

Article

A New Co-Evolution Binary Particle Swarm Optimization with Multiple Inertia Weight Strategy for Feature Selection

Jingwei Too ^{1,*} , Abdul Rahim Abdullah ^{1,*} and Norhashimah Mohd Saad ²

¹ Fakulti Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal 76100, Melaka, Malaysia

² Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal 76100, Melaka, Malaysia; norhashimah@utem.edu.my

* Correspondence: jamesjames868@gmail.com (J.T.); abdul@utem.edu.my (A.R.A.); Tel.: +60-178-845-938 (J.T.)

Received: 11 March 2019; Accepted: 6 May 2019; Published: 8 May 2019



Abstract: Feature selection is a task of choosing the best combination of potential features that best describes the target concept during a classification process. However, selecting such relevant features becomes a difficult matter when large number of features are involved. Therefore, this study aims to solve the feature selection problem using binary particle swarm optimization (BPSO). Nevertheless, BPSO has limitations of premature convergence and the setting of inertia weight. Hence, a new co-evolution binary particle swarm optimization with a multiple inertia weight strategy (CBPSO-MIWS) is proposed in this work. The proposed method is validated with ten benchmark datasets from UCI machine learning repository. To examine the effectiveness of proposed method, four recent and popular feature selection methods namely BPSO, genetic algorithm (GA), binary gravitational search algorithm (BGSA) and competitive binary grey wolf optimizer (CBGWO) are used in a performance comparison. Our results show that CBPSO-MIWS can achieve competitive performance in feature selection, which is appropriate for application in engineering, rehabilitation and clinical areas.

Keywords: feature selection; classification; binary particle swarm optimization; inertia weight; wrapper; binary optimization

1. Introduction

Various pattern recognition studies have shown that a proper selection of features can lead to satisfactory classification performance. However, it is difficult to determine which feature is relevant due to the lack of experience and prior knowledge [1–3]. In addition, a weak feature (a feature that contributes low classification accuracy) might be able to enhance the classification performance when it is combined with other potential features. Moreover, the selection of features is considered as a non-deterministic polynomial-time (*NP*) hard combinatorial problem, where the number of possible solutions increases exponentially with the number of features. Hence, an exhaustive search is impractical [4]. In fact, a feature set with large number of features not only introduces the extra computational complexity, but also significantly degrades the performance of system. Therefore, the feature selection process is critically important for classification tasks.

Feature selection is a technique that aims to find a subset of input features, through which can improve or maintain the classification accuracy [5]. Generally, feature selection can be categorized into filter and wrapper approaches. The former is based on statistical, information theory, distance measurement and intrinsic characteristic of the data. By contrast, the latter evaluates the best combination of features (feature subset) by optimizing the classification performance. As compared to

the wrapper approach, the filter approach does not involve any specific learning algorithm in the process of evaluation, which is more general than the wrapper approach. However, the wrapper approach can always achieve better classification results, which has become a major interest of researches in feature selection [6]. Thus, this study focuses on wrapper feature selection.

Recently, there are many metaheuristic algorithms that have been proposed for wrapper feature selection. Huang et al. [7] proposed a new ant colony optimization (ACO) with minimum redundancy maximum relevance criterion (mRMR) as the heuristic measurement for electromyography signals classification. The authors reported the proposed approach can offer better classification results as compared to principle component analysis (PCA) and original feature sets. Mesa et al. [8] introduced a novel mRMR with F-test Correlation Out (FCO) for channel and feature selection. In the same year, Venugopal et al. [9] applied the genetic algorithm (GA) and information gain (IG) to select the relevant features for measuring the muscle fatigue conditions. Phinyomark et al. [10] employed the sequential forward selection (SFS) for feature selection tasks. Moreover, Purushothaman and Vikas [11] made use of particle swarm optimization (PSO) and ACO to solve the feature selection problem in finger movement recognition. Another recent study proposed a new competitive binary grey wolf optimizer (CBGWO) for electromyography signals classification, which shown to be outperformed binary particle swarm optimization (BPSO), GA and binary grey wolf optimization (BGWO) in evaluating the optimal feature subset [12].

Among those feature selection methods, PSO and BPSO are the most frequently used. This is mainly due to their advantageous of simplicity and low computational complexity, which has become of major interest to researchers in feature selection studies [13,14]. However, BPSO has the limitations of premature convergence, and it is not good at avoiding the local optimal [15–17]. In addition, BPSO suffers from the setting of the inertia weight, thus leading unsatisfactory performance [18]. Therefore, Chuang et al. [13] developed an improved BPSO for gene selection. The proposed approach aimed to reset the global best solution (*gbest*) when it does improve for three iterations. Banka and Dara [19] designed a Hamming distance based BPSO with novel fitness function to tackle the high dimensional feature selection problem. Furthermore, Bharti and Singh [20] integrated the opposition based strategy, chaos theory, mutation and fitness based dynamic inertia weight into BPSO for efficient feature selection in text clustering.

In this paper, our goal was to develop a new variant of BPSO that works effectively in feature selection problems. A new feature selection method namely co-evolution binary particle swarm optimization with multiple inertia weight strategy (CBPSO-MIWS) is proposed in this work. To resolve the limitations of BPSO, two strategies are introduced in CBPSO-MIWS. The first strategy is a co-evolution concept, which partitions the population of particles into several species (sub-populations). In this way, the particles can share information within different species, and this increases the global search capability. The second strategy is a multiple inertia weight strategy, which promotes the use of multiple inertia weight schemes in each species iteratively. Since multiple species are involved in CBPSO-MIWS, each species can perform the search with different inertia weight schemes, which is good at improving diversity. The proposed CBPSO-MIWS was tested with 10 benchmark datasets collected from the UCI machine learning repository. In order to examine the efficiency and efficacy of proposed CBPSO-MIWS, four recent and popular feature selection methods include BPSO, genetic algorithm (GA), binary gravitational search algorithm (BGSA) and CBGWO were used in performance comparison. The experimental result showed that CBPSO-MIWS had promising performance in most of the datasets.

The remainder of this paper is organized as follows: Section 2 details the standard binary particle swarm optimization. Section 3 briefly describes the proposed CBPSO-MIWS and its application for feature selection. Section 4 presents the experimental results. The discussions are shown in Section 5. At last, Section 6 concludes the findings of this work.

