A hybrid Modified Black Widow Optimization and PSO Algorithm: Application in Feature Selection for Cognitive Radio Networks

Sarra Ben Chaabane*, Kais Bouallegue†, Akram Belazi‡, Sofiane Kharbech*, and Ammar Bouallegue*
* Laboratory Sys'Com-ENIT (LR-99-ES21), Tunis El Manar University, Tunis 1002, Tunisia
† Univ Rennes, INSA Rennes, CNRS, IETR - UMR 6164, Rennes, France
‡ Laboratory RISC-ENIT (LR-16-ES07), Tunis El Manar University, Tunis 1002, Tunisia
sarra.benchaabane@enit.utm.tn

***Abstract*—In spectrum sensing issues, like in any other classification problem, the performance of the classification task is significantly impacted by the feature selection. This paper proposes a new hybrid optimization algorithm to optimize feature selection for a Deep Neural Network (DNN) classifier. To surpass the premature convergence problem and improve the exploitation ability of the original Black Widow Optimization Algorithm (BWO), we mix a modified version of BWO and Particle Swarm Optimization (PSO), called MBWPSO. The aim is to enhance the performance of a blind spectrum sensing approach in the context of cognitive radio (CR) for wireless communications. Computer simulations show that the MBWPSO algorithm outperforms the original one and a set of state-of-the-art algorithms (i.e., HS, BBO, PSO, and SA) algorithms. The MBWPSO also exhibits the best performance once applied for feature selection in the above context.**

***Index Terms*—BWO optimization algorithm, deep learning, feature selection, spectrum sensing, wireless communications.**

I. Introduction

Feature selection is an essential preprocessing technique in learning applications, eliminating irrelevant and redundant features [1], [2]. So, it helps to reduce dimensionality and improve the accuracy of classification in a dataset. Therefore, many different search techniques have been proposed to feature selection, such as metaheuristic algorithms. Metaheuristic algorithms have achieved competitive results when solving optimization problems, including feature selection [3]–[5]. There are three basic categories of these algorithms: physics-based [6], evolutionary-based [7], and swarm-based [8]. Black Widow Optimization algorithm (BWO) is a recently swarm-based optimization algorithm that was proposed by Hayyolalam and Kazem [9]. In this paper, we propose a hybrid modified version of BWO and PSO, referred to as MBWPSO, that later on will be utilized for the feature selection.

The main contributions of this paper are two-fold: (i) to overcome premature convergence of BWO and evolution stagnation, we boost up the mutation part of the BWO algorithm into the direction of average mutation and the modified version is named as MBWO; (ii) we merge the MBWO with the PSO algorithm, which requires a perfect balancing between exploration and exploitation to improve the accuracy of the solution.

In order to validate its effectiveness, the proposed algorithm is compared first, with a variety of most popular optimization algorithms, the original version, i.e., BWO, Particle Swarm Optimization (PSO) [10], Biogeography-Based Optimization (BBO) [11], Harmony Search (HS) [12], and Simulated Annealing (SA) [13] on common benchmark functions before evaluating them for feature selection example to give a better misclassification rate for the studied classification problem. The classification problem that we considered in this paper is blind spectrum sensing in Cognitive Radio (CR) networks, based on the deep learning model. Spectrum Sensing (SS) is the technique of monitoring a specific wireless communication frequency band, aiming to detect the presence or absence of primary users [14], [15]. As features, the introduced approach uses the eigenvalues of the covariance matrix of received signals [16], [17].

The rest of the paper is structered as follows. In Section II, we formulate the optimization problem. Section III describes the original BWO algorithm and introduces our motivation and improvements. In Section IV, the proposed algorithm is evaluated by twelve optimization benchmark functions. The performance of the MBWPSO algorithm in feature selection is discussed in Section V. Finally, Section VI summarizes the main findings of this study and suggests directions for future research.

II. Problem Formulation

In order to describe our approach and validate its performance for real-world applications, the MBWPSO algorithm is applied for feature selection in the context of the SS paradigm in CR networks. SS model is the one employed in [16]. The model is based on the eigenvalues of the covariance matrix of the received signal as input features. In CR context, we define two main hypotheses noted $\mathcal{H}_1$ when the Primary User (PU) is present and $\mathcal{H}_0$ when the frequency resource is vacant. Deep Neural Network (DNN) is performed for binary classification, making it a good candidate for the SS paradigm ($\mathcal{H}_0$ or $\mathcal{H}_1$). We applied the DNN Feed-forward model [18]. The feed-forward model is the simplest form of the neural network as information is only processed in one direction. While the data

978-1-6654-9927-9/22/$31.00 ©2022 IEEE

may pass through multiple hidden nodes, it always moves in one direction and never backward. The inputs of the DNN are the the eigenvalues of the covariance matrix of the received signal. The layers include three hidden layers and one output representing the class label ($\mathcal{H}_0$ or $\mathcal{H}_1$). Hence, the entire objective is to find the optimal feature set denoted $\{\mathcal{F}\}^*$ that gives the minimum misclassification rate.

