archive.ics.uci.edu/datasets?search=parkinson**
33. HandPD Dataset, New HandPD Dataset (Accessed in 4 August 2022). **https://wwwp.fc.unesp.br/~papa/pub/datasets/Handpd/**
34. Parkinson's Progression Markers Initiative (Accessed in 1 August 2022). **https://www.ppmi-info.org**
35. T o o, J. Ant Colony Optimization for Feature Selection. 2021 (Retrieved at 26/05/2022). **https://www.mathworks.com/matlabcentral/fileexchange/80278-ant-colony-optimization-for-feature-selection?s_tid=srchtitle**
36. H e r i s, M. K. Artificial Bee Colony. 2015. **https://yarpiz.com/297/ypea114-artificial-bee-colony**
37. T o o, J., A. R. A b d u l l a h. Binary Atom Search Optimisation Approaches for Feature Selection. – Connection Science, Vol. **32**, 2020, No 4, pp. 406-430.

38. M i r j a l i l i, S., S. M. M i r j a l i l i, X.-S. Y a n g. Binary Bat Algorithm. – Neural Computing & Applications, Vol. **25**, 2014, pp. 663-681.
39. T o o, J., A. R. A b d u l l a h, N. M o h d S a a d. Hybrid Binary Particle Swarm Optimization Differential Evolution-Based Feature Selection for EMG Signals Classification. – Axioms, Vol. **8**, 2019, No 3, pp. 1-17.
40. T o o, J., S. M i r j a l i l i. A Hyper Learning Binary Dragonfly Algorithm for Feature Selection: A COVID-19 Case Study. – Knowledge-Based Systems, Vol. **212,** 2020, pp. 1-16.
41. H e r i s, M. K. Firefly Algorithm (FA) in MATLAB. 2015. **https://yarpiz.com/259/ypea112, retrieved in August 2022**
42. T o o, J., A. R. A b d u l l a h, N. M o h d S a a d, N. M. A l i, W. T e e. A New Competitive Binary Grey Wolf Optimizer to Solve the Feature Selection Problem in EMG Signals Classification. – Computers, Vol. **7**, 2018, No 4, pp. 1-18.
43. M i r j a l i l i, S. Moth-Flame Optimization Algorithm: A Novel Nature-Inspired Heuristic Paradigm. – Knowledge-Based Systems, Vol. **89**, 2015, pp. 228-249.
44. T o o, J. Particle Swarm Optimization for Feature Selection, 2021, **https://www.mathworks.com/matlabcentral/fileexchange/78802-particle-swarm-optimization-for-feature-selection?s_tid=prof_contriblnk**
45. T o o, J. Salp Swarm Algorithm for Feature Selection. 2021. **https://www.mathworks.com/matlabcentral/fileexchange/78913-salp-swarm-algorithm-for-feature-selection?s_tid=prof_contriblnk**
46. T o o, J., A. R. A b d u l l a h, N. M. S a a d, N. M. A l i. Feature Selection Based on Binary Tree Growth Algorithm for the Classification of Myoelectric Signals. – Machines, Vol. **6**, 2018, No 4, pp. 1-19.
47. M i r j a l i l i, S., A. L e w i s. The Whale Optimization Algorithm. – Advances in Engineering Software, Vol. **95**, 2016, pp. 51-67. DOI: 10.1016/j.advengsoft.2016.0. **https://www.mathworks.com/matlabcentral/fileexchange/55667-the-whale-optimization-algorithm**
48. T o o, J., S. M i r j a l i l i. General Learning Equilibrium Optimizer: A New Feature Selection Method for Biological Data Classification. – Applied Artificial Intelligence, Vol. **35**, 2021, No 3, pp. 247-263.
49. T o o, J., A. R. A b d u l l a h. A New and Fast Rival Genetic Algorithm for Feature Selection. – The Journal of Supercomputing, Vol. **77**, 2021, pp. 2844-2874.
50. T o o, J. Sine Cosine Algorithm for Feature Selection. 2021. **https://www.mathworks.com/matlabcentral/fileexchange/80671-sine-cosine-algorithm-for-feature-selection?s_tid=prof_contriblnk, retrieved at 09/08/2022**
51. H e r i s, M. K. Teaching-Learning-Based Optimization in MATLAB. 2015. **https://yarpiz.com/83/ypea111-teaching-learning-based-optimization.**
52. S a r e m i, S., S. M i r j a l i l i, A. L e w i s. Grasshopper Optimization Algorithm: Theory and Application. – Advances in Engineering Software, Vol. **105**, 2017, pp. 30-47.