2. Binary Particle Swarm Optimization

Binary particle swarm optimization (BPSO) is a binary version of particle swarm optimization (PSO) that has been proposed to solve the binary optimization tasks [21]. Like PSO, BPSO involves the personal best (*pbest*) and global best (*gbest*) solutions in the velocity and position update. For each particle (solution), the velocity is updated as [13,22]:

$$v_i^d(t+1) = wv_i^d(t) + c_1r_1(pbest_i^d(t) - x_i^d(t)) + c_2r_2(gbest^d(t) - x_i^d(t)) \quad (1)$$

where v is the velocity, x is the solution (position of particle), w is the inertia weight, c_1 and c_2 are the acceleration factors, r_1 and r_2 are two independent random numbers in $[0,1]$, *pbest* is the personal best solution, *gbest* is the global best solution for the entire population, i is the order of particle in the population, d is the dimension of search space, and t is the number of iterations. Note that the velocity is bounded by the maximum velocity, v_{max} and minimum velocity, v_{min} . In this study, the v_{max} and v_{min} were set at 6 and -6 , respectively [13].

Afterward, the velocity is converted into probability value using Equation (2), and the position of particle is updated as shown in Equation (3):

$$S(v_i^d(t+1)) = \frac{1}{1 + \exp(-v_i^d(t+1))} \quad (2)$$

$$x_i^d(t+1) = \begin{cases} 1, & \text{if } rand < S(v_i^d(t+1)) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where *rand* is a random number uniformly distributed between 0 and 1. In BPSO, *pbest* and *gbest* play an important role in guiding the particle to move toward the global optimum. Considering the minimization function was applied in this paper. Iteratively, the *pbest* and *gbest* are updated as follows:

$$pbest_i(t+1) = \begin{cases} x_i(t+1), & \text{if } F(x_i(t+1)) < F(pbest_i(t)) \\ pbest_i(t), & \text{otherwise} \end{cases} \quad (4)$$

$$gbest(t+1) = \begin{cases} pbest_i(t+1), & \text{if } F(pbest_i(t+1)) < F(gbest(t)) \\ gbest(t), & \text{otherwise} \end{cases} \quad (5)$$

where x is the solution, *pbest* is the personal best solution, *gbest* is the global best solution for the entire population, $F(\cdot)$ is the fitness function, and t is the number of iterations.

3. Co-evolution Binary Particle Swarm Optimization with Multiple Inertia Weight Strategy

Generally speaking, BPSO is a useful optimization tool and it has been successfully applied for many feature selection tasks. However, BPSO suffers from the premature convergence and slow convergence rate [15–17]. Additionally, one of the major drawbacks of BPSO is the setting of the inertia weight [18]. In order to solve the limitations of BPSO, a new co-evolution binary particle swarm optimization with a multiple inertia weight strategy (CBPSO-MIWS) is proposed in this work.

Among birds or fishes, there can be several types of species. Instead of working on a specific species, the co-evolution between different types of species can lead to more efficient local and global search capability. In CBPSO-MIWS, the population of particles is equally divided into several sub-populations, where each sub-population is assumed to consist of one species. For example, a population of 30 particles is partitioned into three sub-populations (species), where each sub-population is comprised of 10 particles. The example is illustrated in Figure 1.

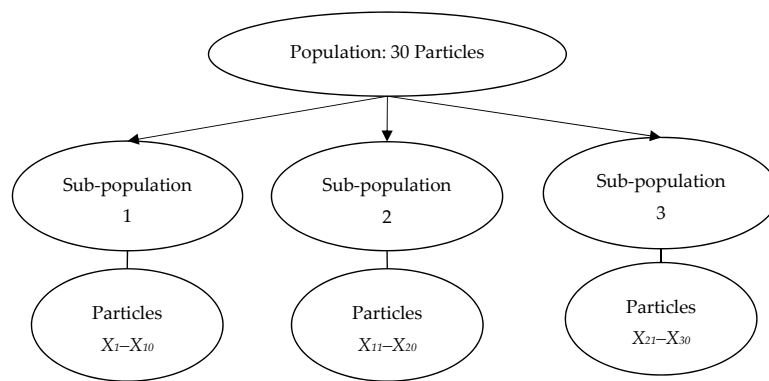


Figure 1. The example of structure of co-evolution binary particle swarm optimization with multiple inertia weight strategy (CBPSO-MIWS).

3.1. Multiple Inertia Weight Strategy

Briefly, inertia weight is one of the important parameters in BPSO, which is useful in balancing the exploration and exploitation behavior [18]. A smaller value of inertia weight ensures high exploitation. By contrast, a larger value of inertia weight guarantees high exploration. In order to achieve optimal performance, a proper balance between exploration and exploitation is critically essential. According to the literature, several types of inertia weight schemes have been proposed to enhance the performance of PSO. However, an inertia weight scheme that performs better in problem *A* might not work effectively in problem *B*. To date, there is no universal inertia weight strategy that can provide an optimal performance for all engineering problems.

To resolve the issues above, a multiple inertia weight strategy (MIWS) was introduced. MIWS consists of several inertia weight schemes, which takes advantage of different inertia weight strategies in the evaluation process. Since multiple species are involved in CBPSO-MIWS, a search with different kind of inertia weight strategies can be performed by each species, which is beneficial in enhancing the diversity and avoiding the local optimal. In other words, instead of using a fixed inertia weight strategy, each sub-population carries out the search with different inertia weight to seek out the global optimal solution. In this study, four inertia weight schemes were implemented and they are listed as follows [23–26]:

Inertia weight scheme 1 (IWS 1):

$$w = w_{\max} - (w_{\max} - w_{\min})\left(\frac{t}{T_{\max}}\right) \tag{6}$$

Inertia weight scheme 2 (IWS 2):

$$w = 0.5 + \frac{1}{2}r_3 \tag{7}$$

Inertia weight scheme 3 (IWS 3):

$$w = \left\{ \frac{(T_{\max} - t)^p}{(T_{\max})^p} \right\} (w_{\max} - w_{\min}) + w_{\min} \tag{8}$$

Inertia weight scheme 4 (IWS 4):

$$w = w_0 \tag{9}$$

where w_{\max} and w_{\min} are bound on inertia weight, r_3 is a random number uniformly distributed in $[0,1]$, p is the nonlinear modulation index, w_0 is the initial inertia weight, t is the number of iteration and T_{\max} is the maximum number of iterations. These inertia weight schemes were chosen due to their promising performances and low complexity in previous works. It is worth noting that other inertia weight schemes are also applicable in CBPSO-MIWS. As for simplicity, we only consider four inertia weight schemes in this paper.