It can be formulated as the optimization problem shown as:

$$\boldsymbol{f}^* = \arg\min_{\boldsymbol{f}} \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} [\![y_i \neq \mathcal{D}(\boldsymbol{f})]\!], \tag{1}$$

where $\boldsymbol{f}$ is the selected features vector, the Iverson bracket $[\![.]\!]$ is a function that takes a truth value inside and returns 1 or 0 accordingly, $|.|$ denotes the cardinality of the dataset $\mathcal{S}$. y present the true class. $\mathcal{D}$ is a function that returns the output class of the DNN classifier applying the f vector.

III. Improvements on Black Widow Optimization Algorithm

A. BWO, the Basic Algorithm

BWO is inspired by the black widow, which is a female spider. Black widows spin at night. To attract the males, black widows determine some specific locations of the net. Black widows eat the male partner. Then, they move the laid eggs to a nearby location. The offsprings struggle with each other for life. The offsprings stay with the parents for a period. In this period, they may even eat their parent. Like evolutionary population algorithms, BWO has an initial population. Each solution is considered as a widow. The objective function is computed for each candidate solution. To start the optimization process, an initial population of spiders in a matrix should be defined. Then, the pairs of parents are chosen from the initial population matrix to provide the offsprings in the procreation stage. The procreation process is represented as follows:

$$\begin{cases} \mathbf{y}_1 = \alpha \mathbf{x}_1 + (1-\alpha)\,\mathbf{x}_2 \\ \mathbf{y}_2 = \alpha \mathbf{x}_2 + (1-\alpha)\,\mathbf{x}_1, \end{cases} \tag{2}$$

where $\mathbf{x}_1$ and $\mathbf{x}_1$ are parents, α presents procreate rate, $\mathbf{y}_1$ and $\mathbf{y}_2$ are offspring.

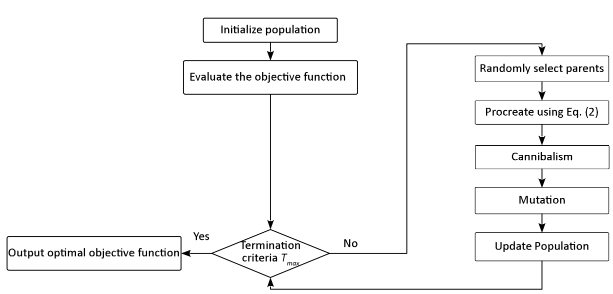


Fig. 1: Flowchart of the BWO algorithm.

The next stage is cannibalism. The black widow eats its male partner. The BWO uses the cannibalism rate to eliminate weak solutions from the search space. Then, the mutation is applied, each solution randomly exchanges two elements. Then, the newly produced member is stored, and the initial population of BWO is updated. Fig. 1 depicts the flowchart of BWO algorithm.

B. Motivation and Improvements

The original BWO algorithm has population initialization, procreate, cannibalism, and mutation steps. The mutation rule is essential in the BWO algorithm to generate optimal convergence speed. But, the performance of the mutation stage is deficient. So, to boost up this mutation step, increase the convergence performance of the BWO algorithm, and avoid trapping, we propose to apply the average direction strategy discussed in [19]. The mutation rules are performed via three ingredients, such as the kth generation with the current solution in the population $\mathbf{x}_i[k]$, the kth generation with the best solution in the population $\mathbf{x}_b[k]$, and the kth generation with randomly chosen individuals $\mathbf{x}_r[k]$. The former two ingredients are deterministic, while the last one is stochastic. Mutation rules are better constructed via balancing the above three ingredients. For better comparable fitness value, the surviving individual in the k generation is contrasted to the $(k-1)$ generation. So, the newly developed algorithm is referred to as the Modified Black Widow Optimization (MBWO) algorithm. The below equation illustrates the averaged mutation rule with a devised direction.

$$\begin{aligned} \mathbf{x}_i[k+1] = \mathbf{x}_i[k] + m_1\,(\mathbf{x}_{avg}[k] - \mathbf{x}_i[k-1]) + m_2 \\ \times (\mathbf{x}_{r1}[k] - \mathbf{x}_{r2}[k]) \end{aligned} \tag{3}$$

Where $\mathbf{x}_{avg}[k]$ represents the best first individuals in the kth generation. $\mathbf{x}_{r1}[k]$ and $\mathbf{x}_{r1}[k]$ are two random individuals selected from the current generation.The parameters m_1, and m_2 fix the interval range [0.5, 1]