Appendix A. Results from the S-shaped and V-shaped transfer functions

The values displayed in bold indicate the optimal outcomes.

Table 9. The results of average fitness for each dataset, and transfer function

Dataset	S1	S2	S3	S4	V1	V2	V3	V4
D1	0.06027	0.05929	0.06037	0.06228	0.05749	0.05832	**0.05338**	0.05421
D2_S	0.06508	0.06822	0.07338	0.06686	0.06768	0.06512	**0.06385**	0.06690
D2_M	0.05992	**0.05716**	0.06554	0.06438	0.06054	0.06112	0.05798	0.05856
D3_S	0.15848	0.15530	**0.14256**	0.14607	0.17481	0.17176	0.16378	0.15534
D3_M	0.14306	0.14194	**0.13584**	0.14511	0.15267	0.15205	0.14599	0.14123
D4	0.3491	**0.34154**	0.35054	0.34283	0.35271	0.34762	0.34258	**0.34154**
D5	0.08854	0.08592	**0.08522**	0.08755	0.09008	0.08931	0.08968	0.08599
D6	0.11497	0.12096	0.12025	0.11752	0.11551	0.11371	0.11914	**0.10983**
D7	0.32395	**0.3226**	0.32349	0.32566	0.32869	0.32774	0.32760	0.32313
D8	0.16934	0.17679	0.18291	0.17965	0.16723	0.16270	**0.15493**	0.16745

Table 10. The standard deviation of fitness for each dataset, and transfer function

Dataset	S1	S2	S3	S4	V1	V2	V3	V4
D1	0.02302	0.02300	0.02496	0.02819	0.02146	0.02226	**0.01874**	0.02274
D2_S	0.01840	0.02083	0.025240	0.02082	0.01864	0.01794	**0.01782**	0.01861
D2_M	0.01598	0.01763	0.02238	0.01637	0.01514	0.01645	0.01515	**0.01388**
D3_S	0.03406	0.03564	0.03281	0.02990	0.03616	**0.02799**	0.03449	0.03104
D3_M	0.02132	0.0225	**0.02114**	0.02431	0.02308	0.02489	0.02989	0.02509
D4	0.03609	0.03427	**0.03182**	0.03448	0.03616	0.03317	0.03419	0.03223
D5	0.01093	0.01111	0.01227	0.01105	0.01286	0.01204	**0.01051**	0.01301
D6	0.02274	0.02440	0.02716	0.03290	**0.01905**	0.03097	0.02417	0.02630
D7	0.01469	0.01530	0.01461	0.01394	0.01299	0.01254	**0.01089**	0.01138
D8	0.03893	**0.03386**	0.03548	0.03908	0.03589	0.03888	0.03445	0.03755

Table 11. The average accuracy for each dataset, and transfer function

Dataset	S1	S2	S3	S4	V1	V2	V3	V4
D1	0.94040	0.94149	0.94068	0.93879	0.94301	0.94215	**0.94723**	0.94669
D2_S	0.93597	0.93311	0.92781	0.93448	0.93344	0.93620	**0.93724**	0.93445
D2_M	0.94174	**0.94433**	0.93570	0.93720	0.94057	0.94037	0.94354	0.94291
D3_S	0.84258	0.84557	**0.85924**	0.85561	0.82582	0.82890	0.83709	0.84620
D3_M	0.85802	0.85950	**0.86557**	0.85587	0.84789	0.84857	0.85506	0.86000
D4	0.64852	**0.65643**	0.64757	0.65530	0.64487	0.65018	0.65526	0.65627
D5	0.91173	0.91482	**0.91593**	0.91374	0.90995	0.91083	0.91060	0.91458
D6	**0.88458**	0.87919	0.88054	0.88351	0.88417	0.88620	0.88058	0.89040
D7	0.67397	**0.67564**	0.67502	0.67289	0.66911	0.67010	0.67034	0.67477
D8	0.82955	0.82282	0.81723	0.82066	0.83184	0.83644	**0.84429**	0.83193