There are several types of inertia weight schemes implemented in CBPSO-MIWS. The question is, which inertia weight scheme (IWS) should be selected for each species in the process of evaluation? Generally, it is extremely difficult to choose a proper inertia weight scheme since the optimal inertia weight might be highly depended on the data and model specification. In order to resolve this problem, we adopted a random selection strategy, which randomly selects an inertia weight scheme for each species in each iteration. Mathematically, the random selection strategy can be represented as follows:

$$IWS = \begin{cases} IWS 1, & \text{if } rand(1,4) = 1 \\ IWS 2, & \text{if } rand(1,4) = 2 \\ IWS 3, & \text{if } rand(1,4) = 3 \\ IWS 4, & \text{if } rand(1,4) = 4 \end{cases} \quad (10)$$

where IWS is the inertia weight scheme and $rand(1,4)$ is a random number generated-either 1, 2, 3 or 4. In this way, all inertia weight schemes have equal probability to be selected by each sub-population in each iteration. This, in turn, will not only enhance the diversity of the swarm, but also prevent the algorithm from being trapped in the local optimal.

The pseudocode of CBPSO-MIWS is demonstrated in Algorithm 1. In the first step, an initial population of N particles is randomly initialized (either 1 or 0), and the velocity of each particle is initialized to zero. Next, the population of particles is equally divided into ns sub-populations, where ns is the number of species. Afterward, the fitness of each particle for each species is evaluated. The $pbest_n$ and $gbest_n$ of each species are set, and the overall best particle from all species is known as $Gbest$. In each iteration, for each species, the inertia weight scheme (IWS) is selected, as shown in the Equation (10). Then, the inertia weight is computed based on the selected IWS. For each particle in each species, the velocity and position are updated using Equation (1) and (3), respectively. In the next step, the fitness of each particle of each species is evaluated. The $pbest_n$ and $gbest_n$ are again updated. At the end of each iteration, the overall global best particle, $gbest$ is updated. The algorithm is repeated until the maximum number of iterations is reached. Finally, the overall best particle is selected as the optimal feature subset.

Algorithm 1. Pseudocode of CBPSO-MIWS

Input: $N, T_{max}, v_{max}, v_{min}, ns, c_1$ and c_2

- 1) Initialize a population of particles, X_i ($i = 1, 2 \dots, N$)
- 2) Divide the population into ns sub-populations/species, S_n ($n = 1, 2 \dots, ns$)
- 3) Evaluate the fitness of particles for each species, $F(S_n)$ using fitness function
- 4) Define the global best particle of each species as $gbest_n$ ($n = 1, 2 \dots, ns$), and select the overall global best particle from $gbest_n$ and set it as $Gbest$
- 5) Set the personal best particles for each species as $pbest_n$ ($n = 1, 2 \dots, ns$)
- 6) **for** $t = 1$ to the maximum number of iteration, T_{max}
- 7) **for** $n = 1$ to the number of sub-population/species, ns
 - // Multiple Inertia Weight Strategy //
 - 8) Randomly select one IWS using Equation (10)
 - 9) Compute the inertia weight based on the selected IWS
 - 10) **for** $i = 1$ to the number of particles in each species
 - 11) **for** $d = 1$ to the number of dimension, D
 - // Velocity and Position Update // #Note that $pbest_i$ is selected from $pbest_n$
 - 12) Update the velocity of particle as shown in Equation (1)
 - 13) Convert the velocity into probability value using Equation (2)
 - 14) Update the position of particle as shown in Equation (3)
 - 15) **next** d
 - 16) Evaluate the fitness of particle by applying the fitness function
 - 17) Update $pbest_{n,i}$ and $gbest_n$
- 18) **next** i
- 19) **next** n
- 20) Update $Gbest$
- 21) **next** t

Output: Overall global best particle

3.2. Proposed CBPSO-MIWS for Feature Selection

In this section, the application of CBPSO-MIWS for solving the feature selection problem is described. Feature selection is known as a NP hard combinatorial problem, where the number of possible solutions increases exponentially with the number of features, D . Hence, an exhaustive search that requires a very high computational complexity was impractical. In this paper, a new CBPSO-MIWS is proposed to tackle the feature selection problem in classification tasks. Ultimately, our main goal was to select k potential features from a large available feature set, where $k < D$. As for feature selection, the solution is represented in binary form, which can be either bit 0 or 1. Bit 1 shows that the feature is selected, while the bit 0 exhibits the unselected feature [13]. Figure 2 illustrates an example of a solution with 10 dimensions. As can be observed, four features (2nd, 3rd, 6th and 10th) are selected from the original feature set.

1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
0	1	1	0	0	1	0	0	0	1

Figure 2. An example of a solution with 10 dimensions.

Figure 3 demonstrates the flowchart proposed for CBPSO-MIWS feature selection and classification tasks. Firstly, the benchmark feature set is acquired from UCI machine learning dataset. Then, the proposed CBPSO-MIWS is applied to evaluate the most informative feature subset. In CBPSO-MIWS, a population of initial solutions with D dimensions are randomly generated, where D represents the number of features. The fitness of the initial solutions is evaluated, and the best solution (G_{best}) is set. Iteratively, the solutions are updated as shown in Algorithm 1. As for wrapper feature selection, the fitness function that considered both classification performance and feature size is utilized, it can be defined as follows:

$$Fitness = \alpha \frac{|S|}{|T|} + (1 - \alpha)Error \tag{11}$$

and:

$$Error = \frac{\text{Number of wrongly classified instances}}{\text{Total number of instances}} \tag{12}$$

where $|S|$ is the length of feature subset, $|T|$ is the total number of features in each dataset, $Error$ is the classification error rate and α is the parameter in $[0,1]$ to control the influence of classification performance and feature size. In this paper, we set α to 0.01 according to [2,3]. Note that the $Error$ is computed by using k -nearest neighbor (KNN) with Euclidean distance and $k = 5$. The KNN was chosen since it offers a very low computational complexity, and it is easy to implement [1,3,27]. In the final feature selection step, the global best solution (best feature subset) that comprises of the optimal features is produced. After that, the feature subset is fed into the KNN classifier for the classification process.

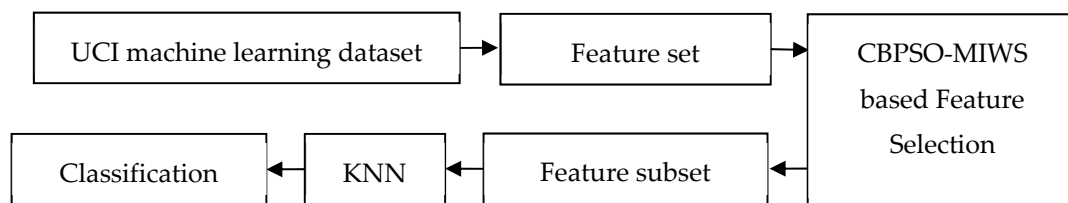


Figure 3. Overview of proposed CBPSO-MIWS for feature selection and classification.