C. PSO Optimization Algorithm

The Particle Swarm Optimization algorithm (PSO) was discovered by James Kennedy and Russell C. Eberhart [10]. This algorithm is inspired by simulation of social psychological expression of birds and fishes. The velocity of the $i-th$ particle is defined as the change of its position. The position and velocity vectors of all particles are generated randomly. Then, each particle moves in the design space using the best position experienced by that particle ($\mathbf{x}^{pBest}$) and the best solution obtained by all particles ($\mathbf{x}^{gBest}$). In each iteration, the swarm is updated by the following equations:

$$\begin{aligned} \mathbf{v}_i[k+1] = w\mathbf{v}_i[k] + c_1 r_1 \left(\mathbf{x}_i^{pBest} - \mathbf{x}_i[k]\right) + c_2 r_2 \\ \times \left(\mathbf{x}^{gBest} - \mathbf{x}_i[k]\right), \end{aligned} \tag{4}$$

$$\mathbf{x}_i[k+1] = \mathbf{x}_i[k] + \mathbf{v}_i[k+1], \tag{5}$$

where $\mathbf{v}_i[k]$ is velocity of the particle at iteration k, $\mathbf{x}_i[k]$ is the position of particles at iteration k. w presents the inertia weight. r_1 and r_2 are random number in the interval [0, 1]. c_1 and c_2 are the acceleration coefficients.

D. Proposed Hybride MBWO and PSO Algorithm

Both MBWO and PSO algorithms contain many advantages, but they have a few difficulties. Generally, strong local exploration is presented in MBWO, but the performance of global exploitation is poor. Fig. 2(a) and Fig. 3(a) show the evaluated solutions using the MBWO algorithm for optimizing $f5$ and $f9$ (two selected benchmark functions [9]), respectively. Their correspondent convergence visualizations are plotted in Fig. 2(b) and Fig. 3(b), respectively. As shown, the algorithm fails to reach the best solution (i.e., the $(0, 0)$ pair) and cannot converge to the global optima.

To overcome these issues, we combine the MBWO with the PSO algorithm, increasing the convergence performance and leading to optimal results. The combination of MBWO and PSO algorithm is named as Modified Black Widow PSO (MBWPSO) algorithm. Fig. 4 depicts the flowchart of MBWPSO algorithm. The entire population is divided into two stages (i.e., PSO and MBWO algorithm stages). MBWO algorithm is used for the exploration phase. The exploration phase means the capability of the algorithm to try out a large number of possible solutions. The position of particle responsible for finding the optimum solution of the complex nonlinear problem is replaced with the position of Black Widow. MBWO directs the particles faster toward optimal value, as illustrated in (6). The best individuals are shared by combining the MBWO and PSO algorithms stages and forming a new population. The Hybridization of MBWO with PSO merges the best strength of both PSO in exploitation and BWO in the exploration phase to obtain the best possible solution to the problem that avoids local stagnation. Fig. 2(c) and Fig. 3(c) show the evaluated solutions using the MBWPSO algorithm for optimizing $f5$ and $f9$, respectively, their correspondent convergence visualizations are plotted in Fig. 2(d) and Fig. 3(d), respectively. As can be seen from these figures, the MBWPSO achieved a satisfactory level in leading to the best solution. Moreover, it is robust in converging toward the global optima

$$\mathbf{v}_i[k+1] = w\mathbf{v}_i[k] + c_1 r_1 \left(\mathbf{x}_{MBWO}[k] - \mathbf{x}_i[k]\right) + c_2 r_2 \times \left(\mathbf{x}_{gBest} - \mathbf{x}_i[k]\right), \quad (6)$$

IV. Benchmarking of BWO and MBWPSO Algorithms

The performance of the proposed MBWPSO algorithm is tested by solving 12 benchmark functions under dimension 30 (dimension of agent) reported in [9]. These are grouped into unimodal functions ($f_1 - f_6$) with one global optimum and multimodal functions ($f_7 - f_{12}$) with many local optima. For all tests, the population size is set to 40. In addition to the original BWOA algorithm, the proposed MBWPSO algorithm is compared with BBO, HS, PSO, and SA algorithms. For all computer simulations of this paper, we used Matlab version 9.9.0.1538559 (R2020b) Update 3.