Table 12. The average number of features for each dataset, and transfer function

Dataset	S1	S2	S3	S4	V1	V2	V3	V4
D1	2.8	3	3.6	3.7	2.35	**2.3**	2.5	3.15
D2_S	**2.05**	2.4	2.3	2.4	2.15	2.35	**2.05**	2.4
D2_M	2.7	2.45	2.25	2.65	**2.05**	2.5	2.5	2.45
D3_S	3.15	2.9	3.85	3.75	**2.85**	***2.85***	3	3.7
D3_M	3	3.4	3.3	2.9	**2.5**	2.55	3	3.15
D4	**2.95**	3.65	4.25	4.1	**2.95**	3.35	3.35	3.25
D5	87.1	119.2	149.75	161.95	**69.75**	77.65	87.8	107
D6	**3.15**	6.1	8.95	9.9	3.8	4.7	4.1	5.95
D7	3.05	3.85	4.55	4.75	**2.9**	2.95	3.2	3
D8	**3.2**	7.45	10.65	11.35	4.05	4.2	4.2	5.75

Appendix B. Results from the statistical tests

The bolded values show that there is a significant difference (*p*-value < 0.05)

Table 13. The *p*-value results of BVPL against the 10 first metaheuristics

Dataset	GOA	PSO	ABC	ACO	ALO	ASO	BA	DE	DF	FA
D1	**1.57×10^{-6}**	**6.19×10^{-4}**	**6.14×10^{-9}**	**3.96×10^{-2}**	**3.82×10^{-1}**	5.68×10^{-2}	**1.77×10^{-2}**	5.86×10^{-1}	**5.18×10^{-4}**	**1.28×10^{-6}**
D2_S	**1.67×10^{-4}**	**1.36×10^{-3}**	**1.59×10^{-6}**	2.91×10^{-1}	**3.33×10^{-2}**	**3.44×10^{-2}**	**1.51×10^{-4}**	8.63×10^{-2}	**3.88×10^{-3}**	**5.40×10^{-5}**
D2_M	**3.65×10^{-5}**	**4.54×10^{-4}**	**8.16×10^{-5}**	4.90×10^{-1}	**2.42×10^{-3}**	3.92×10^{-1}	**1.28×10^{-4}**	**6.79×10^{-4}**	**7.32×10^{-3}**	**1.65×10^{-6}**
D3_S	**5.20×10^{-4}**	**9.69×10^{-6}**	**2.91×10^{-8}**	7.41×10^{-1}	**3.45×10^{-2}**	**6.08×10^{-3}**	**6.03×10^{-4}**	1.34×10^{-2}	**1.07×10^{-3}**	**5.26×10^{-8}**
D3_M	**3.17×10^{-4}**	**1.52×10^{-7}**	**1.78×10^{-6}**	3.61×10^{-1}	**3.42×10^{-4}**	**3.69×10^{-2}**	**2.90×10^{-5}**	**2.66×10^{-3}**	**3.19×10^{-4}**	**5.54×10^{-7}**
D4	**4.17×10^{-9}**	**2.70×10^{-12}**	**1.86×10^{-11}**	**1.77×10^{-2}**	**3.81×10^{-8}**	**3.13×10^{-8}**	**3.47×10^{-6}**	**3.48×10^{-6}**	**7.72×10^{-9}**	**6.74×10^{-8}**
D5	**1.75×10^{-6}**	**8.47×10^{-10}**	**2.41×10^{-10}**	**4.78×10^{-3}**	**2.83×10^{-6}**	**1.84×10^{-5}**	**6.11×10^{-3}**	**4.37×10^{-6}**	**2.43×10^{-9}**	**6.98×10^{-10}**
D6	**2.05×10^{-5}**	**2.17×10^{-8}**	**8.26×10^{-8}**	7.42×10^{-1}	4.66×10^{-1}	**2.01×10^{-6}**	**7.37×10^{-5}**	**2.52×10^{-3}**	**4.47×10^{-9}**	**2.38×10^{-8}**
D7	6.75×10^{-1}	4.33×10^{-1}	**2.58×10^{-3}**	**5.19×10^{-8}**	**1.12×10^{-2}**	1.72×10^{-1}	6.97×10^{-2}	**3.12×10^{-2}**	4.41×10^{-1}	**1.98×10^{-3}**
D8	**1.44×10^{-6}**	**3.91×10^{-8}**	**1.06×10^{-7}**	3.97×10^{-1}	7.25×10^{-1}	**2.56×10^{-5}**	**1.29×10^{-4}**	**1.43×10^{-3}**	**1.09×10^{-7}**	**1.96×10^{-7}**
+/-/=	9/1/0	9/1/0	10/0/0	4/6/0	8/2/0	7/3/0	9/1/0	7/3/0	9/1/0	10/0/0