4. Results

4.1. Dataset and Parameter Setting

In this paper, ten benchmark datasets collected from the UCI machine learning repository (<https://archive.ics.uci.edu/ml/index.php>) were used to validate the performance of CBPSO-MIWS.

Table 1 lists the ten benchmark datasets used in this work. For each dataset, the features were first normalized in the range between 0 and 1. In the process of fitness evaluation, the dataset was randomly partitioned into 80% for the training set and 20% for the testing set [4].

Table 1. The ten utilized benchmark datasets.

No	UCI Dataset	Number of Instances	Number of Features	Number of Classes
1	Breast Cancer Wisconsin	699	9	2
2	Diabetic Retinopathy	1151	19	2
3	Glass Identification	214	10	6
4	Ionosphere	351	34	2
5	Libras Movement	360	90	15
6	Musk 1	476	167	2
7	Breast Cancer Coimbra	116	9	2
8	Lung Cancer	32	56	3
9	Parkinson's Disease	756	754	2
10	Seeds	210	7	3

Furthermore, in order to measure the effectiveness of the proposed CBPSO-MIWS, four recent and popular feature selection methods include BPSO [13,22], genetic algorithm (GA) [28], binary gravitational search algorithm (BGSA) [29] and competitive binary grey wolf optimizer (CBGWO) [12] were used in comparison. GA is an evolutionary algorithm that utilizes the selection, crossover and mutation operators to evolve solutions. On one hand, BGSA is a binary version of the gravitational search algorithm (GSA), and it advances the resolution by calculating the total force to update acceleration, velocity and position of particles. Finally, CBGWO is an improved version of BGWO, in which the competition and leader enhancement strategy are integrated. The parameter settings of BPSO, GA, BGSA, CBGWO and CBPSO-MIWS are listed in Table 2. In the experiment, we tested the CBPSO-MIWS by using different numbers of species, ns (1, 2, 3, 4, and 5), and we found that the best result was obtained when $ns = 3$ and 4. To ensure fair comparison, the population size and the maximum number of iterations were fixed at 10 and 100 for each feature selection method. All the analysis was done in MATLAB 2017 software (MathWorks, Massachusetts and United States) by using a computer with Intel Core i3 3.3 GHz and 8.0 GB RAM.

Table 2. Parameter setting of BPSO, GA, BGSA, CBGWO and CBPSO-MIWS.

Parameters	Values				
	Proposed Method (CBPSO-MIWS)	Binary Particle Swarm Optimization (BPSO)	Genetic Algorithm (GA)	Binary Gravitational Search Algorithm (BGSA)	Competitive Binary Grey Wolf Optimizer (CBGWO)
Population size, N	10	10	10	10	10
Maximum number of iterations, T_{max}	100	100	100	100	100
Number of runs	20	20	20	20	20
Number of species, ns	3	-	-	-	-
w_{max}	0.9	-	-	-	-
w_{min}	0.4	-	-	-	-
w_0	0.9	-	-	-	-
c_1	2	2	-	-	-
c_2	2	2	-	-	-
v_{max}	6	6	-	6	-
v_{min}	-6	-6	-	-	-
p	1.2	-	-	-	-
CR	-	-	0.8	-	-
MR	-	-	0.01	-	-
w	-	0.9-0.4	-	-	-
G_0	-	-	-	100	-

4.2. Evaluation Metrics

The CBPSO-MIWS and other feature selection methods were executed for 20 independent runs in order to obtain useful statistical results. For performance evaluation, the best fitness, worst fitness, mean fitness, standard deviation of fitness (STD), accuracy, and feature size were recorded. These parameters can be calculated as follows [27,30,31]:

$$\text{Best Fitness} = \min_{t=1}^{T_{max}} F_t \quad (13)$$

$$\text{Worst Fitness} = \max_{t=1}^{T_{max}} F_t \quad (14)$$

$$\text{Mean Fitness} = \frac{1}{T_{max}} \sum_{t=1}^{T_{max}} F_t \quad (15)$$

$$\text{STD} = \sqrt{\frac{\sum_{t=1}^{T_{max}} (F_t - \mu)^2}{T_{max}}} \quad (16)$$

$$\text{Accuracy} = \frac{\text{Number of correctly predicted instances}}{\text{Total number of instances}} \times 100\% \quad (17)$$

where F is the fitness values of best solution, μ is the mean, t is the number of iteration and T_{max} is the maximum number of iterations. In the final step, the average of the parameters over 20 independent runs were calculated and presented as the experimental results.

4.3. Experimental Results

Figures 4 and 5 illustrate the convergence curves of five different feature selection methods for 10 datasets. Note that the fitness is the average fitness value obtained from 20 runs. In these Figures, the proposed CBPSO-MIWS is marked with diamond shape on green line. It is observed that the performance of CBPSO-MIWS was superior for most datasets. For dataset 3, 5, 7, 8 and 9, it can be clearly seen that CBPSO-MIWS outperformed other feature selection methods, which converged faster and deeper, to seek out the optimal solution. Such improvement is mostly coming from co-evolution and multiple inertia weight strategy, which greatly enhances the performance of CBPSO-MIWS in feature selection. Moreover, CBPSO-MIWS can usually achieve better fitness value than BPSO. This implies that the proposed approach overtakes BPSO by overcoming the limitations of BPSO in both premature convergence and the setting of inertia weight. On the other hand, CBGWO provided competitive performance in this work, especially for datasets 2, 4 and 6.

Table 3 outlines the results of best fitness, worst fitness, mean fitness, STD, accuracy and feature size of five different feature selection methods for 10 datasets. Note that the best parameter value for each method is bolded. In Table 3, the lower the values of best fitness, worst fitness and mean fitness are, the better the performance is. The STD represents the robustness and consistency of the algorithm. Thus, the feature selection method that scores the lowest STD value has very good robustness, which can produce highly consistent results. On one hand, a higher value of accuracy indicates that more samples have been successfully predicted. For instance, the feature size with a smaller number of features reveals less features are selected. Successively, CBPSO-MIWS showed a very competitive performance in best, worst and mean fitness values. The experimental results highlight that CBPSO-MIWS was very good in selecting the relevant features, thus leading to promising performance.