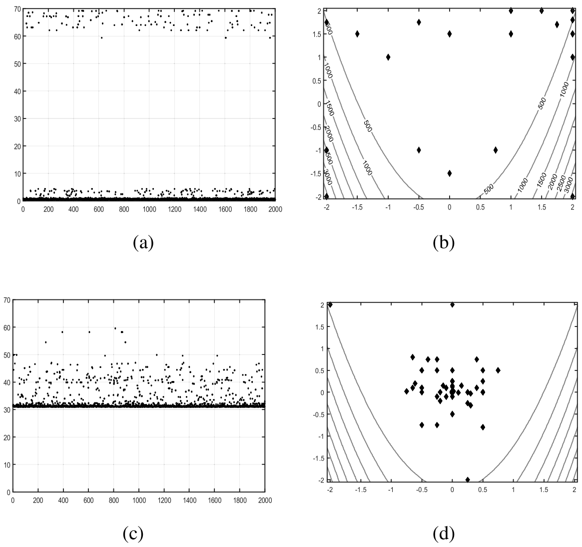


Fig. 2: Average best solution (i.e., the achieved minimum, described by the y-axis) in terms of the number of objective function evaluations (x-axis) for the benchmark function f_5 using (a) MBWO and (c) MBWPSO; corresponding convergence visualizations (b,d) in accordance with (a,c). (b,d) are the contour plots of the function f_5, the x and y-axis represent the dimensions of the function, and the markers represent the evaluated points by running the optimization algorithms.

A. Comparison Based on Solution Accuracy

Table I describes the performance of the MBWPSO through the best mean values (Mean), the standard deviations (SD), and the standard errors of means (SEM). The unimodal functions ($f_1 - f_6$) allow evaluating the exploitation capability of the studied meta-heuristic algorithms. In most of these functions, MBWPSO is the best optimizer and successfully reaches the global optima. The present algorithm can hence provide perfect exploitation. Nevertheless the unimodal functions, the multimodal functions ($f_7 - f_{12}$), include many local optima. Therefore, this kind of test function is beneficial to evaluate a given algorithm's exploration capability. From the reported results, we can conclude that the proposed algorithm achieves efficient exploitation and exploration for the tested benchmark functions.

B. Comparison Based on Convergence

The convergence rates of the comparative algorithms are listed in Table II. These rates are estimated using the mean number of function evaluations (MeanFES) and the success rate (SR). For most benchmark functions, MBWPSO presents the highest SR and the lowest MeanFES required to reach an acceptable solution. Except for (f_6, f_{11}, and f_{12}) functions, despite the difficulty of these multimodal functions to converge, MBWPSO nearly keeps the same values as the original BWOA. For f_6, the PSO has the best convergence speed. Fig.

mation and achieve more immediate stability for the above benchmark functions.

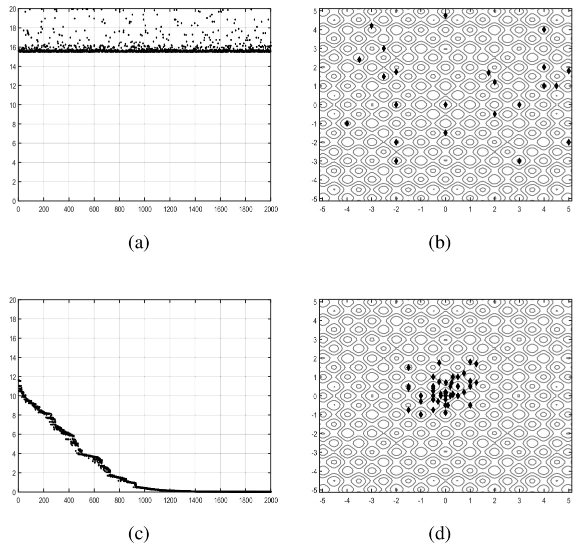


Fig. 3: Average best solution (i.e., the achieved minimum, described by the y-axis) in terms of the number of objective function evaluations (x-axis) for the benchmark function f_9 using (a) MBWO and (c) MBWPSO; corresponding convergence visualizations (b,d) in accordance with (a,c). (b,d) are the contour plots of the function f_9, the x and y-axis represent the dimensions of the function, and the markers represent the evaluated points by running the optimization algorithms.

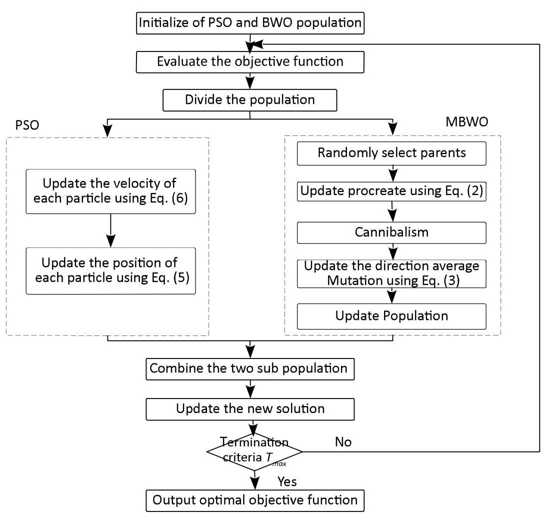


Fig. 4: Flowchart of the MBWPSO algorithm.