Table 14. The *p*-value results of BVPL against the 10 last metaheuristics

Dataset	GWO	HHO	MFO	TGA	TLBO	WOA	EOA	GA	SCA	SSA
D1	5.19×10^{-1}	1.32×10^{-1}	**1.87×10^{-3}**	2.17×10^{-1}	**4.39×10^{-5}**	3.63×10^{-1}	**3.08×10^{-3}**	9.24×10^{-1}	**4.69×10^{-4}**	**2.04×10^{-6}**
D2_S	7.95×10^{-2}	**1.22×10^{-2}**	**3.82×10^{-2}**	4.65×10^{-1}	**6.54×10^{-6}**	6.33×10^{-1}	**9.32×10^{-4}**	5.70×10^{-1}	**2.31×10^{-4}**	**3.36×10^{-7}**
D2_M	**4.65×10^{-3}**	**5.07×10^{-3}**	**3.87×10^{-5}**	**1.27×10^{-2}**	**3.89×10^{-4}**	**1.23×10^{-2}**	**2.89×10^{-3}**	5.81×10^{-2}	**1.08×10^{-4}**	**4.43×10^{-7}**
D3_S	3.09×10^{-1}	2.53×10^{-1}	**1.08×10^{-2}**	**2.32×10^{-1}**	**3.12×10^{-3}**	1.78×10^{-1}	**6.14×10^{-3}**	2.00×10^{-1}	**4.44×10^{-3}**	**8.29×10^{-8}**
D3_M	2.28×10^{-1}	2.82×10^{-1}	6.77×10^{-2}	9.58×10^{-2}	**1.28×10^{-4}**	1.68×10^{-1}	**5.09×10^{-5}**	1.43×10^{-1}	**2.39×10^{-2}**	**5.16×10^{-7}**
D4	**1.03×10^{-6}**	**2.73×10^{-5}**	**3.92×10^{-7}**	**7.60×10^{-6}**	**4.60×10^{-10}**	**1.53×10^{-5}**	**5.45×10^{-6}**	**6.14×10^{-9}**	**1.49×10^{-5}**	**6.77×10^{-10}**
D5	**3.25×10^{-3}**	**5.16×10^{-6}**	**1.46×10^{-4}**	**1.79×10^{-5}**	**3.24×10^{-8}**	**7.76×10^{-5}**	**7.94×10^{-6}**	**6.86×10^{-7}**	**6.23×10^{-6}**	**2.86×10^{-8}**
D6	**1.36×10^{-2}**	**1.23×10^{-3}**	**3.18×10^{-3}**	**1.22×10^{-3}**	**3.75×10^{-5}**	**7.54×10^{-4}**	**6.80×10^{-3}**	**1.23×10^{-3}**	**3.96×10^{-2}**	**2.38×10^{-6}**
D7	**1.15×10^{-4}**	**3.56×10^{-4}**	**4.10×10^{-2}**	**9.28×10^{-6}**	1.37×10^{-1}	**7.27×10^{-6}**	2.21×10^{-1}	**1.25×10^{-2}**	5.98×10^{-2}	5.48×10^{-2}
D8	**7.40×10^{-3}**	**4.19×10^{-4}**	1.46×10^{-1}	**4.76×10^{-8}**	**1.76×10^{-8}**	1.29×10^{-1}	**1.33×10^{-2}**	**3.17×10^{-4}**	**4.71×10^{-2}**	**2.67×10^{-6}**
+/-/=	6/4/0	7/3/0	8/2/0	7/3/0	9/1/0	5/5/0	9/1/0	5/5/0	9/1/0	9/1/0

Received: 05.09.2023; Second Version:07.11.2023; Accepted: 15.11.2023