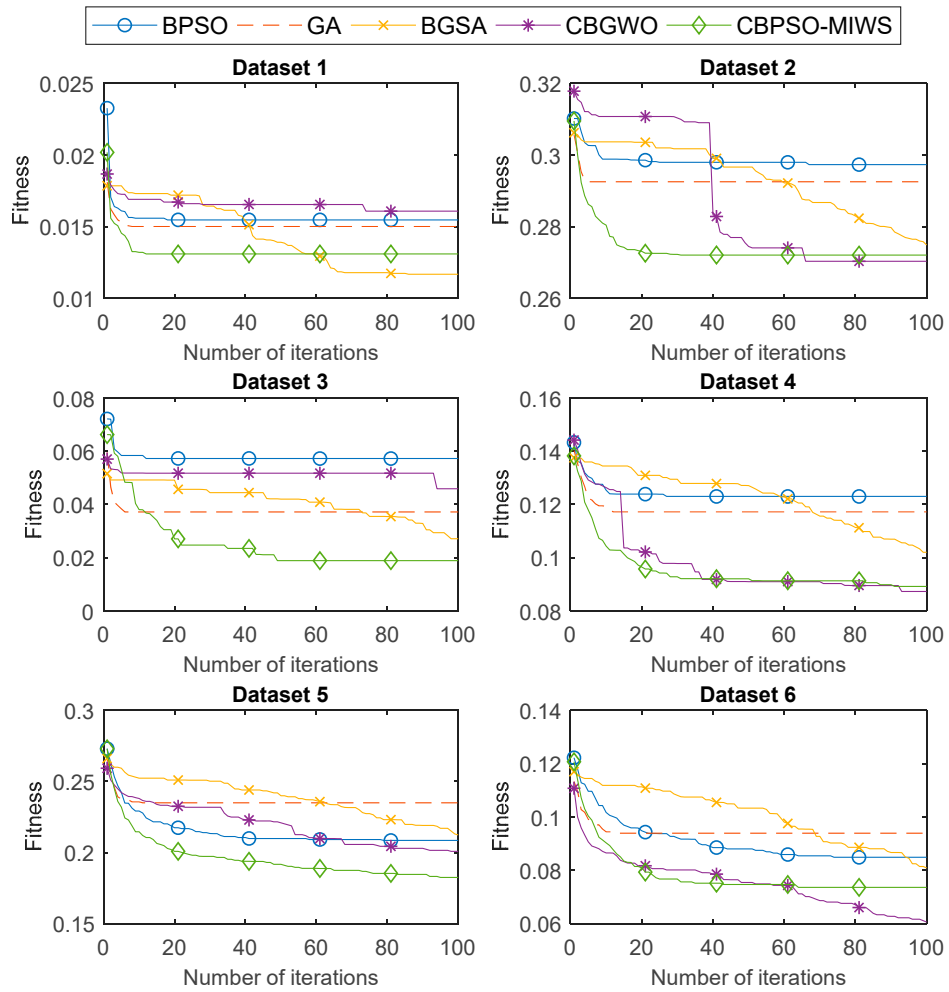


Figure 4. Convergence curves of five feature selection methods on datasets 1 to 6.

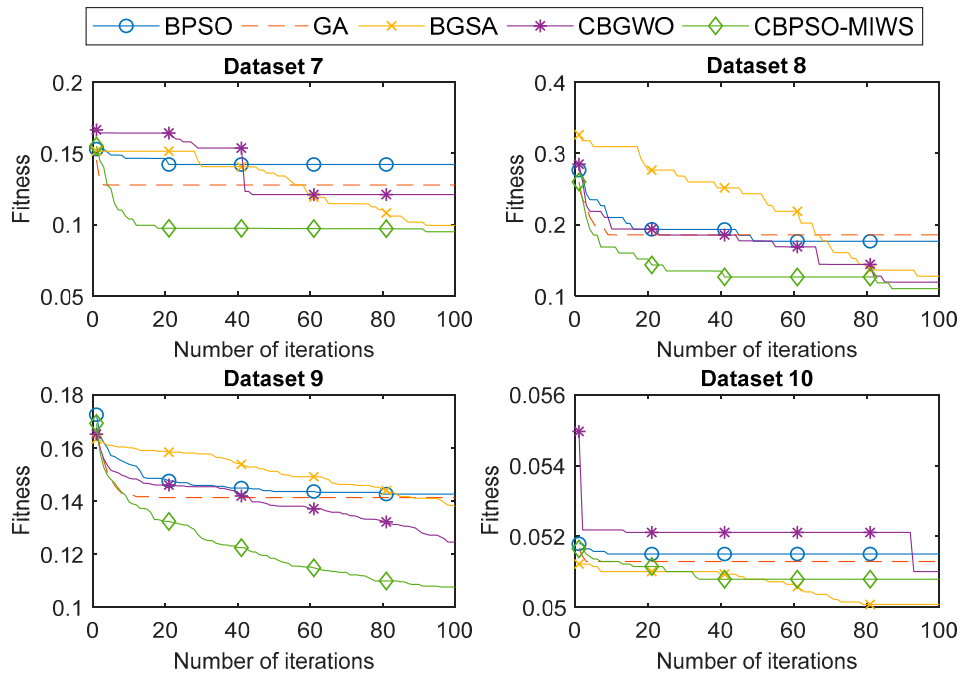


Figure 5. Convergence curves of five feature selection methods on datasets 7 to 10.

Table 3. Experimental results of five feature selection methods for 10 datasets.