5 shows the convergence curves of the tested algorithms for the benchmark functions used to manifest the convergence performance more intuitively. The present algorithm required the fastest convergence speed and the highest convergence precision for most test functions compared to other algorithms. The MBWPSO algorithm can search for optimal approximation and achieve more immediate stability for the above benchmark functions.

C. *Statistical tests*

To prove the overall performance of a given algorithm, statistical tests are required. Since, in our experiments, non-parametric statistical tests should be provided to compare the comparative algorithms quantitatively. As a result, we applied the Friedman and the Quade tests [20], [21]. Fig. 6 shows the average rankings of the tested algorithms based on the standard errors of means (SEM). As it is shown in this figure, MBWPSO is the best ranked. The proposed algorithm is very efficient in balancing exploitation and exploration of the search space and surmounts the original BWO algorithm in solving the benchmark functions.

V. MBWPSO FOR FEATURE SELECTION IN SS

In this section, we measure the performance of the proposed algorithm in providing the best features for SS in CR networks using a DNN classifier. SS model is the one used in [16]. In this model, the input features are the eigenvalues of the covariance matrix of the received signal. For all tests, the number of PU, the number of antennas, the number of observed samples, and the number of trials are set to 1, 20, 250, and 500, respectively. Hence, the training set is set to 20 features and 1000 samples (500 for each class $\mathcal{H}_0$ or $\mathcal{H}_1$). Table III illustrates the solution accuracy for all algorithms. One can see that MBWPSO achieves the best mean. Table IV notes that the proposed algorithm based on the feature selection approach remains the best (minimum) misclassification rate in differents SNR. Confusion matrices illustrated in Fig. 8 and Fig. 7 give in-depth results and ensure that the proposed system is still the most efficient method, compared to the original, in the considered context.

TABLE I: Mean, SEM and SD for functions $f_1 - f_{12}$.

		BBO	HS	PSO	SA	BWO	MBWPSO
f_1	Mean	2.70e-09	6.86e-13	**0.00e+00**	3.78e-42	0.00019	**0.00e+00**
	SD	**0.00e+00**	4.73e-13	**0.00e+00**	1.35e-42	0.0004	**0.00e+00**
	SEM	**0.00e+00**	8.63e-14	**0.00e+00**	2.42e-43	7.53e-05	**0.00e+00**
f_2	Mean	50.168	2581.3	0.00012	2.22e-21	0.0001	**2.40e-280**
	SD	**0.00e+00**	710.76	0.0006	1.2201e-20	0.0004	**0.00e+00**
	SEM	**0.00e+00**	129.76	0.00012	2.22e-21	7.89e-05	**0.00e+00**
f_3	Mean	23.7669	2735.4	4.20e-06	1.51e-39	772.83	**0.00e+00**
	SD	**0.00e+00**	855.06	7.30e-06	6.15e-40	402.50	**0.00e+00**
	SEM	**0.00e+00**	156.11	1.33e-06	1.12e-40	73.487	**0.00e+00**
f_4	Mean	6.30e-08	6.24e-13	**0.00e+00**	2.82e-43	7.0964	**3.37e-271**
	SD	**0.00e+00**	2.70e-13	**0.00e+00**	9.84e-44	6.5918	**0.00e+00**
	SEM	**0.00e+00**	4.94e-14	**0.00e+00**	1.79e-44	1.2035	**0.00e+00**
f_5	Mean	77.138	357.99	30.675	96.562	2327.2	**27.461**
	SD	**0.00e+00**	406.82	26.444	161.78	4100.6	**0.5359**
	SEM	**0.00e+00**	74.276	4.8281	29.538	748.66	**0.0978**
f_6	Mean	0.1295	2.5113	2.96e-15	**0.00e+00**	1.64e-06	1.2144
	SD	**0.00e+00**	1.5099	1.58e-14	**0.00e+00**	5.13 e-06	0.6027
	SEM	**0.00e+00**	0.27568	2.89e-15	**0.00e+00**	9.37e-07	0.11005
f_7	Mean	-8341.6	**-12568.5**	-6984.9	-10475.5	-11357.4	-5033.7
	SD	**0.00e+00**	0.8838	849.335	345.46	929.45	1300.8
	SEM	**0.00e+00**	0.1613	155.06	63.072	169.694	237.5
f_8	Mean	53.854	0.6166	51.439	34.657	0.0008	**0.00e+00**
	SD	**0.00e+00**	15.392	8.4992	345.46	0.0034	**0.00e+00**
	SEM	**0.00e+00**	2.8102	1.5517	63.0728	0.0006243	**0.00e+00**
f_9	Mean	0.1281	0.4766	1.0305	7.87e-15	0.0106	**1.24e-15**
	SD	**0.00e+00**	0.2000	0.8714	6.48e-16	0.0530	**1.084e-15**
	SEM	**0.00e+00**	0.0365	0.1591	1.18e-16	0.0095	**1.97e-16**
f_{10}	Mean	0.1557	0.9832	0.0220	0.0058	0.1839	**0.00e+00**
	SD	**0.00e+00**	0.1034	0.0255	0.0086	0.2373	**0.00e+00**
	SEM	**0.00e+00**	0.0188	0.0046	0.0015	0.0433	**0.00e+00**
f_{11}	Mean	0.0005	0.0167	0.1904	0.0069	**1.62e-06**	0.1111
	SD	**0.00e+00**	0.0237	0.6255	0.0263	5.66e-06	0.1601
	SEM	**0.00e+00**	0.0043	0.1142	0.0048	1.034e-06	0.0292
f_{12}	Mean	0.0069	0.2118	0.0554	**0.0007**	0.0011	1.6838
	SD	**0.00e+00**	0.0726	0.1332	0.0263	0.0027	0.4946
	SEM	**0.00e+00**	0.0132	0.0243	0.0005	0.0011	0.0903
	Best for	0/12	1/12	2/12	2/12	0/12	**8/12**

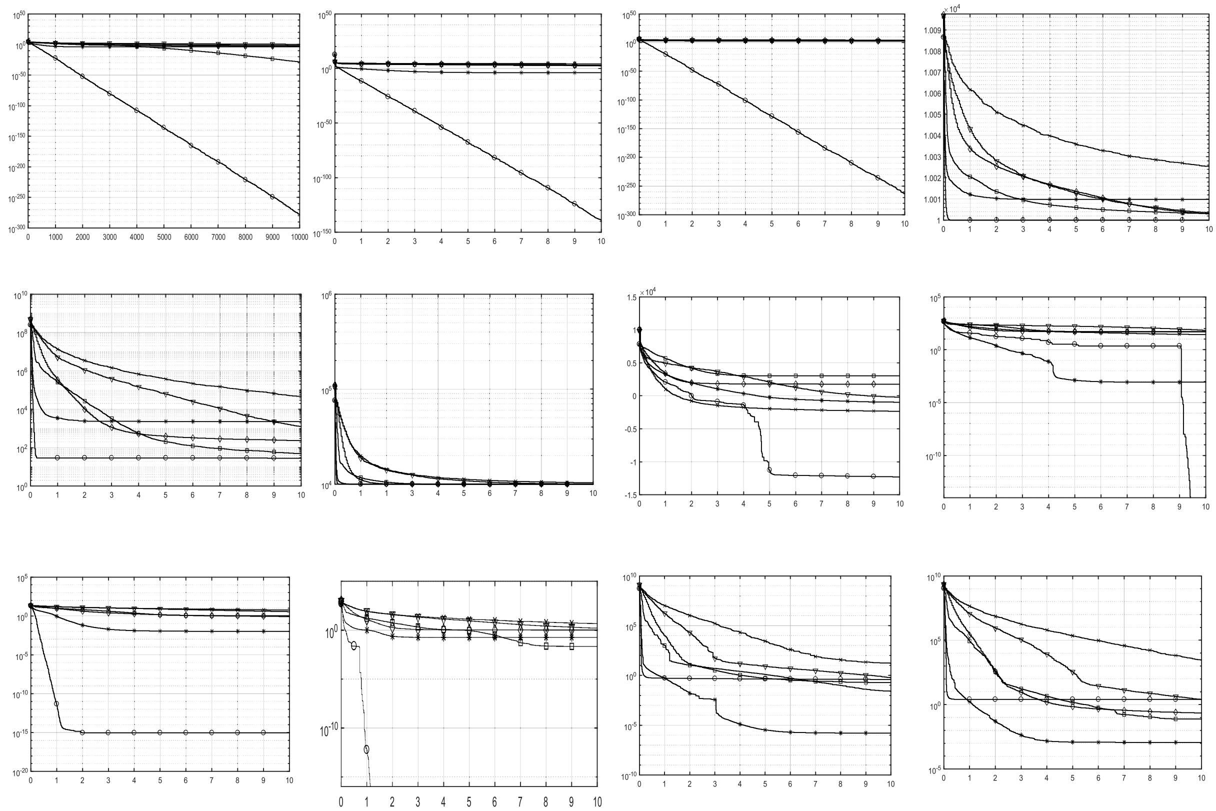
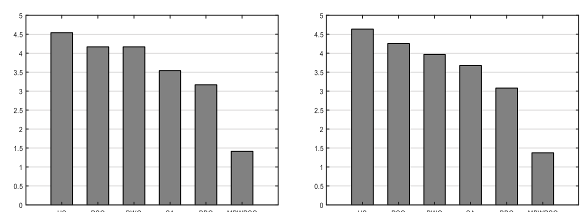


Fig. 5: Convergence results (average best solution in terms of FES‡) for the benchmark functions $f_1 - f_{12}$ in row-major order. BBO[◇], PSO[□], HS[×], SA[▽], BWO[⋆], and MBWPSO[○].
‡ The x-axis values are to be multiplied by 10^3.