Dataset	Feature Selection Method	Best Fitness	Worst Fitness	Mean Fitness	STD	Accuracy (%)	Feature Size
1	BPSO	0.0155	0.0233	0.0156	0.0009	98.96	4.70
	GA	0.0150	0.0181	0.0151	0.0004	99.00	4.60
	BGSA	0.0117	0.0179	0.0143	0.0026	99.29	4.15
	CBGWO	0.0161	0.0187	0.0165	0.0006	98.96	5.25
	Proposed	0.0131	0.0202	0.0133	0.0009	99.14	4.15
2	BPSO	0.2973	0.3102	0.2984	0.0025	70.41	8.40
	GA	0.2925	0.3056	0.2928	0.0016	70.89	8.30
	BGSA	0.2749	0.3062	0.2934	0.0108	72.70	8.70
	CBGWO	0.2703	0.3178	0.2876	0.0193	73.11	7.80
	Proposed	0.2721	0.3095	0.2740	0.0063	72.89	7.00
3	BPSO	0.0572	0.0720	0.0576	0.0024	94.65	4.20
	GA	0.0371	0.0595	0.0375	0.0027	96.63	3.75
	BGSA	0.0271	0.0515	0.0412	0.0083	97.56	2.90
	CBGWO	0.0458	0.0570	0.0513	0.0021	95.70	3.25
	Proposed	0.0189	0.0662	0.0250	0.0129	98.37	2.75
4	BPSO	0.1229	0.1432	0.1239	0.0035	88.00	14.10
	GA	0.1172	0.1402	0.1180	0.0037	88.57	13.65
	BGSA	0.1020	0.1374	0.1225	0.0117	90.07	12.55
	CBGWO	0.0873	0.1441	0.0978	0.0145	91.50	10.80
	Proposed	0.0892	0.1381	0.0951	0.0103	91.36	12.35
5	BPSO	0.2084	0.2730	0.2147	0.0124	79.44	44.50
	GA	0.2349	0.2660	0.2357	0.0042	76.74	41.65
	BGSA	0.2123	0.2661	0.2386	0.0150	79.03	42.30
	CBGWO	0.2008	0.2592	0.2191	0.0162	80.21	43.90
	Proposed	0.1825	0.2729	0.1958	0.0170	82.01	39.95
6	BPSO	0.0849	0.1222	0.0907	0.0092	91.89	77.05
	GA	0.0939	0.1133	0.0946	0.0032	91.00	80.15
	BGSA	0.0809	0.1170	0.1006	0.0116	92.32	80.10
	CBGWO	0.0606	0.1107	0.0753	0.0109	94.32	71.70
	Proposed	0.0736	0.1207	0.0782	0.0099	93.05	80.30
7	BPSO	0.1422	0.1531	0.1434	0.0031	86.09	4.05
	GA	0.1278	0.1454	0.1280	0.0018	87.61	4.65
	BGSA	0.0995	0.1517	0.1296	0.0227	90.43	4.30
	CBGWO	0.1211	0.1665	0.1371	0.0203	88.26	4.40
	Proposed	0.0950	0.1552	0.1000	0.0118	90.87	4.15
8	BPSO	0.1768	0.2766	0.1894	0.0262	82.50	20.10
	GA	0.1857	0.2519	0.1879	0.0113	81.67	23.60
	BGSA	0.1276	0.3261	0.2233	0.0789	87.50	21.70
	CBGWO	0.1193	0.2849	0.1693	0.0452	88.33	21.35
	Proposed	0.1102	0.2600	0.1359	0.0355	89.17	16.45
9	BPSO	0.1425	0.1725	0.1460	0.0060	86.09	366.40
	GA	0.1413	0.1659	0.1421	0.0038	86.23	368.10
	BGSA	0.1380	0.1633	0.1512	0.0079	86.56	371.65
	CBGWO	0.1245	0.1652	0.1394	0.0092	87.88	338.75
	Proposed	0.1075	0.1692	0.1217	0.0138	89.60	347.10
10	BPSO	0.0515	0.0518	0.0515	0.0001	95.24	3.05
	GA	0.0513	0.0516	0.0513	0.0000	95.24	2.90
	BGSA	0.0501	0.0512	0.0506	0.0005	95.24	2.05
	CBGWO	0.0510	0.0550	0.0521	0.0006	95.24	2.70
	Proposed	0.0508	0.0516	0.0509	0.0003	95.24	2.55

From Table 3, CBPSO-MIWS offered mean fitness with a very low STD value. This again expresses the high consistency of CBPSO-MIWS in feature selection. Another important result is the accuracy. Based on the result obtained, CBPSO-MIWS ranked first in six datasets, which outperformed other methods in feature selection problems. By applying the CBPSO-MIWS to evaluate the optimal feature subset, a high classification performance can be guaranteed. As for feature size, it shows that roughly half of the features can be eliminated for all methods, which indicates that some of the features are redundant and they badly degraded the classification result. The experimental result clearly evinced the impact of feature selection in classification tasks.

Since the classification result of each subject is the average accuracy obtained from 20 independent runs, intuitively, the two sample *t*-test with 95% confidential level was applied to examine whether the classification performance achieved by the proposed CBPSO-MIWS was significantly better than the other methods. In the statistical test, a null hypothesis indicates that the classification performance of two different methods were similar. If the *p*-value is less than 0.05, then the null hypothesis is rejected, which claims that there is a significant difference in classification performance among two different methods. Table 4 demonstrates the results of the *t*-test with *p*-values by using the CBPSO-MIWS as the reference method. In this Table, “*” indicates the performance of proposed method was significantly better, while “**” means the performance of CBPSO-MIWS was significantly worse. As can be observed, the classification performance of CBPSO-MIWS was significantly better than BPSO and GA (*p*-value < 0.05) on eight datasets, respectively. On one side, CBPSO-MIWS significantly outperformed BGSA and CBGWO (*p*-value < 0.05) for four datasets.

Table 4. The result of *t*-test with *p*-values (The CBPSO-MIWS is used as reference algorithm).

Dataset	<i>p</i> -Value			
	BPSO	GA	BGSA	CBGWO
1	0.36414	0.22519	0.03557 **	0.20295
2	0.00061 *	6.00×10^{-5} *	0.54053	0.53562
3	0.00162 *	0.02147 *	0.28239	0.00183 *
4	1.00×10^{-5} *	0.00000 *	0.00271 *	0.72344
5	0.00016 *	0.00000 *	0.00000 *	0.00176 *
6	0.00548 *	1.00×10^{-5} *	0.02268 *	0.00281 **
7	0.00012 *	0.00000 *	0.38880	3.00×10^{-5} *
8	0.00197 *	0.00963 *	0.50274	0.74359
9	0.00000 *	0.00000 *	0.00000 *	1.00×10^{-5} *
10	1.00000	1.00000	1.00000	1.00000

In this study, the most important measurement is the accuracy, which indicates goodness of the features selected by the proposed method in classification tasks. For the ease of understanding, the accuracies obtained by the 10 datasets were averaged, and the result of mean accuracy is displayed in Figure 6. In this Figure, the error bars represent the standard deviation value. Averaged across 10 datasets, it shows that the best mean accuracy was achieved by CBPSO-MIWS (90.17%), followed by CBGWO (89.35%). When inspecting the results, the classification performance of CBPSO-MIWS was superior against BPSO, GA and BGSA, and slightly better than CBGWO. This again validates the effectiveness of the CBPSO-MIWS in selecting the significant features.