TABLE II: MeanFES and SR by comparative algorithms for functions $f_1 - f_{12}$.

		BBO	HS	PSO	SA	BWO	IBWO
f_1	MeanFES	2613	7069	3058.03	15599.6	411.6	**182.3**
	SR (%)	**100**	**100**	**100**	**100**	**100**	**100**
f_2	MeanFES	NaN	NaN	85553.4	45554.3	2102.3	**287.9**
	SR (%)	0	0	96.67	**100**	**100**	**100**
f_3	MeanFES	NaN	NaN	87508.4	45331.6	NaN	**367.4**
	SR (%)	0	0	86.67	**100**	0	**100**
f_4	MeanFES	35234.3	33307.6	4624.5	28975.6	NaN	**294.3**
	SR (%)	**100**	**100**	**100**	**100**	0	**100**
f_5	MeanFES	32561	NaN	11651.3	16338.1	NaN	**171.1**
	SR (%)	20	0	80	70	0	**100**
f_6	MeanFES	NaN	NaN	**10041.7**	40970.3	3685	NaN
	SR (%)	0	0	**100**	**100**	93.3	0
f_7	MeanFES	51.4	43.4	39	23.2	**7.93**	237.5
	SR (%)	**100**	**100**	**100**	**100**	**100**	**100**
f_8	MeanFES	NaN	NaN	NaN	NaN	4316.6	**2053.2**
	SR (%)	0	0	0	0	73.3	**100**
f_9	MeanFES	NaN	NaN	14386.9	58031.6	5281.9	**491.1**
	SR (%)	0	0	33.3	**100**	56.67	**100**
f_{10}	MeanFES	NaN	NaN	10017.5	42272.1	4762.6	**373.2**
	SR (%)	0	0	30	60	16.67	**100**
f_{11}	MeanFES	NaN	NaN	8677.8	28846.7	**2989**	NaN
	SR (%)	0	0	30	93.3	**93.3**	0
f_{12}	MeanFES	NaN	NaN	10454.5	34378.1	**3821.8**	NaN
	SR (%)	0	0	43.3	93.3	**66.6**	0
Best for	MeanFES	0/12	0/12	1/12	0/12	3/12	**8/12**
	SR (%)	3/12	3/12	4/12	7/12	5/12	**9/12**

When an algorithm cannot reach an acceptable solution over the fixed number of runs, the value is marked as 'NaN'.



(a) (b)

Fig. 6: Average ranking of comparative algorithms by Friedman test (a), and Quade test (b)

VI. Conclusion

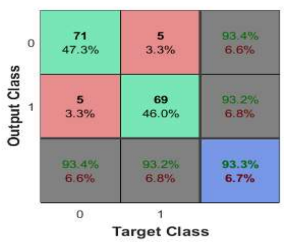
In this work, we proposed a hybrid modified version of the BWO and PSO algorithm, dubbed MBWPSO, to optimize feature selection in the context of spectrum sensing for wireless communications using a deep learning method. The MBWPSO relies mainly on the excellent balance between local and global searches. Simulation results on benchmarking functions and feature selection in Spectrum Sensing (SS), as a case study, showed that the proposed algorithm outperforms the other approaches used for comparison in terms of solution accuracy and convergence. Due to its reliability, the proposed

TABLE III: Mean, SEM and SD comparison for differents SNR.

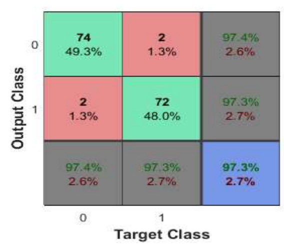
		BWO	MBWPSO
SNR=-25 dB	Mean	4.20e-1	**3.90e-1**
	SD	1.46e-2	**1.35e-2**
	SEM	4.50e-3	**4.2e-3**
SNR=-20 dB	Mean	4.00e-1	**3.80e-1**
	SD	1.39e-2	**1.30e-2**
	SEM	4.30e-3	**4.1e-3**
SNR=-15 dB	Mean	3.80e-1	**3.70e-1**
	SD	1.30e-2	**1.28e-2**
	SEM	4.10e-3	**4.00e-3**
SNR=-10 dB	Mean	6.50e-2	**3.80e-2**
	SD	1.32e-3	**2.20e-3**
	SEM	5.12e-4	**4.10e-4**

TABLE IV: Misclassification rate for differents SNR.

	SNR (dB)				
Algorithm	-25	-20	-15	-10	-5
BWO	0.42	0.40	0.38	0.065	0
MBWPSO	**0.39**	**0.38**	**0.37**	**0.038**	**0**

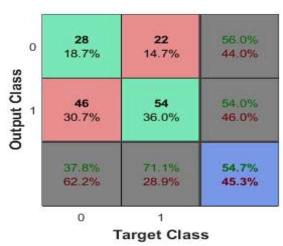


(a)

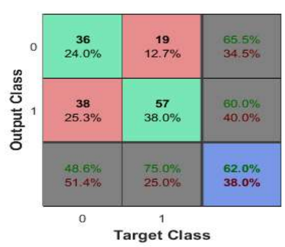


(b)

Fig. 7: Confusion matrices of the classification accuracy (SNR=-10 dB) given by BWO (a), MBWPSO (b)



(a)



(b)

Fig. 8: Confusion matrices of the classification accuracy (SNR=-25 dB) given by BWO (a), MBWPSO (b)

approach can be applied to railway environments with high-Speed channels and impulsive noise.

References

[1] J. Miao and L. Niu, "A survey on feature selection," *Procedia Computer Science*, vol. 91, pp. 919–926, 2016.

[2] T. Hamed, R. Dara, and S. C. Kremer, "An accurate, fast embedded feature selection for svms," in *2014 13th International Conference on Machine Learning and Applications*. IEEE, 2014, pp. 135–140.

[3] S. Pourbahrami, "Improving pso global method for feature selection according to iterations global search and chaotic theory," *arXiv preprint arXiv:1811.08701*, 2018.

[4] S. Ben Chaabane, A. Belazi, S. Kharbech, A. Bouallegue, and L. Clavier, "Improved salp swarm optimization algorithm: Application in feature weighting for blind modulation identification," *Electronics*, vol. 10, no. 16, pp. 2002, 2021.

[5] S. B. Chaabane, S. Kharbech, A. Belazi, and A. Bouallegue, "Improved whale optimization algorithm for svm model selection: Application in medical diagnosis," in *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2020, pp. 1–6.

[6] A. Biswas, K. Mishra, S. Tiwari, and A. Misra, "Physics-inspired optimization algorithms: a survey," *Journal of Optimization*, vol. 2013, 2013.

[7] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary computation*, vol. 3, no. 1, pp. 1–16, 1995.

[8] R. S. Parpinelli and H. S. Lopes, "New inspirations in swarm intelligence: a survey," *International Journal of Bio-Inspired Computation*, vol. 3, no. 1, pp. 1–16, 2011.

[9] V. Hayyolalam and A. A. P. Kazem, "Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 87, pp. 103249, 2020.

[10] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. IEEE, 1995, pp. 39–43.

[11] D. Simon, "Biogeography-based optimization," *IEEE transactions on evolutionary computation*, vol. 12, no. 6, pp. 702–713, 2008.

[12] B. Alatas, "Chaotic harmony search algorithms," *Applied mathematics and computation*, vol. 216, no. 9, pp. 2687–2699, 2010.

[13] E. Aarts, J. Korst, and W. Michiels, "Simulated annealing. ek burke y g. kendall, editores, search methodologies," 2005.

[14] Y.-L. Foo, "Sensing criteria of tv white space devices under nakagami-rice fading," in *2019 25th Asia-Pacific Conference on Communications (APCC)*. IEEE, 2019, pp. 32–35.

[15] L. L. Nguyen, A. Fiche, R. Gautier, and E. Radoi, "Impacts of the transmitter signal key parameters on the compressed sensing spectrum reconstruction for iot cognitive radio applications," in *2019 25th Asia-Pacific Conference on Communications (APCC)*. IEEE, 2019, pp. 461–466.

[16] K. Bouallegue, M. Crussiere, and S. Kharbech, "Svm assisted primary user-detection for non-cooperative cognitive radio networks," in *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2020, pp. 1–5.

[17] K. Bouallegue, M. Crussiere, and I. Dayoub, "On the impact of the covariance matrix size for spectrum sensing methods: beamforming versus eigenvalues," in *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2019, pp. 1–5.

[18] S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari, "Output range analysis for deep feedforward neural networks," in *NASA Formal Methods Symposium*. Springer, 2018, pp. 121–138.

[19] L. Hong, C. Liu, J. Cui, and F. Liu, "Mutation strategy based on step size and survival rate for evolutionary programming," *Discrete Dynamics in Nature and Society*, vol. 2021, 2021.

[20] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, May 2010.

[21] Z. Liu, E. Blasch, and V. John, "Statistical comparison of image fusion algorithms: Recommendations," *Information Fusion*, vol. 36, pp. 251–260, July 2017.