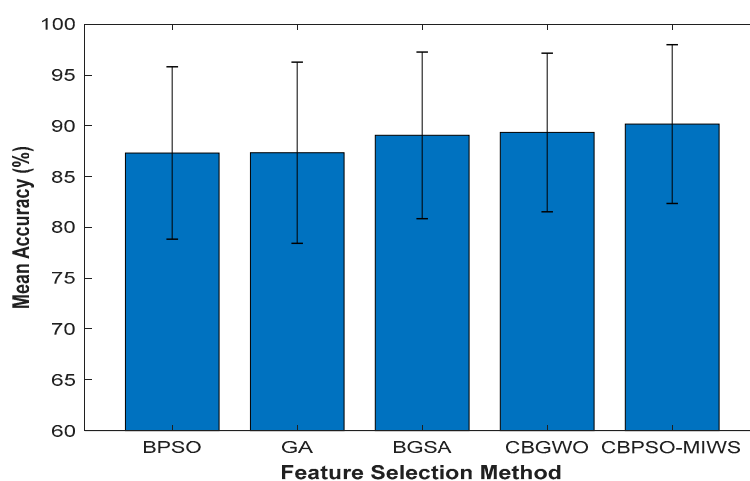


Figure 6. Mean accuracy of five different feature selection methods over 10 datasets.

Table 5 exhibits the average computational cost of five feature selection methods. As can be observed, the highest computational result was achieved by GA. In comparison with BPSO, CBPSO-MIWS was more time consuming. Nevertheless, CBPSO-MIWS contributed better classification performance. On the other hand, CBGWO offered the fastest processing speed in this work. This expected because CBGWO utilizes a competition strategy, in which only half of the population is used in evaluation process. Even though the computational complexity of CBPSO-MIWS was higher than CBGWO, CBPSO-MIWS could often affirm promising results. For instance, only four simple inertia weight schemes are utilized in CBPSO-MIWS. Hence, it is believed that the performance of CBPSO-MIWS can show great improvement when more formidable inertia weight schemes are implemented.

Table 5. The average computational cost of five feature selection methods.

Dataset	Average Computational Time (s)				
	BPSO	GA	BGSA	CBGWO	CBPSO-MIWS
1	5.603	8.952	5.613	4.524	6.861
2	15.321	24.499	14.646	12.388	17.731
3	1.687	2.380	1.693	1.331	2.091
4	2.465	3.804	2.435	1.951	3.208
5	2.884	4.182	3.082	2.213	3.663
6	4.043	6.008	4.390	3.036	4.931
7	1.233	1.858	1.629	1.010	1.496
8	1.177	1.654	1.439	0.916	1.492
9	13.496	19.645	13.851	9.849	16.273
10	1.528	2.476	1.611	1.211	2.057

5. Discussion

In the present study, a new co-evolution binary particle swarm optimization with multiple inertia weight strategy (CBPSO-MIWS) has been proposed for wrapper feature selection. In the proposed scheme, the co-evolution concept and multiple inertia weight strategy are adopted to heighten the performance of CBPSO-MIWS in feature selection. Owing to the co-evolution strategy, the particles are able to share the information while performing the search on different search spaces. This in turn will maximize the global search capability in the process of evaluation. On one hand, the multiple inertia weight strategy promotes the usage of different weight components, which is good at improving the diversity of the algorithm. By making full use of these mechanisms, CBPSO-MIWS has the ability to prevent premature convergence, thus leading to promising results.

This study has shown the impact of feature selection in classification tasks. CBPSO-MIWS not only eliminated redundant and irrelevant features, but also enhanced the classification performance. It is worth noting that CBPSO-MIWS can be applied without prior knowledge. CBPSO-MIWS automatically selects the best feature subset for each dataset, and then stores the selected features for real time application. The experimental results evidently show the superiority of CBPSO-MIWS in the feature selection problem. The findings of the current work indicate that CBPSO-MIWS is a powerful method, which can select the optimal feature subset that best describes the target in the classification process. Thus, CBPSO-MIWS can be a useful tool in engineering, rehabilitation and clinical applications.

There were several limitations in this work. First, only four types of inertia weight schemes were used in this research. It must be mentioned that other inertia weight schemes are also applicable in CBPSO-MIWS. Second, the number of species/sub-population, ns is fixed at 3 in the present work. The users are encouraged to test different numbers of species in order to achieve the optimal performance. Note that the number of species is related to the population size. A larger number of species is recommended if the population size is larger, which ensures the goodness of the multiple inertia weight strategy in the optimization task.

6. Conclusions

Feature selection is an effective way to improve classification performance with a minimal number of features. In this paper, we have proposed a new variant of BPSO, namely co-evolution binary particle swarm optimization with a multiple inertia weight strategy (CBPSO-MIWS) for wrapper feature selection. The main contribution was the proposal of a co-evolution concept and multiple inertia weight strategy (MIWS) into CBPSO-MIWS, which improved the diversity and prevented the algorithm from having premature convergence. Ten benchmark datasets from the UCI repository were used to test the proposed approach, and the results were compared with other recent and popular feature selection approaches. Based on the results obtained, CBPSO-MIWS can provide competitive and promising performances against other approaches. In comparison with BPSO, CBPSO-MIWS usually selected a subset of minimal features that gave the highest accuracy in this work. Thus, it is concluded CBPSO-MIWS can be useful in engineering, rehabilitation and clinical applications.

This research makes open to the public, different future research directions where co-evolution and multiple inertia weight strategies can be implemented in other optimization algorithms. In future, more inertia weight schemes can be added into CBPSO-MIWS for performance enhancement. Moreover, the adaptive scheme that can automatically adjust the number of species can be implemented in CBPSO-MIWS for the extension of future work.

Author Contributions: Conceptualization, J.T.; Formal analysis, J.T.; Funding acquisition, A.R.A.; Investigation, J.T.; Methodology, J.T.; Software, J.T.; Supervision, A.R.A.; Validation, J.T.; Writing—original draft, J.T.; Writing—review & editing, J.T., A.R.A. and N.M.S.

Funding: This research and the Article Processing Charge were funded by the Ministry of Higher Education (MOHE) Malaysia under grant number GLuar/STEVIA/2016/FKE-CeRIA/100009.

Acknowledgments: The authors would like to thank Skim Zamalah UTeM and the Ministry of Higher Education Malaysia for funding this research under grant GLuar/STEVIA/2016/FKE-CeRIA/100009.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xue, B.; Zhang, M.; Browne, W.N. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Appl. Soft Comput.* **2014**, *18*, 261–276. [[CrossRef](#)]
2. Al-Madi, N.; Faris, H.; Mirjalili, S. Binary multi-verse optimization algorithm for global optimization and discrete problems. *Int. J. Mach. Learn. Cybern.* **2019**, 1–21. [[CrossRef](#)]
3. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary ant lion approaches for feature selection. *Neurocomputing* **2016**, *213*, 54–65. [[CrossRef](#)]
4. Faris, H.; Mafarja, M.M.; Heidari, A.A.; Aljarah, I.; Al-Zoubi, A.M.; Mirjalili, S.; Fujita, H. An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. *Knowl-Based Syst.* **2018**, *154*, 43–67. [[CrossRef](#)]
5. Hafiz, F.; Swain, A.; Patel, N.; Naik, C. A two-dimensional (2-D) learning framework for Particle Swarm based feature selection. *Pattern Recognit.* **2018**, *76*, 416–433. [[CrossRef](#)]
6. Tran, B.; Xue, B.; Zhang, M. Variable-Length Particle Swarm Optimisation for Feature Selection on High-Dimensional Classification. *IEEE Trans. Evol. Comput.* **2018**, *1*. [[CrossRef](#)]
7. Huang, H.; Xie, H.B.; Guo, J.Y.; Chen, H.J. Ant colony optimization-based feature selection method for surface electromyography signals classification. *Comput. Biol. Med.* **2012**, *42*, 30–38. [[CrossRef](#)] [[PubMed](#)]
8. Mesa, I.; Rubio, A.; Tubia, I.; De No, J.; Diaz, J. Channel and feature selection for a surface electromyographic pattern recognition task. *Expert Syst. Appl.* **2014**, *41*, 5190–5200. [[CrossRef](#)]
9. Venugopal, G.; Navaneethakrishna, M.; Ramakrishnan, S. Extraction and analysis of multiple time window features associated with muscle fatigue conditions using sEMG signals. *Expert Syst. Appl.* **2014**, *41*, 2652–2659. [[CrossRef](#)]
10. Phinyomark, A.; N Khushaba, R.; Scheme, E. Feature Extraction and Selection for Myoelectric Control Based on Wearable EMG Sensors. *Sensors* **2018**, *18*, 1615. [[CrossRef](#)]

11. Purushothaman, G.; Vikas, R. Identification of a feature selection based pattern recognition scheme for finger movement recognition from multichannel EMG signals. *Australas Phys. Eng. Sci. Med.* **2018**, *41*, 549–559. [[CrossRef](#)]
12. Too, J.; Abdullah, A.; Mohd Saad, N.; Mohd Ali, N.; Tee, W. A New Competitive Binary Grey Wolf Optimizer to Solve the Feature Selection Problem in EMG Signals Classification. *Computers* **2018**, *7*, 58. [[CrossRef](#)]
13. Chuang, L.Y.; Chang, H.W.; Tu, C.J.; Yang, C.H. Improved binary PSO for feature selection using gene expression data. *Comput. Biol. Chem.* **2008**, *32*, 29–38. [[CrossRef](#)] [[PubMed](#)]
14. Xue, B.; Zhang, M.; Browne, W.N. Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. *IEEE Trans. Cybern.* **2013**, *43*, 1656–1671. [[CrossRef](#)]
15. Gou, J.; Lei, Y.X.; Guo, W.P.; Wang, C.; Cai, Y.Q.; Luo, W. A novel improved particle swarm optimization algorithm based on individual difference evolution. *Appl. Soft. Comput.* **2017**, *57*, 468–481. [[CrossRef](#)]
16. Dong, W.; Zhou, M. A Supervised Learning and Control Method to Improve Particle Swarm Optimization Algorithms. *IEEE Trans. Syst. Man. Cybern. Syst.* **2017**, *47*, 1135–1148. [[CrossRef](#)]
17. Jensi, R.; Jiji, G.W. An enhanced particle swarm optimization with levy flight for global optimization. *Appl. Soft Comput.* **2016**, *43*, 248–261. [[CrossRef](#)]
18. Adeli, A.; Broumandnia, A. Image steganalysis using improved particle swarm optimization based feature selection. *Appl. Intell.* **2018**, *48*, 1609–1622. [[CrossRef](#)]
19. Banka, H.; Dara, S. A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation. *Pattern Recognit. Lett.* **2015**, *52*, 94–100. [[CrossRef](#)]
20. Bharti, K.K.; Singh, P.K. Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering. *Appl. Soft Comput.* **2016**, *43*, 20–34. [[CrossRef](#)]
21. Kennedy, J.; Eberhart, R.C. A discrete binary version of the particle swarm algorithm. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Orlando, FL, USA, 12–15 October 1997. [[CrossRef](#)]
22. Too, J.; Abdullah, A.R.; Mohd Saad, N.; Tee, W. EMG Feature Selection and Classification Using a Pbest-Guide Binary Particle Swarm Optimization. *Computation* **2019**, *7*, 12. [[CrossRef](#)]
23. Unler, A.; Murat, A. A discrete particle swarm optimization method for feature selection in binary classification problems. *Eur. J. Oper. Res.* **2010**, *206*, 528–539. [[CrossRef](#)]
24. Taherkhani, M.; Safabakhsh, R. A novel stability-based adaptive inertia weight for particle swarm optimization. *Appl. Soft Comput.* **2016**, *38*, 281–295. [[CrossRef](#)]
25. Chatterjee, A.; Siarry, P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Comput. Oper. Res.* **2006**, *33*, 859–871. [[CrossRef](#)]
26. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998. [[CrossRef](#)]
27. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, *172*, 371–381. [[CrossRef](#)]
28. Huang, C.L.; Wang, C.J. A GA-based feature selection and parameters optimization for support vector machines. *Expert Syst. Appl.* **2006**, *31*, 231–240. [[CrossRef](#)]
29. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. BGSa: Binary gravitational search algorithm. *Nat. Comput.* **2010**, *9*, 727–745. [[CrossRef](#)]
30. Zawbaa, H.M.; Emary, E.; Grosan, C. Feature Selection via Chaotic Antlion Optimization. *PLoS ONE* **2016**, *11*, e0150652. [[CrossRef](#)]
31. Sayed, G.I.; Hassanien, A.E. Moth-flame swarm optimization with neutrosophic sets for automatic mitosis detection in breast cancer histology images. *Appl. Intell.* **2017**, *47*, 397–408. [[CrossRef](#)]

